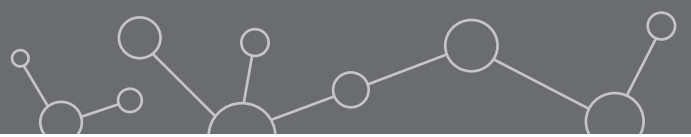


MANUAL

VERSION 7.6



Contents

1	Installation	3
1.1	32-bit or 64-bit installation	5
1.2	System requirements	7
1.2.1	Hardware requirements	7
1.2.1.1	Minimum hardware requirements	7
1.2.1.2	Recommended hardware configuration	7
1.2.2	Operating system	8
1.2.3	Microsoft .NET Framework 2.0 SP2	9
1.2.4	Microsoft .NET Framework 3.5 SP1	9
1.2.5	Microsoft Visual C++ 2012 Redistributable	9
1.2.6	Permissions	9
1.2.7	Security software	9
1.2.7.1	Anti-Virus	9
1.2.7.2	Firewall and proxy servers	10
1.3	Installation procedure	13
1.3.1	Installing a new BioNumerics instance	13
1.3.1.1	Prerequisites	13
1.3.1.2	Existing instances detected	14
1.3.1.3	Welcome dialog	14
1.3.1.4	Software End User License Agreement	14
1.3.1.5	Customer information	15
1.3.1.6	Setup Type	16
1.3.1.7	Choose destination location	16
1.3.1.8	Select features	17
1.3.1.9	NetKey+ connection settings	18
1.3.1.10	Confirm installation	19
1.3.1.11	NetKey+ configuration	20
1.3.1.12	Setup INI XML file	20
1.3.2	Updating a BioNumerics instance	20
1.3.2.1	Welcome dialog	20
1.3.2.2	Software End User License Agreement	21
1.3.2.3	Customer information	21
1.3.2.4	Choose destination location	22
1.3.2.5	Select features	23
1.3.2.6	NetKey+ connection settings	25
1.3.2.7	Confirm update	25
1.3.3	Maintenance installation	26
1.3.3.1	Select instance to maintain	26
1.3.3.2	Maintenance options	26
1.3.3.3	Modify maintenance mode	26
1.3.3.4	Repair maintenance mode	28

1.3.3.5	Remove maintenance mode	28
1.3.4	Installing Protection Keys	29
1.3.4.1	Protection Key Types	29
1.3.4.2	Install Protection Key Driver	29
1.3.4.3	Activate Sentinel HASP SL key	30
1.3.5	Setup log	37
1.3.6	Silent installation	37
1.3.6.1	Purpose	37
1.3.6.2	Procedure	37
1.3.6.3	Setup INI XML file format	39
1.3.7	Silent un-installation	40
1.3.7.1	Purpose	40
1.3.7.2	Procedure	40
1.4	NetKey+ configuration	43
1.4.1	Introduction	43
1.4.2	Installing and starting the NetKey+ service on the server	43
1.4.3	Configuring licenses	46
1.4.4	Running sessions on the clients	51
1.4.5	Monitoring sessions	52
1.4.6	Logging data	53
1.4.7	Resetting the NetKey+ settings	53
1.4.8	Repairing the NetKey+ service	54
1.4.9	Overview of configuration rights	55
1.4.10	Usage statistics	55
1.4.10.1	Usage information parse tool	55
1.4.10.2	Example	57
1.5	Installation process	61
1.5.1	Overview	61
1.5.2	Setup dialog list	61
1.5.3	Setup processes	61
1.5.3.1	Read command line options	61
1.5.3.2	Read global variables	62
1.5.3.3	Write global variables	62
1.5.3.4	Save Setup INI XML file	62
1.5.3.5	Read requested features	62
1.5.3.6	Save Setup Log	63
1.5.3.7	OnMoveData	63
1.5.3.8	Feature functions	63
1.5.4	Setup Process list	66
1.6	Command line options	69
1.6.1	Running BioNumerics from the command line	69
1.6.2	Running the startup program from the command line	70
1.7	Granting access to BioNumerics databases	71
2	Concepts	73
2.1	The concepts of BioNumerics	75
2.1.1	The programs	75
2.1.2	The database and the experiments	75

2.1.3	Multi-database setup	76
2.1.4	Modules and features	76
2.1.5	BioNumerics plugins	78
2.1.5.1	Introduction	78
2.1.5.2	Plugins included with the installation	80
2.1.5.3	Database plugins	81
2.1.6	Script languages	83
2.1.7	Example databases	84
2.2	About this guide	87
2.2.1	Documentation formats	87
2.2.2	Conventions	89
2.2.3	Toolbars	91
2.2.4	Floating menus	91
2.2.5	Shortcut keys	92
2.3	The BioNumerics user interface	93
2.3.1	Introduction to the BioNumerics user interface	93
2.3.2	The BioNumerics main window	93
2.3.3	General settings and preferences	95
2.3.3.1	Introduction	95
2.3.3.2	General preferences	96
2.3.3.3	BioNumerics windows preferences	96
2.3.3.4	Calculation preferences	97
2.3.3.5	Comparison preferences	97
2.3.3.6	Color maps preferences	98
2.3.3.7	Resetting preferences and settings to factory defaults	99
2.3.4	Display of panels	100
2.3.5	Configuring toolbars	102
2.3.6	The Experiments panel	103
2.3.7	Zoom sliders	103
2.3.8	Navigator pane	104
3	The BioNumerics database	107
3.1	Administering databases	109
3.1.1	The BioNumerics Startup program	109
3.1.2	The BioNumerics home directory	110
3.1.3	Creating a new database	112
3.1.4	Removing databases	115
3.1.5	Backing up and restoring databases	116
3.1.5.1	Backups to protect against data loss	116
3.1.5.2	Databases created by BioNumerics	116
3.1.5.3	Databases not created by BioNumerics	116
3.2	Database objects	119
3.2.1	Introduction	119
3.2.2	Object views	120
3.2.3	Object access settings	123
3.2.4	Object selections	124
3.2.5	Adding and removing object information fields	125
3.2.6	Editing object information	126

3.2.7	Displaying information fields	126
3.2.8	Sorting object grid panels	128
3.2.9	Actions on objects	128
3.2.10	Searching for objects using statements	128
3.2.11	Finding an object in an object grid panel	130
3.2.12	Exporting object information	131
3.2.13	Object attachments	131
3.2.14	Object queries	135
3.2.15	Cross-links between database objects	138
3.2.15.1	Introduction	138
3.2.15.2	Creating object cross-links	139
3.2.15.3	Managing object cross-links	141
3.2.15.4	Visualizing object cross-links	142
3.3	Database entries	143
3.3.1	Introduction	143
3.3.2	Adding and removing database entries	143
3.3.3	Creating information fields	144
3.3.4	Entering information in entry fields	146
3.3.5	Importing information from external sources	147
3.3.5.1	Introduction	147
3.3.5.2	Importing entry information	149
3.3.5.3	Importing entry mapping information	157
3.3.5.4	Managing import templates	160
3.3.5.5	Advanced template options	162
3.3.6	Information field properties	168
3.3.7	Configuring the database entries panel	173
3.3.8	Selections of database entries	175
3.3.9	Entry searches based on information and experiment data	176
3.3.9.1	Introduction	176
3.3.9.2	Database field	176
3.3.9.3	Database field range	177
3.3.9.4	Experiment presence	177
3.3.9.5	Fingerprint bands	177
3.3.9.6	Character value	178
3.3.9.7	Subsequence	179
3.3.9.8	Trend data parameter	179
3.3.9.9	Attachment	180
3.3.9.10	Logical operators	180
3.3.10	Levels and dependencies	181
3.3.10.1	Introduction	181
3.3.10.2	Creating database levels	183
3.3.10.3	Summarizing information and experiments	184
3.3.10.4	Editing entry dependencies	185
3.3.10.5	Importing data in a leveled database	186
3.4	Database exchange tools	189
3.4.1	Solutions for data exchange: bundles and XML files	189
3.4.2	Database exchange using Bundles	189
3.4.3	Database exchange using XML files	192
3.4.3.1	Introduction	192
3.4.3.2	Export data exchange data	192
3.4.3.3	Import data exchange data	194

3.5	User management	199
3.5.1	Introduction	199
3.5.2	Users and user groups	200
3.5.3	User authentication	205
3.5.4	User activity log	207
3.6	Audit trails and versioning	209
3.6.1	Introduction	209
3.6.2	Audit trail settings	210
3.6.3	The Audit trail window	212
3.6.4	Digital signatures	215
3.7	The BioNumerics relational database	219
3.7.1	Principles	219
3.7.2	Configuring the relational database	220
3.7.3	Data migration	226
3.7.3.1	Introduction	226
3.7.3.2	Migrating data with the MigrationGUI script	226
3.7.3.3	The Migration.exe command line tool	227
3.7.4	BioNumerics 64-bit versions and MS Access	228
3.7.5	Protecting databases at the DBMS level	229
3.7.5.1	Introduction	229
3.7.5.2	Access databases	229
3.7.5.3	Other relational databases	230
3.8	BioNumerics process templates	231
3.8.1	The Process data dialog	231
3.8.2	Process templates	232
3.8.2.1	Comparison	232
3.8.2.2	Sequence read set type	232
3.8.2.3	Sequence type	234
3.8.2.4	Spectrum type	235
4	Fingerprint types	237
4.1	Setting up fingerprint type experiments	239
4.1.1	Introduction	239
4.1.2	Creating a new fingerprint type	239
4.1.3	Importing and processing gel images	241
4.1.3.1	Import and image pre-processing	241
4.1.3.2	Processing steps	244
4.1.3.3	Defining pattern strips on the gel	245
4.1.3.4	Defining densitometric curves	251
4.1.3.5	Normalizing a gel	254
4.1.3.6	Defining bands and quantification	260
4.1.3.7	Advanced band search using a size-dependent threshold	263
4.1.3.8	Adding gel lanes to the database	264
4.1.3.9	Quantification of bands	266
4.1.4	Importing and processing capillary sequencer curves	268
4.1.4.1	Import	268
4.1.4.2	Processing steps	289
4.1.4.3	Data visualization	289

4.1.4.4	Defining reference bands	293
4.1.4.5	Creating a reference system	295
4.1.4.6	Normalization	296
4.1.4.7	Defining data bands	297
4.1.4.8	Dealing with bleed through	299
4.1.5	Editing the fingerprint type settings	300
4.1.5.1	Overview of available settings	300
4.1.5.2	Fingerprint data type	301
4.1.5.3	General settings	302
4.1.5.4	Layout settings	302
4.1.5.5	Comparative quantification settings	302
4.1.5.6	Comparison settings	302
4.1.5.7	Reference systems and calibration curves	303
4.1.5.8	Standard profile	307
4.1.5.9	Active zones on fingerprints	307
4.1.5.10	Band classes	307
4.1.6	Importing fingerprint information from a text file	310
4.1.7	Fingerprint experiment display	313
4.1.8	Exporting fingerprint data	313
4.2	Cluster analysis of fingerprints	315
4.2.1	Fingerprint comparison settings	315
4.2.2	Fingerprint display functions	319
4.2.3	Defining 'active zones' on fingerprints	319
4.2.4	Optimization of similarity coefficient parameters	321
4.2.5	Exporting fingerprint information	323
4.3	Band matching and polymorphism analysis	325
4.3.1	Introduction	325
4.3.2	Creating a band matching	326
4.3.3	Manual editing of bands	329
4.3.4	Manual editing of a band matching	330
4.3.5	Adding entries to a band matching	331
4.3.6	Saving band classes to the fingerprint type	332
4.3.7	Band and band class filters	332
4.3.8	Exporting band matching information	334
4.3.9	Tools to display selective band classes	335
4.3.10	Creating a band matching table for polymorphism analysis	335
4.3.11	Finding discriminative bands between entries	337
5	Spectrum types	339
5.1	Setting up spectrum type experiments	341
5.1.1	Creating a new spectrum type	341
5.1.2	Settings of a spectrum type experiment	341
5.1.3	Importing and preprocessing of spectra	344
5.1.3.1	Introduction	344
5.1.3.2	Importing spectrum experiments	344
5.1.3.3	The preprocessing window	355
5.2	Making spectrum preprocessing templates	361
5.2.1	Introduction	361

5.2.2	Managing the workflow	361
5.2.3	The preprocessing operators and their parameters	363
5.2.3.1	Import/Export	364
5.2.3.2	Resampling	365
5.2.3.3	Processing	365
5.2.3.4	Peaks	367
5.3	Summary spectra	369
5.3.1	Creating summary spectra	369
5.3.2	Processing summary spectra	373
5.3.3	The summary processing operators and their parameters	378
5.3.3.1	Background	378
5.3.3.2	Import/Export	378
5.3.3.3	Disable spectra	378
5.3.3.4	Peak matching	378
5.3.3.5	Summary	379
5.4	Cluster analysis of spectra	381
5.4.1	Spectrum comparison settings	381
5.4.2	Spectrum display functions	384
5.5	Peak matching	385
5.5.1	Introduction	385
5.5.2	Creating a peak matching	385
5.5.3	Managing peak class types	387
5.5.4	Managing peak classes	387
5.5.5	Managing peak class views	389
5.5.6	Creating a peak matching table	391
5.5.7	Finding discriminative peaks between entries	391
6	Character types	393
6.1	Setting up character type experiments	395
6.1.1	Defining a new character type	395
6.1.2	Editing a character type	396
6.1.2.1	The Character type window	396
6.1.2.2	General character type settings	397
6.1.2.3	Adding and removing characters	399
6.1.2.4	Enabling and disabling characters	401
6.1.2.5	Rearranging characters	402
6.1.2.6	Character ranges and color scales	402
6.1.2.7	Character mappings	403
6.1.2.8	Character comparison settings	405
6.1.2.9	Character type information fields	405
6.1.2.10	Importing character information from a text file	406
6.1.2.11	Creating and managing character views	408
6.1.3	Importing character data	410
6.1.3.1	Import options for character data	410
6.1.3.2	Importing fields and characters	410
6.1.3.3	Importing characters (db style)	419
6.1.3.4	Import of character data by quantification of images	422
6.1.4	Character experiment card	433

6.1.5	Exporting character data	435
6.2	Cluster analysis of characters	439
6.2.1	Selecting characters for comparison	439
6.2.2	Character comparison settings	442
6.2.3	Character display functions	448
6.2.4	Calculating the diversity of characters	448
6.2.5	Exporting character information	449
6.2.6	Character bar graphs	449
7	Trend data types	451
7.1	Setting up trend data type experiments	453
7.1.1	Introduction	453
7.1.2	Defining a new trend data type	454
7.1.3	Editing a trend data type	454
7.1.3.1	The Trend data type window	454
7.1.4	Trend curve models and parameters	458
7.1.4.1	Linear function	458
7.1.4.2	Logarithmic function	458
7.1.4.3	Exponential function	458
7.1.4.4	Power function	459
7.1.4.5	Hyperbolic function	459
7.1.4.6	Gaussian function	460
7.1.4.7	Logistic growth function	460
7.1.4.8	Gompertz function	461
7.1.4.9	Michaelis–Menten function	462
7.1.5	Importing trend data	462
7.1.5.1	Introduction	462
7.1.5.2	The Import wizard	463
7.1.6	Displaying trend data	467
7.1.7	Comparison settings	471
7.1.8	Additional trend data comparison parameters	471
7.2	Cluster analysis of trend data	475
7.2.1	Trend data comparison settings	475
7.2.2	Display options for trend data	477
7.2.3	Exporting trend data	478
7.3	Trend analysis	479
7.3.1	Introduction	479
7.3.1.1	Exploring trends among trend data	479
7.3.1.2	The Trend analysis window	479
7.3.2	Analysis of trend data	480
7.3.2.1	Using the data tree	480
7.3.2.2	Reports	481
7.3.2.3	Displaying trend data in channels	484
7.3.2.4	Calculating functions on trend curves	486
7.3.2.5	Regression and correlation analysis	488

8	Sequence types	495
8.1	Setting up sequence type experiments	497
8.1.1	Defining a new sequence type	497
8.1.2	Editing a sequence type	498
8.1.2.1	Sequence type settings	498
8.1.2.2	Converting to a reference mapped sequence type	499
8.1.2.3	The reference sequence and excluding regions	500
8.1.3	Importing sequence data	501
8.1.3.1	Import options for sequence data	501
8.1.3.2	Importing and assembling sequences in batch	501
8.1.3.3	Importing sequence assemblies from BAM or SAM files	543
8.1.3.4	Importing FASTA sequences from text files	556
8.1.3.5	Importing sequences from EMBL/GenBank files	563
8.1.3.6	Downloading sequences from internet	572
8.1.3.7	Assembling sequences from trace files	581
8.1.4	Exporting sequence data	596
8.1.5	The Sequence experiment card	598
8.1.6	The Sequence Editor	599
8.1.6.1	Introduction	599
8.1.6.2	General functionality	600
8.1.6.3	Annotation	606
8.1.6.4	Header	610
8.1.6.5	Sequence Search	612
8.1.6.6	Contig information	613
8.1.7	The Genome viewer	613
8.1.7.1	Introduction	613
8.1.7.2	The Genome panel	614
8.1.7.3	The Tracks and Features panels	616
8.1.7.4	Export options	616
8.2	Analysis tools for individual sequences	619
8.2.1	Introduction	619
8.2.2	Frame analysis	619
8.2.2.1	Frame Analysis panel	619
8.2.2.2	Frame Analysis features	619
8.2.3	Restriction analysis	622
8.2.3.1	Restriction Analysis panel	622
8.2.3.2	Restriction Analysis features	622
8.2.4	Primer analysis	627
8.2.4.1	Introduction	627
8.2.4.2	Creating a locus	627
8.2.4.3	Searching for primers and PCR products	630
8.2.4.4	Degeneracy and melting temperature plots	637
8.2.4.5	Adding primers to the oligo nucleotide database	637
8.3	Multiple alignment and cluster analysis of sequences	639
8.3.1	An introduction to sequence analysis	639
8.3.2	Calculating a cluster analysis based on a pairwise alignment	640
8.3.3	Calculating a multiple alignment	642
8.3.4	Multiple alignment display options	644
8.3.5	Editing a multiple alignment	645
8.3.6	Drag-and-drop manual alignment	646

8.3.7	Inserting and deleting gaps	646
8.3.8	Removing common gaps in a multiple alignment	649
8.3.9	Editing sequences in a multiple alignment	649
8.3.10	Finding a subsequence	649
8.3.11	Adding entries to and deleting entries from a multiple alignment	650
8.3.12	Automatically realigning selected sequences	651
8.3.13	Calculating a clustering based on a multiple alignment	651
8.3.14	Exporting a multiple alignment	652
8.3.15	Converting sequence data to categorical character sets	654
8.3.16	Writing comments in an alignment	655
8.4	Sequence alignment and mutation analysis	657
8.4.1	Introduction	657
8.4.2	Creating a new alignment project	657
8.4.3	The Alignment window	658
8.4.4	General functions	660
8.4.5	Adding and removing entries	661
8.4.6	Aligning sequences	661
8.4.7	Calculating a consensus sequence	665
8.4.8	Display options for sequences and curves	666
8.4.9	Editing an alignment	668
8.4.10	Calculating a global cluster analysis	669
8.4.11	Calculating a maximum parsimony cluster analysis	670
8.4.12	Dendrogram display functions	671
8.4.13	Cluster significance tools	671
8.4.14	Matrix display functions	672
8.4.15	Printing and exporting a sequence alignment	673
8.4.16	Finding sequence positions in an alignment	675
8.4.17	Sequence translation	675
8.4.18	Subsequence search	676
8.4.19	Mutation search	678
8.4.20	Defining bookmarks in a sequence alignment	680
8.4.21	Primer analysis	681
8.4.21.1	Introduction	681
8.4.21.2	General primer analysis	681
8.4.21.3	Discriminative primer design	683
8.5	Sequence databases	685
8.5.1	Restriction enzyme database	685
8.5.2	Oligo nucleotide database	688
8.5.3	Molecular weight markers	693
8.6	Pairwise chromosome comparisons	695
8.6.1	Introduction	695
8.6.2	Creating a new chromosome comparison project	697
8.6.3	Defining seed patterns and calculation of chromosome comparisons	698
8.7	Multiple chromosome alignments	707
8.7.1	Introduction	707
8.7.2	Calculating a chromosome alignment	707
8.7.3	Alignment overview and Alignment detail panel	711
8.7.4	Mutation analysis	714
8.7.5	Sequence search panel	718

8.7.6	Feature search panel	720
8.7.7	dNdS search panel	721
8.8	Genome annotation	725
8.8.1	Introduction	725
8.8.2	Creating a new annotation project	725
8.8.3	The Annotation window	726
8.8.4	Specifying a query sequence	726
8.8.5	Annotation steps	727
8.8.6	Frame analysis settings	728
8.8.7	Homology screening settings	729
8.8.8	Feature annotation settings	730
8.8.9	Calculating an annotation project	732
8.8.10	General functions	735
8.8.11	Editing an annotation	736
8.8.12	BLAST tools	737
8.9	BLAST analysis	739
8.9.1	Introduction	739
8.9.2	Creating a new BLAST analysis	740
8.9.2.1	Background	740
8.9.2.2	BLAST analysis from the main window	740
8.9.2.3	BLAST analysis from the Sequence editor	741
8.9.2.4	BLAST analysis from the Alignment window	741
8.9.2.5	BLAST analysis from the Chromosome comparison window	741
8.9.2.6	BLAST analysis from the Annotation window	742
8.9.2.7	The BLAST search settings	743
8.9.3	The BLAST window	747
8.9.4	BLAST databases	748
8.10	Whole genome single nucleotide polymorphism analysis	751
8.10.1	An introduction to wgSNP analysis	751
8.10.1.1	Definitions	751
8.10.1.2	Prerequisites	752
8.10.1.3	Workflow	752
8.10.2	SNP filtering	753
8.10.2.1	Introduction	753
8.10.2.2	Initiating a SNP filtering from the Main window	753
8.10.2.3	Initiating a SNP filtering from the Comparison window	756
8.10.2.4	The SNP filtering window	757
8.10.3	Analyses on filtered SNP matrices	764
8.10.3.1	Introduction	764
8.10.3.2	Calculating a clustering based on a SNP matrix	765
8.10.3.3	Exporting SNP data	766
9	Sequence read sets	767
9.1	Setting up sequence read set experiments	769
9.1.1	Introduction	769
9.1.2	Creating a new sequence read set experiment	769
9.1.3	Editing a sequence read set experiment	770
9.1.3.1	Sequence read sets data type window	770

9.1.3.2	General sequence read set experiment settings	770
9.1.3.3	Sequence read set comparison settings	772
9.1.4	Importing sequence read sets as files	772
9.1.4.1	Introduction	772
9.1.4.2	Importing sequence read sets from a FASTA file	773
9.1.4.3	Importing sequence read sets from a FASTQ file	773
9.1.4.4	Importing sequence read sets from a .fna and .qual file	774
9.1.4.5	The Import wizard	774
9.2	The sequence read set experiment card	783
9.3	Cluster analysis of sequence read sets	787
9.3.1	Sequence read set comparison settings	787
9.3.2	Sequence read set display settings	789
9.4	Preprocessing sequence read sets	791
9.4.1	Demultiplexing	791
9.4.2	Split paired-end reads	791
9.4.3	Trimming	795
9.4.4	Chimera detection	798
9.4.5	Primer removal	803
9.4.6	Sequence selection	805
9.5	Sequence read set analyses	811
9.5.1	De novo assembly	811
9.5.2	Resequencing assembly	818
9.5.3	Map to reference	824
9.5.4	Run custom template	827
9.5.5	Single-sample diversity analysis	828
9.5.6	Identification against a taxonomic database	832
9.6	Exporting sequence read sets	835
10	Matrix types	837
10.1	Setting up matrix type experiments	839
10.1.1	Defining a new matrix type	839
10.1.2	Matrix comparison settings	839
10.1.3	Importing matrix data	839
10.1.3.1	Introduction	839
10.1.3.2	The Import wizard	840
10.2	Cluster analysis of matrix data	843
10.2.1	Comparison settings	843
10.2.2	Matrix display functions	844
11	Composite data sets	845
11.1	Setting up composite data sets	847
11.1.1	Introduction	847
11.1.2	Defining a new composite data set	847
11.1.3	Composite data set comparison settings	849

11.2 Cluster analysis of composite data sets	851
11.2.1 Principles	851
11.2.2 Calculating a dendrogram from a composite data set	852
11.2.3 Composite data set display functions	857
11.2.4 Finding discriminative characters between entries	857
11.2.5 Transversal clustering	858
 12 Whole Genome Maps	 861
12.1 Background	863
12.2 Setting up Whole Genome Map experiment types	865
12.2.1 Defining a new Whole Genome Map type	865
12.2.2 Importing Whole Genome Map data	866
12.2.2.1 Introduction	866
12.2.2.2 The Import wizard	866
12.2.3 Editing Whole Genome Map experiment type settings	871
12.2.4 The Whole Genome Map experiment card	873
12.3 Whole Genome Maps: display and search options	875
12.3.1 Display settings	875
12.3.2 Fragment annotation: customizable labels	877
12.3.3 Fragment search and selection	879
12.3.3.1 Introduction	879
12.3.3.2 Fragment search	879
12.3.3.3 Fragment filtering	880
12.3.3.4 Activate / Inactivate selected fragments	881
12.3.3.5 Visualization of selected fragments	882
12.3.3.6 Selecting matching fragments	882
12.4 Clustering of Whole Genome Map data	883
12.4.1 Starting up a cluster analysis	883
12.4.2 Whole Genome Map comparison settings	883
12.4.3 Optimization of similarity coefficient parameters	886
12.4.4 Exporting Whole Genome Map information from a comparison	888
12.5 Alignment of Whole Genome Maps	889
12.5.1 Background	889
12.5.2 Pairwise alignment	889
12.5.2.1 Introduction	889
12.5.2.2 Fragment match based alignment of whole genome maps	889
12.5.2.3 Alignment tools	890
12.5.2.4 Pattern match based alignment of whole genome maps	891
12.5.2.5 Pattern match statistics	893
12.5.3 Multiple alignment	894
12.5.3.1 Introduction	894
12.5.3.2 Calculating a multiple alignment	894
12.5.3.3 Alignment tools	895
12.5.3.4 Further analysis	896
12.6 Finding discriminating fragments	897
12.6.1 Background	897
12.6.2 Discriminating fragments for a selection of entries	897

12.6.3	Pattern Match Classes	900
13	Basic cluster analysis	903
13.1	Cluster analysis: an introduction	905
13.1.1	Similarity-based cluster analysis	905
13.1.2	Degeneracy of dendrograms	906
13.1.3	Cluster analysis sensu lato	907
13.2	Comparisons in BioNumerics	909
13.2.1	The Comparison window	909
13.2.2	Adding and removing entries	911
13.2.3	Rearranging entries in a comparison	912
13.2.4	Saving and loading comparisons	912
13.2.5	Experiment type aspects	913
13.2.6	Calculating a dendrogram	914
13.3	General comparison functions	917
13.3.1	Dendrogram display functions	917
13.3.2	Matrix display functions	919
13.3.3	Pairwise comparisons	921
13.3.4	Working with comparison groups	922
13.3.5	Local composite data sets	926
13.3.6	Tree degeneracy	927
13.3.7	Cluster significance tools	930
13.3.7.1	Introduction	930
13.3.7.2	Error flags	930
13.3.7.3	Cophenetic correlation	931
13.3.7.4	Bootstrap analysis	931
13.3.7.5	Cluster cutoff	932
13.3.7.6	Consensus tree	933
13.3.8	Group statistics	933
13.3.8.1	K-means partitioning	933
13.3.8.2	Group separation statistics	935
13.3.9	Printing and exporting a cluster analysis	937
13.3.9.1	Introduction	937
13.3.9.2	The Comparison print preview window	937
13.3.9.3	Printing and exporting options	938
13.3.9.4	Print templates	940
13.3.10	Displaying rendered trees	941
13.3.11	Analysis of the congruence between techniques	942
13.3.12	Identifying unknown entries	946
13.3.13	Exporting data from a comparison	946
14	Charts	949
14.1	Introduction	951
14.2	Chart components	953
14.3	Creating a chart in BioNumerics	955
14.3.1	Introduction	955

14.3.2	Creating charts from the BioNumerics main window	955
14.3.3	Creating charts from the Entry edit window	955
14.3.4	Creating charts from the Pairwise comparison window	956
14.3.5	Creating charts from the Comparison window	956
14.3.6	Create chart dialog box	957
14.3.7	Create chart wizard	958
14.4	Chart types and components	963
14.4.1	Profile chart	963
14.4.1.1	Profile chart type	963
14.4.1.2	Profile chart components	963
14.4.1.3	Profile chart properties	964
14.4.2	Bar Graph	965
14.4.2.1	Bar Graph type	965
14.4.2.2	Bar Graph components	965
14.4.2.3	Bar Graph properties	966
14.4.3	Value histogram	967
14.4.3.1	Value histogram type	967
14.4.3.2	Value histogram components	967
14.4.3.3	Value histogram properties	967
14.4.4	Box and Whiskers chart	968
14.4.4.1	Box and Whiskers chart type	968
14.4.4.2	Box and Whiskers chart components	969
14.4.4.3	Box and Whiskers chart properties	969
14.4.5	Scatter chart	970
14.4.5.1	Scatter chart type	970
14.4.5.2	Scatter chart components	970
14.4.5.3	Scatter chart properties	971
14.4.6	Profile difference chart	973
14.4.6.1	Profile difference chart type	973
14.4.6.2	Profile difference chart components	973
14.4.6.3	Profile difference chart properties	974
14.4.7	3D Scatter chart	974
14.4.7.1	3D Scatter chart type	974
14.4.7.2	3D Scatter chart components	975
14.4.7.3	3D Scatter chart properties	975
14.4.8	Frequency bar graph	976
14.4.8.1	Frequency bar graph type	976
14.4.8.2	Frequency bar graph components	976
14.4.8.3	Frequency bar graph properties	976
14.4.9	Frequency bar graph (colored)	977
14.4.9.1	Frequency bar graph (colored) type	977
14.4.9.2	Frequency bar graph (colored) components	977
14.4.9.3	Frequency bar graph (colored) properties	978
14.4.10	Contingency chart	979
14.4.10.1	Contingency chart type	979
14.4.10.2	Contingency chart components	979
14.4.10.3	Contingency chart properties	980
14.4.11	3D Contingency table	980
14.4.11.1	3D Contingency table type	980
14.4.11.2	3D Contingency table components	981
14.4.11.3	3D Contingency table properties	981
14.4.12	ANOVA chart	982

14.4.12.1	ANOVA chart type	982
14.4.12.2	ANOVA chart components	982
14.4.12.3	ANOVA chart properties	982
14.4.13	Component summary (mean)	984
14.4.13.1	Component summary (mean) type	984
14.4.13.2	Component summary (mean) components	984
14.4.13.3	Component summary (mean) properties	984
14.4.14	Component summary (quantile)	985
14.4.14.1	Component summary (quantile) type	985
14.4.14.2	Component summary (quantile) components	986
14.4.14.3	Component summary (quantile) properties	986
14.4.15	Component summary (range count)	987
14.4.15.1	Component summary (range count) type	987
14.4.15.2	Component summary (range count) components	987
14.4.15.3	Component summary (range count) properties	988
14.4.16	Pie chart histogram table	989
14.4.16.1	Pie chart histogram table type	989
14.4.16.2	Pie chart histogram table components	990
14.4.16.3	Pie chart histogram table properties	990
14.5	Charts window	993
14.5.1	Window layout	993
14.5.2	Editing components of a chart	994
14.5.3	Derived properties	995
14.5.3.1	Introduction	995
14.5.3.2	The Create derived property wizard	995
14.5.3.3	Data Set Derived Properties	996
14.5.4	Changing properties of a chart	1008
14.5.5	Creating fits on charts	1008
14.5.5.1	The Add new fit wizard	1008
14.5.5.2	Available fit models	1010
14.5.6	Displaying and managing multiple charts	1020
14.5.7	Synchronizing selections between chart and database	1020
14.5.8	Working with chart templates	1021
14.5.9	Copying and exporting charts	1022
14.5.10	Pivot tables	1023
14.6	Tutorials	1025
14.6.1	Displaying character sets as bar graph	1025
14.6.2	Plotting densitometric profiles	1026
14.6.3	Plotting a character for multiple entries	1027
14.6.4	Creating a scatter chart for two characters	1031
14.6.5	Creating a contingency table	1031
15	Identification	1035
15.1	Principles	1037
15.2	Fast matching methods	1039
15.2.1	Fast band-based database screening of fingerprints	1039
15.2.2	Fast character-based identification	1041
15.2.3	Fast sequence-based identification	1041

15.3 Identification projects	1043
15.3.1 Creating an identification project1043
15.3.2 Editing an identification project1044
15.4 Classifiers	1047
15.4.1 Creating a new classifier1047
15.4.2 Training classifiers1054
15.4.3 Classifier settings1054
15.4.4 Cross-validation analysis1055
15.4.5 Optimizing classifier parameters1056
15.5 Identifying entries using classifiers	1059
15.6 Decision networks	1067
15.6.1 Introduction1067
15.6.2 Creating a new decision network1067
15.6.3 Operators1069
15.6.4 Building a decision network1069
15.6.5 Display and output options for decision networks1076
15.6.6 Working with layers in a decision network1076
15.6.7 Using confidence values1078
15.6.8 Building decisions relying on multiple states1078
15.6.9 Creating charts from a decision network1080
15.6.10 Executing a decision network from the main window1081
15.6.11 Decision trees1083
16 Advanced cluster analysis	1087
16.1 Background information	1089
16.1.1 Introduction1089
16.1.2 The problem of degeneracy1090
16.1.3 The problem of input data imperfection1091
16.1.4 The resampling framework1091
16.1.5 Data resampling techniques1091
16.1.6 Tree summarizing techniques1093
16.1.7 Trees and networks1093
16.2 The Advanced clustering wizard	1099
16.2.1 Understanding the work flow of the advanced clustering wizard1099
16.2.2 Steps in the advanced clustering wizard1099
16.2.2.1 Basic analysis settings and analysis template1099
16.2.2.2 Input data and treatment1102
16.2.2.3 Network creation algorithm1103
16.2.2.4 Hypothetical nodes1104
16.2.2.5 Priority rules1105
16.2.2.6 Evolutionary model1106
16.2.2.7 Split importance1106
16.2.2.8 Resampling procedure1106
16.2.2.9 Summary creation1108
16.2.3 An example1109
16.3 The Advanced cluster analysis window	1111
16.3.1 Introduction1111

16.3.2	Display settings1111
16.3.2.1	Node labels and sizes1113
16.3.2.2	Node colors1114
16.3.2.3	Branch labels and sizes1115
16.3.2.4	Branch colors1115
16.3.2.5	Branch styles1116
16.3.3	Network editing functions1117
16.3.3.1	Selections on a network1117
16.3.3.2	Creating an unrooted tree1117
16.3.3.3	Creating a rooted tree1118
16.3.3.4	Grouping and ungrouping entries with zero distance1119
16.3.3.5	Optimizing the network layout1120
16.3.3.6	Rotating a network1121
16.3.3.7	Hypothetical nodes1121
16.3.3.8	Parsimonize/Likelihoodize a tree1122
16.3.3.9	Cross links in networks1122
16.3.3.10	Hiding long branches1123
16.3.3.11	Creating partitions1123
16.3.3.12	Creating subnetworks1126
16.3.4	Statistics1127
16.3.5	Analysis templates1128
16.3.6	Analysis functions1129
16.3.7	Exporting and printing the analysis1130
16.4	Tutorials	1133
16.4.1	Introduction1133
16.4.2	UPGMA with error resampling1133
16.4.3	Maximum parsimony with bootstrap resampling1134
16.4.4	Minimum spanning tree with permutation resampling1137
17	Statistics and dimensioning	1141
17.1	Statistics on charts	1143
17.1.1	Introduction1143
17.1.2	Basic terminology1143
17.1.2.1	The use of statistical tests1143
17.1.2.2	Parametric or non-parametric tests1144
17.1.2.3	Categorical or quantitative data1144
17.1.3	Description of tests1144
17.1.3.1	Kolmogorov–Smirnov test for normality1144
17.1.3.2	Parametric test for correlations: Pearson correlation test1146
17.1.3.3	Non–parametric test for correlations: Spearman rank–order correlation test1147
17.1.3.4	T test for paired samples1149
17.1.3.5	Non–parametric test for means: Wilcoxon signed ranks test1150
17.1.3.6	T test for two independent groups1151
17.1.3.7	Mann–Whitney test1151
17.1.3.8	Chi square test for equal category sizes1152
17.1.3.9	Simpson and Shannon–Weiner indices of diversity1153
17.1.3.10	Chi square test for contingency tables1153
17.1.3.11	Parametric test for more than two groups: F test1155
17.1.3.12	Non–parametric test for more than two groups: Kruskal–Wallis test1156
17.1.4	Calculating a statistic on a chart1157

17.2 Multivariate analysis of variance (MANOVA)	1161
17.2.1 What is (M)ANOVA?	1161
17.2.2 ANOVA prerequisites	1162
17.2.2.1 Introduction	1162
17.2.2.2 Use variances only	1163
17.2.2.3 Use variable directions only	1163
17.2.3 Validating ANOVA test results	1163
17.2.3.1 Introduction	1163
17.2.3.2 Testing normality	1164
17.2.3.3 Testing homoscedasticity	1166
17.2.4 Interpreting ANOVA test results	1166
17.2.4.1 Introduction	1166
17.2.4.2 Do the groups have different means?	1167
17.2.4.3 In a two-way ANOVA, how important are the two explanatory variables?	1167
17.2.4.4 What does the separation of groups look like?	1167
17.2.5 Example data set	1167
17.2.5.1 Sample data	1167
17.2.5.2 Creating a new database	1168
17.2.5.3 Creating a new character type experiment	1168
17.2.5.4 Importing data	1169
17.2.6 Performing a MANOVA	1170
17.2.7 The MANOVA window	1173
17.2.7.1 General features	1173
17.2.7.2 Groups	1175
17.2.7.3 Group means	1176
17.2.7.4 Histograms	1176
17.2.7.5 Covariance matrices	1177
17.2.7.6 Testing normality	1178
17.2.7.7 Testing homoscedasticity	1179
17.2.7.8 Test hypothesis	1180
17.2.7.9 Analysis of variance	1180
17.2.7.10 Variable and interaction significance	1180
17.2.7.11 Marginalizing effects	1181
17.2.7.12 Univariate analyses	1181
17.2.7.13 Sum of squares matrices	1181
17.2.7.14 Components	1183
17.2.7.15 Pairwise plots	1183
17.2.7.16 Entry coefficients	1183
17.2.7.17 Component coefficients	1184
17.3 Partition mapping	1187
17.3.1 Introduction and definitions	1187
17.3.2 Example data set	1189
17.3.3 Preparing the database	1190
17.3.4 Performing a partition mapping	1192
17.3.5 The Partition mapping window	1193
17.3.6 Refining a partition mapping	1198
17.3.7 Applying a partition mapping	1199
17.4 Dimensioning techniques	1203
17.4.1 Introduction	1203
17.4.2 Calculating an MDS	1203
17.4.3 Calculating a PCA	1205

17.4.4	Calculating a discriminant analysis	1210
17.4.5	Self-organizing maps	1211
18	High-throughput sequence analysis	1215
18.1	An introduction to the Power Assembler	1217
18.2	Creating a new power assembly	1223
18.3	The Power assembly window	1225
18.3.1	Panel structure	1225
18.3.2	The Project pipeline panel	1226
18.3.2.1	Introduction	1226
18.3.2.2	Working with project templates	1226
18.3.2.3	Adding actions to the project pipeline	1229
18.3.2.4	Removing actions from the project pipeline	1230
18.3.2.5	Changing the action's name and description	1230
18.3.2.6	Loading an existing project structure into the project pipeline	1231
18.3.2.7	Executing actions	1231
18.3.2.8	Aborting the execution of actions	1232
18.3.2.9	Saving the project pipeline	1232
18.3.3	The Report panel	1232
18.3.4	The Action data panel	1234
18.3.5	The Sequence curves panel	1234
18.3.6	The Summary graph panel	1240
18.3.7	The Assembly panel	1242
18.3.8	The Samples panel	1245
18.3.9	The Action design panel	1247
18.3.10	The Execution log panel	1250
18.3.11	Power Assembler general settings	1250
18.3.12	Cleanup project data	1251
18.4	Predefined projects	1253
18.5	Power assembler predefined actions	1255
18.5.1	Import	1255
18.5.1.1	Importing reference sequences	1256
18.5.1.2	Importing read sequences and sequence qualities	1257
18.5.1.3	Importing sample data from TAB-delimited file	1260
18.5.2	Preprocessing	1261
18.5.2.1	Merge samples	1261
18.5.2.2	Demultiplexing	1261
18.5.2.3	Demultiplexing with error correction	1261
18.5.2.4	Split Roche/454 paired ends	1262
18.5.2.5	Set sample entry key from barcode	1263
18.5.2.6	Set entry key from source	1263
18.5.2.7	Set sample entry key from source	1263
18.5.2.8	Set experiment name from source	1263
18.5.2.9	Set sample experiment name from source	1263
18.5.3	Trimming	1263
18.5.3.1	Quality trimming (automatic)	1263
18.5.3.2	Quality trimming	1263
18.5.3.3	Overall quality trimming	1264

18.5.3.4	Tail quality trimming	.1265
18.5.3.5	Length trimming	.1265
18.5.3.6	Remove polyA reads	.1265
18.5.3.7	Remove polyGC reads	.1266
18.5.3.8	Remove reads with long homopolymers	.1266
18.5.4	Statistics	.1266
18.5.4.1	Global reference statistics	.1266
18.5.4.2	Global read statistics	.1267
18.5.4.3	Mapping statistics	.1267
18.5.4.4	Contig statistics	.1267
18.5.5	Mapping	.1267
18.5.6	De novo assembly	.1268
18.5.7	Postprocessing	.1270
18.5.7.1	Determine covered regions	.1270
18.5.8	Export	.1270
18.5.8.1	Export multiple sequence read sets to FASTQ files	.1271
18.5.8.2	Export multiple sequence read sets to database	.1271
18.5.8.3	Export multiple sequences and coverages to database	.1271
18.5.8.4	Export multiple sequences to database	.1271
18.5.8.5	Export sample barcode field to entry information field	.1271
18.5.8.6	Export sample data to tab-delimited file	.1271
18.5.8.7	Export single sequence and coverage to database	.1271
18.5.8.8	Export single sequence to database	.1272
18.6	User-specific actions	1273
18.6.1	Introduction	.1273
18.6.2	The use of action templates	.1274
18.6.3	Creating action templates	.1274
18.6.4	Removing action templates	.1274
18.6.5	Exporting action templates	.1274
18.6.6	Importing action templates	.1275
18.7	Overview of the operators	1277
18.7.1	Introduction	.1277
18.7.2	The operator parameters	.1277
18.7.2.1	Background	.1277
18.7.2.2	General parameters	.1277
18.7.2.3	Background	.1278
18.7.2.4	The data set structure	.1278
18.7.2.5	Data set field types	.1278
18.7.2.6	Data set fields	.1279
18.7.2.7	Sample set fields	.1281
18.7.2.8	Power assembler qualities	.1281
18.7.3	The operator categories	.1282
18.7.4	The operator parameter properties	.1284
18.7.4.1	Displaying parameter properties of an operator	.1284
18.7.4.2	Runtime parameter questioning	.1285
18.7.4.3	Linking parameters to summary graphs	.1288
18.7.4.4	Expressions	.1291
18.7.5	Power assembler operators	.1298
18.7.5.1	Import & Export	.1298
18.7.5.2	Trimming	.1310
18.7.5.3	Preprocessing	.1318

18.7.5.4	Mapping	.1325
18.7.5.5	De novo assembly	.1331
18.7.5.6	Sequence clustering	.1336
18.7.5.7	Region tools	.1337
18.7.5.8	Sequence profiles & curves	.1344
18.7.5.9	Summary graphs	.1349
18.7.5.10	Statistics	.1349
18.7.5.11	Data set tools	.1350
18.7.5.12	Sample tools	.1356
18.7.5.13	Project tools	.1360
19	Metagenomics analysis	1363
19.1	An introduction to metagenomics	1367
19.1.1	Alignment of metagenomics sequences using a reference alignment	.1367
19.1.1.1	Add reference alignment	.1368
19.1.1.2	Edit reference alignment	.1369
19.1.1.3	Remove reference alignment	.1369
19.1.2	Identification of metagenomics sequences using a taxonomy database	.1369
19.1.2.1	Add taxonomic database	.1370
19.1.2.2	Edit taxonomic database	.1372
19.1.2.3	Remove taxonomic database	.1372
19.2	Creating a new metagenomics project	1373
19.2.1	From the main window	.1373
19.2.2	From the Sequence read set experiment window	.1373
19.2.3	From the Metagenomics window	.1374
19.3	Preprocessing analyses for metagenomics data	1375
19.3.1	Chimera detection	.1375
19.3.1.1	Chimera detection : Project element settings	.1379
19.3.2	Primer removal	.1385
19.3.2.1	Primer removal : Project element settings	.1386
19.3.3	Sequence selection	.1388
19.3.3.1	Sequence selection : Project element settings	.1392
19.4	Predefined metagenomics workflows	1395
19.4.1	Identification against a taxonomic database	.1395
19.4.1.1	Input sequences	.1398
19.4.1.2	Phylotype determination: OTU determination by taxonomic identification	.1398
19.4.1.3	Phylotype determination: OTU determination by sequence clustering and taxonomic identification of the OTUs	.1400
19.4.1.4	Save to character set	.1405
19.4.1.5	Identify against taxonomic database : Project element settings	.1406
19.4.2	Single-sample diversity analysis	.1415
19.4.2.1	Input sequences	.1419
19.4.2.2	OTU determination by sequence clustering	.1419
19.4.2.3	OTU determination by taxonomic identification	.1423
19.4.2.4	OTU determination by sequence clustering and taxonomic identification of the OTUs	.1424
19.4.2.5	Single-sample diversity analysis	.1430
19.4.2.6	Single-sample diversity analysis : Project element settings	.1433

19.5 The Metagenomics window	1449
19.5.1 Panel structure1449
19.5.2 The Project panel1450
19.5.2.1 Introduction1450
19.5.2.2 Working with project templates1451
19.5.2.3 Executing the metagenomics analysis projects1453
19.5.2.4 The project settings1453
19.5.3 The Execution log panel1455
19.5.4 The Report list panel1455
19.5.4.1 Working with report templates1456
19.5.5 The Data source overview panel1458
19.5.6 The Report panel1460
19.5.7 Specific data visualization windows1462
19.5.7.1 Data set grid window1462
19.5.7.2 Charts window1463
19.5.7.3 Rich text table window1463
19.5.7.4 Rich text editor window1463
19.5.7.5 Taxonomic tree window1464
 20 Matrix mining	 1469
20.1 An introduction to the Matrix mining window	1471
20.1.1 Introduction1471
20.1.2 The Matrix mining window1471
20.1.3 Discovering the Main view1471
 20.2 General functionality	 1475
20.2.1 Views in the matrix mining1475
20.2.2 Selections in the matrix mining1476
20.2.2.1 Manual selection functions1476
20.2.2.2 Automatic selection functions1476
20.2.3 Groups in the matrix mining1477
20.2.4 Subsets in the matrix mining1480
20.2.5 Scopes and aspects in the matrix mining1481
20.2.5.1 Introduction1481
20.2.5.2 Collapse by field1481
20.2.5.3 Collapse by group1482
20.2.5.4 Transfer text fields1483
20.2.5.5 Transfer selection1483
 20.3 Layers	 1487
20.3.1 Introduction1487
20.3.2 General1487
20.3.3 Data Transformation1487
20.3.3.1 Log Transformation1487
20.3.3.2 Average values1488
20.3.4 Normalization1488
20.3.4.1 Normalize columns/rows1488
20.3.4.2 Quantile Normalization1490
20.3.4.3 Remove effect1490
20.3.5 Filtering1491
20.3.5.1 Clip internal1491

20.3.5.2	Clip external	.1492
20.3.5.3	Absent values	.1493
20.4	Profiles	1495
20.4.1	Introduction	.1495
20.4.2	Creating profiles	.1495
20.4.3	Displaying and handling profiles	.1496
20.4.4	Statistics wizard	.1497
20.4.5	Plot wizard	.1501
20.5	Hierarchical clustering	1503
20.5.1	Introduction	.1503
20.5.2	Calculating a cluster analysis	.1503
20.5.3	Displaying and handling dendrograms	.1507
20.6	Partitioning	1509
20.6.1	Introduction	.1509
20.6.2	Partitioning methods	.1509
20.6.2.1	Manual partitioning	.1509
20.6.2.2	Automatic partitioning	.1509
20.6.3	Calculating a Partitioning	.1510
20.6.4	Partitioning view	.1512
20.6.4.1	Discovering the Partitioning view	.1512
20.6.4.2	Selections and view	.1512
20.6.4.3	Splitting in partitions	.1514
20.6.4.4	Handling cells	.1515
20.6.4.5	Printing and exporting	.1515
20.7	Dimensioning techniques	1517
20.7.1	Introduction	.1517
20.7.2	Calculating a PCA	.1517
20.7.3	PCA view	.1518
20.7.3.1	Discovering the 2D view	.1518
20.7.3.2	Selections and views	.1520
20.7.3.3	Handling components	.1520
20.7.3.4	Printing and exporting	.1521
20.7.3.5	Discovering the 3D view	.1521
20.8	Self-organizing map	1523
20.8.1	Introduction	.1523
20.8.2	Calculating a SOM	.1523
20.8.3	SOM view	.1526
20.8.3.1	Discovering the SOM view	.1526
20.8.3.2	Selections and views	.1526
20.8.3.3	Handling cells	.1527
20.8.3.4	Printing and exporting	.1529
21	Appendix	1531
21.1	Relational database table structure	1533
21.1.1	Introduction	.1533
21.1.2	Table ACTIONS	.1533
21.1.3	Table ALIGNPROJ	.1533

21.1.4	Table ANNOTATION	.1534
21.1.5	Table ANTEMPLATES	.1534
21.1.6	Table ATTACHMENTS	.1535
21.1.7	Table AUTOSQNUMBERS	.1535
21.1.8	Character Values table	.1535
21.1.9	Character Experiments table	.1536
21.1.10	Character Fields table	.1536
21.1.11	Table CHROMOCOMP	.1536
21.1.12	Table CLASSIFIERDATA	.1537
21.1.13	Table CLASSIFIERSETTINGS	.1537
21.1.14	Table COMPAREXTS	.1537
21.1.15	Table COMPARISONS	.1537
21.1.16	Table DBSCHEMAS	.1538
21.1.17	Table DBSETTINGS	.1538
21.1.18	Table DECISNTW	.1538
21.1.19	Table ENLEVELS	.1539
21.1.20	Table ENRELATIONS	.1539
21.1.21	Table ENRELATIONTYPES	.1539
21.1.22	Table ENTRYFLD	.1540
21.1.23	Table ENTRYINFOFIELDS	.1540
21.1.24	Table ENTRYTABLE	.1541
21.1.25	Table ERRORS	.1541
21.1.26	Table EVENTLOG	.1541
21.1.27	Table EXPERATTACH	.1542
21.1.28	Table EXPERIMENTS	.1542
21.1.29	Table FPRBNDCLS	.1543
21.1.30	Table FPRINT	.1543
21.1.31	Table FPRINTFILES	.1544
21.1.32	Table FPRINTREGION	.1545
21.1.33	Table GROUPRIGHTS	.1545
21.1.34	Table MATRIXVALS	.1546
21.1.35	Table OBJACTIONS	.1546
21.1.36	Table OBJQUERIES	.1546
21.1.37	Table OBJSETTINGS	.1546
21.1.38	Table OLIGOSEQ	.1547
21.1.39	Table PAPIPL	.1547
21.1.40	Table PARECSET	.1547
21.1.41	Table PASMBL	.1548
21.1.42	Table REPTEMPLATES	.1548
21.1.43	Table SEARCHDATA	.1549
21.1.44	Table SEQTRACEFILES	.1549
21.1.45	Table SEQUENCES	.1549
21.1.46	Table SIGNATURES	.1550
21.1.47	Table STSETTINGS	.1550
21.1.48	Table SUBSETMEMBERS	.1551
21.1.49	Table SUBSETS	.1551
21.1.50	Table TRENDATA	.1551
21.1.51	Table TRENDXPERS	.1551
21.1.52	Table USERGROUPS	.1552
21.1.53	Table USERKEYS	.1552
21.1.54	Table USERLOG	.1552
21.1.55	Table USERMEMB	.1553

21.1.56 Table USERS1553
21.1.57 Audit trail tables1553
21.1.58 Indices in the database1554
21.1.58.1 Using indices1554
21.1.58.2 ENTRYTABLE1554
21.1.58.3 EXPERIMENTS1554
21.1.58.4 FPRINTFILLES1554
21.1.58.5 FPRINT1554
21.1.58.6 Character values table1554
21.1.58.7 Character fields table1554
21.1.58.8 SEQUENCES1554
21.1.58.9 MATRIXVALS1555
21.1.58.10SUBSETMEMBERS1555
21.2 Regular expressions	1557
21.2.1 Understanding regular expressions1557

NOTES

SUPPORT BY APPLIED MATHS

While the best efforts have been made in preparing this manuscript, no liability is assumed by the authors with respect to the use of the information provided.

Applied Maths will provide support to research laboratories in developing new and highly specialized applications, as well as to diagnostic laboratories where speed, efficiency and continuity are of primary importance. Our software thanks its current status for a part to the response of many customers worldwide. Please contact us if you have any problems or questions concerning the use of BioNumerics[®], or suggestions for improvement, refinement or extension of the software to your specific applications:

Applied Maths NV

Keistraat 120
9830 Sint-Martens-Latem
Belgium
PHONE: +32 9 2222 100
FAX: +32 9 2222 102
E-MAIL: info@applied-maths.com
URL: <http://www.applied-maths.com>

Applied Maths, Inc.

11940 Jollyville Road, Suite 115N
Austin, Texas 78759
U.S.A.
PHONE: +1 512-482-9700
FAX: +1 512-482-9708
E-MAIL: info-US@applied-maths.com

LIMITATIONS ON USE

The BioNumerics[®] software, its plugin tools and their accompanying guides are subject to the terms and conditions outlined in the License Agreement. The support, entitlement to upgrades and the right to use the software automatically terminate if the user fails to comply with any of the statements of the License Agreement. No part of this guide may be reproduced by any means without prior written permission of the authors.

Copyright ©1998, 2018, Applied Maths NV. All rights reserved.

BioNumerics[®] is a registered trademark of Applied Maths NV. All other product names or trademarks are the property of their respective owners.

BioNumerics[®] uses following third-party software tools and libraries:

- The Python[®] 2.7.4 release from the Python Software Foundation (<http://www.python.org/>).
- A library for XML input and output from the Apache Software Foundation (<http://www.apache.org>).
- NCBI toolkit version 2.2.10 (<http://www.ncbi.nlm.nih.gov/BLAST/>).
- The Boost c++ libraries (<http://www.boost.org/>).
- Samtools for interacting with SAM / BAM files (<http://www.htslib.org/download/>)
- The 7-Zip command line version (7za.exe) from 7-Zip, copyright 1999-2010 Igor Pavlov. <http://www.7-zip.org/>
- Velvet for Windows, source code can be downloaded from <http://www.applied-maths.com/download/open-source>.
- Ray for Windows, source code can be downloaded from <http://www.applied-maths.com/download/open-source>.
- Mothur for Windows, source code can be downloaded from <http://www.applied-maths.com/download/open-source>.
- Cairo 2D graphics library version 1.12.14 (<http://cairographics.org/>).
- Crypto++ Library version 5.5.2 (<http://www.cryptopp.com/>).
- libSVM library for Support Vector Machines (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>).
- SQLite version 3.7.17 (<http://www.sqlite.org/>).
- Gecko engine version 21 (<https://developer.mozilla.org/en-US/docs/Mozilla/Gecko>).
- pymzML Python[®] module for high throughput bioinformatics on mass spectrometry data (<https://github.com/pymzml/pymzML>).
- Numpy Python[®] library version 1.8.1 (<http://www.numpy.org/>).
- BioPython Python[®] library version 1.64 (<http://www.biopython.org/>).
- PIL Python library[®] version 1.1.7 (<http://www.pythonware.com/products/pil/>).
- The SPAdes genome assembler version 3.7.1 (<http://bioinf.spbau.ru/spades>).

Part 1

Installation

Chapter 1.1

32-bit or 64-bit installation

From version 7.5 on, is BioNumerics available in a 32-bit and a 64-bit version. All earlier BioNumerics versions were 32-bit applications.

- Running the setup file `bionumerics_7.5_<DetailedVersion>_setup.exe` will install the 32-bit version of BioNumerics on a supported 32-bit or 64-bit Windows operating system.
- Running the setup file `bionumerics_7.5_<DetailedVersion>_setup_x64.exe` will install the 64-bit version of BioNumerics on a supported 64-bit Windows operating system.

On a 32-bit Windows operating system, please install the 32-bit BioNumerics version.

On 64-bit Windows systems, it is recommended to install the 64-bit BioNumerics version. However, since BioNumerics 64-bit is not compatible with MS Access (see [3.7](#)), it is **not recommended** to install the 64-bit version in any of the following scenarios:

- Data from one or more external MS Access databases should be imported in a BioNumerics database on a routine basis.
- When using an MS Access connected database in BioNumerics that is shared by other applications (e.g. data in the database are manipulated directly in MS Access, via a web service, etc.).



Please note that the 64-bit BioNumerics version requires 64-bit ODBC drivers when using relational databases such as MS SQL Server, MySQL or Oracle (see [3.7](#)).

Chapter 1.2

System requirements

1.2.1 Hardware requirements

1.2.1.1 Minimum hardware requirements

The minimum hardware requirements for running the BioNumerics application are the cumulative requirements needed to run the Operating System, the BioNumerics application and any third-party software that will run concurrently (e.g. Microsoft Office).

The typical minimum hardware requirements for a computer running Windows Vista, Microsoft Office 2003 and the BioNumerics application are:

- **Processor:** 1.6 gigahertz (GHz) processor or higher
- **Processor Type:** Intel Pentium Dual Core or higher compatible processor
- **Memory:** 1 GB or higher
- **Hard disk:** 2 GB of free disk space (application files only)
- **Display:** WXGA (1280 x 800) or higher resolution monitor, True Color (32 bit)
- **USB port:** Depending on the license type a free USB port may be required

For *standalone licenses*, each computer that will run BioNumerics must have an available USB port for connecting the Sentinel hardware security key. For *network licenses*, the computer that will be running the NetKey+ server program must have a free USB port for attaching the hardware security key. *Internet licenses* do not require a hardware security key, hence an USB port is not needed.

A 64-bit processor and Windows version are required for systems with more than 4 GB of RAM installed.



The actual hardware requirements will largely depend on the features that will be used in BioNumerics, the database platform used to store the BioNumerics data and the size of the data. For example, the Power Assembler feature of the Sequence data module requires a 64-bit processor and a minimum of 8 GB installed memory.

1.2.1.2 Recommended hardware configuration

The recommended hardware configuration for a computer running the latest Windows and Office versions, and the BioNumerics application are:

- **Processor:** 1.8 gigahertz (GHz) processor or higher
- **Processor Type:** Intel Core 2 Duo Processor or higher compatible processor
- **Memory:** 2 GB or higher
- **Hard disk:** 2 GB of free disk space (application files only), fast hard drive for storing database files (e.g. 7200 RPM SATA drive)
- **Display:** WXGA+ (1440 x 900) or higher resolution monitor, True Color (32 bit), graphics card with dedicated video memory

When purchasing a new computer that will run BioNumerics, make sure that you choose a 64-bit Windows version to allow for future memory expansion. At least 4 GB of RAM should be installed when purchasing a new system.

A recent graphics card with dedicated video memory is recommended. Choosing a basic Windows theme instead of a Windows 7 or Vista Aero theme may be required if the computer is not equipped with sufficient dedicated video memory.



Some features of BioNumerics may require hardware specifications that exceed the above recommendations. For example, the Power Assembler feature of the Sequence data module requires a 64-bit processor, a minimum of 8 GB installed memory and a fast storage system (SSD).

1.2.2 Operating system ---

Generally, Applied Maths will support installing BioNumerics on Windows operating system versions for which the Microsoft Extended Support Phase (see <http://support.microsoft.com/gp/lifeselect>) has not been retired. This will allow you to obtain support and security updates from Microsoft for the target operating system.

- Windows Vista (with Service Pack 2)
- Windows 7
- Windows 8
- Windows 10
- Windows 2008 Server (RTM with Service Pack 2 or R2)
- Windows 2012 Server

Applied Maths recommends installing BioNumerics on a workstation or server with the latest Microsoft service packs installed. BioNumerics can be installed on 64-bit versions of Windows if the WoW64 (Windows 32-bit On Windows 64-bit) subsystem is installed and enabled.

The NetKey+ licensing server program should preferably be installed on a computer running Windows Server 2012 or 2008 with the latest service pack and security updates installed. If a Windows Server computer is not available, then the NetKey+ program can be installed on a Windows 7 or later client operating system.

1.2.3 Microsoft .NET Framework 2.0 SP2

The Microsoft .NET Framework 2.0 Service Pack 2 is required to be able to run the BioNumerics Setup. New installation functions have been added to the AppliedMaths.SetupFramework.dll .NET assembly, and this library requires the Microsoft .NET Framework 2.0 runtime.

The Setup will install the Microsoft .NET Framework 2.0 SP2 on Windows Vista and Windows Server 2008 RTM.

Note that the Setup will attempt to install the Microsoft .NET Framework 3.5 Service Pack 1 Windows feature on Windows 7, Windows Server 2008 R2 and later versions, instead of installing Microsoft .NET Framework 2.0 SP2.

1.2.4 Microsoft .NET Framework 3.5 SP1

Microsoft .NET Framework 3.5 Service Pack 1 is a cumulative update that contains many new features building incrementally upon .NET Framework 2.0, 3.0, 3.5, and includes .NET Framework 2.0 Service Pack 2 and .NET Framework 3.0 Service Pack 2 cumulative updates.

The Setup will attempt to install the Microsoft .NET Framework 3.5 Service Pack 1 Windows feature on Windows 7, Windows Server 2008 R2 and later versions.

1.2.5 Microsoft Visual C++ 2012 Redistributable

The Setup will install the Microsoft Visual C++ 2012 Redistributable Package on the target computer prior to installing any application files. The redistributable is required to be able to run C++ applications like BioNumerics.

On 32-bit computers only the x86 version will be installed. On 64-bit computers the x86 and x64 version of the Microsoft Visual C++ 2012 Redistributable Package will be installed.

1.2.6 Permissions

The user running the BioNumerics Setup package must have full Administrator privileges on the computer(s) where the Setup program will run. In addition the user must have MODIFY NTFS folder permissions and FULL CONTROL share permissions (if applicable) on the database home directory, for example when this folder will be located on a file server and will be accessed via a file share.

1.2.7 Security software

1.2.7.1 Anti-Virus

To optimize the performance of the BioNumerics Setup program it is recommended to temporarily disable the real-time protection or on-access scanning features while running the installer.

Anti-virus software may also affect the performance of the BioNumerics application. If you notice a significant difference in responsiveness when the anti-virus tool is enabled compared to when the tool is disabled, it may be recommended to exclude the anti-virus tool from scanning the BioNumerics executables (bn-start.exe and bn.exe), the DLL and BXT sub-folders and specific file extensions (*.dll, *.mdb, *.bpl) in the

application and database folders.

In addition, the anti-virus software must be properly configured to be compatible with the database platform used to host the BioNumerics databases. Most database software vendors require that the directories containing data and log files are excluded from anti-virus scanning.

1.2.7.2 Firewall and proxy servers

For BioNumerics internet and evaluation licenses, network filtering software and firewall devices may need to be configured to allow access to TCP port 80 on the Applied Maths license servers.

Currently, the following license servers are active to verify internet licenses:

- license1.applied-maths.com (81.246.4.66)
- license2.applied-maths.com (81.246.4.69)
- license3.applied-maths.com (71.42.72.154)
- license4.applied-maths.com (71.42.72.154)

The BioNumerics application requires access to the above internet domain names and public IP addresses to be able to validate internet and evaluation licenses. Note that the IP addresses of the license servers may change in the future, hence firewall exception rules based on the internet domain name should be preferred.

In addition, several BioNumerics plugins require access to specific internet domains to be able to download relevant data:

- .applied-maths.com
- .pubmlst.org (for the *MLST online plugin*)
- .pasteur.fr (for the *MLST online plugin*)
- .mlst.ucc.ie (for the *MLST online plugin*)
- .ridom.de (for the *Spa typing plugin*)

If applicable for your configuration, you may need to grant the BioNumerics application internet access to the above domain names.

If internet access is only allowed through a proxy server, the corresponding settings must be properly configured for the Microsoft Internet Explorer browser (see Figure 1.2.1). The BioNumerics application will use the same settings when connecting to the internet. In other words, if an automatic configuration script (*.pac file) or a static proxy server address has been configured for Internet Explorer, BioNumerics will inherit these LAN settings to connect to the internet.

Network licenses of BioNumerics require that a NetKey+ server has been configured to manage the license sessions. All computers running BioNumerics must be configured to allow access to the listening TCP port on the NetKey+ server computer. Also, the server computer must allow incoming access for the TCP ports used by the NetKey+ server program. For details please check 1.4.

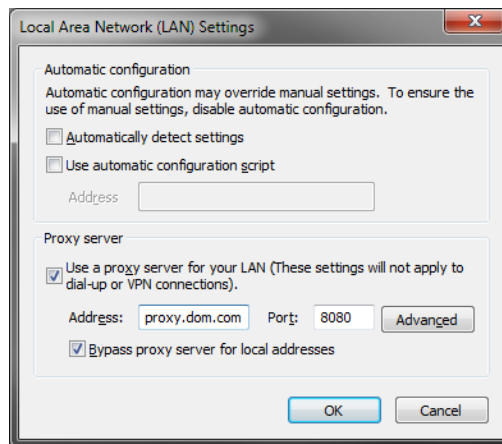


Figure 1.2.1: The *LAN Settings* dialog box.

Chapter 1.3

Installation procedure

1.3.1 Installing a new BioNumerics instance

1.3.1.1 Prerequisites

The *Prerequisites dialog* shows the items that are required to be installed on the local computer before any of the BioNumerics features can be installed (see Figure 1.3.1).

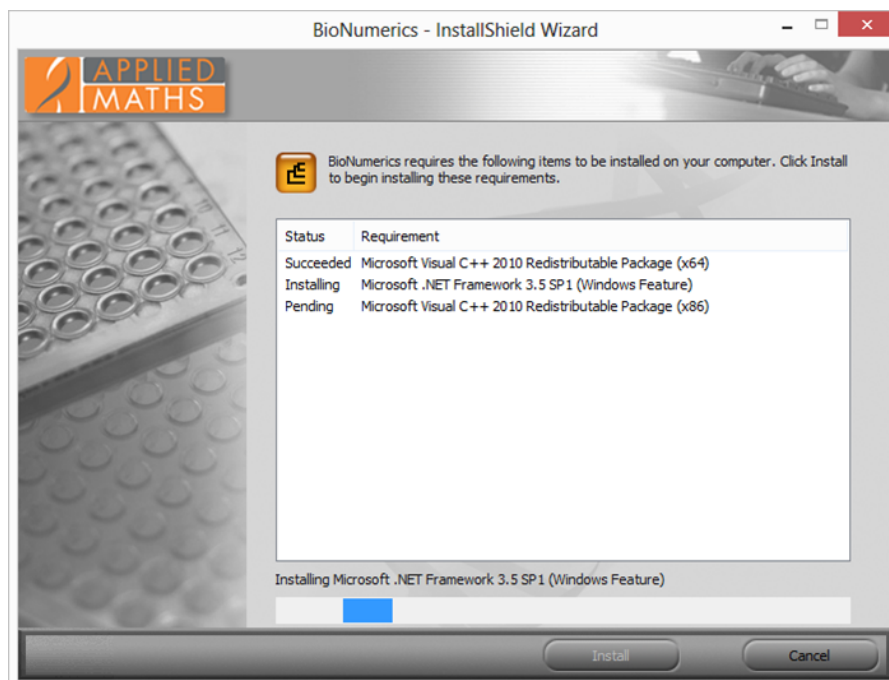


Figure 1.3.1: The *Prerequisites dialog*.

Click on the **<Install>** button to start the installation of the missing prerequisites.



It is recommend to install the Setup prerequisites as described in 1.2 prior to launching the Setup in **Silent installation** mode (see 1.3.6). For example the silent installation will fail if the Setup is not able to download and install the Microsoft .NET Framework 3.5 SP1.

1.3.1.2 Existing instances detected

The BioNumerics 6.5 or later Setup package supports installing multiple instances of the same application side-by-side. Each BioNumerics instance will have a dedicated application installation path, and will have a set of start menu and desktop shortcuts. If an instance of BioNumerics 6.5 or later is already installed then the *Existing Installed Instances Detected* dialog will appear when launching the Setup executable (see Figure 1.3.2).

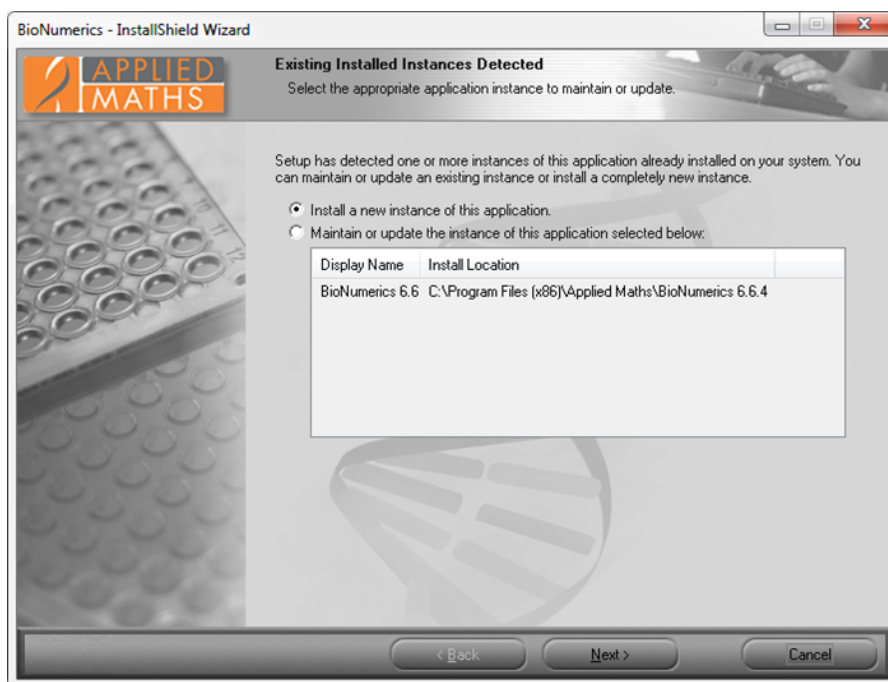


Figure 1.3.2: The *Existing Installed Instances Detected* dialog.

This dialog allows you to choose between installing a new BioNumerics instance, and changing an existing instance. Choose the ***Install a new instance of this application*** option to install a new instance of the BioNumerics application.



The above dialog will not appear if BioNumerics 6.1 or older versions are already installed since these applications were installed with a Setup program that was not yet multi-instance aware. In this case the welcome dialog will be displayed with an update message.

1.3.1.3 Welcome dialog

If no instance of BioNumerics is detected on the local computer, the *Welcome dialog box* will display the version number of BioNumerics that is included with the Setup package when launching the Setup executable. Please verify that you are installing the correct version and click **<Next>** to continue.

1.3.1.4 Software End User License Agreement

The next dialog will display the Software End User License Agreement (EULA) (see Figure 1.3.3). Please read the EULA carefully and click the top ***I accept the terms of the license agreement*** radio button and the **<Next>** button to continue the installation. Click **<Cancel>** if you do not agree with the license agreement; this will abort the installation. The Software End User License Agreement document can be printed to the default printer by clicking the **<Print>** button. The **<Save>** button allows you to browse to a folder where you want to save the Applied Maths EULA.PDF Acrobat document.



Figure 1.3.3: The License Agreement dialog box.

1.3.1.5 Customer information

The *Customer information dialog box* allows you to enter the user and organization names, and the BioNumerics license string (see Figure 1.3.4). You must enter a valid license string to be able to continue with the installation. In addition, the user and organization names cannot be empty. The license string is provided on the sleeve of the installation DVD or you may have obtained it electronically.

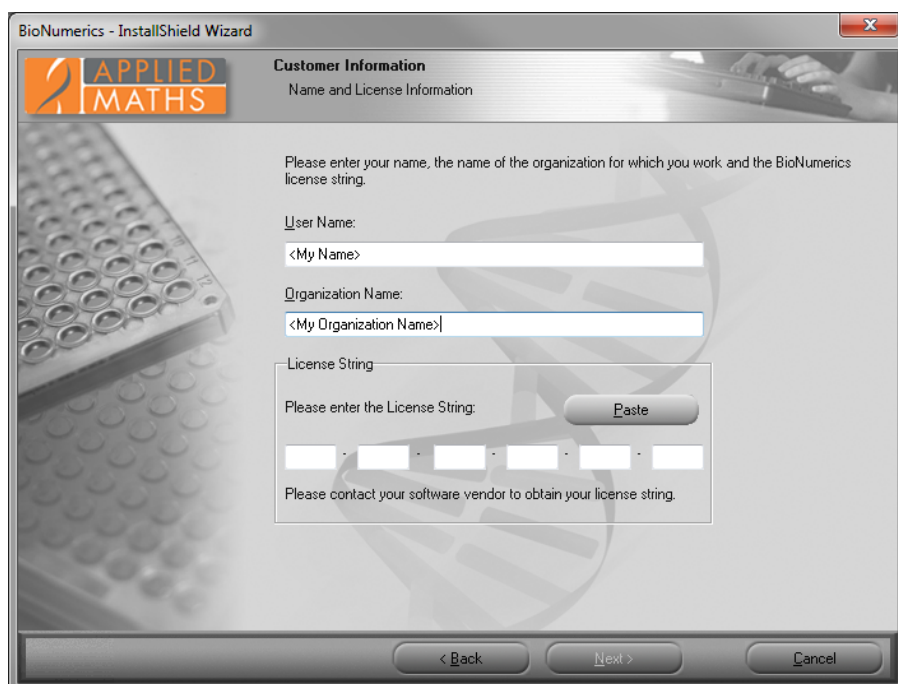


Figure 1.3.4: The Customer Information dialog box.

1.3.1.6 Setup Type

In the *Setup Type* dialog you can choose between a **Default** and a **Custom** setup configuration.

The **Default** setup configuration will install all BioNumerics features with default settings, including the destination paths for the application and data. Note that the **Default** setup configuration does not include the **NetKey+ server program**.

The **Custom** setup configuration allows you to select the features you want to install and choose the target paths for the application and data. The **Custom** setup configuration also allows you to install the **NetKey+ server program** in case of a network license.

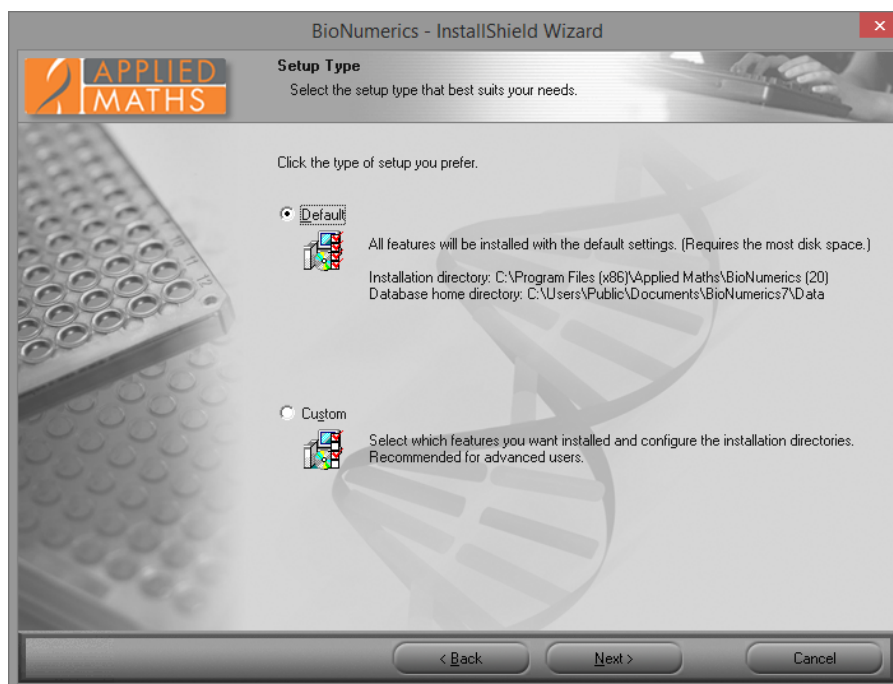


Figure 1.3.5: Choose setup type.

1.3.1.7 Choose destination location

The installation directory for the BioNumerics application and the database home directory can be entered in the *Choose Destination Location* dialog box (see Figure 1.3.6).

The top **<Browse>** button allows you to navigate to a custom installation path for the BioNumerics application. A BioNumerics shortcut will be created on the desktop when the option **Create a BioNumerics shortcut on the desktop** is checked.

Two default locations can be selected for the database home directory: **In Common Documents** and **In My Documents**. The **In Common Documents** option will store the BioNumerics databases in the public documents folder. As a result, the databases will be available to all users on the local computer. The **In My Documents** option will store the BioNumerics databases in the personal documents folder and by default the databases will only be available to the current user.

The third **Custom** option allows you to enter a path on the local computer or even on a remote file server via a permanent network drive. The lower **<Browse>** button will be enabled if the **Custom** radio button has been selected for the database home directory. Note that all BioNumerics users that will access data in the database home directory must have MODIFY NTFS permissions. In addition, the FULL CONTROL permissions must be granted at the file share level when the directory is located on a remote file server.

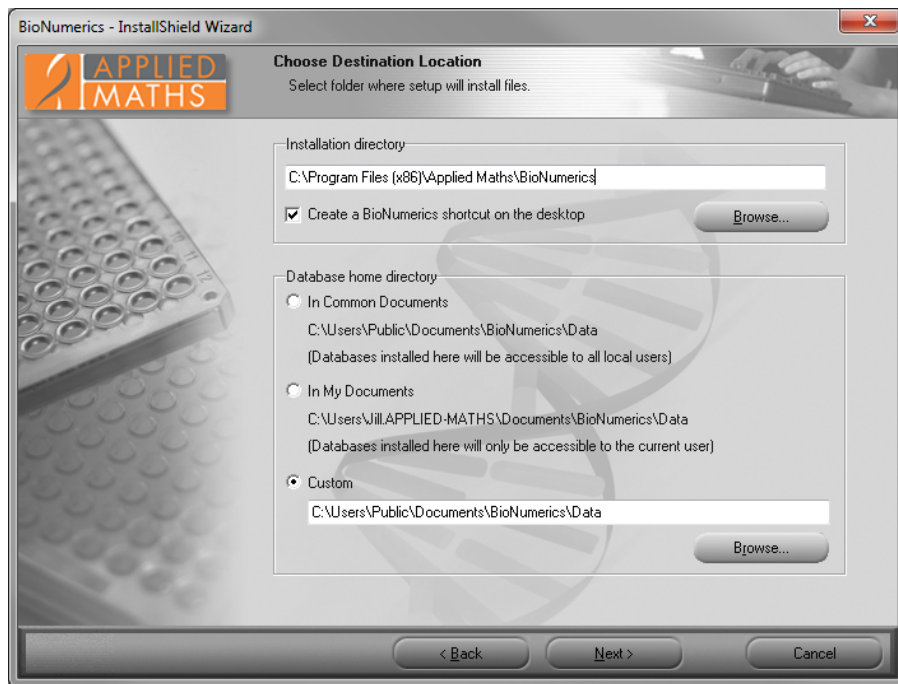


Figure 1.3.6: The *Choose Destination Location* dialog box.

1.3.1.8 Select features

The BioNumerics features that you want to install on the local computer can be selected in the *Select Features dialog box* (see Figure 1.3.7). Clicking on a feature in the left pane will display a short description in the right pane. Tick the appropriate check boxes for the features you want to install.

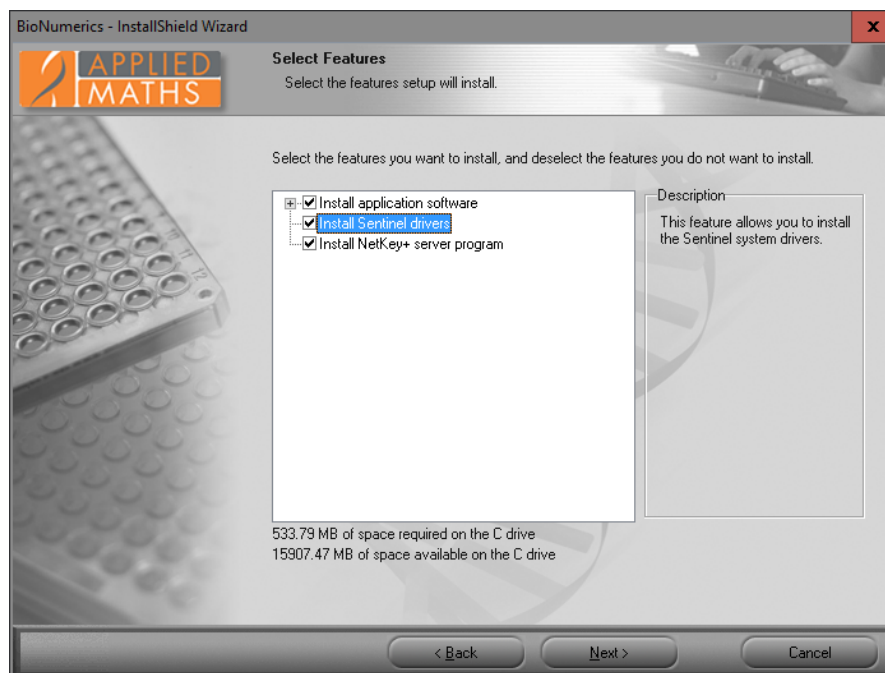


Figure 1.3.7: The *Select Features* dialog box.

Install application software:

- In case of a *standalone license*, the **Application software** needs to be installed on each computer that you want to use to run the software. Please note that only on the computer where the dongle is attached to, you will be able to work with the software.
- In case of an *internet license*, the **Application software** needs to be installed on the computer that you want to use to run the software. Please note that a permanent and stable internet connection is required to run the internet license.
- In case of a *network license*, the **Application software** needs to be installed on the computers in the network that you want to use to run the software.

Install Sentinel drivers:

The **Install Sentinel drivers** feature will install version 7.5.7 of the Sentinel System Driver. In addition this feature will also install the Sentinel Run-time Environment (previously known as HASP) version 6.51 if the NetKey+ server program feature has been selected for installation. The Sentinel Run-time Environment will not be installed if a standalone license string was entered in the *Customer Information dialog box*.

- In case of a *standalone license*, the **Sentinel drivers** need to be installed on each computer that you want to use to run the software.
- In case of an *internet license*, you only need an internet connection to run the software. The **Install Sentinel drivers** option does not need to be checked.
- In case of a *network license*, the **Sentinel drivers** only need to be installed on the NetKey+ server computer in the network.

Install NetKey+ server program:

- The **NetKey+ server program** feature will only be visible and available for installation if a network license string has been entered in the *Customer Information dialog box* (see Figure 1.3.4). The **NetKey+ server program** feature must only be installed on the computer in the network where the hardware security key will be connected to.

A message will appear when selecting the **Sentinel drivers** feature and the minimum required version is already installed (see Figure 1.3.8).

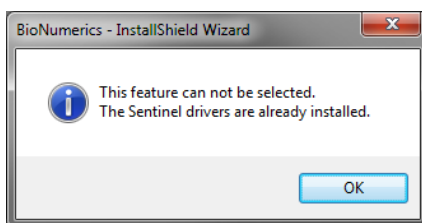


Figure 1.3.8: Sentinel drivers are already installed.

1.3.1.9 NetKey+ connection settings

After pressing the **<Next>** button, the *NetKey+ connection settings dialog box* will appear (see Figure 1.3.9) if a network license string was entered in the *Customer Information dialog box* (see Figure 1.3.4), and if the BioNumerics application feature was selected for installation (see Figure 1.3.7).

The **NetKey+ Server name**, **Server port** and **Admin port numbers** can be entered in the *NetKey+ connection settings dialog box* (see Figure 1.3.9). These parameters will allow the BioNumerics application to connect to the NetKey+ server and request a session for the client computer.

- **NetKey+ Server name:** Name of the computer where the NetKey+ license service is running.
- **Server port number:** TCP listening port number of the NetKey+ service running on the NetKey+ server.
- **Server admin port number:** TCP listening port number for configuring the NetKey+ server. This can be the same number as for the Server port, but to increase security a different TCP port number can be configured for administrating the NetKey+ license server. This way the Windows firewall on the NetKey+ server can be configured to only allow remote NetKey+ administration from specific computers.

Please make sure that you enter available TCP port numbers for the NetKey+ Server and admin ports. The Setup will display the following message if the TCP port is already in use: "TCP port 80 is already in use. Please choose an available TCP port".

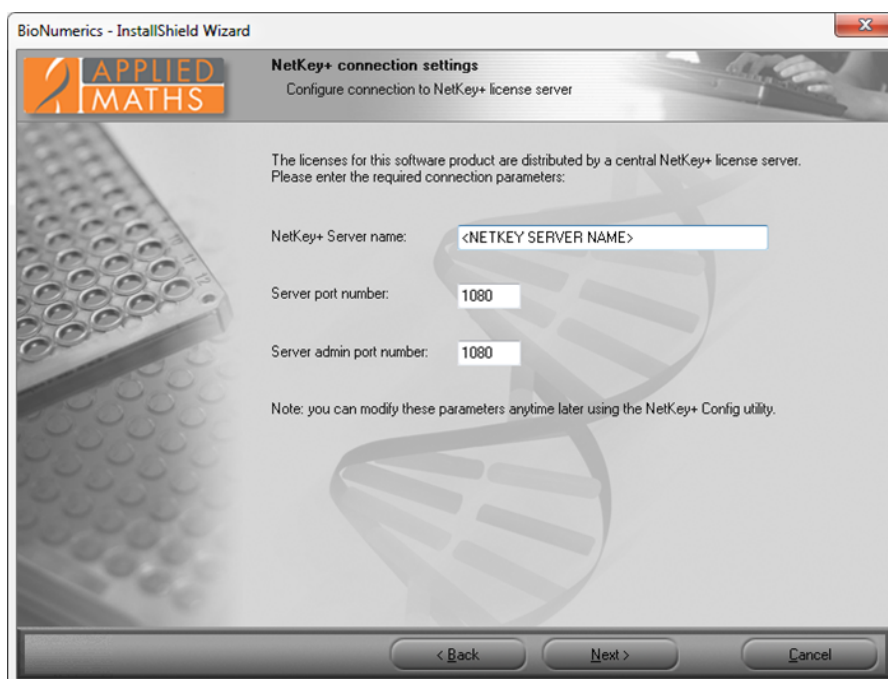


Figure 1.3.9: The *NetKey+ connection settings dialog box*.

After the BioNumerics application has been installed, the Setup will save the server name and TCP port number to the `NetKey.ini` text file on the client computer. The `NetKey.ini` file is located in the folder containing application data for all users (CommonAppDataFolder, for which the path is typically `C:\ProgramData\Applied maths\NetKey+`).

1.3.1.10 Confirm installation

After clicking **<Next>**, the *Ready to install BioNumerics dialog box* will appear. Click **<Install>** to start the installation. The **<Back>** button allows you to review the installation settings, and clicking **<Cancel>** will cause the installation wizard to exit without modifying your system.

The *Setup Status dialog box* will be displayed after clicking the **<Install>** button. This dialog will show the name of the feature that is being installed, and the name of the file that is being copied.

The *Install Complete dialog box* will appear after the installation has finished. Click **<Finish>** to exit the Setup program.

1.3.1.11 NetKey+ configuration

If a network license string has been entered in the *Customer Information dialog box* (see Figure 1.3.4), and the **NetKey+ server program** feature was selected for installation (see Figure 1.3.7), the Setup will ask if you want to run the NetKey+ Configuration tool (see Figure 1.3.10). This tool allows you to install and subsequently start the NetKey+ service. Click **<Yes>** if you want to start the NetKey+ Configuration tool. Click **<No>** if you do not want to specify the NetKey+ settings at this time. More information about the NetKey+ Configuration tool can be found in 1.4.

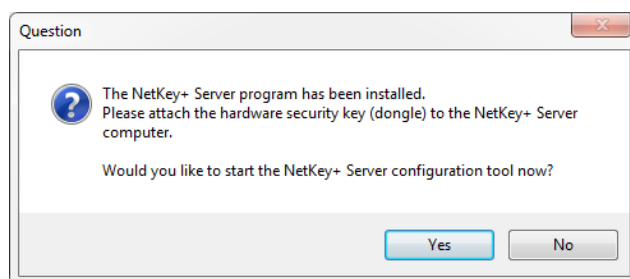


Figure 1.3.10: Launch the NetKey+ Configuration tool.

1.3.1.12 Setup INI XML file

After the dialog sequence, the Setup will record all settings to a Setup INI XML file. This file will be saved to the SetupLogs sub-folder of the BioNumerics installation directory. The file name format is Setup_x_ini.XML, where x is a counter to make sure that the file name is unique in the SetupLogs folder.

Each time the Setup program has been launched, and features were installed or removed, a Setup INI XML file will be created. The file will not be created if the Setup was canceled during the initial dialog sequence.

Please attach the Setup log and INI XML files to your e-mail message when reporting Setup issues to the Applied Maths help desk.

After a manual installation of BioNumerics, the Setup INI XML file can subsequently be used to perform silent installations (see 1.3.6).

1.3.2 Updating a BioNumerics instance

1.3.2.1 Welcome dialog

1.3.2.1.1 Updating a 6.1 or older instance of BioNumerics

If no existing BioNumerics 6.5 or later instances were detected and an older version of BioNumerics was already installed, then the update *Welcome dialog box* will be displayed when launching the Setup executable (see Figure 1.3.11). The *Welcome dialog box* will show the version number of the installed instance of BioNumerics and the version that is included in the Setup package.

Click **<Next>** if you want to update the existing version. If you enter the installation directory of the currently installed version in the *Choose Destination Location* dialog box, then the older version will be replaced by the newer version.



Figure 1.3.11: The *Welcome* dialog.

1.3.2.1.2 Updating a 6.5 or later instance of BioNumerics

If an instance of BioNumerics (version 6.5 or higher) is already installed, then the *Existing Installed Instances Detected* dialog box will appear when launching the Setup executable (see Figure 1.3.12).

Choose the ***Maintain or update the instance of this application selected below*** option to perform an update of the BioNumerics application.

1.3.2.2 Software End User License Agreement

The next dialog will display the Software End User License Agreement (EULA) (see Figure 1.3.13). Please read the EULA carefully and click the top ***I accept the terms of the license agreement*** radio button and the **<Next>** button to continue the installation. Click **<Cancel>** if you do not agree with the license agreement, this will abort the installation. The Software End User License Agreement document can be printed to the default printer by clicking the **<Print>** button. The **<Save>** button allows you to browse to a folder where you want to save the Applied Maths EULA.PDF Acrobat document.

1.3.2.3 Customer information

If you are installing a new major or minor version of BioNumerics, the *Customer Information* dialog box will be displayed after clicking the **<Next>** button (see Figure 1.3.14). This dialog allows you to update the license string for the new version of BioNumerics. By default, a new license string is required for each new minor or major version of BioNumerics. For example, updating version 6.6.4 to 7.0.0 will require a new license string, while updating 6.5.0 to version 6.5.1 will not. You must enter a valid license string to be able to continue with the installation. In addition, the user and organization names cannot be empty.

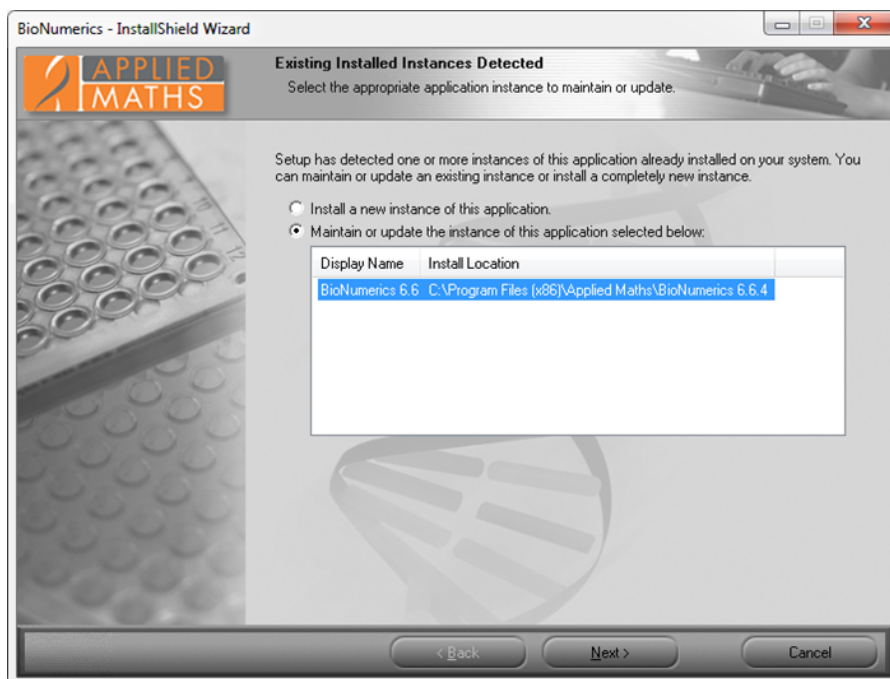


Figure 1.3.12: The *Existing Installed Instances Detected* dialog box.



Figure 1.3.13: The *License Agreement* dialog box.

1.3.2.4 Choose destination location

The *Choose Destination Location* dialog box (see Figure 1.3.15) will only appear when upgrading a BioNumerics version older than 6.5 (see 1.3.2.1.1). If you enter the installation directory of the currently installed version, then this version will be replaced by the newer version.



The *Choose Destination Location* dialog box will not appear when upgrading a BioNumerics 6.5 or newer instance (see 1.3.2.1.2).

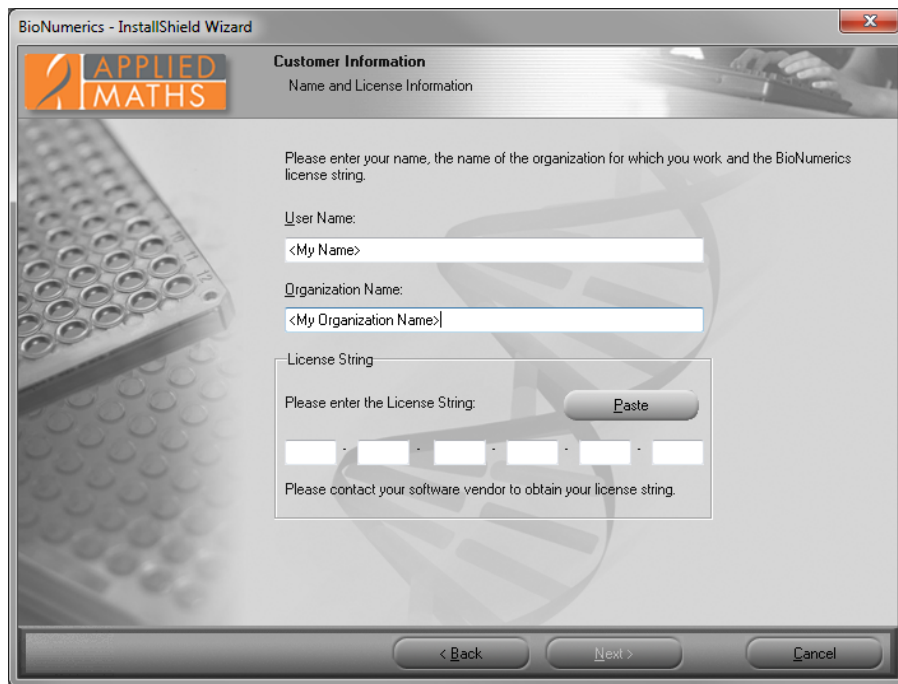


Figure 1.3.14: The *Customer Information* dialog box.

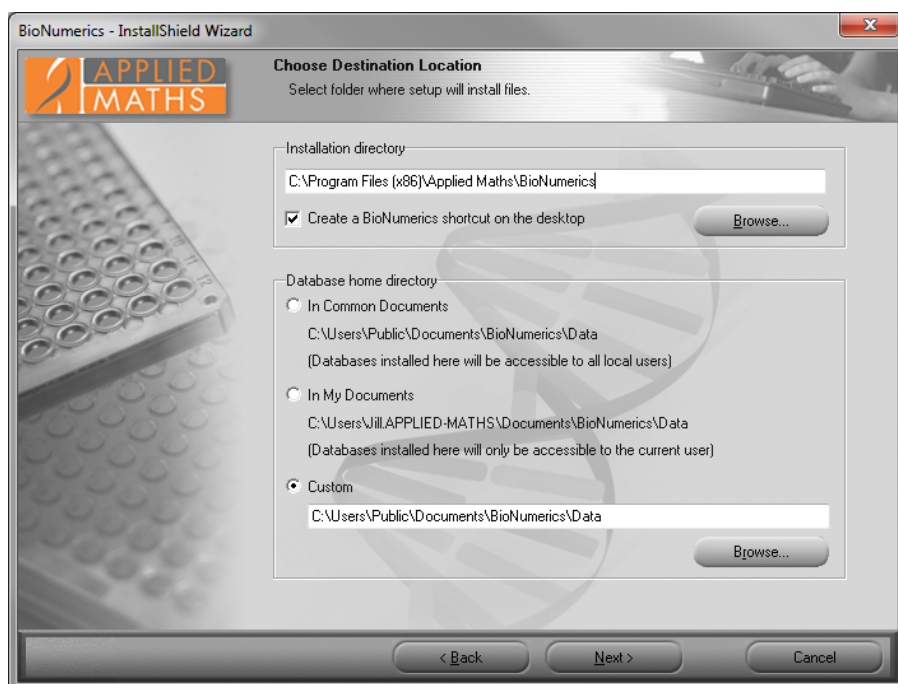


Figure 1.3.15: The *Choose Destination Location* dialog box.

1.3.2.5 Select features

After clicking *<Next>*, the *Select Features* dialog box (see Figure 1.3.16) will be displayed allowing you to choose which features to update or to uninstall. Typically you should accept the default feature selection, or select additional features to install.

Install application software:

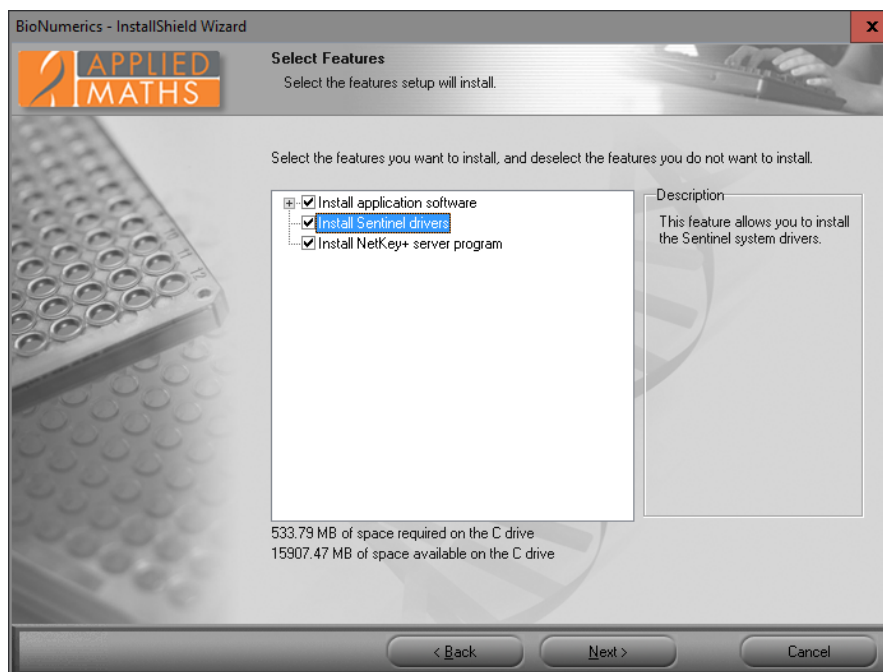


Figure 1.3.16: The *Select Features* dialog box.

- In case of a *standalone license*, the **Application software** needs to be installed on each computer that you want to use to run the software. Please note that only on the computer where the dongle is attached to, you will be able to work with the software.
- In case of an *internet license*, the **Application software** needs to be installed on the computer that you want to use to run the software. Please note that a permanent and stable internet connection is required to run the internet license.
- In case of a *network license*, the **Application software** needs to be installed on the computers in the network that you want to use to run the software.

Install Sentinel drivers:

The **Install Sentinel drivers** feature will install version 7.5.7 of the Sentinel System Driver. In addition this feature will also install the Sentinel Run-time Environment (previously known as HASP) version 6.51 if the NetKey+ server program feature has been selected for installation. The Sentinel Run-time Environment will not be installed if a standalone license string was entered in the *Customer Information dialog box*.

- In case of a *standalone license*, the **Sentinel drivers** need to be installed on each computer that you want to use to run the software.
- In case of an *internet license*, you only need an internet connection to run the software. The **Install Sentinel drivers** option does not need to be checked.
- In case of a *network license*, the **Sentinel drivers** only need to be installed on the NetKey+ server computer in the network.

Install NetKey+ server program:

- The **NetKey+ server program** feature will only be visible and available for installation if a network license string has been entered in the *Customer Information dialog box* (see Figure 1.3.14). The

NetKey+ server program feature must only be installed on the computer in the network where the hardware security key will be connected to.



De-selecting already installed features in the *Select Features dialog box* (see Figure 1.3.16) will cause these features to be uninstalled during the update. A message box will appear if you de-select the main BioNumerics application feature (see Figure 1.3.17). Select **<No>** if you do not want to uninstall the BioNumerics feature.

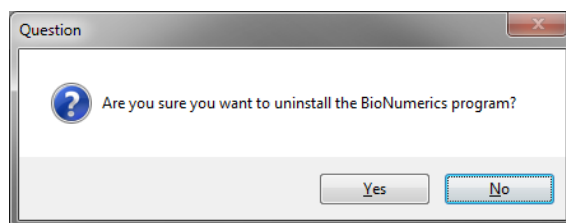


Figure 1.3.17: Warning message.

1.3.2.6 NetKey+ connection settings

After pressing the **<Next>** button, the *NetKey+ connection settings dialog box* will appear if a network license string was entered in the *Customer Information dialog box* (see Figure 1.3.4), and if the BioNumerics application feature was selected for installation (see Figure 1.3.7).

Typically during an update you can accept the **NetKey+ Server name** and **Port numbers** from the previous installation. These parameters will allow the BioNumerics application to connect to the NetKey+ server and request a session for the client computer.

- **NetKey+ Server name:** Name of the computer where the NetKey+ license service is running.
- **Server port number:** TCP listening port number of the NetKey+ service running on the NetKey+ server.
- **Server admin port number:** TCP listening port number for configuring the NetKey+ server. This can be the same number as for the Server port, but to increase security a different TCP port number can be configured for administrating the NetKey+ license server. This way the Windows firewall on the NetKey+ server can be configured to only allow remote NetKey+ administration from specific computers.

1.3.2.7 Confirm update

Click **<Next>** to start the update. The *Setup Status dialog box* will be displayed. Newer files will be copied to the target system for the selected features. Any feature that was de-selected will cause the corresponding files and shortcuts to be uninstalled.

The *Update Complete dialog box* will appear after the update has finished. Click **<Finish>** to exit the Setup program.

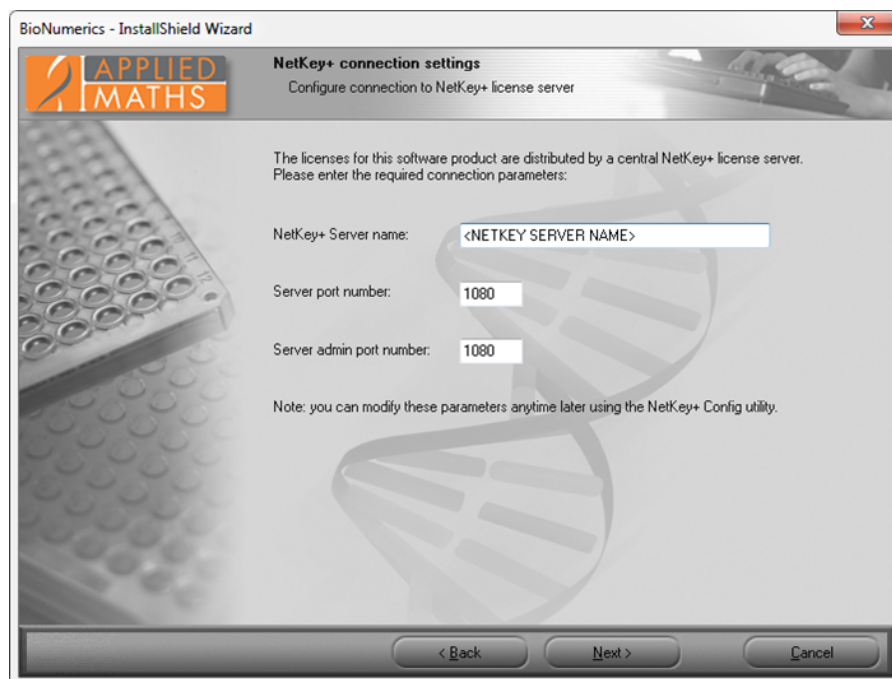


Figure 1.3.18: The *NetKey+ connection settings* dialog box.

1.3.3 Maintenance installation

1.3.3.1 Select instance to maintain

If an instance of BioNumerics (version 6.5 or higher) is already installed, then the *Existing Installed Instances Detected* dialog box will appear when launching the Setup executable (see Figure 1.3.19).

This dialog allows you to choose between installing a new BioNumerics instance, or changing an existing instance. Choose the ***Maintain or update the instance of this application selected below*** option to perform a maintenance of the BioNumerics application.

1.3.3.2 Maintenance options

After selecting the BioNumerics instance that needs to be modified, the *Welcome* dialog box will display the maintenance options (see Figure 1.3.20).

- **Modify:** Select **Modify** to install a feature that was not installed during the previous installation (see 1.3.3.3).
- **Repair:** Choose **Repair** to repeat the previous installation of the BioNumerics application. The same features selected during the previous setup will be re-installed (see 1.3.3.4).
- **Remove:** Choose **Remove** to remove all BioNumerics files and shortcuts that were created during previous installations of the selected BioNumerics instance (see 1.3.3.5).

1.3.3.3 Modify maintenance mode

The *Customer Information* dialog box will appear after selecting the **Modify** option and clicking **<Next>** in the *Welcome* dialog box (see Figure 1.3.20). This dialog allows you to update the user and organization

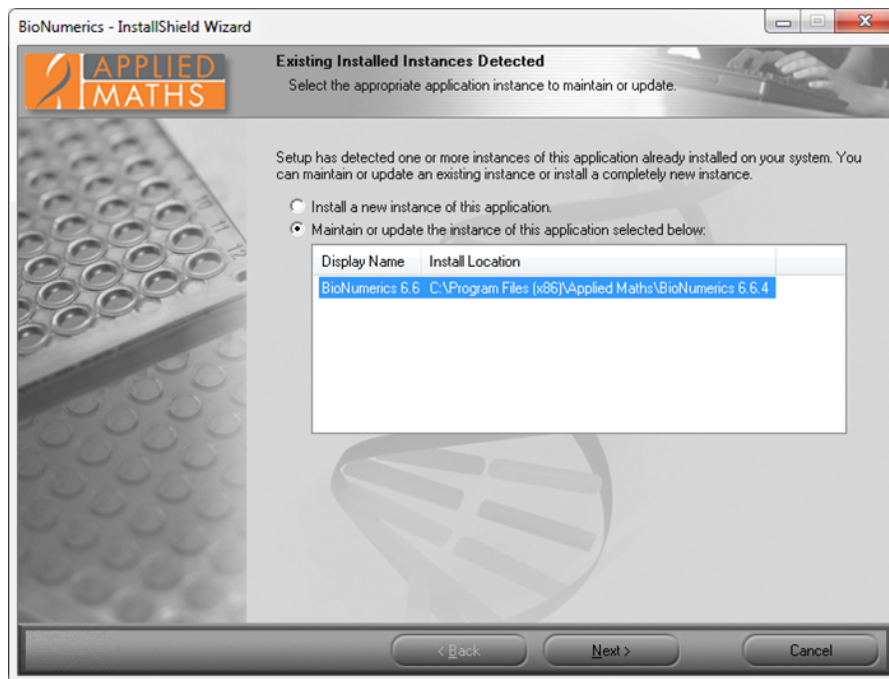


Figure 1.3.19: The *Existing Installed Instances Detected* dialog box.

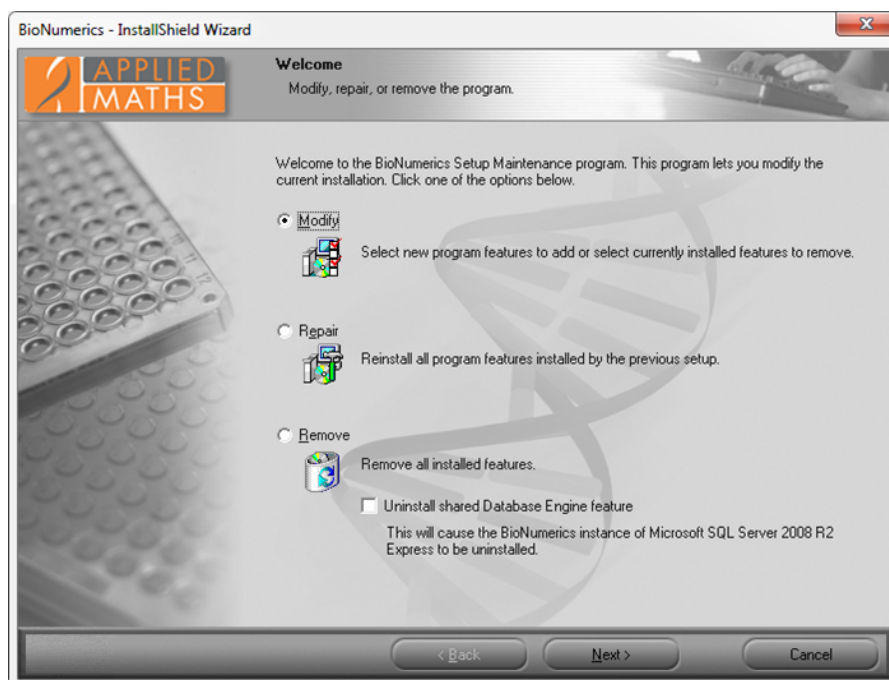


Figure 1.3.20: The *Welcome* dialog box.

names, and the BioNumerics license string. You must enter a valid license string to be able to continue with the installation.

Next, the *Select Features dialog box* will be displayed, allowing you to choose which features to install or to uninstall.



De-selecting already installed features in the *Select Features dialog box* will cause these features to be uninstalled during the update. A message box will appear if you de-select the main BioNumerics application feature. Select **<No>** if you do not want to uninstall the BioNumerics application.



The recommended method for uninstalling an instance of BioNumerics is to choose the **Remove** option in the *Welcome dialog box* (see Figure 1.3.20). De-selecting the BioNumerics application feature in the **Modify** maintenance mode will uninstall the application, but will not delete any uninstall information from the registry and file system. A message box will appear asking you to confirm that you want to uninstall the BioNumerics application. Other features that remained selected, like the NetKey+ server features, will not be removed from the target system.

After pressing **<Next>** the *NetKey+ connection settings dialog box* will appear if a network license string was entered in the *Customer Information dialog box*, and if the BioNumerics application feature was selected for installation in the *Select Features dialog box*.

Click **<Next>** to start applying the changes. Files will be copied to the target system for new features that have been selected. Any feature that was de-selected will cause the corresponding files and shortcuts to be uninstalled.

The *Maintenance Complete dialog box* will appear after all changes have been executed. Click **<Finish>** to exit the Setup program.

1.3.3.4 Repair maintenance mode

After choosing the **Repair** option in the *Welcome dialog box* (see Figure 1.3.20) and clicking **<Next>**, the Setup program will re-install all features that were selected during the previous installation. All corresponding files, shortcuts and registry settings will be re-created on the computer where the Setup is running.

If a network license string has been entered, and the **NetKey+ server program** feature was selected for installation, the Setup will ask if you want to run the NetKey+ Configuration tool. This tool allows you to connect to the NetKey+ server to verify and update the license information. In addition, the tool allows you to repair the NetKey+ service (see 1.4.7 and 1.4.8). Click **<Yes>** if you want to start the NetKey+ Configuration tool. Click **<No>** if you do not want to change the NetKey+ settings at this time. More information about the NetKey+ Configuration tool can be found in 1.4.

The *Maintenance Complete dialog box* will appear after all changes have been executed. Click **<Finish>** to close the Setup program.

1.3.3.5 Remove maintenance mode

The **Remove** option in the *Welcome dialog box* (see Figure 1.3.20) allows you to completely uninstall the selected instance of BioNumerics. All BioNumerics files and shortcuts that were created during previous installations of the selected BioNumerics instance will be deleted. In addition, the uninstall information for the selected instance will be removed from the computer.

A confirmation dialog will appear, asking you to confirm the removal of the selected BioNumerics instance (see Figure 1.3.21). Click **<Yes>** to start the deletion of the BioNumerics application. Select **<No>** to return to the previous *Welcome dialog box*.



Completely uninstalling an instance of BioNumerics which includes the NetKey+ server program may affect other BioNumerics users that use the corresponding NetKey+ service to request license sessions. Make sure that no other users are using the NetKey+ service prior to uninstalling the NetKey+ server program feature, or completely uninstalling the BioNumerics instance.

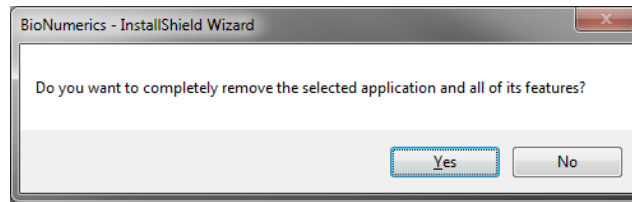


Figure 1.3.21: Confirm removal of selected features.

The *Uninstall Complete dialog box* will be displayed after the selected BioNumerics instance has been removed. Click the <**Finish**> button to exit the Setup program.



The Setup will not delete BioNumerics program folder because it contains the SetupLogs sub-folder holding the log files for each Setup that has been run. Also any file that has been added to the program folder, and which was not originally installed by the Setup program, will not be deleted from the hard drive.

1.3.4 Installing Protection Keys

1.3.4.1 Protection Key Types

Starting from BioNumerics version 7.0 the NetKey+ server supports two types of SafeNet protection keys:

- **SentinelSuperPro** provider: hardware-based Sentinel SuperPro USB protection key. The Sentinel USB dongle is used to protect standalone and network licenses of BioNumerics running on computers either equipped with a physical USB port, or with a network-attached USB hub. The USB dongle has been tested with network-attached USB hubs from Digi (AnywhereUSB) and Silex (USB Device Servers).
- **SentinelHasp** provider: software based Sentinel HASP protection keys. The software-based Sentinel HASP SL key is used to protect network licenses of BioNumerics, more specific to provide a software protection key for the NetKey+ license server program running on a computer that is not equipped with a free physical USB port. This is particularly useful if the NetKey+ license service is running on a virtualized operating system and a network-attached USB hub is not available.

1.3.4.2 Install Protection Key Driver

The BioNumerics Setup includes the latest version of the SafeNet drivers available at the time of the product release. When installing older BioNumerics versions it is recommended to download and install the latest version of the SafeNet driver before attaching the USB dongle.

The drivers for the Sentinel USB dongle can be downloaded from the following web site: <http://www.applied-maths.com/sentineldriver>.

The Sentinel Run-time Environment for the HASP SL or HL protection keys can be downloaded from the following web site: <http://www.applied-maths.com/haspdriver>.

The above URLs will redirect you to the appropriate download page on the SafeNet Sentinel customer web site.

After installing the drivers and connecting the USB dongle, the protection key should appear under **Universal Serial Bus controllers** in the Windows device manager (see Figure 1.3.22).

The Windows device manager can be accessed via "Control Panel > System and Security > Administrative Tools > Computer Management".

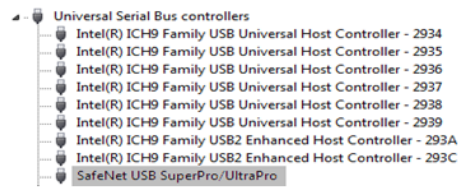


Figure 1.3.22: Universal Serial Bus controllers.

If the USB dongle is not listed in Windows device manager then download and install the latest version of the driver from the SafeNet web site and reboot the computer. Please contact the Applied Maths support team (info@applied-maths.com) if Windows still is unable to detect the protection key after reboot.

1.3.4.3 Activate Sentinel HASP SL key

1.3.4.3.1 Introduction

The first step in installing a software-based Sentinel HASP SL key is adding the license string using the NetKey+ configuration tool. If the added license string corresponds with a software lock protection key then the **<Activate>** button (Figure 1.3.23) will be available, which allows downloading and installing the SentinelHasp key on the NetKey+ server computer. If the license key with the *SentinelHasp* provider is already listed in the NetKey+ configuration tool on the NetKey+ server then the software lock (SL) key is already activated.

Clicking the **<Activate>** button will display the *Activate Hasp license dialog* (see Figure 1.3.23). It is recommended to activate the software lock (SL) key using automatic activation. This requires an active internet connection on the computer running the NetKey+ configuration tool.

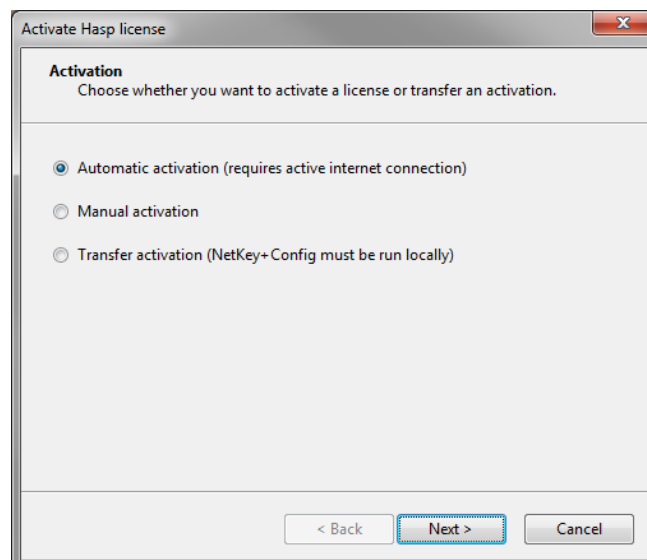


Figure 1.3.23: Activate Hasp license.

This dialog enables an Administrator to perform an automatic or manual activation of a HASP software lock (SL) license string, or to transfer an existing protection key to another computer. Note that the NetKey+ server never needs an active internet connection for the activation, an active internet connection is recommended for the NetKey+ configuration tool however.

1.3.4.3.2 Automatic Activation

It is recommended to activate the software lock (SL) key using automatic activation. This requires an active internet connection on the computer running the NetKey+ configuration tool. Note that the SL key can only be activated once, however it is possible to transfer the lock to a different computer afterwards (see 1.3.4.3.4).

Click <**Next**> to display the *Customer details dialog*, and enter the contact person's name, email address and organization name you want to use to register the software activation. If possible please use the contact details of the person who ordered the software at Applied Maths NV.

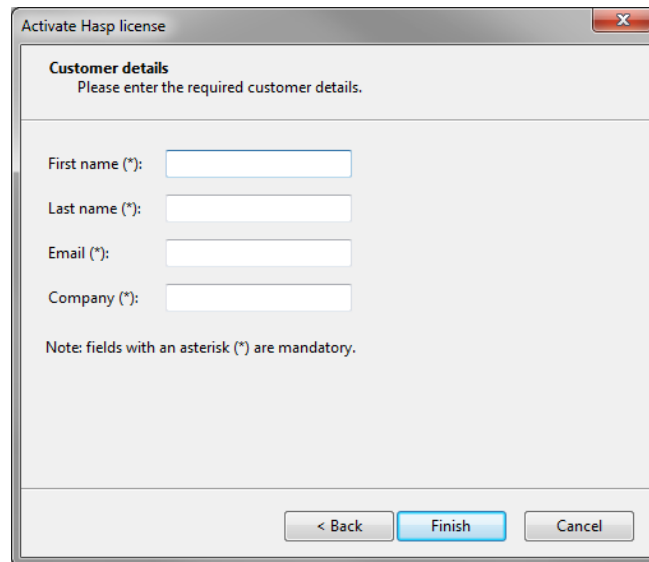


Figure 1.3.24: The *Customer details dialog*.

Click <**Finish**> to start the activation process. The NetKey+ configuration tool will connect to a secured license server to check if there is a SentinelHasp soft lock protection key available for the entered license string. If a protection key is available the NetKey+ configuration tool will connect to a secured activation server to upload a fingerprint of the NetKey+ computer, and subsequently download and install the corresponding soft lock key. Hence the computer running the activation process must be able to access the following web sites on the internet:

- <https://ssllicense.applied-maths.com>: Secured License Server
- <https://activate.applied-maths.com>: Secured Activation Server

Select **Server** in the left panel of the NetKey+ configuration tool. If the automatic activation was successful the software-based protection key with **SentinelHasp** as the provider should appear within a minute or so in the list of available license keys.

Note that the installed SentinelHasp soft lock key is only valid for a specific target computer, and can by default only be activated once. Afterwards the protection key can be moved to another NetKey+ server, for example when installing a new NetKey+ server computer.

If an error message appears during the activation process, you can look up the NetKeyConfigLog.txt log file in your temp folder, and send the file as an email attachment to activate@applied-maths.com. If receiving the vendor-to-customer (v2c) file from the Applied Maths activation server succeeded, but applying it to the NetKey+ server failed, a backup v2c file is created in the temp folder, with a name formatted like NetKeyConfig_autoActivate_backup_#.v2c. The activation can then be completed manually by using this file and the **Activate with confirmation file** option in the *Manual Activation dialog*.

- License Activation log file path: temp \NetKeyConfigLog.txt
- License Activation backup v2c file: temp \NetKeyConfig_autoActivate_backup_0.v2c



A complete system or full backup scheme must be in place to protect the NetKey+ license server where a soft lock key has been activated. Changing the hardware configuration (e.g. MAC address, CPU, hard drive) will cause the protection key to render invalid; hence the protection key must be transferred to another (intermediate) computer before modifying the hardware, and transferred back to the source computer after the hardware component(s) have been replaced. This also applies to virtual environments, for example moving a virtual NetKey+ server guest image to another host server may invalidate the protection key. Hence the key must be transferred to another (intermediate) computer before moving the guest image to another host server, and transferred back after the virtual guest image has been moved. In case of doubt please contact the support team before changing the hardware configuration of a NetKey+ server that contains a SentinelHasp soft lock protection key.

1.3.4.3.3 Manual Activation

If no internet connection is available on the NetKey+ server computer or on any of the computers where the NetKey+ configuration tool is installed, then the **Manual Activation option** can be selected in the *Activate Hasp license dialog* (see Figure 1.3.23 (It is recommended to activate the software lock (SL) key using automatic activation. This requires an active internet connection on the computer running the NetKey+ configuration tool).

Click <Next> to display the *Manual activation dialog*.

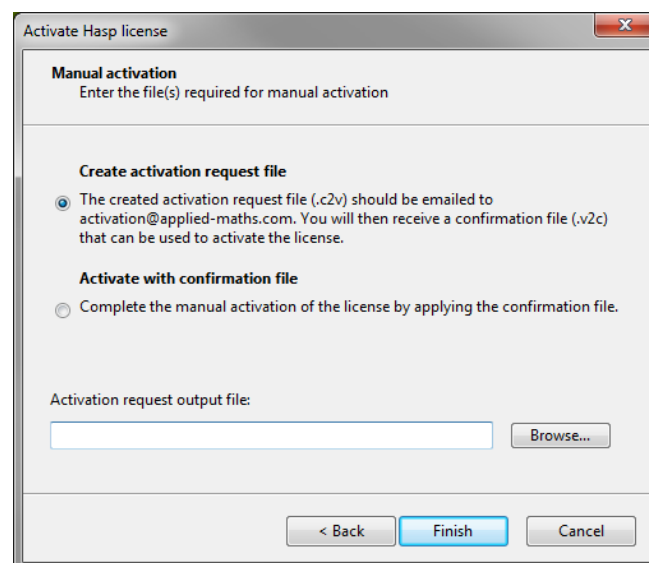


Figure 1.3.25: The *Manual activation dialog*.

Select the **Create activation request file** option in the *Manual Activation dialog*.

Click <Browse> to enter the path and file name for the activation request file.

Click <Finish> to save the customer-to-vendor (*.c2v) activation request file, and include the file as an email attachment and send an email to activation@applied-maths.com. If your email system does not allow sending *.c2v files you can change the file extension to *.txt.

After receiving the vendor-to-customer (*.v2c) confirmation file from Applied Maths the activation process can be completed:

Select the **Activate with confirmation file** option in the *Manual Activation dialog*.

Click <**Browse**> to select the *.v2c confirmation file, and click <**Finish**> to install the SentinelHasp soft lock key on the NetKey+ server;

Select **Server** in the left panel of the NetKey+ configuration tool. If the manual activation was successful the software-based protection key with **SentinelHasp** as the provider should appear within a minute or so in the list of available license keys.

1.3.4.3.4 Transfer Sentinel HASP SL key

To be able to transfer a software-based protection key the NetKey+ server and the configuration tool must be installed on both the source and destination computers, and the NetKey+ configuration tool must be started locally on both computers. A license key with the **SentinelHasp** provider must be listed in the NetKey+ configuration tool on the source NetKey+ server to be able to transfer the key to another NetKey+ server computer.

Transferring the Sentinel HASP SL key is a three-step process:

The first step is creating the protection key request file on the target computer:

1. Start the NetKey+ configuration tool on the target computer.
2. Select **Licenses** in the left pane, select the license string.
3. Click the <**Activate**> button.
4. Select **Transfer activation** in the *Activate Hasp license dialog* (see Figure 1.3.26 and click <**Next**>).
5. In the top **Computer** section select **NetKey+ is running on the destination computer**.
6. In the **Step** section select **Step 1: Create request file - Destination computer**.
7. Click the <**Browse**> button to enter the path and file name of the protection key request file.
8. Click <**Finish**> to save the protection key request file.

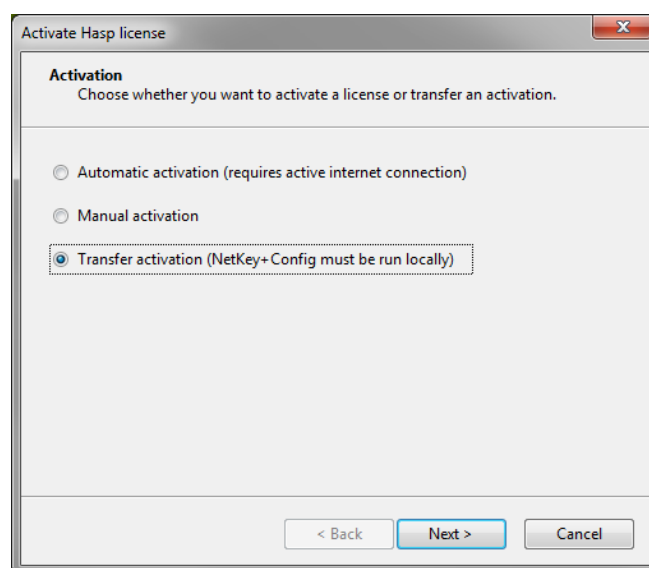


Figure 1.3.26: Transfer activation.

The second step is creating the protection key transfer file on the source computer:

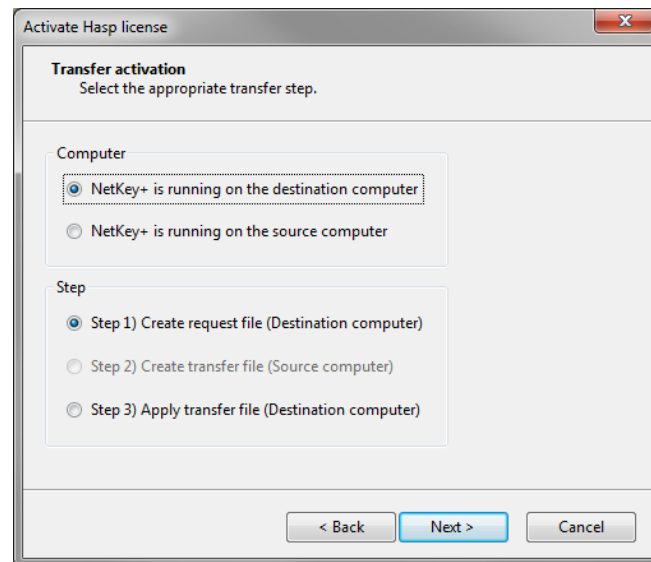


Figure 1.3.27: Transfer activation.

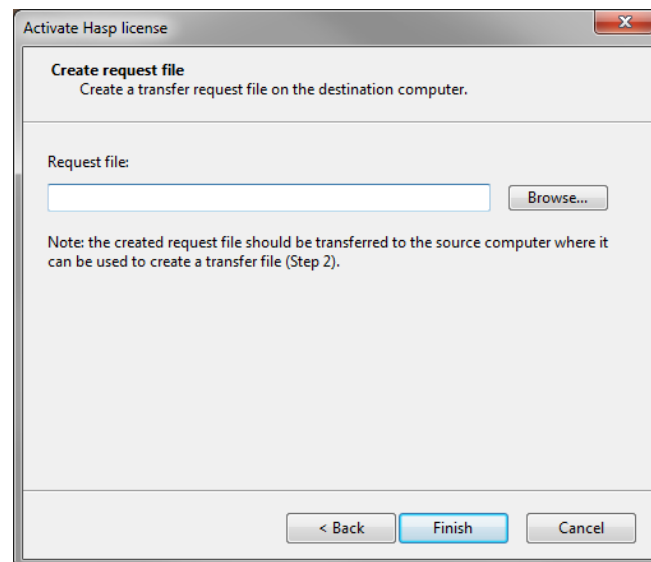


Figure 1.3.28: Create request file.

1. Copy the protection key request file from the target to the source computer.
2. Start the NetKey+ configuration tool on the source computer.
3. Select **Licenses** in the left pane, select the license string.
4. Click the <**Activate**> button.
5. Select **Transfer activation** in the *Activate Hasp license dialog* (see Figure 1.3.26, and click <**Next**>).
6. In the top **Computer** section select **NetKey+ is running on the source computer**.
7. In the **Step** section select **Step 2: Create transfer file - Source computer**.
8. Click the first <**Browse**> button to select the *.id protection key request file.
9. Click the second <**Browse**> button to enter the path and file name of the *.v2c transfer file.

10. Click **<Finish>** to remove the protection key from the local computer and to save the *.v2c transfer file.

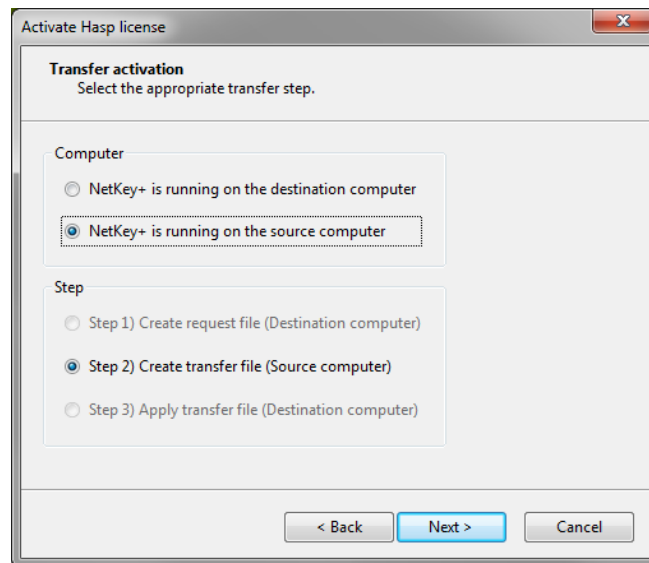


Figure 1.3.29: Transfer activation.

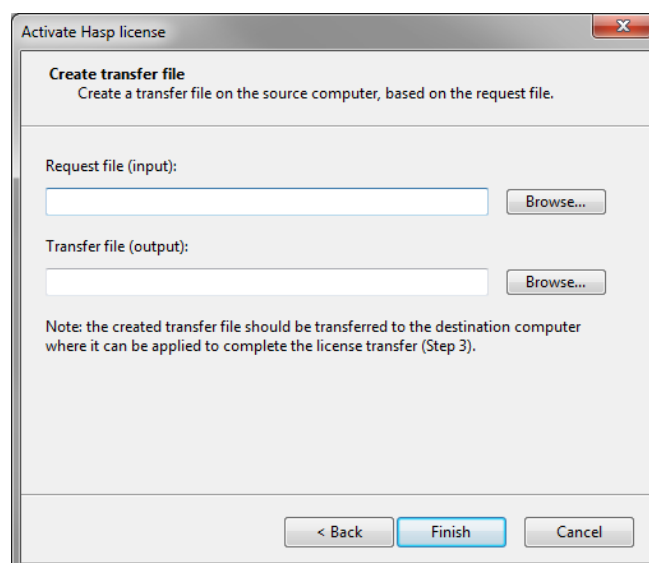


Figure 1.3.30: Transfer activation.



Upon completion of this step the software lock is effectively removed from the source computer. This means that the license connected to this lock will be deactivated. The software lock has not been transferred to the target computer yet however, at this point the lock can be thought of as "residing in the transfer file". Until the transfer file has been applied on the target computer it is therefore crucial not to accidentally remove it. As a back-up measure a copy of the transfer file is stored in the temp folder, with name e.g. NetKeyConfig_transferFile_backup_0.v2c.

The third and last step is to apply the protection key transfer file on the target computer:

1. Copy the *.v2c transfer file from the source to the target computer.
2. Start the NetKey+ configuration tool on the target computer.

3. Select **Licenses** in the left pane, select the license string.
4. Click the <**Activate**> button.
5. Select **Transfer activation** in the *Activate Hasp license dialog* (see Figure 1.3.26, and click <**Next**>.
6. In the top **Computer** section select **NetKey+ is running on the destination computer**.
7. In the **Step** section select **Step 3: Apply transfer file - Destination computer**.
8. Click <**Browse**> and browse to path where the copied *.v2c transfer file is located and select the file.
9. Click <**Finish**> to install the protection key on the local computer.

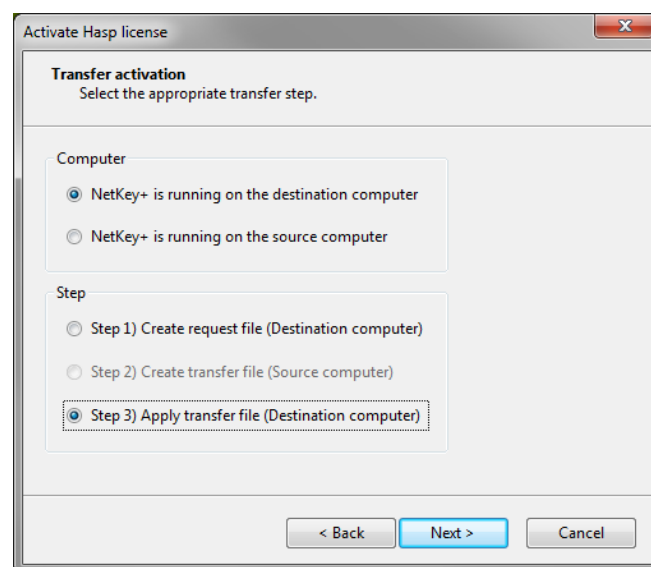


Figure 1.3.31: Transfer activation.

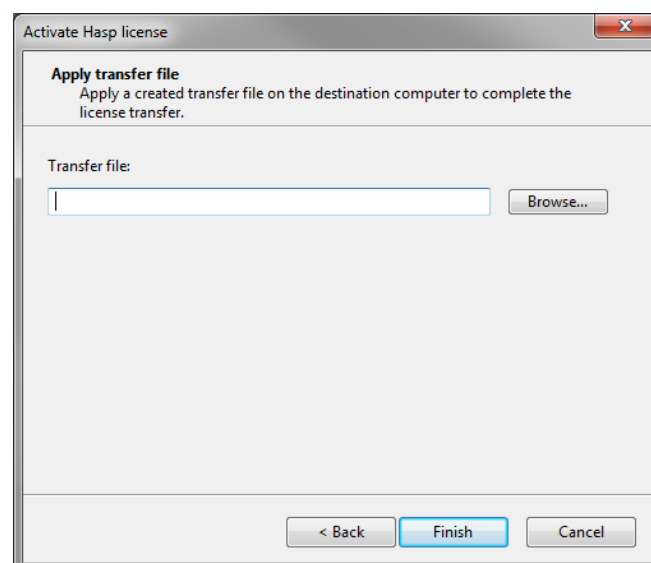


Figure 1.3.32: Transfer activation.

1.3.5 Setup log

All messages generated while the Setup is running are written to the Setup log XML file. The name of each XML element indicates the message type:

- `<message />`: This is an information message and can safely be ignored.
- `<warning />`: This is a warning message, usually indicating that some user action may be required to resolve the issue.
- `<error />`: This indicates that a severe error has occurred. User action is required to resolve the issue. Severe errors may cause the Setup to abort.

The Setup log XML file is best viewed with a recent version of the Microsoft Internet Explorer browser (see Figure 1.3.33). This will allow you to expand and collapse specific message tables in the XML document. Error and warning messages will be expanded by default, and will be displayed at the top of the browser window. Hence you do not need to scroll down to verify if an error has occurred.

A yellow information bar may appear in Internet Explorer with the following message: "To help protect your security, Internet Explorer stopped this site from installing an ActiveX control on your computer. Click here for options". Right-click the information bar and select **Allow Blocked Content...** A *Security Warning message box* will appear. Click **<Yes>** to confirm that you want to enable the active content in the Setup log XML file.

If the Setup is running in normal (non-silent) installation mode and a warning or error event has occurred, the Setup will automatically display the Setup log XML file in Internet Explorer. Additional messages will continue to be written to the log file, and the file will automatically be updated in Internet Explorer. If you have scrolled down on the Setup log web page, your current position will be lost after the web page has been refreshed.

The Setup log XML file is located in the SetupLogs sub-folder of the BioNumerics program folder. For example:

- C: \Program Files \Applied Maths \BioNumerics \SetupLogs \Setup_1_log.XML
- C: \Program Files (x86) \Applied Maths \BioNumerics \SetupLogs \Setup_1_log.XML

1.3.6 Silent installation

1.3.6.1 Purpose

Running the BioNumerics Setup in "silent installation" mode allows running the BioNumerics Setup program without an end-user interface. No dialogs will be displayed in silent mode, and all messages, including errors, will be logged to the Setup log file. All information required to run the Setup needs to be recorded to a properly formatted Setup_x_ini.XML file. This file must subsequently be invoked through Setup.exe command line parameters.

The silent installation mode can be helpful for mass-deployment of BioNumerics, for creating identical configurations and to automate repetitive behavior.

1.3.6.2 Procedure

Each installation of BioNumerics 6.5 or later not only creates a Setup log XML file, but also a Setup INI XML file (see 1.3.1.12 for more details). This Setup INI XML file recorded during a manual install of



Figure 1.3.33: The BioNumerics setup log.

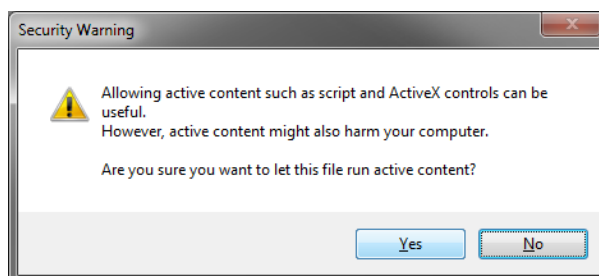


Figure 1.3.34: Security warning.

BioNumerics can subsequently be used to perform silent installations.

The Setup INI XML file is located in the SetupLogs sub-folder of the BioNumerics installation directory. The file name is formatted like Setup_x_ini.XML. Check the file modification date to determine which INI

XML file was created during the latest installation.

The BioNumerics 6.5 or later versions of the Setup program accept the following command line parameters to invoke the silent installation mode:

```
"<path to Setup files>\Setup.exe" /s --ini="<path to Setup_x.ini.XML file>"
```

- The /s command line parameter instructs the InstallShield runtime engine to suppress the *Existing Installed Instances Detected dialog box* if BioNumerics version 6.5 or later is already installed.
- The --ini parameter instructs the Setup script to read the installation settings from the INI XML file, and to hide all dialogs.
- The double hyphen is required to differentiate between InstallShield runtime engine and custom InstallScript command line parameters.
- The slash parameters are used by the runtime engine.
- The double hyphen custom parameters are used by the installation script.
- Optionally the --logdir command line parameter can be specified to override the log_dir path recorded in the Setup INI XML file.

```
"<path to Setup files>\Setup.exe" /s --ini="<path to Setup_x.ini.XML file>" --logdir="<path to log folder>"
```

Example (all command line parameters should be on a single line):

```
"C:\Users\Public\Documents\Applied Maths\BioNumerics \Setup.exe" /s
--ini="C:\Users\Public\Documents\Applied Maths\Setup_1.ini.XML"
--logdir="C:\Users\Public\Documents\Applied Maths\SetupLogs"
```

During silent installations, no error or warning messages are displayed when the Setup is running. The installation Administrator should check the Setup log XML file to verify that no errors have occurred, and that no further action is required to complete the BioNumerics installation on the target computer.



The Microsoft .NET Framework 2.0 SP2 or 3.5 SP1 and Windows Installer 4.5 prerequisites described in [1.2](#) should be installed prior to launching the Setup in "silent installation" mode. For example the silent installation will fail if the Setup is not able to download and install the Microsoft .NET Framework 3.5 SP1.

1.3.6.3 Setup INI XML file format

The information recorded in the Setup_x.ini.XML file has the format as displayed in [Figure 1.3.35](#).

The root XML node of the Setup INI file is the *setup* node. The attributes in the *setup* node are only used for information purposes, for example to display which BioNumerics Setup version created the Setup INI file. The *setup* node also contains *property* sub-elements, one for each property that is required to configure the Setup.

Setup properties typically contain Setup-related configuration values which are not feature-specific, or which are shared by multiple features.

The *start* XML element contains a time stamp indicating when the file was created.

Each feature that was selected for installation has a corresponding *feature* element with the *display_name* attribute. The attribute value must match the feature name displayed in the *Select Features dialog box*. The *feature* element may contain *property* sub-elements, one for each property that is required to configure the parent feature.

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<setup name="\BnSoftwareName" version="7.0.1" date="2012-12-12" time="00:00:00">
  <start date="12-12-2012" time="00:00:00"/>
  <feature display_name="Install application software">
    <property netkey_server="localhost"/>
    <property netkey_server_port="1080"/>
    <property netkey_config_port="1080"/>
    <property netkey_refresh_rate="30"/>
    <property desktop_shortcut="1"/>
  </feature>
  <feature display_name="Install Database Engine">
    <property enable_remote_access="1"/>
    <property restart_sql_server_service="0"/>
    <property user_database_directory="C:\Program Files\Microsoft SQL Server\MSSQL10_
50.\BNSOFTWARENAME\MSSQL"/>
    <property sysadmin_group="CN=Domain Users,CN=Users,DC=domain,DC=local"/>
  </feature>
  <property log_dir="C:\Program Files (x86)\Applied Maths\BnSoftwareName 7.0\SetupLogs"/>
  <property install_dir="C:\Program Files (x86)\Applied Maths\BnSoftwareName 7.0"/>
  <property database_home_dir="C:\Users\Public\Documents\BnSoftwareName\Data"/>
  <property registered_user="user name"/>
  <property registered_organization="organization name"/>
  <property license_string="license string"/>
  <feature display_name="Microsoft SQL Server 2008 R2 Native Client"/>
  <feature display_name="Microsoft System CLR Types for SQL Server 2008 R2"/>
  <feature display_name="Microsoft SQL Server 2008 R2 Shared Management Objects"/>
  <feature display_name="HPC Pack 2008 MS-MPI Redistributable Package"/>
  <feature display_name="Install sample database"/>
  <feature display_name="Install sample and tutorial data"/>
  <feature display_name="Install Sentinel drivers"/>
  <!-- The NetKey+ service should only be deployed to a single license server computer,
  hence silent deployment of this feature usually is not desired. -->
  <!--
  <feature display_name="Install NetKey+ server program"/>
  -->
</setup>

```

Figure 1.3.35: Setup INI XML file format.

1.3.7 Silent un-installation

1.3.7.1 Purpose

The silent un-installation procedure uninstalls the BioNumerics program without displaying the graphical user interface.

1.3.7.2 Procedure

Go to the *Programs and Features* Windows control panel and check the BioNumerics Program display name (e.g. BioNumerics 7.6 x64 in Figure 1.3.36).

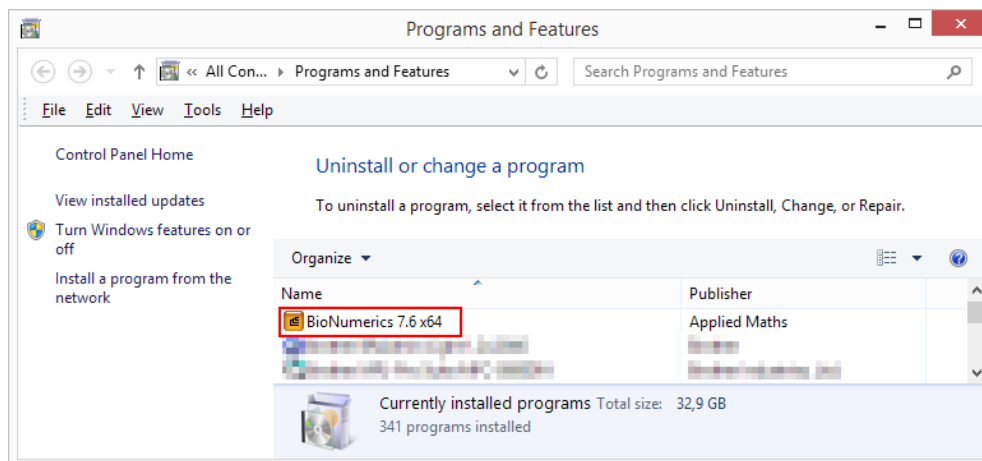


Figure 1.3.36: Programs and Features.

Open the Windows registry by typing **regedit** in the search tab of the Windows Start Menu. The actual program (regedt32.exe) is located in the following location: C:\Windows\System32\regedt32.exe.

Browse to the **Uninstall** registry key (see 1 in Figure 1.3.37):

- On 64-bit Windows version:

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall

- On 32-bit Windows version:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall

Choose **Edit > Find** and search for the GUID subkey containing the **DisplayName** registry value that matches with the Program display name (e.g. search for BioNumerics 7.6 x64) obtained from the first step. See 2 in Figure 1.3.37.

In this subkey, copy the string text of the **UninstallString** registry value (see 3 in Figure 1.3.37). For example: "C:\Windows\system32>"C:\Program Files (x86)\InstallShield Installation Information\64306F67-16C0-48EE-9F30-B3D336E7FA35\setup.exe" -runfromtemp -l0x0409 -removeonly"

Append the "/silent" command line parameter to the **UninstallString** text to hide the confirmation dialogs: For example: "C:\Windows\system32>"C:\Program Files (x86)\InstallShield Installation Information\64306F67-16C0-48EE-9F30-B3D336E7FA35\setup.exe" -runfromtemp -l0x0409 -removeonly /silent"

Start a command prompt as an administrator and run the uninstall command.

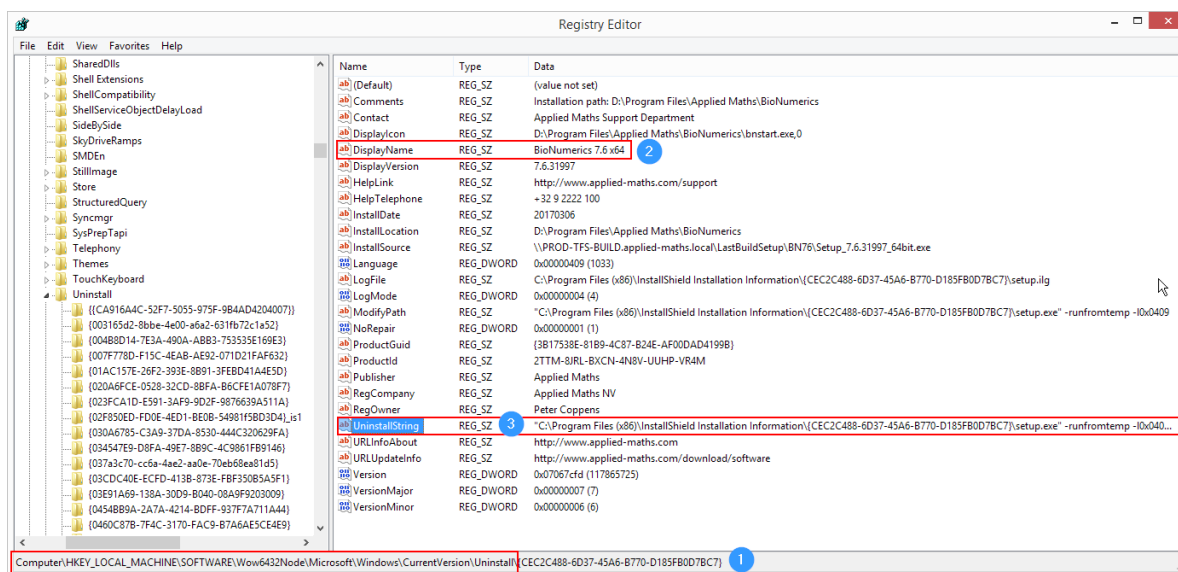


Figure 1.3.37: Registry Editor.

To automate the above steps you will need to develop a vbScript or Powershell script for retrieving the correct **UninstallString** text value from the registry, appending the "/silent" command line parameter and executing the command with administrator privileges.



The **UninstallString** (<GUID>) is unique for each instance of BioNumerics, and thus is also machine specific. Hence you cannot use the same uninstall command line for removing BioNumerics from multiple computers.

Chapter 1.4

NetKey+ configuration

1.4.1 Introduction

If a network license has been purchased, the *NetKey+ server program* and the *Sentinel drivers* must be installed on a computer where the hardware security will be connected to (i.e. the *server* computer) (see [1.3](#)).

After installation of these features on the server computer, the NetKey+ service needs to be installed and started using the NetKey+ Configuration tool (NetKey+Config.exe) (see [1.4.2](#)).

Once started, the license(s) can be configured in the NetKey+ Configuration tool (see [1.4.3](#)) and the NetKey+ service can start distributing sessions to the requesting BioNumerics applications running on the client computers (i.e. the computers with the application software installed) (see [1.4.4](#)).

1.4.2 Installing and starting the NetKey+ service on the server

If a network license string has been entered in the *Customer Information dialog box*, and the NetKey+ server program feature was selected for installation in the *Select Features dialog box*, the Setup will ask if you want to run the NetKey+ Configuration tool (see [Figure 1.4.1](#)). This tool allows you to install and subsequently start the NetKey+ service.

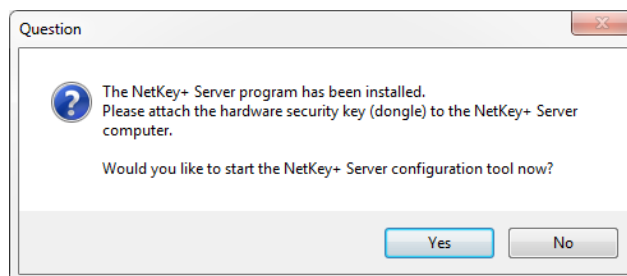



Figure 1.4.1: Run the NetKey+ Configuration tool.

Click <Yes> to start the NetKey+ Configuration tool. This will run the tool with Windows elevated privileges (**Run as administrator**) and the *Login window* will be displayed (see [Figure 1.4.2](#)).



The NetKey+ Configuration tool can also be called by (double-)clicking on the NetKey+Config.exe application in the installation directory of BioNumerics. Alternatively, press the **<Settings>** button () in the startup window of BioNumerics- if the application software has been installed - and select **NetKey+ configuration** from the drop-down list.



The configuration tool can be run as NetKey+ **User** or NetKey+ **Administrator** in combination with or without Windows elevated privileges. An overview of all tools that are accessible in the NetKey+ Configuration program for the four different login options is given in [1.4.9](#).



To run a program with Windows elevated privileges in Windows Vista, Windows 7 or Server 2008, right-click on the application and select "Run as administrator".

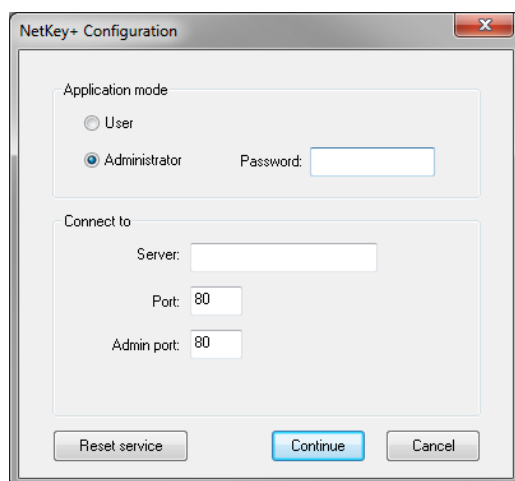


Figure 1.4.2: The *Login window*.

Choose the **Administrator** mode in the *Application mode panel*. This mode will allow you to install and start the NetKey+ service.

The first time the service will be started, a password will be prompted for. This **Password** is required the next time someone wants to access the configuration program in **Administrator** mode. When the service has not been started yet, the **Password** field can be left empty.

Enter the local computer name or "localhost" as the **Server** name in the *Connect to panel* to indicate that the NetKey+ service will be installed on the computer where the tool is running.

The server **Port** number is an available TCP port number that will be used by the NetKey+ server and clients to exchange session information. The **Admin port** is an available TCP port number that will be used to by the NetKey+ server and configuration tool to configure the service settings. The default suggested TCP port number for both ports is 80. Any other port numbers can be specified.



An HTTP-based protocol is used for the communication between the NetKey+ server, the NetKey+ Configuration tool and the BioNumerics application. Both TCP ports must be enabled on the Windows firewall or any other security tool that may block access to these ports, both on the NetKey+ server computer and on each computer where BioNumerics is installed. The NetKey+ server TCP ports may not be used by any other application or service. For example, no websites should be hosted on the IIS server using a NetKey+ TCP port number.

Clicking the **<Reset service>** button will stop the NetKey+ service on the server computer and will delete all current NetKey+ settings, including the Administrator password (see [1.4.7](#) for more information). This operation is not applicable if the service is not already installed.

Clicking the **<Continue>** button will save the connection settings to the NetKey.ini text file, and to the NetKey+_Config.txt XML file. These files are located in the folder containing application data for

all users (CommonAppDataFolder, for which the path is typically C: \ProgramData \Applied maths \NetKey+).

Select **Connection** in the left panel to display the server connection settings (*Server connection panel*) and service status (*Service panel*) (see Figure 1.4.3).

The **Refresh rate** determines how often the information displayed in the NetKey+ Configuration tool is updated. The default value is 30 seconds.

The **Service status** text box displays the current status of the NetKey+ windows service. The status should be "Not installed" if this is the first time the BioNumerics Setup is running on the server computer.

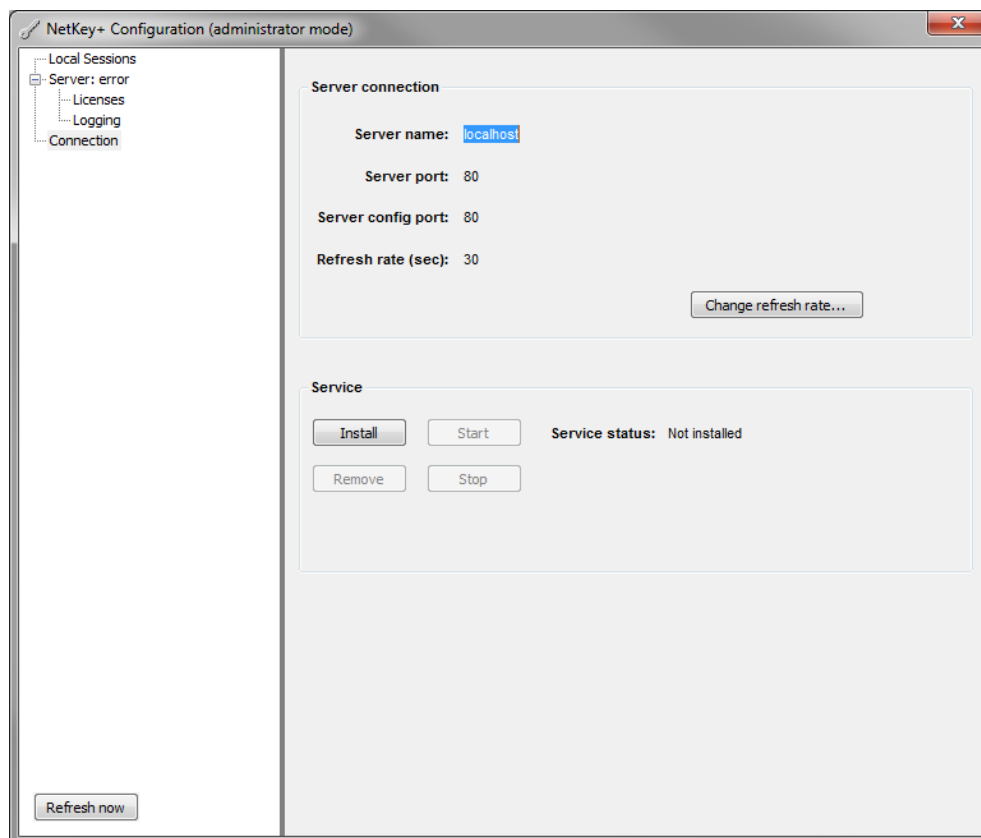


Figure 1.4.3: The NetKey+ Configuration tool window.

Click the **<Install>** button in the lower *Service panel* to install the NetKey+ Windows service. Next click the **<Start>** button to start the NetKey+ service.

The *Change server password dialog* will be displayed during a first-time installation of the service, allowing you to enter and confirm a new NetKey+ server password (see Figure 1.4.4). A user will be required to enter the NetKey+ server password each time the configuration tool is started in **Administrator** application mode. After the user clicks **<Continue>** in the *Login window*, the configuration tool will connect to NetKey+ server via the specified Server config **Port** (or **Admin port**) to verify the credentials.

After clicking **<OK>** in the *Change server password dialog box*, the password is encrypted and stored in the NetKey+_Config.txt XML file. The Service status will change to "Started" if no error has occurred. In case of error, the NetKey+_LOG.txt log file can be checked to verify the error message (see 1.4.6). The log file is stored in the same ProgramData or Application Data folder as the NetKey.ini file, depending on the Windows version.

Once the service has been installed and started, the service can be stopped by pressing the **<Stop>** button, and can be removed by clicking the **<Remove>** button in the lower *Service panel*.

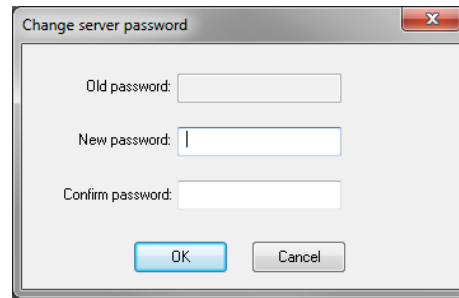


Figure 1.4.4: Specify server password.



The *Service panel* will be disabled (grayed out) if the configuration tool is launched without Windows elevated privileges.



The NetKey+ server program is a Windows service. Once the service is installed, it can be managed from the Services control panel. On Windows Vista or later, the Services control panel can be accessed from Control Panel > System and Security > Administrative Tools > Services.

Select **Server** in the left panel (see Figure 1.4.5).

The ports displayed in the upper panel are the TCP listening ports on the NetKey+ server computer. These port numbers must correspond with the port numbers saved in the `NetKey.ini` file. The next time the configuration tool is launched, the program will read the port numbers from the `NetKey.ini` file and automatically display the numbers in the *Login window*. The *Uptime* value displays the amount of time the NetKey+ service has been up and running.

The hardware or software (HASP) security keys detected on the server computer are displayed in the lower *Available license keys panel*. At least one key should be listed, if not please check the dongle drivers.

Click the **<Edit Settings>** button to display the *Server properties window* (see Figure 1.4.6), and to change the TCP port numbers.



If the NetKey+ Configuration tool or the BioNumerics application is unable to communicate with the NetKey+ service through the specified port numbers then check your security settings to make sure that the TCP ports are accessible. For example, if a software firewall has been enabled on the NetKey+ server or on the BioNumerics client computer, then the firewall may need to be configured to allow traffic for the Applied Maths executables and/or the applicable TCP port numbers.

Continue with 1.4.3 if you want to set up the BioNumerics license string(s) to allow access for the client computers.

Click the "x" sign in the top right corner or press **ALT+F4** to close the NetKey+ Configuration tool. Closing the NetKey+ Configuration tool will not affect the current status of the NetKey+ service. If the service is running, then clients will be able to connect to the NetKey+ server if the configuration was successful.

1.4.3 Configuring licenses

Adding and configuring licenses can only be done by running the NetKey+ Configuration tool in **Administrator** application mode, with or without Windows elevated privileges (**Run as administrator**) (see Table 1.4.1). After selecting the **Administrator** mode in the *Login window*, the correct NetKey+ server password can be entered in the **Password** field (see Figure 1.4.2).

The settings in the lower *Connect to panel* correspond with the settings stored in the `NetKey.ini` file. These settings can be changed if the tool was started with Windows elevated privileges. Click the **<Continue>**

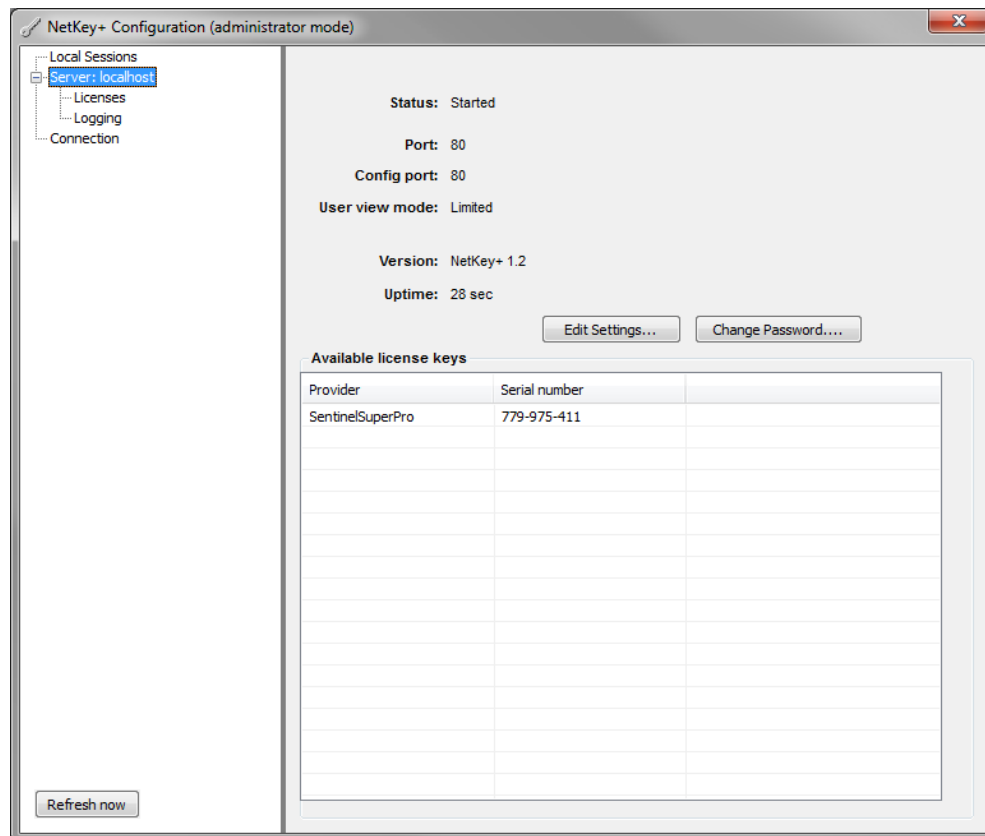


Figure 1.4.5: The NetKey+ settings and available keys.

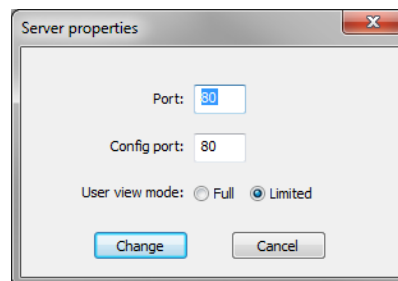


Figure 1.4.6: Edit the server properties.

button to connect to the NetKey+ server.

Select **Licenses** under the **Server** option in the left panel (see Figure 1.4.7). Click the <Add> button to add a new BioNumerics license string to the list of installed licenses.

In the *License properties dialog box*, enter the 6 x 4 characters **License String** in the input fields (see Figure 1.4.8). Alternatively, use the <P> button to paste the contents of the clipboard in the **License** fields. The license string is provided on the sleeve of the installation DVD or the string may have been delivered electronically. An error message will pop up when attempting to add an invalid license string (e.g. a standalone license string, a second license string for the same key, ...) to the license list.

Press <Add> to insert the new license string into the list of installed licenses. The added license string will be displayed in the **String** column (see Figure 1.4.7). The number of concurrent sessions that are granted to the license is shown in the **Allowed sessions** column. If the corresponding protection key is present in the **Available license keys** list (see Figure 1.4.5), the state of the license is set to **Active**. If the key is not detected on the server computer, the state is set to **Valid**. The last **Sessions in use** column displays the total

- **Filter by Computer Name:** TCP/IP Host name of the client computer (with or without domain name).
- **Filter by User Name:** Windows login name without domain name.
- **Filter by IP:** Single or a range of IPv4 addresses of client computers.

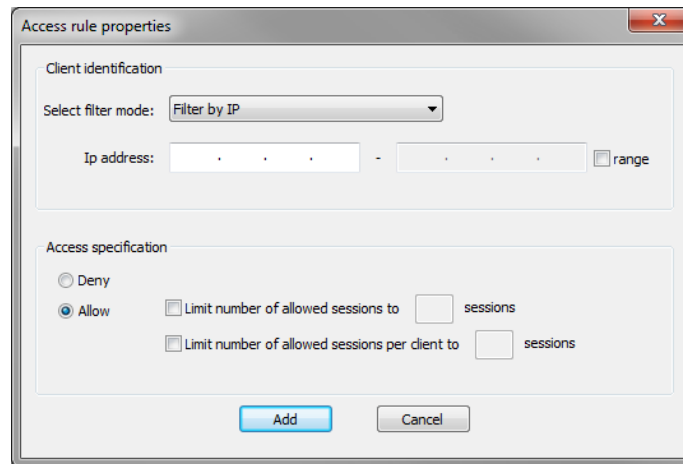


Figure 1.4.11: The *Access rule properties* dialog box.

A range of IPv4 addresses can be specified if the **Range** option is checked in the upper *Client identification* panel. Optionally a limit on the number of allowed concurrent sessions can be specified in the lower *Access specification* panel when the **Allow option** is checked (see Figure 1.4.11). Note that NetKey+ does not support the **Filter by IP** filter mode for IPv6 addresses.

Pressing the **<Add>** button adds the rule to the **Access rules** list (see Figure 1.4.10). Each access rule is identified by a unique identifier (**Id**). The filter mode is displayed in the **Client filter** column, and the **Sessions** column displays the number of allowed concurrent sessions to all clients. If no limit has been set this column will display **Limit by license**. The **Sessions per client** column displays the number of allowed concurrent sessions for each client. If no limit has been set this column will also display **Limit by license**. Both these sessions columns will display **Deny** if this has been specified as the Access specification. The **Connected** sessions column shows the number of currently connected sessions. The number of sessions that are queued on a waiting list are shown in the last **Waiting sessions** column.

The access properties for a selected rule can be modified by clicking the **<Change>** button. If multiple access rules have been specified for a license, the order of the rules can be changed with the **<Up>** and **<Down>** buttons.

When a client tries to open a session, a *session request* is sent to the server, containing computer information of the client (computer name, Windows user name, IP address, and MAC address) and the license string. The server checks the access rules of the license string that is sent with the session request, and based on the access rules, the server grants or denies the client access to the license. Each session that is granted access to a license is identified by a unique identifier, the *session ID*. The session identifier is sent to the client, and the session is launched on the client computer or the session is put on a waiting list in case the number of allowed sessions (on the client) is reached. The client stores the *session ID* of the session and closes the connection with the server computer. On regular time intervals, a *renew session request* of each connected session and session that is put on hold is sent to the server. Based on these renew session requests, the server keeps track of the status of the sessions on the client computers. The server might disconnect a session if the **Usage time**, **Idle time** or **Timeout** value for a session is reached:

- **Usage time:** The *Usage time* (or *time in use*) of each session that is granted access to a license is recorded by the server program. The usage time is the total connection time for each connected

session, or in case of a session present in the waiting queue, the time the session has been put on hold. In case there is a waiting list, a connected session for which the usage time exceeds the maximum usage time (default 120 min., see Figure 1.4.8) will be closed in favor of the first session in the waiting list. The usage time of the session that was put on hold, but now is launched by the software, is reset. A session that exceeds the maximum usage time limit will not be closed as long as there is no waiting list.

- **Idle time:** The *Idle time* of each connected session is also recorded by the server program. The idle time starts running as soon as the session is running on a client computer. The status of the session is checked each time a *renew session request* is sent to the server: when the session is in use, the idle time is reset; if no user activity is recorded, the idle time keeps running. A session for which the idle time exceeds the maximum idle time (default 60 min., see Figure 1.4.8) will be closed in favor of the first session in the waiting list. A session that exceeds the idle time limit will not be closed by the server as long as there is no waiting list.
- **Timeout:** The *Timeout* of a connected session starts running when the server stops receiving *renew session requests* for the session (e.g. caused by a crash, network problems, ...). A session that exceeds the timeout time (default 5 min., Figure 1.4.8) is closed.

If a session is disconnected by the server, e.g. due to idle time or maximum usage limit, a warning box flashes, warning the client that the session is removed from the list of connected sessions. The session halts automatically after a few seconds.

To change the default suggested *Usage time*, *Idle time* and *Timeout* values for a license, select the license from the list in the left panel and press the **<Change>** button to call the *License properties dialog box* (see Figure 1.4.8).

1.4.4 Running sessions on the clients

After the Setup has finished installing the BioNumerics application, configured with a network license, on the client computers (see 1.3), the BioNumerics application should start on the client computers if the following conditions are met:

1. The NetKey+ service is running on the NetKey+ server computer (see 1.4.2).
2. The correct NetKey+ server name and TCP port number have been specified on the client computer.
3. If present, the security software (e.g. firewall) has been configured to allow access to the NetKey+ TCP port.
4. The TCP port is not in use by another application.
5. There is a matching access rule that grants the client access to the license (see 1.4.3).

If a client is allowed access to the license, but the session limit is reached (see 1.4.3), the session is added to the waiting queue. A message pops up on the client computer, stating how many sessions have to close before the session can be launched by the software (see Figure 1.4.12). As soon as one of the connected sessions of the corresponding license is closed on one of the clients, the first session in the waiting list automatically opens on the client computer, and all waiting numbers of the remaining sessions in the waiting queue are updated. Press the **<Close Application>** button if you wish to remove the session from the waiting list.

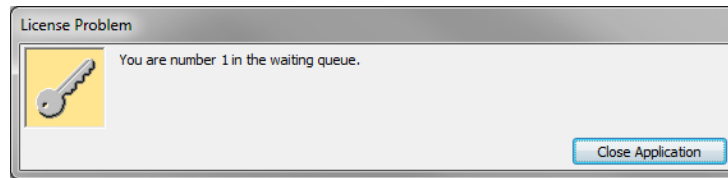


Figure 1.4.12: Waiting queue.

1.4.5 Monitoring sessions

A list of all sessions that are running on the client computers and that are put on hold, can be consulted in the *NetKey+ Configuration window* when logged in as **Administrator** or as **User** with **Full** view mode (see Table 1.4.1). Selecting the **Sessions** option in the left panel, shows the sessions in the right panel (see Figure 1.4.13). Each connected session and session present in the waiting queue is identified by a unique *session identifier* (**ID** column). The access rule ID that grants access to the license is displayed in the **Linked rule** column. Information of the associated client computer is shown in the **Client Id**, **Name** and **IP address** columns. The **Status** of each connected session is set to **Connected**. When a session is put on the waiting list (**Waiting** status), the number of sessions that have to close before this session can be launched by the software is displayed in the **Wait number** column. Detailed session information is shown in the right panel when selecting a session in the left panel.

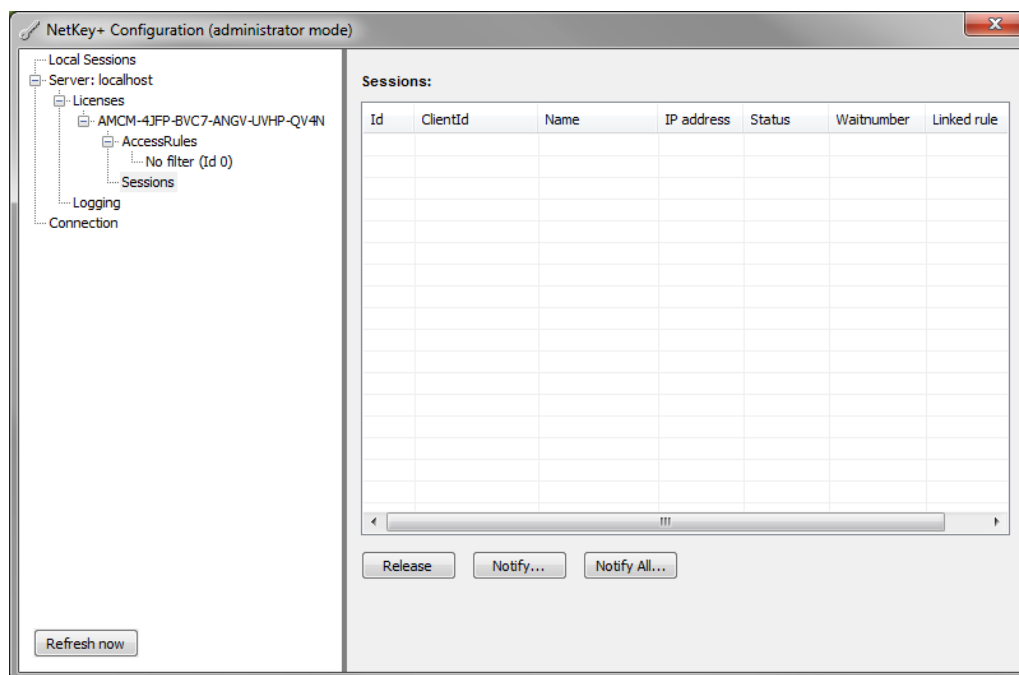


Figure 1.4.13: List of connected sessions and sessions that are present in the waiting queue.

In **Administrator** mode, messages can be sent to any or all connected clients, for example in case the server computer will be shut down or if a session will be disconnected (see Table 1.4.1). To send a message to a client, select a session of the client in the *Sessions panel* (see Figure 1.4.13), and press the **<Notify>** button (see Figure 1.4.13). Alternatively, select the session under the **Sessions** option in the left panel and select the **<Notify>** button. Enter a message string and press **<OK>** (see Figure 1.4.14). The message is sent to the corresponding client. A message can be sent to all users with **<Notify All>**. All active users will receive the message in a dialog box.

All connected sessions on the clients and sessions present in the waiting queue, can be closed by the **Ad-**

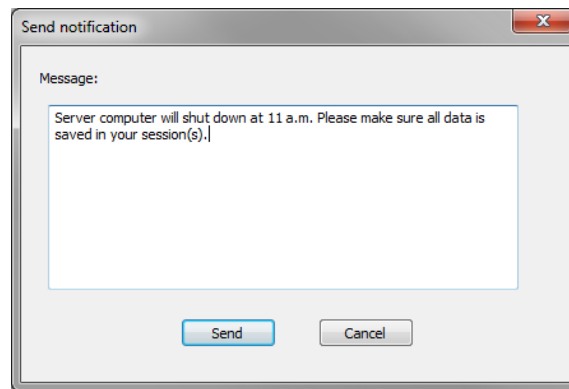


Figure 1.4.14: Notification message.

ministrator (see Table 1.4.1). To close a session, select the session in the *Sessions panel* (see Figure 1.4.13), and disconnect the session with **<Release>**. Alternatively, select the session under the *Sessions* option in the left panel and select the **<Release>** button.

A list of all sessions that are running on the *local* computer and that are put on hold, can be consulted in the *NetKey+ Configuration window* when logged in as *Administrator* or as *User* with **Full** or **Limited** view mode. Selecting the **Local Sessions** option in the left panel, shows all connected local sessions and local sessions that are present in the waiting queue below the **Local Sessions** option in the left panel (see Figure 1.4.13). The *Status* (**Connected** or **Waiting**) and *Time in use*, are shown next to each local session. Detailed session information is shown in the right panel when selecting a local session in the left panel.

1.4.6 Logging data

When the NetKey+ Configuration program is launched in *Administrator* mode or in *User* mode with **Full** view, the **Logging** option is displayed in the left panel (see Table 1.4.1 and Figure 1.4.15).

Pressing the **Logging** option in the left panel shows all logged information in the right panel. This logged information is stored in a text file called `NetKey+_Log.txt`. This file is located in the folder containing application data for all users (CommonAppDataFolder, for which the path is typically `C: \ProgramData \Applied maths \NetKey+`).

When *verbose logging* is enabled, additional information messages are logged in the text file (see Figure 1.4.15). Selecting the **<Change>** button changes the verbose logging status. To clear the log file, press the **<Clear log>** button.



Enabling/disabling verbose logging (**<Change>**) and clearing the log file (**<Clear log>**) is only possible in *Administrator* mode (see Table 1.4.1).

1.4.7 Resetting the NetKey+ settings

When the NetKey+ Configuration tool is run with Windows elevated privileges (**Run as administrator**), the **<Reset service>** button is displayed in the *Login window* (see Figure 1.4.2). This button allows you to delete all current NetKey+ settings, including the Administrator password. Furthermore this operation will delete all licensing information and access rules you may have configured previously. Hence the reset service function should be used with caution.

Use the following steps to stop the NetKey+ service and delete the NetKey+ settings:

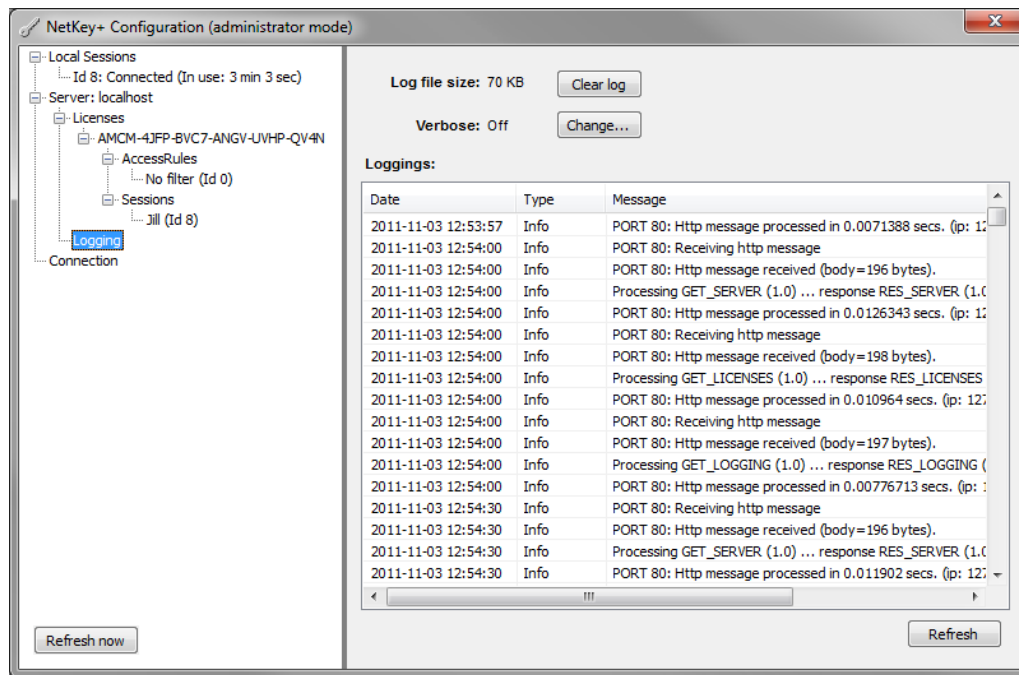


Figure 1.4.15: Logging information.

1. Click the **<Reset service>** button in the *Login window* (see Figure 1.4.2).
2. Click **<Yes>** in the confirmation dialog (see Figure 1.4.16) to delete the current NetKey+ configuration. All NetKey+ settings will be deleted after clicking **<Yes>**.
3. Select the **Administrator** option in the upper *Application mode panel*.
4. Verify and update the **Port** and **Admin port** TCP port numbers if needed. Make sure that the TCP port numbers are not in use on the NetKey+ server computer.
5. Click **<Continue>** and select **Connection** in the left panel to display the **Service** settings.
6. Click **<Start>** in the lower *Service panel*. This brings up the *Change server password dialog*.
7. Enter a secure NetKey+ Administrator password in the **New password** and **Confirm password** text boxes. This password will be required to be able to start the NetKey+ Configuration tool in Administrator application mode.
8. Restart the NetKey+ Configuration tool. Select the **Administrator** option in the upper *Application mode panel* and enter the Administrator **Password** created in the previous step.
9. Click **<Continue>** to connect to the NetKey+ service.

Now you are ready to start configuring the access rules for your BioNumerics license.

1.4.8 Repairing the NetKey+ service

The following steps allow you to repair the NetKey+ service without deleting the current configuration:

1. Select the **Administrator** option in the upper *Application mode panel*.
2. Enter the NetKey+ Administrator **Password** and click the **<Continue>** button.

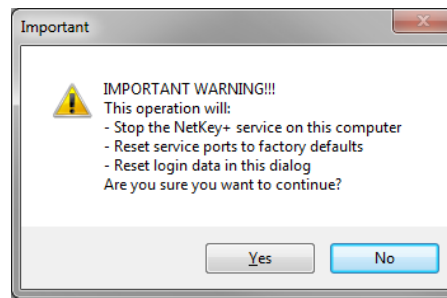


Figure 1.4.16: Warning message.

3. Select **Connection** in the left panel to display the **Service** settings. Click the **<Remove>** button in the lower **Service panel** to uninstall the NetKey+ Windows service.
4. Click **<Install>** to re-install the NetKey+ service.
5. Next, click the **<Start>** button to start the NetKey+ service.
6. Close the NetKey+ Configuration tool.

1.4.9 Overview of configuration rights

The NetKey+ Configuration program (`NetKey+Config.exe`) is available on the server computer and on all client computers that have the application software installed. This configuration tool can be run as *NetKey+ user* or *NetKey+ administrator*, in combination with or without *Windows elevated rights*. An overview of all rights for the four different login options are shown in the table below.

1.4.10 Usage statistics

1.4.10.1 Usage information parse tool

The NetKey+ server program comes with a standalone command line tool called `ParseUsage.exe`. This tool will transform the NetKey+ log file (see 1.4.6) to a tab-delimited text file. This text file can easily be imported in MS Excel, which can be used to create usage statistics.

On the NetKey+ server computer, open a command prompt or a Windows PowerShell window and navigate to the NetKey+ installation folder (see 1.3.1.7).

Enter the command `"ParseUsage "` and press **Enter** to see how to use the `ParseUsage.exe` tool. The result is depicted in Figure 1.4.17.



For Windows PowerShell, start any command line with `". \ "`. For example, `"ParseUsage "` in a command prompt becomes `". \ParseUsage "` in PowerShell.

Table 1.4.2 lists all available options for the `ParseUsage.exe` command line tool.

For the `ParseUsage.exe` tool to work, at least the path for the output file should be specified, e.g. `"ParseUsage out=C:\LogFiles\NetKey+.TXT "`.



In case a file path contains one or more spaces, it should be enclosed with double quotes in the Windows command prompt or PowerShell.

The output of `ParseUsage.exe` is a tab-delimited text file with seven fields:

	Windows elevated rights	Windows user rights
NetKey+ admin (password required)	<ul style="list-style-type: none"> • Configure licenses, passwords, logging • Monitor all sessions • View log information • Start/stop service only when run on the server computer • Configure ports 	<ul style="list-style-type: none"> • Configure licenses, passwords, logging • Monitor all sessions • View log information
NetKey+ user (no password)	<ul style="list-style-type: none"> • Limited user view: Monitor own sessions, Configure ports • Full user view: Monitor own sessions, View session information from other clients, View log information, Configure ports 	<ul style="list-style-type: none"> • Limited user view: Monitor own sessions • Full user view: Monitor own sessions, View session information from other clients, View log information

Table 1.4.1: Running the NetKey+ configuration tool with different rights.

```

C:\Windows\system32\cmd.exe

C:\Program Files (x86)\Applied Maths\BioNumerics66beta>ParseUsage.exe
Please provide argument 'out'.

ParseUsage out=<filename> [inp=<filename>] [begin=<date>] [end=<date>] [Lic=<lic
ense string>] [IP=<ip address>] [User=<user>]

- out: output text file
- inp: optional, input file name, default %ProgramData%\Applied Maths\netkey+N
etKey+_LOG.txt
- begin: optional, begin date in output file, format YYYY-MM-DD
- end: optional, end date in output file, format YYYY-MM-DD
- Lic: optional, filter on specific license string
- IP: optional, filter on specific ip address
- User: optional, filter on specific user

C:\Program Files (x86)\Applied Maths\BioNumerics66beta>_

```

Figure 1.4.17: Windows command prompt with "ParseUsage" executed.

- **Start:** Time stamp for the start of a session.
- **End:** Time stamp for the end of a session.
- **Duration (s):** Total time that the session lasted (in seconds).
- **Lic:** License string used.
- **IP:** IP address (IPv4) of the computer where the session was in use.
- **User:** Windows user name.
- **ID:** Session ID as generated by the NetKey+ server program.

Option	Description
out	The location and name of the output file Example: "out=c:\NetkeyReports\usage_Q1_2011.txt "
inp	The location of the NetKey+_LOG.txt file (optional) Default value: "%ProgramData%\Applied Maths\netkey+\NetKey+_LOG.txt " Example: "inp=c:\Logfiles\Netkey+\Netkey+_LOG_2011.txt "
begin	A begin date in the format YYYY-MM-DD (optional) Example: "begin=2011-01-01 "
end	An end date in the format YYYY-MM-DD (optional) Example: "end=2011-03-31 "
Lic	A filter on a specific license string (optional) Example: "Lic=ABCD-82FP-234N-2N8V-VVHP-UR99 "
IP	A filter on a specific client IP address (IPv4) (optional) Example: "IP=192.168.001.010 "
User	A filter on a specific user name (optional) Example: "User=John "

Table 1.4.2: Options for ParseUsage.exe.

1.4.10.2 Example

We will illustrate the use of ParseUsage.exe with following (hypothetical) example:

In a research institute there are two types of BioNumerics network licenses, one with all modules (for 3 simultaneous users) and another one with only the Fingerprint data module and the Tree and network inference module (for 5 simultaneous users). The institute has bought this for multiple users belonging to three different labs. Since each lab has its own annual budget, the institute would like to charge the labs for their usage of the different BioNumerics licenses. Invoicing is done after the end of each quarter. The financial department has calculated that the total cost of the 3-user network license is 500 euro per quarter and the cost of the 5-user network license is 350 euro per quarter. Each lab should be billed the respective portion of each license.

- **LAB1 users:** Peter S., Jake, Tim
- **LAB2 users:** Jane, Peter V., Sophie, Anna
- **LAB3 users:** Tom, Catherine, Luke

An example NetKey+ log file, named Netkey+_LOG_demo.TXT, can be downloaded from the Applied Maths website (<http://www.applied-maths.com/download/sample-data>, click on "Example NetKey+ log file").

As the NetKey+ server program logs all opened sessions, we will use the ParseUsage.exe tool to create a usage report for the first quarter of 2011.

10.1 On the command line specify: "ParseUsage out=c:\Users\Public\Documents\usage_Q1_2011.txt inp=c:\Users\Public\Documents\Netkey+_LOG_demo.txt begin=2011-01-01 end=2011-03-31 " and press **Enter**.



Obviously, if the example Netkey+_LOG_demo.TXT file is located in a different directory, the command line should be adapted accordingly.



The instructions given below are for Microsoft Excel 2010. For other versions of Excel, we refer to the corresponding user manual.

10.2 Open the usage_Q1_2011.TXT file with MS Excel and add a column for the Lab according to the list of lab members shown above (see Figure 1.4.18 for an example).

	A	B	C	D	E	F	G	H
1	Start	End	Duration(s)	Lic	IP	User	ID	Lab
2	1/01/2011 9:34	1/01/2011 12:24	10182	XYZQ-82XP-134N-2N9V-WWHP-UP99	192.168.001.026	Anna	fbdbce11a39b	LAB2
3	1/01/2011 9:34	1/01/2011 11:24	6574	ABCD-82FP-234N-2N8V-VVHP-UR99	192.168.001.031	Jane	82473553507a	LAB2
4	1/01/2011 9:34	1/01/2011 11:22	6473	XYZQ-82XP-134N-2N9V-WWHP-UP99	192.168.001.016	Luke	8e944d3e9c49	LAB3
5	1/01/2011 9:34	1/01/2011 11:32	7064	ABCD-82FP-234N-2N8V-VVHP-UR99	192.168.001.020	Luke	9ee0fc6cdb16	LAB3
6	1/01/2011 9:34	1/01/2011 11:03	5337	ABCD-82FP-234N-2N8V-VVHP-UR99	192.168.001.032	Tim	13ff03553f56	LAB1
7	2/01/2011 9:34	2/01/2011 10:32	3476	ABCD-82FP-234N-2N8V-VVHP-UR99	192.168.001.010	PeterS	991bb33ad03f	LAB1
8	2/01/2011 9:34	2/01/2011 12:11	9416	XYZQ-82XP-134N-2N9V-WWHP-UP99	192.168.001.033	Sophie	39becb89b6a6	LAB2
9	3/01/2011 9:34	3/01/2011 13:03	12512	ABCD-82FP-234N-2N8V-VVHP-UR99	192.168.001.029	Anna	698505cefb1e	LAB2

Figure 1.4.18: The parsed usage file in MS Excel.

10.3 Select the whole range that contains data and insert a "PivotTable" with "PivotChart" in Excel.

10.4 Click <OK>.

10.5 Choose 'Lic' and 'Lab' as Category fields (Axis field) and 'Duration (s)' as the Values field (Sum). The result is depicted in Figure 1.4.19.

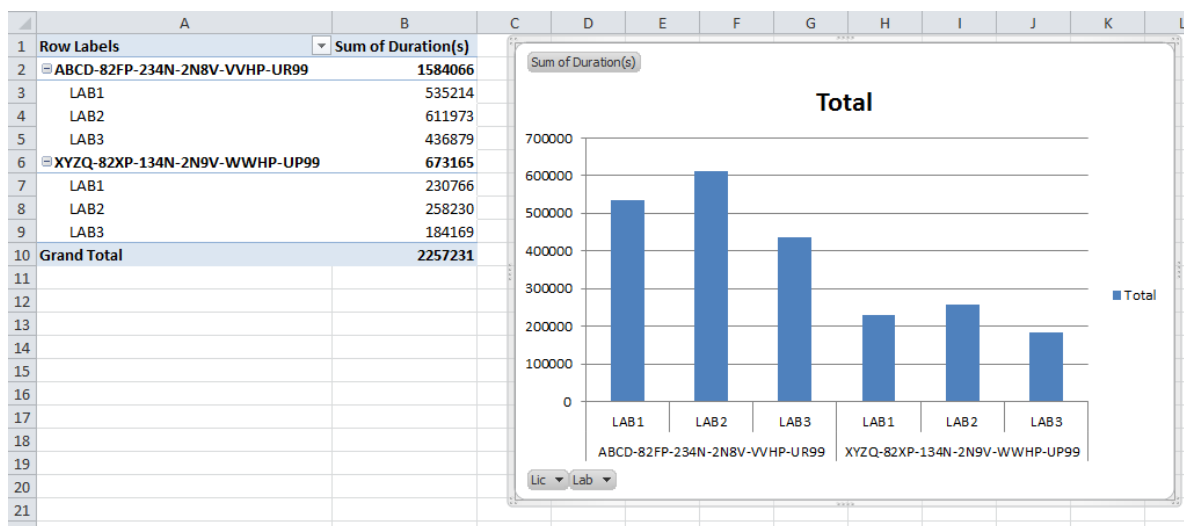


Figure 1.4.19: Resulting PivotTable and PivotChart in MS Excel.

Currently, usage times are expressed in absolute values (seconds), but we can change the display setting for the "Sum of Duration (s)" to relative values.

10.6 Right-click on the "Sum of Duration (s)" cell and choose "Show values as > % of Parent Total..." with the base field 'Lic'.

10.7 You can then easily add a 'Cost' column to this PivotTable and see the respective value per lab per license (see Figure 1.4.20).

	A	B	C
1	Row Labels	Sum of Duration(s)	Cost
2	ABC82FP-234N-2N8V-VVHP-UR99	100,00%	€ 350,00
3	LAB1	33,79%	€ 118,26
4	LAB2	38,63%	€ 135,22
5	LAB3	27,58%	€ 96,53
6	XYZQ-82XP-134N-2N9V-WWHP-UP99	100,00%	€ 500,00
7	LAB1	34,28%	€ 171,40
8	LAB2	38,36%	€ 191,80
9	LAB3	27,36%	€ 136,79
10	Grand Total		

Figure 1.4.20: Calculated license costs per lab and per license.

Chapter 1.5

Installation process

1.5.1 Overview

The purpose of this chapter is to provide a general technical explanation on the Setup behavior, and a basic Setup flow diagram of the installation processes. This chapter contains a partial list of the main functions that are applied in the InstallShield installation script. It is not intended to provide a detailed description of all functions implemented in the installation script.

The BioNumerics installation process can be divided into three main blocks: the initial dialog sequence, the feature installation or removal processes and a final sequence running a cleanup process and showing the finish dialog. A subset of dialogs D1 to D9 is displayed during the initial dialog sequence when the Setup is running in normal (non-silent) mode. Next, the *OnMoveData* process will install the selected features, and uninstall the de-selected features.

The Setup will call the appropriate functions for each feature that is being installed or removed: *<feature>_Installing* and *<feature>_Installed* during installation, and *<feature>_UnInstalling* and *<feature>_UnInstalled* during removal. Each *<feature>_** feature function will either call the *FeatureStart* or the *FeatureEnd* function to create the feature node in the Setup log XML file with the proper time stamp elements. The feature nodes contain the information, warning and error messages for a specific feature.

In normal (non-silent) mode the final sequence will display the finish dialog. The *CleanUp* function will display the Setup log file in Internet Explorer if warning or error messages were written to the Setup log file.

1.5.2 Setup dialog list

The following table lists the dialogs that are displayed during a normal Setup, and that are invoked by the InstallShield engine and installation script (see Table 1.5.1). This does not include the dialogs from the NetKey+ Configuration tool.

1.5.3 Setup processes

1.5.3.1 Read command line options

When the Setup executable is launched the Setup engine will first attempt to detect if a previous instance of the software is already installed. If the same or another version of the software is already installed the Setup will initially display the *Existing Installed Instances Detected dialog box*. Next, the engine will launch the InstallShield installation script.

Number	Dialog name	Dialog image	Related section
D1	Existing Instances		1.3.1.3
D2	Dlg_SdWelcome		1.3.1.3
D3	Dlg_Start / SdWelcomeMaint	Figure 1.3.20	1.3.3.2
D4	Dlg_SdLicense2		
D5	Dlg_SdSetLicense	Figure 1.3.4	1.3.1.5
D6	Dlg_SdPathOptions	Figure 1.3.6	1.3.1.7
D7	Dlg_SdFeatureTree	Figure 1.3.7	1.3.1.8
D8	Dlg_SdNetKey	Figure 1.3.9	1.3.1.9
D9	Dlg_SdStartCopy2		
D10	SdFinish / SdFinishReboot		

Table 1.5.1: The Setup dialog list.

One of the first initialization steps in the installation script is to read the optional command line options used to launch the Setup executable. Currently, the Setup supports the `--ini` and `--logdir` command line parameters. See [1.3.6](#) for more details.

1.5.3.2 Read global variables

After parsing the optional command line parameters the Setup will call the *ReadGlobalVariables* function. This function will:

- Read database home directory from the registry or InstallShield log file.
- Read the Setup INI XML file and check if the file contains a valid license string. The Setup will run in silent mode if the license string is valid. The Setup will abort if a Setup INI XML file has been specified using the `-ini` command line parameter, and the file does not contain a valid license string.
- Read the paths of the Setup log, installation and home directories from the Setup INI XML file.
- Read the requested features listed from the Setup INI XML file. The NetKey+ feature will only be available for installation if a valid network license has been specified in the Setup INI XML file.

1.5.3.3 Write global variables

The *WriteGlobalVariables* function will save the paths of the Setup log, installation and home directories to the Setup INI XML object, if the Setup is running in normal (non-silent) mode. This function will also save the registered user and organization names, and the license string to the Setup INI XML object.

1.5.3.4 Save Setup INI XML file

If the Setup is running in normal (non-silent) mode, the *XML_SaveIni* function will save the contents of the INI XML object from memory to the Setup INI XML file.

1.5.3.5 Read requested features

In silent mode, the *ReadGlobalVariables* function will read the requested features listed in the Setup INI XML file. The NetKey+ server program feature will only be available for installation if a valid network license has been specified in the Setup INI XML file.

1.5.3.6 Save Setup Log

The first time the *XML_SaveLogFile* function is called the Setup will generate a unique file name for the Setup log XML file. Next, the Setup will copy the following style sheet files to the Setup log folder: *processlogs.xsl*, *applied-maths.css*, *amheader.jpg* and *amlogo.gif*.

Finally, the *XML_SaveLogFile* function will save the contents of the Setup log XML object from memory to the Setup log XML file.

1.5.3.7 OnMoveData

The *OnMoveData* function is the main Setup process that handles the file transfer. First, the function will display the progress bar dialog and create the uninstall information in the registry. Next, the function will call the *CheckLicense* function to check and save the license string to the HKEY_LOCAL_MACHINE hive of the registry (if a valid license string was entered).

Subsequently, the *OnMoveData* process will call the *FeatureTransferData* function to install or remove feature files. The *FeatureTransferData* function will launch the *<feature>_Installing* or *<feature>_UnInstalling* function before installing or removing a feature. After a feature has been installed or removed the Setup will call the *<feature>_Installed* or *<feature>_UnInstalled* function.

Finally, the *OnMoveData* function will call the *LaunchNetKey* function to launch the NetKey+ server configuration tool if the corresponding feature was selected for installation.

1.5.3.8 Feature functions

Each feature can be linked to four event handlers:

- The *OnInstalling* event handler responds to the *Installing* event that is generated just before the corresponding feature is installed. This handler is linked to a *<feature>_Installing* function.
- The *OnUnInstalling* event handler responds to the *UnInstalling* event generated just before the corresponding feature is removed from the target system. This handler is linked to a *<feature>_UnInstalling* function.
- The *OnInstalled* event handler responds to the *Installed* event that is generated just after the corresponding feature has been installed. This handler is linked to a *<feature>_Installed* function.
- The *OnUnInstalled* event handler responds to the *UnInstalled* event generated just after the corresponding feature has been removed from the target system. This handler is linked to a *<feature>_UnInstalled* function.

Each *<feature>_Installing* and *<feature>_UnInstalling* function will call the *FeatureStart* function to create a *feature* node and a *start* time stamp element in the Setup log XML file. In addition, each *<feature>_Installed* and *<feature>_UnInstalled* function will call the *FeatureEnd* function to create an *end* time stamp element in the Setup log XML file.

The feature event handler functions that call other function in addition to the *FeatureStart* and *FeatureEnd* function are described in the next sections.

The *Application_Installing* event handler function is called by the Setup just before the main BioNumerics application feature is installed. First, this process will call the *DeleteOldFiles* function to delete legacy files from the BioNumerics program folder, which are no longer included in the current Setup package.

Next, the *Application_Installing* function will run the *vcredist_x86.exe* executable to install the Microsoft Visual C++ 2008 Redistributable Package (x86).

The *Application_Installed* event handler function is called by the Setup immediately after the application feature has been installed. This function will write the database home directory to the HKEY_CURRENT_USER hive.

If a network license string was entered, the *Application_Installed* function will read the NetKey+ server properties from the Setup INI XML file, and create or overwrite the NetKey.ini file in the common application data folder.

Finally, the function will create the shortcuts in the Startup menu and desktop folder.

The *Application_UnInstalled* event handler function is called by the Setup just after the main BioNumerics application feature has been removed. This function will call the *DeleteOldFiles* function to delete legacy files from the BioNumerics program folder, which are no longer included in the current Setup package.

The *Sentinel_Installed* event handler function is called by the Setup after the Sentinel drivers placeholder feature has been installed. This process will first call the *IsSentinelInstalled* function to check if the minimum required version of the Sentinel System Drivers is already installed. If the required version is not installed, or in repair maintenance mode, the *Sentinel_Installed* function will call the *HasDongles* function to check if hardware security keys are connected to the target computer. The appropriate warning messages will appear if existing hardware security keys were detected.

Next, the function will call the *MSI_InstallProduct* function to install the Sentinel System Driver Windows Installer package (e.g. Sentinel System Driver Installer 7.5.1.msi).

The *NetKey_Installing* event handler function is called by the Setup just before the NetKey+ server program feature is installed. First, this function will stop the NetKey+ service if it already exists on the target system. This will make sure that existing files are no longer in use, and will allow the Setup to overwrite these files if needed.

Next, the *NetKey_Installing* process will call the *IsOldNetKeyInstalled* function to delete conflicting versions of the NetKey+ service.

Finally, the function will grant full NTFS permissions to the built-in "NT AUTHORITY\SYSTEM" account for the Applied Maths common application data folder. This way the NetKey+ service running with the SYSTEM account will have sufficient privileges to create and modify files in the NetKey+ sub-folder.

The *NetKey_Installed* handler function is called by the Setup just after the NetKey+ server program feature has been installed. If the NetKey+ sub-folder in the Applied Maths common application data folder already contains a NetKey+_CONFIG.txt file, then the Setup will call the *WMI_ServiceStart* function to start the NetKey+ service.

The *NetKey_UnInstalling* event handler function is called by the Setup just before the NetKey+ server program feature is removed from the target system. This process will first call the *WMI_ServiceExists* function to verify if the NetKey+ service exists. If the service exists, then the Setup will check if the path of the service executable matches the program folder configured for the current instance. If both paths are equal then the function will call *WMI_ServiceStop* to stop the NetKey+ service.

If the running NetKey+ service is installed in a different folder than the program folder of the current BioNumerics instance then the service will not be stopped.

The *NetKey_UnInstalled* event handler function is called by the Setup just after the NetKey+ server program feature has been removed. This process will first call the *WMI_ServiceExists* function to verify if the NetKey+ service exists. If the service exists, then the Setup will check if the path of the service executable matches the program folder configured for the current instance. If both paths are equal, then the function will call the built-in *ServiceRemoveService* InstallShield function to remove the NetKey+ service.

If the running NetKey+ service is installed in a different folder than the program folder of the current BioNumerics instance, then the service will not be removed.

The *DeleteOldFiles* function will delete legacy files from the BioNumerics program folder, which are no

longer included in the current Setup package. Only legacy files with the following file extensions will be deleted from the program folder: .BXT,.DLL,.EXE,.AVI,.PYC and .XML.

The *IsSentinelInstalled* function will check the Windows Installer database to verify if the minimum required version of the Sentinel System Driver Installer is already installed. If the USB Driver feature is not installed, then the function assumes that the Sentinel System Driver package is incomplete, and will instruct the Setup to re-install the package.

The *HasDongles* function will launch the *setlic.exe* executable to verify if hardware security keys or dongles are connected to the target system. The function will check the exit code of the *setlic.exe* program to verify if dongles were detected.

In silent mode, the *CheckLicense* function will first attempt to read the license string from the Setup INI XML file. Next, the function will read the license string from the HKEY_LOCAL_MACHINE hive of the registry if the current string is empty. If the license string is still empty, the Setup will use the license string from the previous installation (in maintenance mode).

If the license string has the correct length, the Setup will launch the *setlic.exe* tool to get the license type of the entered string. The *setlic.exe* license tool will return one of the following constants: LIC_STANDALONE, LIC_NETWORK, LIC_INTERNET or LIC_INVALID.

If the *CheckLicense* function was called by the *OnMoveData* function, and the license type is valid (not LIC_INVALID), then the Setup will save the license string to the HKEY_LOCAL_MACHINE hive of the registry.

The *LaunchNetKey* function is called by the *OnMoveData* function to start the NetKey+ configuration tool after the NetKey+ server program feature has been installed, repaired or updated. The function will use the built-in *LaunchApp* InstallShield function to start the NetKey+Config.exe executable. The Setup will continue after the tool has been launched.

The *IsOldNetKeyInstalled* function will use Windows Management Instrumentation (WMI) queries to verify if other instances of the NetKey+ service are already installed. Optionally, this function can also be used to delete the service if the service name does not match, or if the installation path does not match the current BioNumerics program folder.

The service will not be deleted if the service name is NetKey+, and the path matches with the current BioNumerics program folder.

The *SetFilePermissions* function will use the *xcaccls.vbs* Microsoft Visual Basic script to grant NTFS folder permissions to a specific user. The Setup will launch the *xcaccls.vbs* script using the *cscript.exe* application in the 32-bit version of the Windows system folder.

The *MSI_InstallProduct* function will use the *msiexec.exe* Windows Installer tool to install an MSI package (e.g. Sentinel System Driver Installer 7.5.1.msi).

The *WMI_ServiceStop* function will first call the *WMI_ServiceExists* function to verify that the service exists. The function will attempt to stop the service if the service exists and is running. The *WMI_ServiceStop* function uses the built-in InstallShield functions to control the service on a local computer.

The *WMI_ServiceStart* function will first call the *WMI_ServiceExists* function to verify that the service exists. The function will attempt to start the service if the service exists and is not running. The *WMI_ServiceStart* function uses the built-in InstallShield functions to control the service on a local computer.

The *CleanUp* function will create the end time stamp element in the setup node of the Setup log XML file and close the progress bar dialog. Next, the *CleanUp* function will call the *XML_ShowLogFile* function to save and optionally display the Setup log file in Internet Explorer.

Finally, the *CleanUp* function will unload the *IsGetObj.dll* file from memory and will delete the file from the temporary Setup folder.

1.5.4 Setup Process list

Table [1.5.2](#) shows the main processes and functions that are used in the installation script, and that are displayed in the simplified Setup flow diagram (see Figure [1.5.1](#)).

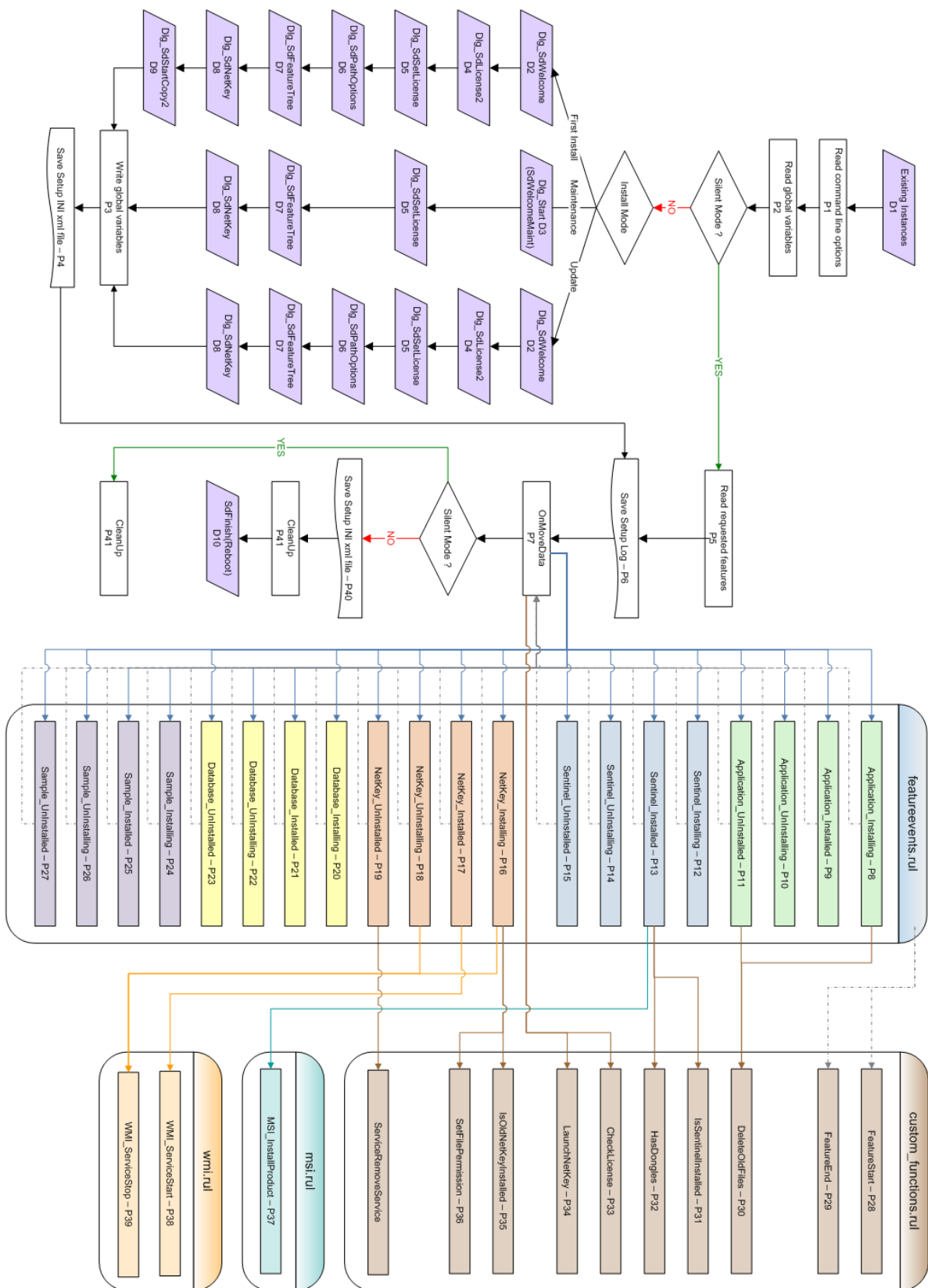


Figure 1.5.1: The Setup flow diagram.

Process number	Process name	Related section number
P1	Read command line options	1.5.3.1
P2	Read global variables	1.5.3.2
P3	Write global variables	1.5.3.3
P4	Save Setup INI xml file	1.5.3.4
P5	Read requested features	1.5.3.5
P6	Save Setup Log	1.5.3.6
P7	OnMoveData	1.5.3.7
P8	Application_Installing	1.5.3.8
P9	Application_Installed	1.5.3.8
P10	Application_UnInstalling	
P11	Application_UnInstalled	1.5.3.8
P12	Sentinel_Installing	
P13	Sentinel_Installed	1.5.3.8
P14	Sentinel_UnInstalling	
P15	Sentinel_UnInstalled	
P16	NetKey_Installing	1.5.3.8
P17	NetKey_Installed	1.5.3.8
P18	NetKey_UnInstalling	1.5.3.8
P19	NetKey_UnInstalled	1.5.3.8
P20	Database_Installing	
P21	Database_Installed	1.5.3.8
P22	Database_UnInstalling	
P23	Database_UnInstalled	
P24	Sample_Installing	
P25	Sample_Installed	
P26	Sample_UnInstalling	
P27	Sample_UnInstalled	
P28	FeatureStart	
P29	FeatureEnd	
P30	DeleteOldFiles	1.5.3.8
P31	IsSentinelInstalled	1.5.3.8
P32	HasDongles	1.5.3.8
P33	CheckLicense	1.5.3.8
P34	LaunchNetKey	1.5.3.8
P35	IsOldNetKeyInstalled	1.5.3.8
P36	SetFilePermissions	1.5.3.8
P37	MSI_InstallProduct	1.5.3.8
P38	WMI_ServiceStart	1.5.3.8
P39	WMI_ServiceStop	1.5.3.8
P40	Save Setup INI xml file	1.5.3.4
P41	CleanUp	1.5.3.8

Table 1.5.2: The Setup process list.

Chapter 1.6

Command line options

1.6.1 Running BioNumerics from the command line

The BioNumerics software (bn.exe, see [2.1](#)) can be started from the command line. This can be done by opening a command prompt, navigating to the BioNumerics installation directory (or opening the command prompt immediately in this directory) and entering `bn.exe`. See [Figure 1.6.1](#) for an example.

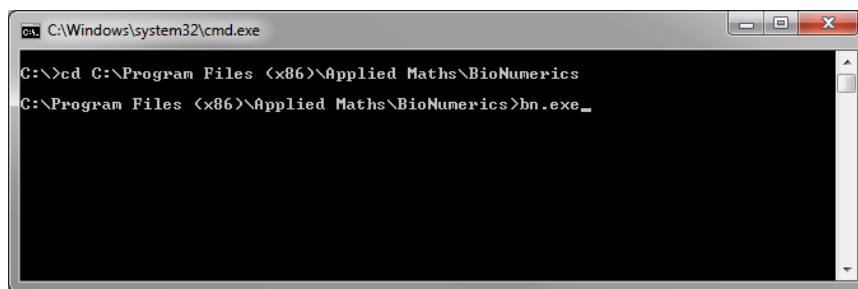


Figure 1.6.1: Running BioNumerics from the command line.

When the executable is called without any options, the program will open the last-opened database (as read from the Windows registry). However, the flexibility associated with running BioNumerics from the command line comes with the additional options that can be specified. Following is a list of available options with their values:

- `-database=<DBNAME>`: The BioNumerics database that will be opened, `<DBNAME>` is the name of the database folder (without the path).
- `-homedir=<HOMEDIR>`: The BioNumerics home directory (see [3.1](#)), `<HOMEDIR>` is the full path to the home directory.
- `-bnuser=<USERNAME>`: The BioNumerics database user (see [3.5](#)).
- `-bnpwd=<PWD>`: The password for the specified database user.
- `-licensestring=<LIC>`: The license string (see [1.4.3](#)) needed to activate the software license.
- `-runbnstart=(0|1)`: Whether or not the startup program (see [3.1](#)) should be ran after the main program is closed.
- `-logfile`: Allows to specify a custom log file, different from the default `BNLOG.TXT`. The custom log file needs to reside in the BioNumerics home directory.

- `-id=<ID>`: The ID which should be written in the protection dongle.
- `-script=<PATH>`: Runs a script and does not open the *Main* window by default, `<PATH>` is the full path to the script file.
- `-openmain=(0|1)`: Whether or not the *Main* window should be opened. This option is only valid in combination with the `-script` option.
- `-dbmanagement=backup`: Takes a backup of the BioNumerics database specified under `-database=<DBNAME>`. Requires the specification of a backup file (see below).
- `-backupfile=<PATH>`: Only used in combination with `-dbmanagement=backup`. `<PATH>` is the directory in which the backup file (.bnbk) will be created. The .bnbk file name will be generated automatically and will consist of the database name and the date when the backup is made.
- `-silent=(0|1)`: Only used in combination with `-dbmanagement=backup`. In silent mode (`-silent=1`), no error messages will be displayed during the creation of a database backup.



As always the case with the Windows command prompt, file paths that contain spaces should be enclosed with double quotes.

The command line syntax is quite flexible:

Options can be provided "as is" or they can start with a "-" (hyphen) or "/" (slash). Examples:

```
bn.exe database=DemoBase
bn.exe -database=DemoBase
bn.exe /database=DemoBase
```

Options are not case sensitive. Examples:

```
bn.exe database=DemoBase
bn.exe DataBase=DemoBase
bn.exe DATABASE=DemoBase
```

Options and their values can optionally be quoted. Examples:

```
bn.exe "database=DemoBase"
bn.exe database="DemoBase"
bn.exe database=DemoBase
```

Option names and their values can be separated with ":" or "=". Examples:

```
bn.exe database=DemoBase
bn.exe database:DemoBase
```

1.6.2 Running the startup program from the command line

The BioNumerics software startup program (BnStart.exe, see 3.1) can be started from the command line. Following options are available:

- `-homedir=<HOMEDIR>`: the BioNumerics home directory, `<HOMEDIR>` is the full path to the home directory
- `-licensestring=<LIC>`: the license string (see 1.4.3), needed to activate the software license

These options will be passed on to bn.exe (see 1.6.1).

Chapter 1.7

Granting access to BioNumerics databases

During the installation of the BioNumerics application, the Setup will create a Windows group named *BioNumerics Database Administrators* (Figure 1.7.1). This local Windows group has Full control NTFS permissions on the local Database home directory.

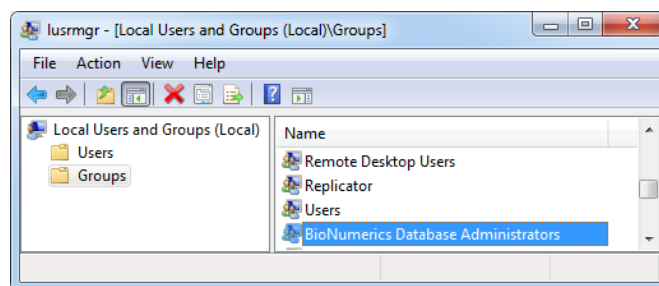


Figure 1.7.1: The BioNumerics Database Administrators Windows group.

By default, the following users and groups are members of the local BioNumerics Database Administrators Windows group:

- User running the BioNumerics Setup
- NT AUTHORITY\Authenticated Users

The Local Users and Groups management console in Figure 1.7.1 can be started by running `lusrmgr.msc` on a Windows Command Prompt. Double-click on the BioNumerics Database Administrators Windows group to view the current group members (Figure 1.7.2). Click the **<Add>** or **<Remove>** button to change the group members.

If you do not want all authenticated users to have full access to the BioNumerics Databases you can simply remove the NT AUTHORITY\Authenticated Users group from the BioNumerics Database Administrators group, and replace it with specific users or groups that require database access. For example, you could add all BioNumerics users to an Active Directory Security Group, and add this group to the local BioNumerics Database Administrators Windows group to grant full access to the databases.

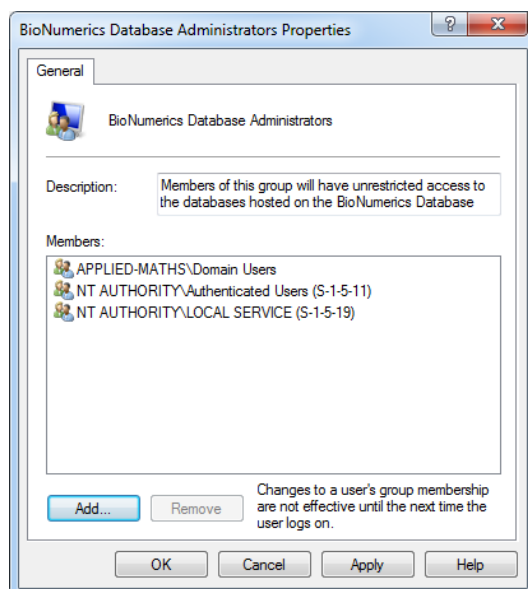


Figure 1.7.2: Properties of the BioNumerics Database Administrators Windows group.

Part 2

Concepts

Chapter 2.1

The concepts of BioNumerics





2.1.1 The programs

The BioNumerics software is composed of two executable units: a **Startup program** that creates and manages the *databases* and associated directories and that starts the **Analyze program** with a selected database. All import and analysis functions are done in the Analyze program.

2.1.2 The database and the experiments

The basis of BioNumerics is a *database* consisting of *entries*. The entries correspond to the individual organisms or samples under study: animals, plants, fungi, bacterial or viral strains, organic samples, tissue samples,.... Each database entry is characterized by a unique *key*, assigned either automatically by the software or manually, and by a number of user-defined information fields. The organization and functions of BioNumerics databases are discussed in 3. Each entry in a database may be characterized by one or more *experiments* that can be linked easily to the entry. What we call experiments in BioNumerics are in fact the experimental data that are the numerical results of the biological experiments or assays performed to estimate the relationship between the samples. In BioNumerics, experiments are divided in eight classes: *Fingerprint types*, *Spectrum types*, *Character types*, *Sequence types*, *Sequence read sets*, *Trend data types*, *Whole genome maps*, and *Matrix types*.

- **Fingerprint types** 📊 (4.1): Any densitometric record seen as a one-dimensional profile of peaks or bands can be considered as a fingerprint type. Examples are of course gel and capillary electrophoresis patterns, but also gas chromatography or HPLC profiles, spectrophotometric curves, etc.. Fingerprint types can be derived from TIFF or bitmap files as well, which are two-dimensional bitmaps. The condition is that one must be able to translate the patterns into densitometric curves.
- **Spectrum types** 📊 (5): This experiment type shares some features with the Fingerprint types, but is specifically designed to hold spectral data, such as mass spectra (MALDI, SELDI).
- **Character types** 📊 (6.1): Any array of named characters, binary or continuous, with fixed or undefined length can be classified within the character types. The main difference between character types and fingerprint types is that in the character types, each character has a well-determined name, whereas in the electrophoresis types, the bands, peaks or densitometric values are unnamed (a molecular size is NOT a well-determined name!).
- **Sequence types** 📊 (8.1): Within the sequence types, the user can assemble, import and edit nucleic acid (DNA and RNA) sequences and amino acid (protein) sequences.

- **Sequence read sets**  (9): Will hold sets of short sequence reads, generated by high-throughput sequencers such as Roche/454[®] or Solexa/Illumina[®]. Sequence read sets can be pre-processed and assembled into sequences (see 18) or used for characterizing microbial communities via deep sequencing (see 19).
- **Trend data types**  (7.1): Reactions to certain substrates or conditions are sometimes recorded as multiple readings in function of a changing factor, defining a trend. Examples are the kinetic analysis of metabolic and enzymatic activity, real-time PCR, or time-course experiments using microarrays. Although multiple readings per experiment are mostly done in function of time, they can also depend on another factor, for example readings in function of different concentrations.
- **Whole genome maps**  (12): Analyzes high resolution, ordered whole genome restriction maps from single microbial DNA molecules obtained from the ArgusTM Optical Mapping System (<http://www.opgen.com>).
- **Matrix types**  (10.2): This is not a native experiment type, but the result of a comparison between database entries, expressed as similarity values between certain database entries. An example of a matrix type is a matrix of DNA homology values. DNA homology between organisms can only be expressed as pairwise similarity, not as native character data.

Most experiment types correspond to a *module* of the BioNumerics software (see 2.1.4).






Essentially, adding a single organism (entry) with its associated experiments to the database constitutes several steps (see Figure 2.1.1).

2.1.3 Multi-database setup


BioNumerics is a multi-database software, which supports the setup of different users in Windows. It is very important to understand the hierarchical structure of the user, database, and experiment setup in order to make optimal use of these features.

The BioNumerics users are associated with the Windows login users. Each Windows user can specify his/her BioNumerics databases directory, and BioNumerics saves this information in the user's system registry. For example, suppose that user X logs in on a Windows machine with BioNumerics installed. This user can create a directory, and specify this directory as the **home directory** (see 3.1.2) in the Startup program. BioNumerics will save this information in this user's system registry, so that each time the user logs in, BioNumerics will automatically consider the same directory as the home directory. In this way, each Windows user can define his/her own BioNumerics home directory, without interfering with other users. Within this home directory, the user can specify as many databases as desired.

2.1.4 Modules and features

The BioNumerics software consists of five data modules and five analysis modules. The **data modules** Whole Genome Map data module  and Trend data module  correspond to the experiment types described in 2.1.2. The Fingerprint data module  handles both the Fingerprint types and the Spectrum types. The Character data module  deals with Character types as well as Matrix types and the Sequence data module  covers both the Sequence types and the Sequence read sets.

The five **analysis modules** are not necessarily linked to a specific experiment type, but rather offer additional functionality for all or some experiment types:

- The Tree and network inference module  allows the user to create comparisons (see 13) and groups all functionality regarding cluster analysis.

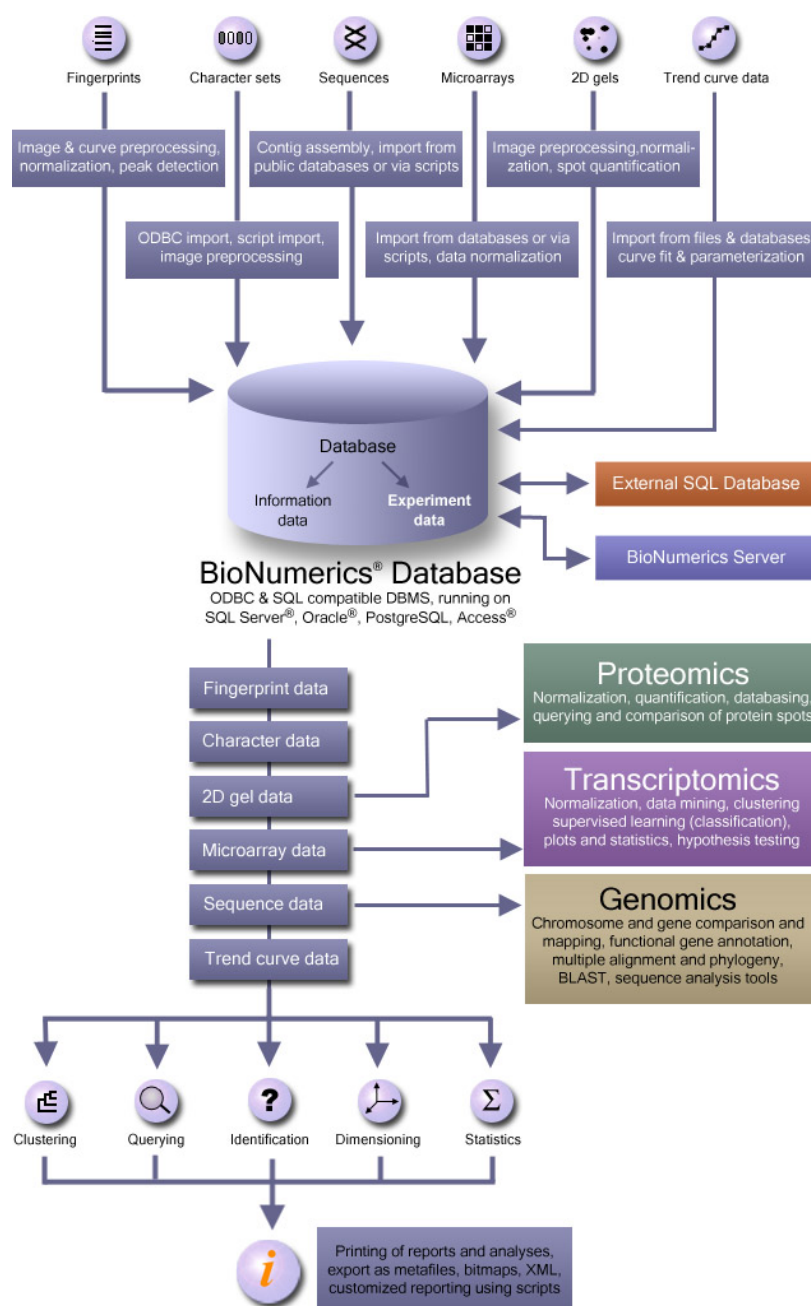


Figure 2.1.1: Flow chart of main steps in the acquisition and analysis of data in BioNumerics.

- The Classifiers and Identification module **ID** allows the user to identify unknown entries using the database, identification projects and decision networks (see 15).
- The Dimensioning and Matrix Mining module **DI** offers several non-hierarchical grouping methods, such as Principal Component Analysis, Self-Organizing Maps and Multidimensional Scaling (see 17.4). In addition, it comprises powerful statistics (see 17) and matrix mining tools (see 20).
- With the Audit trails and Versioning module **AT** every action made to a database *object* can be recorded with indication of the kind, the time of change, and user (see 3.6).
- The Genome Analysis Tools module **GA** allows the user to compare sequences of up to full chromosome length (see 8.6 and 8.7), to perform ORF-based annotation of chromosomes (see 8.8) and to characterize microbial communities by deep sequencing of a phylogenetic marker gene (see 19).

For each section in this manual where specific features are described, the required modules will be indicated in the section title.

The specific BioNumerics package you are working with might not include all modules. To check which modules are present, select **File > About...** in the *Main* window. This pops up the *About* dialog box (see Figure 2.1.2).

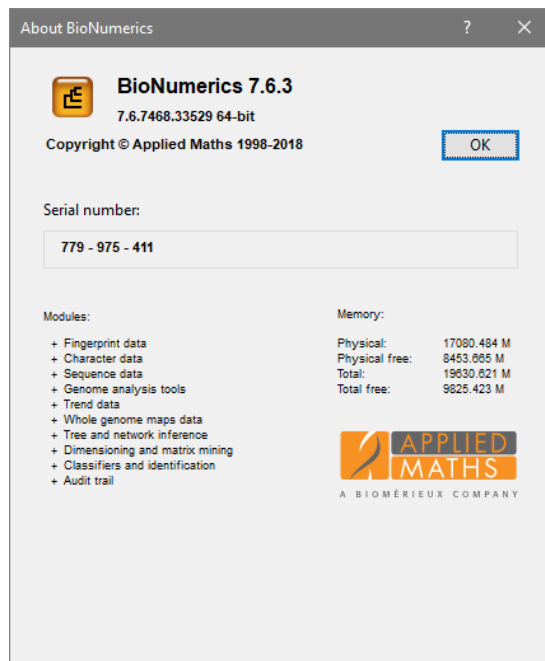


Figure 2.1.2: The *About* dialog box, containing information about the installed BioNumerics package.

This dialog box shows the version of the software, the package serial number, and a list with modules. A module is present in the installed BioNumerics package when the module name is preceded by a plus sign.

In the bottom of the dialog box, information is given about the installed language packs.

2.1.5 BioNumerics plugins

2.1.5.1 Introduction

Plugins offer extra functionality to the BioNumerics core software, often to import or export various types of data, but also to deal with specific applications, such as multi-locus sequence typing (MLST), variable number of tandem repeats (VNTR) analysis, etc.. They can also provide extra functions related to dendrogram analysis, statistics, database management, etc.. Plugins are tools written in the BioNumerics or Python[®] script language, available as binary encoded packages.

If a database is opened for the first time, the *Plugins* dialog box (see Figure 2.1.3) will appear by default. If the database has already been opened previously, the *Plugins* dialog box can be called from the *Main* window by selecting **File > Install / remove plugins...** (🔧).

BioNumerics looks in three different locations to make an inventory of available plugins:

1. In the BPL folder inside the home directory (see 3.1.2).
2. In the relational database (see 2.1.5.3).

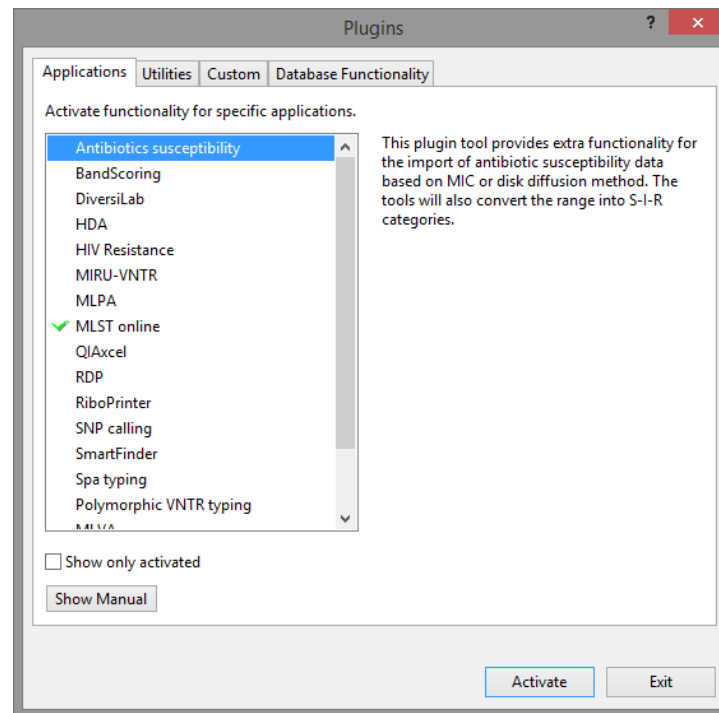


Figure 2.1.3: The *Plugins* dialog box lists the available BioNumerics plugins.

3. In the BPL folder inside the BioNumerics installation directory.

These locations are checked in the order as listed above. When a plugin (as identified by its file name) occurs in more than one location, the last checked location takes precedence. For plugins that are part of the BioNumerics setup, this means that always the version that came with the installation will be used. For development and testing of custom plugins, this system has the advantage that the plugin can temporarily be stored in the home directory or database (no administrator rights needed) instead of in the installation directory.

Regardless of their file location, three types of BioNumerics plugins exist:

- **Applications:** Plugins dealing with specific biological applications or (typing) techniques.
- **Utilities:** Plugins offering extra functionality, which can be used in a more general context.
- **Custom:** Tailor-made plugins for our customers, which are not part of the BioNumerics setup.

These plugin types are listed in their corresponding tab in the *Plugins* dialog box. The last tab, i.e. *Database Functionality*, lists all plugins and scripts that are stored in the relational database and also provides access to online plugins (see 2.1.5.3).

When a particular plugin is highlighted, a short description appears in the right panel. For plugins that are not part of the BioNumerics setup, the version number is shown.

A highlighted plugin can be installed with the **<Activate>** button. The software will ask for confirmation before installation. Some plugins depend on functionality offered by specific BioNumerics modules (see 2.1.4). If a required module is missing, the plugin cannot be installed and an error message will be generated. Once a plugin is installed, it is marked with a green V-sign. It can be removed again with the **<Deactivate>** button.

If documentation exists for the highlighted plugin, the **<Show manual>** button will be active. Pressing this button will open the BioNumerics online help (see 2.2.1) with a description of the plugin's functionality.

If you have finished installing plugins, you can close the dialog box with **<Exit>** (or **<Proceed>** when in the process of creating a new database).

When installed, a plugin installs itself in one or more menus of the software. Plugins might also add buttons, panels and even complete windows to BioNumerics.

2.1.5.2 Plugins included with the installation

The plugins listed below are included in the BioNumerics setup and available from the *Applications tab* and the *Utilities tab* of the *Plugins* dialog box, respectively.

Applications:

- *Antibiotics susceptibility plugin* (CH, ID): Automates the assessment of antibiotic resistance (Sensitive, Intermediate or Resistant) based on user-defined cut-off values for disk diffusion (zone diameter) tests and Minimum Inhibitory Concentration (MIC) values.
- *BandScoring plugin* (FP, CH): Facilitates the scoring of bands, corresponding to genetic markers for eukaryotic organisms. Band presence and zygosity is determined and exported to character data or text files.
- *DiversiLab plugin* (FP, TN, DI): Contains tools for importing and analyzing densitometric traces from the DiversiLabTM system (bioMérieux SA).
- *HDA plugin* (FP, CH, DI): Contains all necessary tools for automated Hetero Duplex Analysis (HDA), based on Conformational Specific Capillary Electrophoresis (CSCE).
- *HIV resistance plugin* (SQ, CH, ID): Runs algorithms to determine HIV-1 resistance to retro-viral drugs based on sequence data. Drug resistance reports are created.
- *MIRU-VNTR plugin* (FP, CH): Offers the tools to import and analyze MIRU-VNTR data for typing of *Mycobacterium* strains.
- *MLPA plugin* (FP, CH): Provides an automated work flow for the analysis of Multiplex Ligation-dependent Probe Amplification and related techniques.
- *MLST online plugin* (SQ, CH): Offers the tools to automatically set up Multi-Locus Sequence Typing (MLST) experiments, linked to online MLST data repositories.
- *QIAxcel plugin* (FP): Contains import tools for densitometric traces generated by the QIAxcel Advanced system (QIAGEN).
- *RDP plugin* (SQ): Offers tools to search for the nearest 16S rRNA sequences in the Ribosomal Database Project (<http://rdp.cme.msu.edu/>).
- *RiboPrinter plugin* (FP): Imports text and XML files generated by the RiboPrinter system (DuPont) as fingerprints.
- *SNP calling plugin* (CH): Performs an automated cluster calling for single-nucleotide polymorphism (SNP) genotyping using TaqMan[®] or related technologies.
- *SmartFinder plugin* (TD, CH): Provides the functionality to import and analyze real-time PCR data based on the SmartFinder[®] technology, developed by PathoFinder B.V..
- *Spa Typing plugin* (SQ, TN): Provides the functionality to perform spa-typing on *Staphylococcus aureus*.

- *Polymorphic VNTR typing plugin* (SQ): Provides the functionality to perform polymorphic VNTR typing (also called tandem repeat sequence Typing).
- *MLVA plugin* (FP, CH): Offers the tools to import and analyze Variable Number of Tandem Repeat (VNTR) and Multi-Locus VNTR Analysis (MLVA) data.
- *WGS tools plugin* (SQ, CH): Client tools for performing Whole Genome Multi Locus Sequence Typing (wgMLST) and reference mapping on the BioNumerics Calculation Engine.

Utilities:

- *Database tools plugin*: Offers additional search functions (fuzzy search, find and replace) and database layout tools.
- *Dendrogram tools plugin* (TN): Contains tools for working with dendrograms in the *Comparison* window.
- *Fingerprint processing reports plugin* (FP): Contains tools for exporting data from the *Fingerprint processing* window.
- *Geographical plugin*: Plots database entries on a geographical map, e.g. for epidemiological investigations.
- *Sequence extraction plugin* (SQ): Extracts a subsequence from a (whole genome) *origin* sequence type and stores the sequence into a *destination* sequence type using a BLAST approach.
- *Sequence translation tools plugin* (SQ): Translates nucleic acid sequences into amino acid sequences.
- *User management tools plugin*: Tools to export and import user management settings between databases.

In addition, some plugins are made available via the Applied Maths website, from which they can be downloaded and stored in the database (see 2.1.5.3).

2.1.5.3 Database plugins

The *Database functionality tab* in the *Plugins* dialog box lists the plugins and scripts (for more information about scripts, see 2.1.6) that are currently stored in the relational database. In a new BioNumerics database, this list will initially be empty (see Figure 2.1.4).

Installation in the relational database has as advantage that no Windows administrator rights are required. Furthermore, in a multi-user database setup, this procedure ensures that all database users work with the same plugin or script version.



Plugins that are included in the BioNumerics setup (see 2.1.5.2) cannot be installed in the database. See 2.1.5.1 for the locations and order in which plugin files are detected.

To remove a plugin or script from the database, highlight it in the list and press the **<Delete>** button. If documentation is available for a plugin or script, pressing **<Show manual>** will open the BioNumerics online help (see 2.2.1) with a description of the plugin's functionality.

To add a new database plugin or script or to update an existing one are installed by pressing the **<Add / Update...>** button. This action opens the *Database plugins* dialog box (see Figure 2.1.5).

Plugins and scripts can be saved in the database from two possible sources: from files on your hard drive, external drive or local file server (**Load from file**) and from the Applied Maths website (**Online plugins**).

In case of local files, pressing the **<Browse...>** button allows you to browse for one or more files of following types:

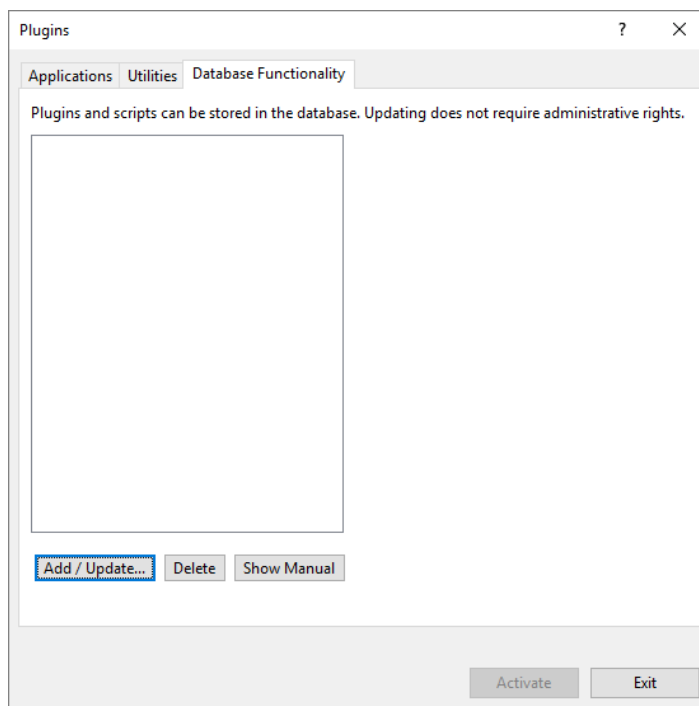


Figure 2.1.4: The *Database functionality* tab in the *Plugins* dialog box.

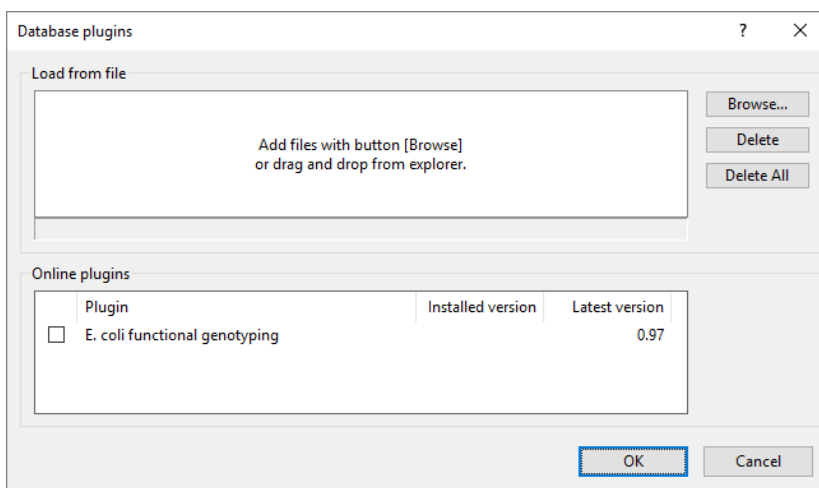


Figure 2.1.5: The *Database plugins* dialog box.

- A BioNumerics plugin, i.e. an encoded text file with the extension .BPL.
- A script in the BioNumerics script language (file extension .BNS) or the Python[®] programming language (file extension .py).
- A ZIP archive containing one or more BioNumerics scripts.

To install or update a plugin available from the Applied Maths website, simply check the corresponding box in the **Online plugins** list. If the plugin is already installed in the database and the version number of the installed version is smaller than the latest version, the check box will be checked automatically.

When the <OK> button is pressed the selected plugins and/or scripts will be saved into the database. In case of online plugins, the plugin needs to be downloaded first. Depending on the size of the plugin file, this

can take from a few seconds up to several minutes. The software will ask for confirmation before actually installing any plugin(s) or before extracting script(s) from a ZIP archive.



For the download of online plugins, a security mechanism is in place that compares the timestamp of the request with the server's time. If the list of **Online plugins** appears empty, please check if the date and time settings on your computer are correct (Windows Control Panel > Date and Time).

2.1.6 Script languages

BioNumerics is a very comprehensive software package which has many data import, export and analysis functions already included in the software. Additional functionality – often related to specific applications – is bundled into convenient plugins (see 2.1.5). However, ultimate flexibility is offered by BioNumerics' own script language and the Python[®] programming language. Both programming languages contain a large array of specialized functions, that allows the user to create scripts (i.e. small programs) for automation of specific tasks, e.g. the import of own data formats.

The *Script editor* window can be opened from the *Main* window by selecting **Scripts > BNS script editor...** (**Ctrl+F10**). For a description of the *Script editor* window and an explanation of the numerous script functions available, we refer to the separate script manual.

The *Python Script* window can be opened from the *Main* window by selecting **Scripts > Python script editor...** (**F10**). For further documentation on the Python[®] language, we like to refer to the literature advised by Python[®] (<http://www.python.org/doc/> and <http://wiki.python.org/moin/PythonBooks>).

The functionality to edit scripts is available in any BioNumerics configuration, but depending on the software configuration (modules present or not, see 2.1.4), some script functions might be disabled.

A number of general scripts are available on the website of Applied Maths. These scripts can be launched from the *Main* window using **Scripts > Browse internet...**

In the *Script from Web* window, click on a category to display the relevant scripts. Using **File > Execute/download** (⚙️), one can toggle between executing a script directly over the internet or saving it on your hard drive for later use. In the latter case, the script code will open in your default text editor (e.g. Notepad), from where the script can be saved in a destination folder of your choice. With **File > Go Back** (⬅️) and **File > Go Forward** (➡️), one can browse through the different script categories. Use **File > Exit** (**Alt+F4**) to close the *Script from Web* window.

Scripts can be saved in any folder on your computer or network drive and can be executed using **Scripts > Run script from file...**

To make the script in the BioNumerics script language appear as a menu item in the **Scripts** menu, it should be saved in one of the following locations:

- The Scripts sub-folder of the BioNumerics *installation directory* (C: \Program Files \Applied Maths \BioNumerics by default): this makes the script available as menu item in all databases. Please note that users without administrator rights might not have access to this directory.
- The Scripts Home sub-folder of the *home directory*: makes the script available in all databases that are located in that directory.
- The Scripts sub-folder of the *database folder*: if you only want the script to appear as a menu item in that specific database.
- As a record in the relational database (via the *Plugins* dialog box; see 2.1.5): makes the script available to all users of this database.

To make a Python[®] script appear as a menu item in the *Scripts* menu, it should be saved in a Python sub-folder of the directories mentioned above.

2.1.7 Example databases

Example or demonstration databases, which are used in tutorials for illustration of BioNumerics functionality, can be downloaded and automatically installed. For this purpose, a **Download example databases** link is present in the lower right corner of the *BioNumerics Startup* window (see 3.1.1). If the computer has an internet connection and is allowed to access the <http://www.applied-maths.com> website, clicking this link will display the *Tutorial databases* window (see Figure 2.1.6).

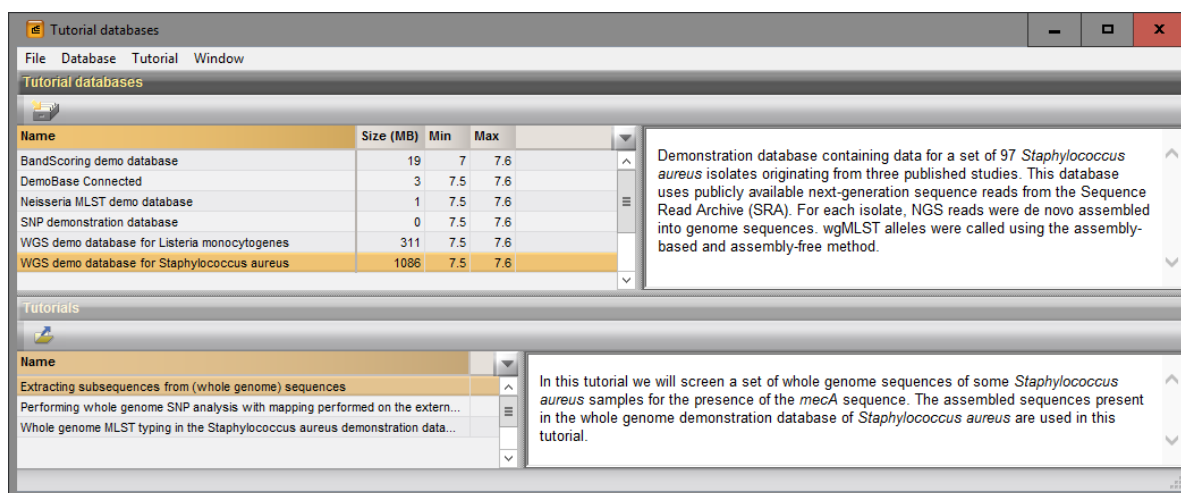


Figure 2.1.6: The *Tutorial databases* window.

The left part of the *Tutorial databases* panel provides an overview of all available tutorial databases with their download size (in MB) and compatibility with BioNumerics versions (the minimum and maximum compatible software version numbers are listed in the respective columns). For the currently highlighted tutorial database, a short description appears on the right-hand side.

Online tutorials (see <http://www.applied-maths.com/tutorials>) that use the highlighted database are listed in the *Tutorials* panel. For the currently highlighted tutorial, a short description appears on the right-hand side. Double-clicking a tutorial from the list or selecting **Tutorial > Open...** (📄) will open this tutorial in the default web browser.

To download a database, select it from the list in the *Tutorial databases* panel and select **Database > Download** (📄). Confirm the download and installation with <Yes>.

In case a database with the same name already exists, a second confirmation will appear, asking for permission to overwrite the existing database with the downloaded one.

Alternatively, if the computer on which BioNumerics is installed has no internet connection, a BioNumerics database back-up file can be downloaded via another computer from <http://www.applied-maths.com/download/sample-data>. The downloaded *.bnbk file can then be restored from the *BioNumerics Startup* window as described in 3.1.5.2.

The tutorial database **DemoBase Connected** contains experimental data on some fictitious (bacterial) genera and will serve repeatedly as a tutorial and example in this guide. The database contains the following experiment types:

- **Fingerprint types**

- **RFLP** Two different RFLP techniques, called **RFLP1** and **RFLP2**, resulting in two patterns for each bacterial strain.
- **AFLP** Amplified Fragment Length Polymorphism profiles (AFLP), run on an AB PRISM 310 Genetic Analyzer (Applied Biosystems).

- **Character types**

- **FAME** Fatty Acid Methyl Ester (FAME) profiles obtained on a Hewlett Packard 5890A gas-liquid chromatography instrument. This is a typical example of an *open* data set: the number of fatty acids found depends on the group of entries analyzed. If more entries are added, more fatty acids will probably be found. Furthermore, FAME profiles are an example of a *continuous* character type: the percentage occurrence of a fatty acid in a bacterium can have any real value between zero and 100%.
- **PhenoTest** This is a fictitious phenotypic test assay that reveals the metabolic activity or enzyme activities of bacteria on 19 different compounds. The first cup of the test is a blank control. This is an example of a *closed* data set: the 20 characters are well-defined, and regardless of the number of entries examined, the number of characters in the experiment will always remain 20. Real examples of such types of assays exist as commercial test panels available on micro plates or galleries. They can be interpreted in two ways. One can read the reactions by eye and score them as positive or negative; in this case the character type is *binary*. If the micro plates are read automatically using a micro plate reader, the reactions in the cups may have any real value between an OD of zero and 2.5 to 3.0, which is a *continuous* character type. In the example database, the reactions are scored as continuous characters.

In addition to the binary and continuous character types, one can also distinguish the *semi-quantitative* character types. These are tests that can have a number of discrete values, e.g. 0, 1, 2, 3, 4, or 5. In practice, a number of continuous character types are interpreted as multi-state characters for convenience.

- **Sequence types**

- **16S rDNA** For all strains, the nearly complete 16S ribosomal RNA gene has been sequenced. The sequences are approximately 1500 bases long, but not all of them are sequenced completely.

- **Matrix types**


- **DNA-Hybrid** A partial homology matrix based upon hybridization of total genomic DNA has been generated for the genera.

Additional example data can be downloaded from the Applied Maths website: <http://www.applied-maths.com/download/sample-data>.

Chapter 2.2

About this guide

2.2.1 Documentation formats

The BioNumerics documentation is available in two different formats: as a PDF file that can be opened in a PDF viewer such as Adobe Reader and as *Online help*, displayed in the *Help* window. The latter window (see Figure 2.2.1) can be called for most dialog boxes in BioNumerics by clicking on the question mark in the upper right corner of the dialog () and for nearly all windows by selecting **Help > Help on window**. The *Help* window also provides direct links to all available PDF manuals.

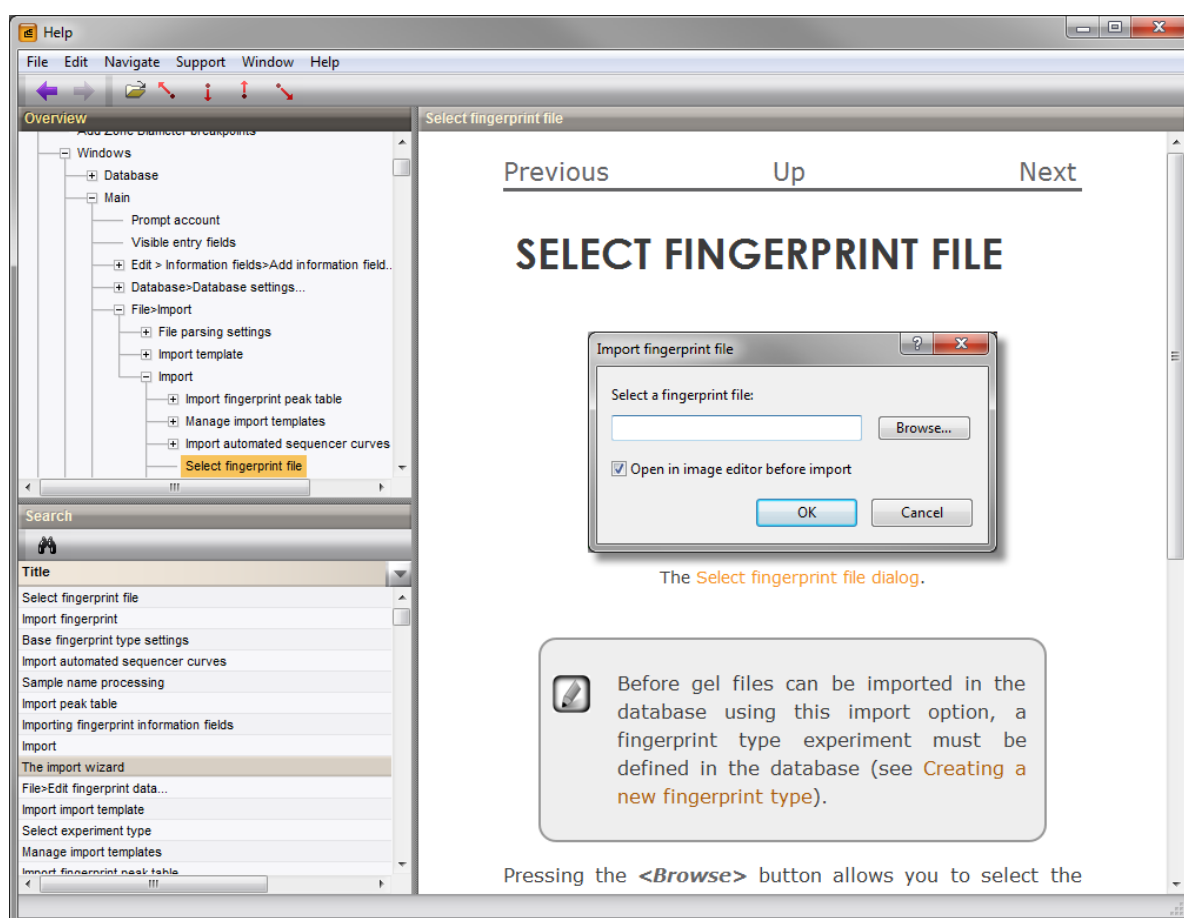



Figure 2.2.1: The *Help* window, displaying help for a BioNumerics dialog box.

This window consists of three dockable panels:


- The *Help panel* shows the actual help text for a certain topic. A help topic can be a dialog box, window, etc. that is being described or any section in the reference manual. The text can contain hyperlinks, linking to other topics.
- The *Overview panel* displays a tree-like overview of all available help topics for easy navigation. The help topic that is currently being displayed in the *Help panel* is highlighted in the tree.
- The *Overview panel* lists the results from a search action.

With the **Previous**, **Up** and **Next** hyperlinks in the header and footer of the *Help panel*, it is possible to browse through the help topics. The same can be done with the commands **Navigate > Go previous** (⏮), **Navigate > Go next** (⏭), **Navigate > Go up** (⏶) and **Navigate > Go down** (⏷) or simply by clicking on a help topic in the *Overview panel*.

A previously displayed help topic (regardless of its position in the tree of the *Overview panel*) can be shown with **Navigate > Go Back** (⏮). Use **Navigate > Go Forward** (⏭) to go to the next page.

A PDF version is available when the  icon appears in the *Overview panel* next to a help topic. Simply click on the icon of a displayed help topic to open the PDF version in your computer's default PDF viewer.



When the  icon of a help topic that is not currently displayed in the *Help panel* is clicked, the help topic will be displayed on the first click. Clicking the icon a second time will open the PDF.

To search the online help for a certain keyword or phrase, select **Navigate > Search...** (🔍, **Ctrl+F**). This action will display the *Search* dialog box.

Enter a search phrase in the *Search* dialog box and press **<OK>**. The results will be listed in the *Search panel* of the *Help* window.

Clicking on a search result will open the help topic in the *Help panel*.

By default, the *Help* window will fetch the help topics from the Applied Maths website. This is the recommended option, as it ensures that the most recent information is being displayed. However, on computers without access to the internet, an offline version of the help can be installed.



As the search tool (**Navigate > Search...** (🔍, **Ctrl+F**)) depends on a web service running on the Applied Maths website, this functionality will not be available when installing an offline version of the help.

First, select **Edit > Settings...** to display the *Settings* dialog box (see Figure 2.2.2).

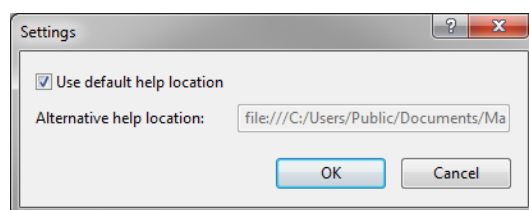


Figure 2.2.2: The *Settings* dialog box, to change the default help location.

With **Use default help location** checked, the software will use the help topics available on the Applied Maths website (online help). When this option is unchecked, an **Alternative help location** can be entered in the corresponding text box. Enter the prefix “file:///”, followed by the directory where the `help.xhtml` is located (see below). Press **<OK>** to accept the setting. It might be necessary to close and re-open the *Help* window before the new help location is used.

An offline version of the online help can be obtained from the Applied Maths help desk. This ZIP archive, containing `help.xhtml`, help topics, images and PDF manuals, needs to be unzipped to the location specified above.

Using **Support** > **Knowledge base**, the Applied Maths Knowledge Base (<http://www.applied-maths.com/knowledge-base>) can be consulted, which contains numerous frequently and less frequently asked questions about BioNumerics.

The Applied Maths help desk can be contacted via **Support** > **Ask a support question**. The BioNumerics version number, detailed version and serial number fields on the web form will be automatically filled in.

To check if an update of the BioNumerics software is available on the Applied Maths website, select **Version** > **Check for new version**. A confirmation message will appear when the software is up-to-date. If not, the *New version* dialog box will pop up (see Figure 2.2.3).

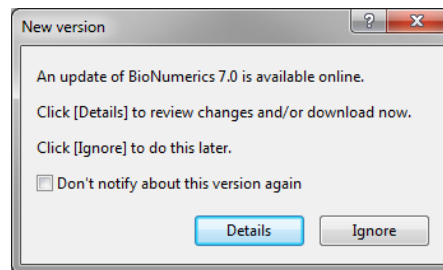


Figure 2.2.3: The *New version* dialog box.

This dialog box appears when a BioNumerics update is available online, either when a database is opened or when selecting **Version** > **Check for new version** in the *Help* window. Pressing <Ignore> will simply close this message. Pressing <Details> will open a web page with information about the update and a download link in your default browser.

When *Don't notify about this version again* is checked, the software will not prompt about the update again until a newer version is made available. This option is not shown when the dialog is called via **Version** > **Check for new version**.

2.2.2 Conventions

In the sections that follow, all menu commands are typed *like this*. Sub-menus are separated from parent menus by a "greater than" sign (>). Button text is always typed <Like this>, i.e. bold-italic and between < and > signs.

For example, the following menu command (Figure 2.2.4) will be indicated as **Edit** > **Information fields** > **Sort by field**.

Throughout this guide, commands will be mostly indicated with their menu command, button and keyboard shortcut (if present), for example: **Edit** > **Move entry up** (↑, Shift+Up).

In Figure 2.2.5, the following buttons are indicated as **Consider absent values as zero** (check box), <OK>, <Cancel>, <Apply> (disabled).

Each window and dialog box described in the guide will be given a name. This name is shown in italic, and usually corresponds to the name in the caption of the window or dialog box. For example, the dialog box in Figure 2.2.5 will be called the *Character settings* dialog box.

Descriptive text, such as explaining the layout of windows, describing the function of available menu items and buttons, or providing background information on the use of different algorithms, etc., is always displayed in normal text layout (such as the present paragraph), without preceding paragraph number. Tutorial text, which guides the user through the program by applying the available analysis functions on example data, is always preceded by a paragraph number for easy reference. This is illustrated in the following

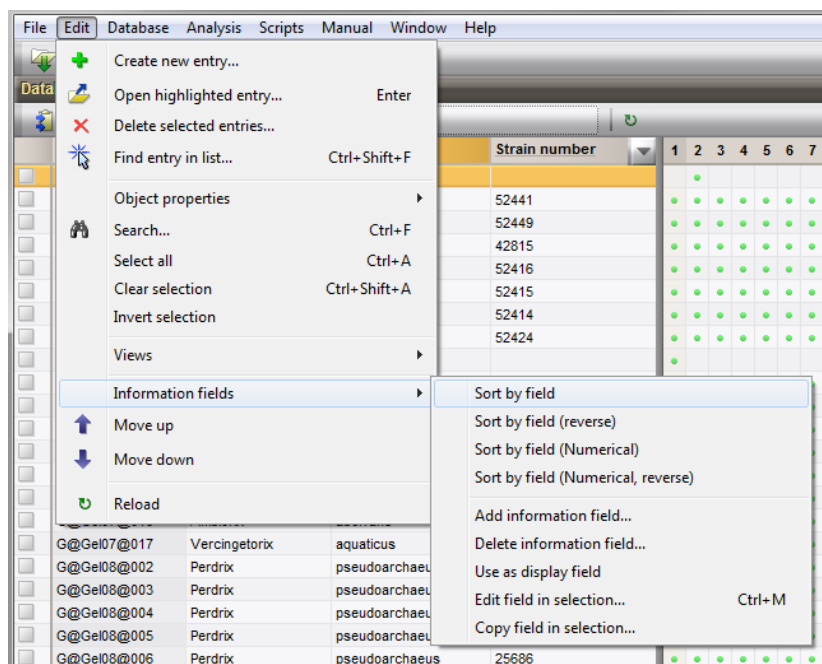


Figure 2.2.4: Illustration of the menu command *Edit* > *Information fields* > *Sort by field*.

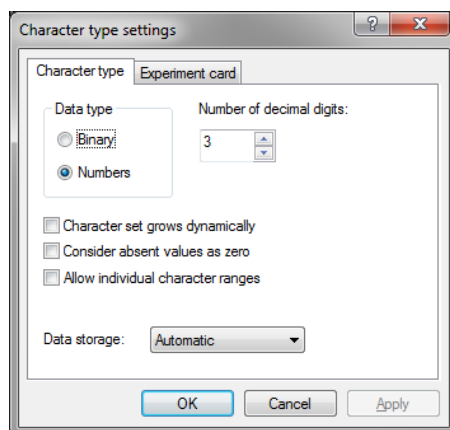



Figure 2.2.5: Illustration of buttons and check boxes.

example:

- 2.1 In the *BioNumerics Startup* window, double-click on **DemoBase Connected** to open the database for analysis. Alternatively, you can select **DemoBase Connected** from the list and press the  button.

Names of databases (e.g. **DemoBase Connected**), experiment types (e.g. **RFLP1**), comparisons, etc. in BioNumerics are typed in **bold face**.

Notes, i.e. remarks outside the normal flow of the text, are indicated in this guide with a "pencil" icon in the left margin as follows:



All actions will be executed on the *active* panel, i.e. the panel that currently has the focus.

Warnings, e.g. for user actions that may have unexpected consequences, are indicated in this guide with an "exclamation" icon in the left margin as follows:



There is no undo function for this action and removed entries are irrevocably lost, together with any experiment information linked to them!

Advanced options, which most probably are not relevant for beginners but might be useful for experienced users, are indicated in this guide with a "wrench" icon in the left margin as follows:



The *User management tools plugin* allows you to copy users, user groups and privileges from one database to another.

2.2.3 Toolbars

In almost every window in BioNumerics, there is a toolbar containing buttons for the most common functions available in the window. Placing the mouse pointer on a button for one second invokes a tool tip to appear (Figure 2.2.6), explaining the meaning of the button.

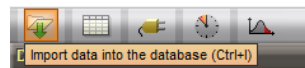


Figure 2.2.6: Tool tip that appears when hovering over a button in a toolbar.

Toolbars can be either displayed or hidden and can be customized to a high degree by the user (see 2.3.5).

2.2.4 Floating menus

In almost every window in BioNumerics, the use of place-specific floating menus is supported. For example, if you *right-click* (clicking the right mouse button) on a database entry, a floating menu is popped up, showing you all the possible menu commands that apply to the selected entry (see Figure 2.2.7).

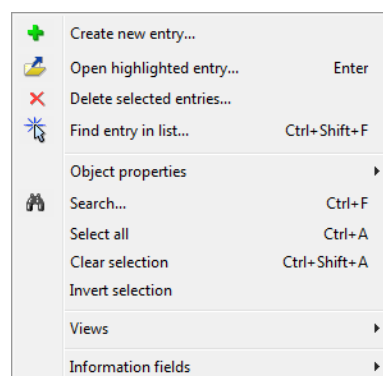


Figure 2.2.7: Floating menu that appears after clicking the right mouse button on a database entry.

The floating menus make the use of BioNumerics easier and more intuitive for beginners, and much faster for experienced users. In describing menu commands in this guide, we generally will not mention the corresponding floating menu command. It is up to the user to try right-clicking in all window panels in order to find out which is more convenient in every specific case: calling the command from the window's menu or toolbar button or from the place-specific floating menu.

2.2.5 Shortcut keys

For a number of often-used functions, a shortcut key combination on the keyboard is available. The shortcut key combinations are displayed in the menus next to the menu commands (see Figure [2.2.4](#) and Figure [2.2.7](#)). Shortcut key combinations (e.g. **F3**, **Ctrl+C**) are typed in **bold** face in the manual.

Chapter 2.3

The BioNumerics user interface

2.3.1 Introduction to the BioNumerics user interface

BioNumerics is a very comprehensive software package. For every specific task in BioNumerics, a *window* is available that groups the relevant functionality for that specific task. From an active window, dialog boxes and other windows can be launched. Each window at its turn consists of several *panels* containing specific information. The BioNumerics user interface is very flexible and can be customized by the user at three different levels:

1. The behavior and general appearance of windows can be set.
2. For each separate window, the toolbars and panels that are displayed can be chosen, as well as the size and the location of panels.
3. In grid panels, columns can be displayed or hidden and the relative position of rows and columns can be chosen.

2.3.2 The BioNumerics main window

The *Main* window appears when a database is opened from the *BioNumerics Startup* window (see [3.1.1](#)). Figure [2.3.1](#) shows the *Main* window for the **DemoBase Connected** demonstration database.

In case the *Plugins* dialog box appears instead of the database (Figure [2.1.3](#)), this means that you are opening this database for the first time. See [2.1.5](#) for further explanation on the installation of plugin tools.

The *Main* window in default configuration consists of a menu, a toolbar for quick access to the most important functions, a status bar, and the following panels:

- The *Database entries* panel, listing all the available entries in the database, with their information fields and their unique keys (see [3.3](#)).
- The *Database design* panel, displaying how the database is structured with levels and dependencies (see [3.3.10](#)).
- The *Experiment types* panel, listing all available experiment types (see [2.3.6](#)).
- The *Experiment presence* panel, which for each database entry shows whether an experiment is available (colored dot) or not. Clicking on a colored dot causes the *Experiment card* window for that experiment to be opened.

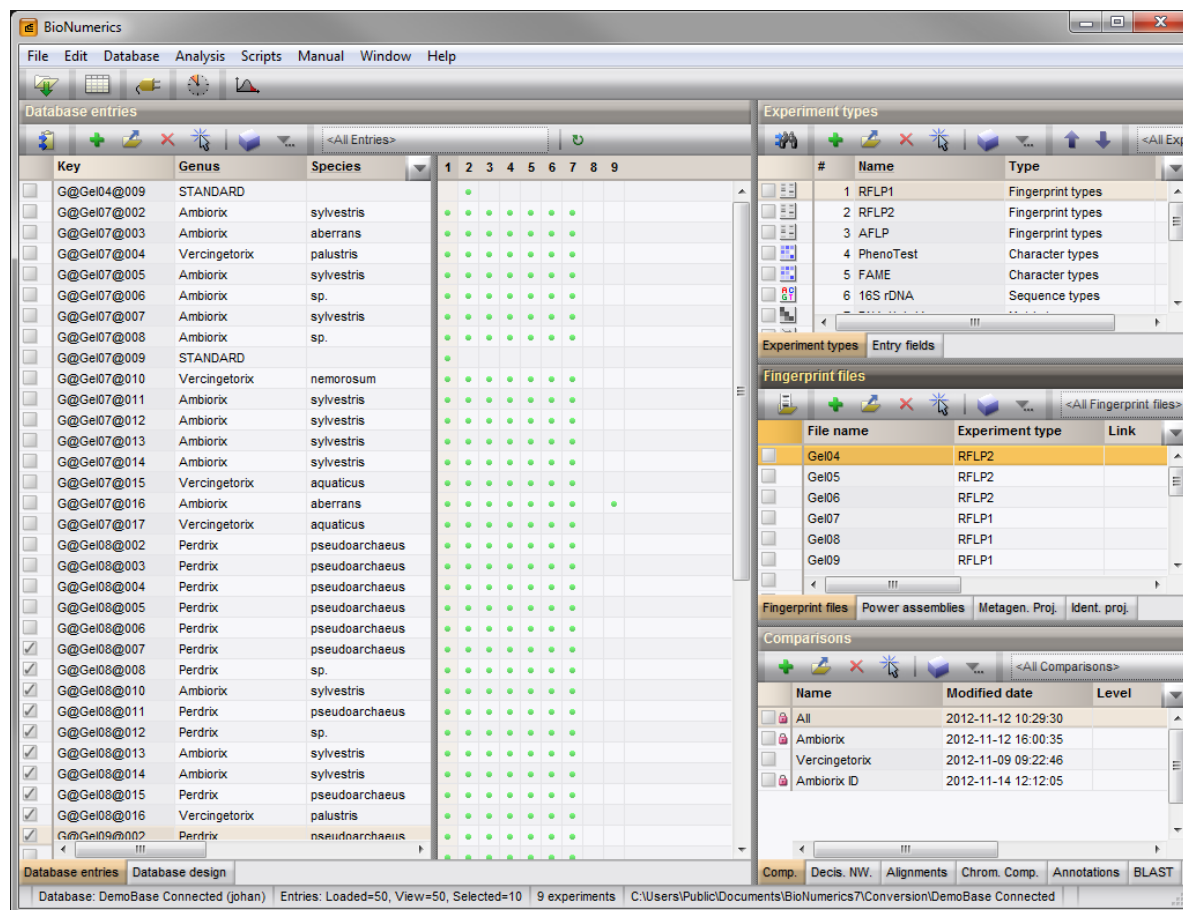


Figure 2.3.1: The *Main* window.

- The *Fingerprint files* panel, showing the available fingerprint files (see 4.1.3 and 4.1.4) for the experiment type selected in the *Experiment types* panel.
- The *Comparisons* panel, listing all comparisons that are saved (see 13.2).
- The *Identification projects* panel, which shows the available identification projects (see 15.3).
- The *Decision networks* panel, which shows the available decision networks (see 15.6).
- The *Entry fields* panel, which shows the available entry information fields in the database (see 3.3.3).
- The *Alignments* panel, listing all sequence alignment projects that are saved (see 8.4).
- The *Chromosome comparisons* panel, listing all chromosome comparisons that are saved (see 8.6 and 8.7).
- The *Annotations* panel, listing all chromosome annotations that are saved (see 8.8).
- The *Power assemblies* panel, listing all chromosome-level assembly projects that are saved (see 18).
- The *Metagenomics projects* panel, listing all metagenomics projects that are saved (see 19).
- The *BLAST projects* panel, listing all BLAST projects that are saved (see 8.9).

The status bar in the bottom of the *Main* window shows the name of the opened database, followed by the BioNumerics user currently logged in. Furthermore, the number of database entries that is loaded in

memory, the number of entries that is currently displayed, the number of selected entries, the total number of experiment types, and the Windows path to the database directory is displayed.



Unless otherwise stated, all screen shots in this manual are taken using default settings. If the *Main* window is displayed differently on your screen than in the screen shot in Figure 2.3.1, then your current settings might be different from the default settings. How to return to the default settings is described in 2.3.4.

2.3.3 General settings and preferences

2.3.3.1 Introduction

A number of general settings and preferences, which modify the appearance and behavior of the software, can be specified in the *Preferences* window (see Figure 2.3.2). The preferences are saved at a database level, allowing the user to specify different preferences for different databases. To call the *Preferences* window, select **File > Preferences...** in the *Main* window.

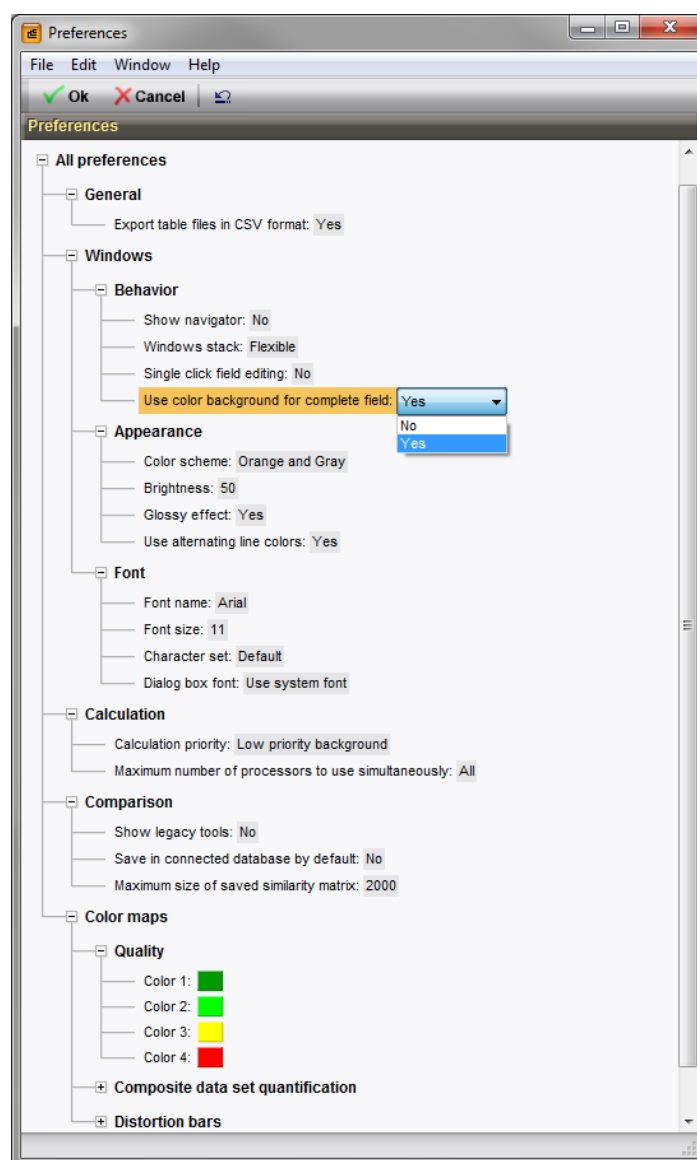




Figure 2.3.2: The *Preferences* window.

In the *Preferences* window, the preferences are hierarchically organized in groups and displayed in an editable tree. By default, all groups are expanded. A group can be collapsed by clicking the  icon left of the group name. A collapsed group can be expanded again by clicking the  icon. When an individual preference is clicked, it will be highlighted and it becomes editable: its value appears in a drop-down list or text box. To edit the value, select a different value from the drop-down list or type a value in the text box.

All preferences are organized into five main groups: General, Windows, Calculation, Comparison, and Color maps and are explained below.

2.3.3.2 General preferences

- **Export table files in CSV format:** When “Yes” is selected for this preference, any export functionality that exports information in tabular format will output Comma Separated Values (CSV) files, that will be opened by the default CSV editor on your computer (often MS Excel). The CSV file format used (i.e. comma or semicolon as separator) is depending on the regional settings of the Windows operating system. With “No” selected, tab-delimited text files will be generated and opened in the default text editor (e.g. Notepad).
- **Number of retained recent items:** The import and processing wizards display recently used items in a separate category “Recently used”. This preference determines the maximum number of recent items in this list.
- **Number of items in pick lists:** The maximum number of items that is displayed in various drop-down lists in the software. Drop-down lists will be organized in multiple columns with values above 27. In case more items are present than specified in this setting, all information will become available via the <More> item at the bottom of the list.

2.3.3.3 BioNumerics windows preferences

These general appearance and behavior preferences have an effect on *all* BioNumerics windows. They are organized in three subgroups: Behavior, Appearance and Font.

Following preferences control the behavior of BioNumerics windows:

- The *Navigator pane* (see 2.3.8) can be enabled or disabled with **Show navigator**.
- With **Windows stack**, the windows stack can be set either “Fixed” or “Flexible”. In fixed mode, the various BioNumerics windows are stacked in a fixed order. For example, a *Comparison* window always appears on top of the *Main* window and in order to view the complete *Main* window, the *Comparison* window needs to be closed or minimized. Furthermore, in the task bar of your Windows operating system, only one tab will appear for the BioNumerics software. In flexible mode, any type of BioNumerics window can be on top of another, regardless of its “rank”. For each BioNumerics window, a separate tab becomes available in the task bar of your operating system.
- **Single click field editing** can be enabled or disabled for editing fields in object grid panels respectively after a single mouse click or after clicking twice (see 3.2.6). Please note that, when **Single click field editing** is enabled, the *Entry* window is opened by double-clicking in the margin or on the ‘Key’ field of the database entry.
- For information fields in the *Database entries* panel, properties can be set (see 3.3.6 for more information). One of these properties includes a different background color for each field state. Select “No” for **Use color background for complete field** to limit the background color to a small rectangle, preceding the actual information field content.

The appearance of BioNumerics windows is controlled by following preferences:

- From the pull-down list next to **Color scheme**, a selection can be made from eight preset color schemes.
- The overall **Brightness** of the windows can be adjusted by entering a percentage.
- The **Glossy effects** of toolbars, panel headers, zoom sliders, etc. can be enabled or disabled.
- In grid panels, even and odd rows can be shaded differently for an improved display. This effect is enabled or disabled with **Use alternating line colors**.

Following preferences are related to the font used in windows and dialog boxes:

- The **Font name** that will be used in all windows, can be specified in the corresponding text box. If the entered name is not recognized, the default font (Arial) will be used.
- In the **Font size** text box, a size can be entered (default size is 11).
- As **Character set**, either “ANSI” (or Windows-1252), “OEM”, “Shift JIS” (a character encoding for the Japanese language), “GB2312” (simplified Chinese), “Big-5” (traditional Chinese) or the “Default” character set that is specified by your Windows operating system can be selected from the drop-down list.
- For the **Dialog box font** used, the options are to use the operating system’s font or the software font. The latter option will use the same font for (most) dialog boxes as set for the windows in BioNumerics (preference **Font name**).

2.3.3.4 Calculation preferences

Following preferences control the calculations in BioNumerics:

- BioNumerics performs almost all its calculations in multi-threaded mode. This means that you can further use BioNumerics or any other program while time-consuming calculations are going on (especially sequence alignments and phylogenetic clustering can take a long processing time). In order to speed up the calculations, or make multi tasking smoother, you may want to modify the **Calculation priority** settings. The drop-down list offers the choice between five priority levels. If “Foreground” is chosen, it will not be possible to run other applications while the calculations are going on. “Idle time background” means that the computer will only process the BioNumerics calculations while it has nothing else to do.
- With the drop-down list next to **Maximum number of processors to use simultaneously** one can specify the number of CPU cores to use simultaneously for the calculations. By default all cores are used (“All”) but this can be changed to all cores except one or two (“All but”), or to one single core (“One”).

2.3.3.5 Comparison preferences

These preferences are relevant to comparisons (see [13.2](#)):

- Enabling **Show legacy tools** will display the tools for calculating minimum spanning, maximum parsimony trees and multivariate analysis of variance (MANOVA) as they were implemented in the software prior to version 6.0.

- When **Save in connected database by default** is enabled, the default option for saving comparisons will be in the connected database. It will still be possible for the user to override this default in the *Save comparison as* dialog box.
- The **Maximum size of saved similarity matrix** can be set to obtain a compromise between comparison file size (or storage space taken in by the comparison in the connected database) and ease of use. The default size of 2,000 means that for comparisons with 2,000 or fewer entries, the similarity matrices will be stored along. For larger comparisons, similarity matrices will not be saved when the comparison is closed, so they need to be recalculated when adding or deleting entries to the comparison.

2.3.3.6 Color maps preferences

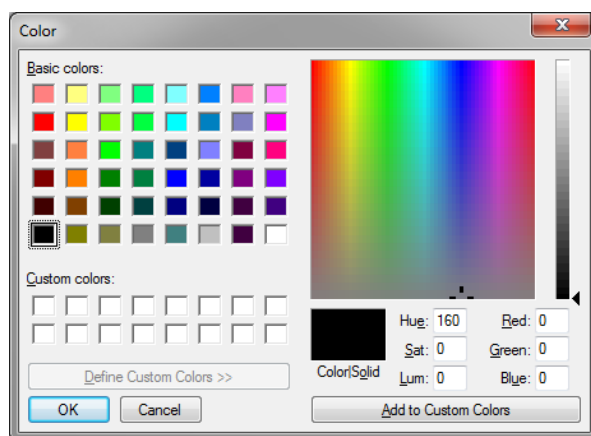


Figure 2.3.3: The *Color* dialog box.

Any desired color can be picked from this dialog using (a combination of) any of the methods below:

- By clicking any of the **Basic colors** in the upper left-hand side of the dialog box.
- By clicking any of the **Custom colors** (if defined) in the lower left-hand side of the dialog box.
- By entering **Red**, **Green** and **Blue** values in the corresponding text boxes.
- By entering **Hue**, Saturation (**Sat**) and Luminosity (**Lum**) values directly in the corresponding text boxes.
- By picking a point in the hue-saturation plot on the right-hand side of the dialog box and selecting a luminosity value using the slider on the far right.

When a color is selected, it can be added to the **Custom colors** by clicking on one of the custom color cells and pressing the **<Add to Custom Colors>** button.

Press **<OK>** to use the selected color in the *Color* dialog box. Conversely, press **<Cancel>** to keep the original color.

Using these preferences, color maps can be composed that are used throughout the software.

Colors can be picked for use in any of the following color maps:

- **Quality**: colors used to indicate the branch quality, as either cophenetic correlation or bootstrap values, on dendrogram branches (see 13.2.6).

- **Composite data set quantification**: the color scale used to display quantitative values in a composite data set (see 11.2.3).
- **Distortion bars**: color scale for the distortion bars, which are used for visual examination of the normalization process during gel and electropherogram preprocessing (see 4.1.3.5 and 4.1.4.6, respectively).
- **Similarity**: the color scale used for similarity matrices, e.g. in the *Comparison* window (see 13.3.2).

To change a color from any of the color maps, highlight the preference that you wish to modify and click on the **<Edit>** button that appears. This pops up the *Color* dialog box.

2.3.3.7 Resetting preferences and settings to factory defaults

To reset an individual preference to its default setting, highlight the preference in the tree and select **Edit > Reset to default** (🔧). A whole group of settings can be reset by highlighting the group name and selecting **Edit > Reset to default** (🔧). To reset all preferences at once, simply highlight **All preferences** and select **Edit > Reset to default** (🔧).

If preferences were modified, select **File > Save settings and close window** (✅ Ok, Ctrl+S) to commit the changes and close the *Preferences* window. It might be necessary to restart BioNumerics to enable the applied changes.

Alternatively, if you do not wish to keep the modified preferences, select **File > Exit** (❌ Cancel) button to cancel the changes.

In many BioNumerics confirmation messages the option **Don't ask again** is available (see Figure 2.3.4 for an example). Advanced users can check this option to prevent the same message from popping up again. Selecting **Edit > Reset all don't ask again** will reset such actions, i.e. all confirmations will again be displayed.

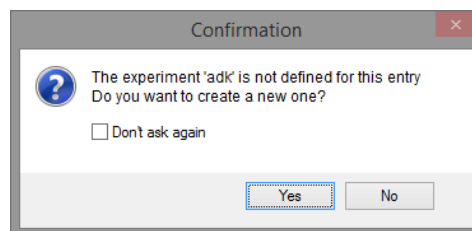


Figure 2.3.4: An example of a confirmation message with **Don't ask again** (DOSTMA) option.

For many dialog boxes in BioNumerics, the last-used settings are stored. Selecting **Edit > Reset all dialog values** will discard all such values.

Selecting **Edit > Reset all grid settings** will reset all grid panels (see 3.2) to their default settings. Grid panel settings include column widths, column ordering, and which columns are being displayed. This command has the same effect as selecting **Restore default configuration** from the menu that pops up when the column properties button (📏) of the grid is clicked, but then for all grid panels at once.

Selecting **Edit > Reset all windows settings** will reset all windows settings i.e. size of the window, panels, toolbars, etc.. This command has the same effect as **Window > Restore default configuration** (see 2.3.4), but for all BioNumerics windows at once.

2.3.4 Display of panels

BioNumerics windows can be customized up to a high degree by the user. All panels can be resized to make optimal use of the display by dragging the horizontal separators between the panels up or down or by dragging the vertical separators left or right.

Two types of panels are available in BioNumerics windows: fixed panels and dockable panels. The displayed information in **fixed** panels is indispensable in any type of analysis. Therefore, these panels are always displayed in their corresponding window. Depending on the nature of the experiment, the type of analysis performed and user preferences, **dockable** panels may not always be essential. Therefore, BioNumerics offers the possibility to either display or hide dockable panels. This feature allows the user to hide infrequently used panels that would otherwise clutter the workspace. For example, if you do not have the Classifiers and Identification module, the *Identification projects* panel in the *Main* window can be hidden for the sake of clarity. Several BioNumerics windows contain dockable panels, which all behave identically. The principles are illustrated here for the *Main* window.

4.1 In the *Main* window, click on the *Identification projects* panel to display the *Identification projects* panel.

4.2 Select **Window > Show / Hide panels** in the *Main* window. This displays a sub-menu, listing all available panels. Panel names for which a check mark is present left of the menu item are shown in the window.

4.3 Click on **Identification projects** in the sub-menu. The *Identification projects* panel is now hidden from the *Main* window.

As an alternative to the above procedure, the same option is available from a context-sensitive menu:

4.4 Right-click on the header of the *Identification projects* panel and select **Close Identification projects** from the floating menu that pops up.

Furthermore, dockable panels can be placed on the screen in one of two modes: floating or docked. **Floating** allows the window to be placed anywhere on the screen, similar to a normal window of a base size (not maximized). The **docked** mode automatically places the panel in one of five locations: top, bottom, left, right, or stacked onto another panel (tabbed view). The position of a panel is controlled with a **docking guide**.

4.5 Click in the header of the *Fingerprint files* panel and - while keeping the mouse button pressed - drag it upwards in the window. As soon as the panel leaves its original position, a docking guide appears in the center of the *Experiment types* panel. Release the mouse button on any place next to the docking guide to leave the panel floating in the window.

A floating window can be repositioned to any place on your monitor.

4.6 Click in the header of the *Fingerprint files* panel and drag it towards the *Experiment types* panel again. Drop the floating panel on the top part of the docking guide that appears (see Figure 2.3.5).

This action will make the *Fingerprint files* panel appear above the *Experiment types* panel in the *Main* window.

The context-sensitive menu can be used to achieve the same effect:

4.7 Right-click the header of the *Fingerprint files* panel and select **Move up in group** to display the *Fingerprint files* panel above the *Experiment types* panel.

4.8 Click in the *Fingerprint files* panel header and drag it towards the *Experiment types* panel again. This time, drop the *Fingerprint files* panel on the center of the docking guide (see Figure 2.3.6).

As a result, the *Fingerprint files* panel is now displayed as a tabbed view with the *Experiment types* panel and *Entry fields* panel (see Figure 2.3.7).



To re-locate a panel that is presently displayed as a tabbed view with other panels, click on the panel tab instead of the panel header to drag the panel to its new position.

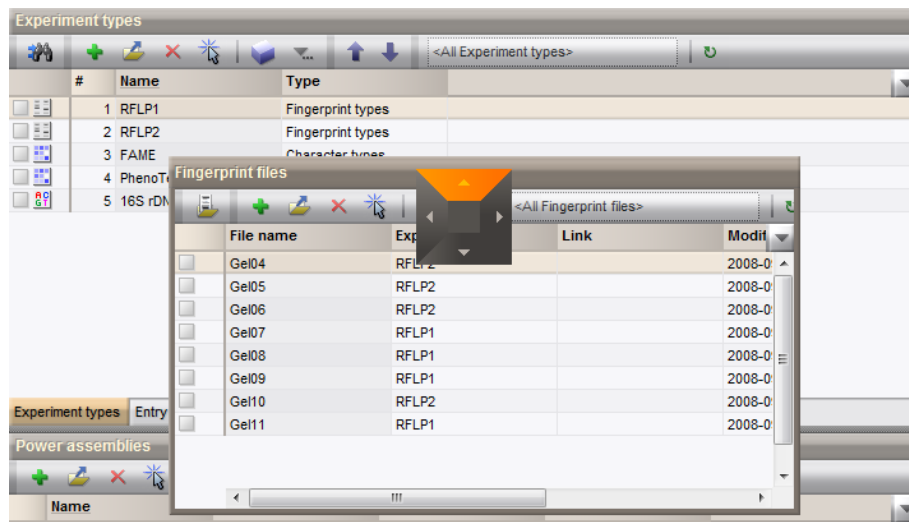


Figure 2.3.5: Docking the *Fingerprint files* panel above the *Experiment types* panel using the docking guide.

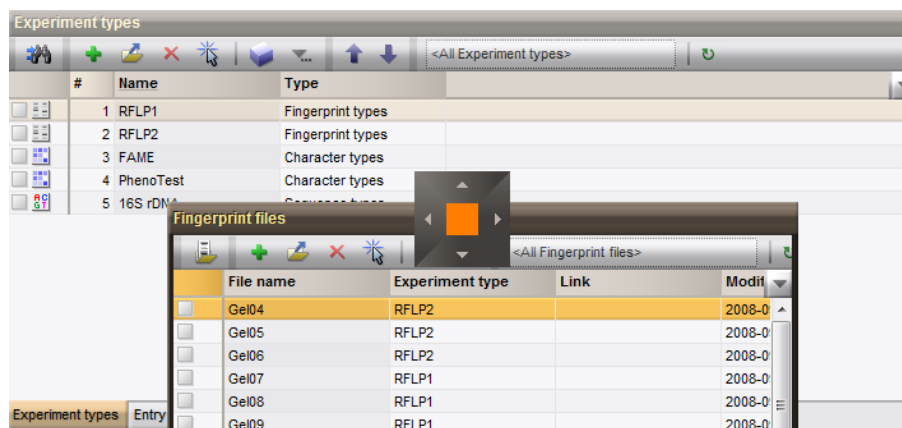


Figure 2.3.6: Docking the *Fingerprint files* panel as a tabbed view with the *Experiment types* panel and *Entry fields* panel.

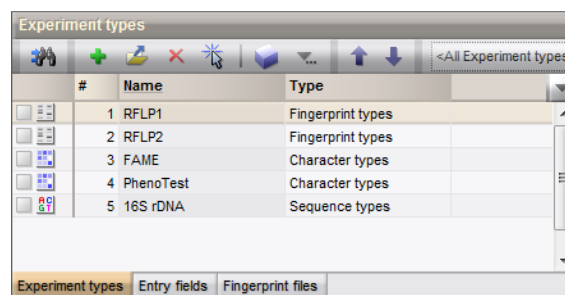


Figure 2.3.7: Result of the action depicted in Figure 2.3.6: tabbed view of the *Experiment types* panel, *Entry fields* panel and *Fingerprint files* panel.

After making some changes to the window configuration, it is always possible to return to the default configuration for the active window. This might be useful e.g. to make comparison with screen shots shown in this manual easier. If you intend to revert to the user-defined configuration afterwards, then you can save the

user-defined configuration first and recall it afterwards.

4.9 Select **Window > Save current configuration** to store the configuration that you have just defined.

4.10 To restore the default configuration, select **Window > Restore default configuration**. The window now appears back in its original configuration.


4.11 Recall the user-defined configuration with **Window > Recall saved configuration**. Notice that the changes you made to the window configuration are introduced again.

In case you do not wish to save the introduced configuration changes, Instruction 4.9 to Instruction 4.11 can be skipped:

4.12 Select **Window > Restore default configuration** to restore the default configuration of the *Main* window again.

Any window configuration can be protected from accidental changes via **Window > Lock configuration**. A check mark is present in the menu left of **Lock configuration** if the configuration of the active window is locked. Configuration changes will be enabled if **Window > Lock configuration** is selected again.

2.3.5 Configuring toolbars

In addition to the pull-down menu's that are available for executing commands, BioNumerics also displays toolbars for frequently used commands. Toolbars consist of buttons that are arranged in groups, according to their function. An example is the *Entries edit toolbar*,  which is located in the header of the *Database entries* panel. Many panels have their own toolbar, grouping panel-specific commands. Toolbars can either be displayed or hidden, a feature which allows the user to hide infrequently used toolbars.



When using Microsoft Windows 7 or 8 as operating system, the corresponding toolbar icons appear left of the menu items as a visual aid. Since earlier operating systems do not support this feature, toolbar icons in the pull-down menus may not be displayed on your computer screen.

5.1 In the *Main* window, select **Window > Show / Hide toolbar** and, for example, click on **Entries edit tools** to hide the *Entries edit toolbar*.

When the toolbar is displayed, a check mark is present next to the corresponding menu item. Toolbars specific to certain panels are listed under the corresponding sub-menus.

5.2 Select **Window > Show / Hide toolbar > Files panel** and click on **File tools** to hide the toolbar specific for the *Fingerprint files* panel.

The position of a toolbar within a window or panel can also be altered.

5.3 In the header of the *Main* window, click on the dark gray area in a toolbar, left of a set of buttons. The mouse pointer will take the shape of a hand on top of two arrows. Drag the toolbar left or right to change the order in which the toolbars appear.

5.4 In the header of the *Main* window, drag another toolbar slightly downwards to make the toolbars appear in two rows.

5.5 Click on a toolbar again and drag it to the left (or right or bottom) part of the window to dock it on the left (or right or bottom) part of the window.

The position of panel-specific toolbars can be customized much in the same way as general toolbars, with the restriction that they cannot be positioned outside their corresponding panel.

Individual buttons can be hidden from their toolbars.

5.6 Right-click on any toolbar. A floating menu appears, listing all buttons of the toolbar (see Figure 2.3.8). By default, all button names are checked in the menu and the corresponding buttons will appear in the toolbar.

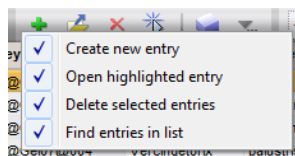


Figure 2.3.8: Floating menu for the *Entries edit toolbar*, listing all available buttons.

5.7 Select the button that you want to hide from the floating menu.

The toolbar button can be displayed again by repeating the above actions.

As for the display of panels, the configuration of toolbars can be restored to default using **Windows > Restore default configuration**. In case you want to save the current configuration, follow Instruction 4.9 to Instruction 4.11.

2.3.6 The Experiments panel

All experiments that are present in a database are listed in the *Experiment types* panel from the *Main* window, together with their 'Name', 'Type' and additional default or user-defined information fields. The *Experiment types* panel is an *object grid panel*, which means that all functionality as described under 3.2 is available for this panel. Each experiment is preceded by an icon, depicting the type it belongs to: for fingerprint types, for spectrum types, for character types, for sequence types, for sequence read sets, for whole genome maps, for trend data types, for matrix types, and for composite data sets. When an experiment is highlighted in the *Experiment types* panel, the corresponding column in the *Experiment presence* panel is highlighted as well.

Selected experiment types can be removed with **Edit > Delete selected objects...** (). The preceding icon of the experiment type(s) will show a red cross for a deleted sequence type) and will only disappear from the list after reloading the database.

2.3.7 Zoom sliders

In many BioNumerics windows, panels containing graphical information can be zoomed in or out to make optimal use of the display. Zooming in or out can be done via **Layout > Zoom in** (, **Ctrl+Page Up**) and **Layout > Zoom out** (, **Ctrl+Page Down**), respectively.

In addition, graphical panels or windows in BioNumerics are equipped with *zoom sliders* in the shape of a narrow vertical or horizontal pane, featuring a colored bar (see Figure 2.3.9 for an example). Increasing the bar size, by dragging it with the mouse, zooms in on the image. Decreasing the bar size with the mouse zooms out on the image. The zoom slider can also be operated by hovering over it and using the scroll wheel of the mouse. Alternatively, press the **Ctrl** or **Shift**-key (in case more than one zoom slider is present) on the keyboard and use the scroll wheel (e.g., **Ctrl+scroll** to zoom vertically and **Shift+scroll** to zoom horizontally). Image proportions are maintained when (as in Figure 2.3.9) the is displayed in the zoom slider. When the or icons are shown in the zoom sliders, horizontal and vertical zooming is performed separately. The gray line in the zoom slider bar corresponds to the original image size (x 1.00). Similar to toolbars, the position of the zoom sliders (left, right, top or bottom) can be changed by clicking on the area above the zoom icons and dragging the zoom sliders in position with the mouse.

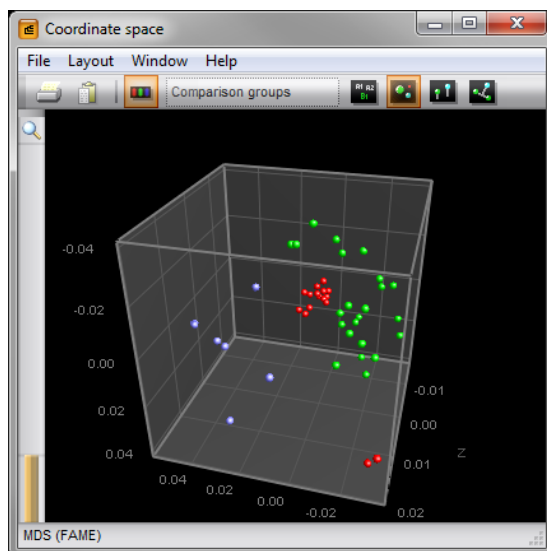


Figure 2.3.9: Zoom slider (left), illustrated for the *Coordinate space* window.

2.3.8 Navigator pane

For both fixed and flexible mode of the windows stack (see 2.3.3), a *Navigator* pane is available. The *Navigator* pane appears when moving the mouse to the far right side on the screen and displays all open BioNumerics windows in a tree-like hierarchical structure to facilitate navigation between windows (Figure 2.3.10). The active window is shown in orange type, inactive windows are shown in white type.

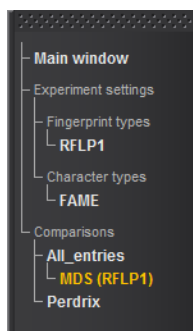


Figure 2.3.10: Navigator pane of a BioNumerics session in which two *Fingerprint type* window and two *Comparison* window are open. For one comparison, an MDS is also available. The *Coordinate space* window is currently active.

The *Navigator* pane can be enabled from the *Preferences* window (see 2.3.3).

The position of the *Navigator* pane on the computer display (top, bottom, left or right) can be modified by clicking on the structured part in the *Navigator* pane and dragging it to the desired position with the mouse.

Other display properties of the *Navigator* pane can be set by right-clicking in the structured part and selecting them from the drop-down menu:

- Uncheck **Always on top** if you want the *Navigator* pane to appear *stacked* with other open windows instead of *on top* of all open windows.
- Uncheck **Auto hide** if you want the *Navigator* pane to be permanently displayed.

- To disable the *Navigator* pane, select ***Disable*** from the floating menu and press <***OK***> in the confirmation dialog box that appears.

Part 3

The BioNumerics database

Chapter 3.1

Administering databases

3.1.1 The BioNumerics Startup program

In order to facilitate the use of BioNumerics in different research projects, it is possible to set up multiple *databases*. These databases can be managed by the **Startup program**.

When BioNumerics is launched from the Windows start panel or when the BioNumerics shortcut () on your computer's desktop is double-clicked, the **Startup program** is run. This program shows the *BioNumerics Startup* window (see Figure 3.1.1).

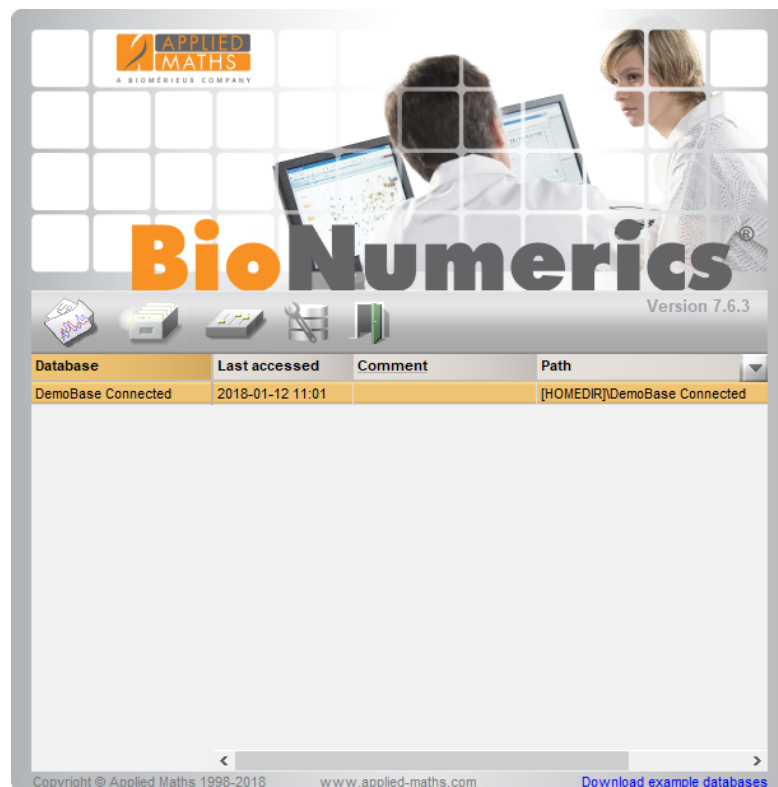



Figure 3.1.1: The *BioNumerics Startup* window.

It allows you to open a database and run the BioNumerics main application with  or by simply double-clicking on a database name in the list. Note that it is also possible to navigate to the list of databases using the **Arrow Up** and **Arrow Down** keys and to open the highlighted database by pressing the **Enter** key on

the keyboard.


A new BioNumerics database can be created with the  button. See 3.1.3 for more instructions on how to create a new database.

Various settings can be customized after pressing the  button. From the menu that pops up, following commands are available:

- **Change home directory...**: Sets the location of the home directory, where BioNumerics will look for its databases (see 3.1.2).
- **License settings...**: Modifies the BioNumerics license settings.
- **NetKey+ configuration...**: Configures the NetKey+ software (see 1.4) in case of a network license.
- **Check serial number...**: Will display the protection key's serial number on the screen, which is useful for support purposes.




In order to modify the license settings or to make changes to the NetKey+ connection settings, it might be necessary to run the Startup program as administrator.

Database management actions can be undertaken on the currently highlighted database in the list after pressing the  button. Following actions are available:

- **Delete database**: Will remove the database (see 3.1.4).
- **Backup database**: Will take a backup of the database (see 3.1.5).
- **Restore database**: Restores a database from a backup (see also 3.1.5).




Use the  button when you are finished running the BioNumerics applications.

The database name, the last accessed date, an optional comment and the path where the database is located are shown in a grid panel. Hence, the *BioNumerics Startup* window has all functionality from a grid panel, which is discussed in 3.2.7.

3.1.2 The BioNumerics home directory

A BioNumerics database can be located on the local computer or anywhere on the network, as long as BioNumerics has sufficient privileges to write to the database and its associated folders. BioNumerics recognizes and inventories the available databases by looking in the *home directory*. This is a folder that can be specified by the user and which contains a *database descriptor file* for each database. The database descriptor files have the extension ".dbs" and basically contain a tag [DIR] under which the full database path or network location is written. The different steps in opening a database are schematically represented in Figure 3.1.2.

Note that each Windows user may specify a different home directory, which can even be on a different computer in the network. The BioNumerics home directory is saved with the system registry of the user.

The current home directory can be changed from the *BioNumerics Startup* window by pressing the settings button () and selecting **Change home directory...**. This pops up the *Home directory* dialog box (Figure 3.1.3).

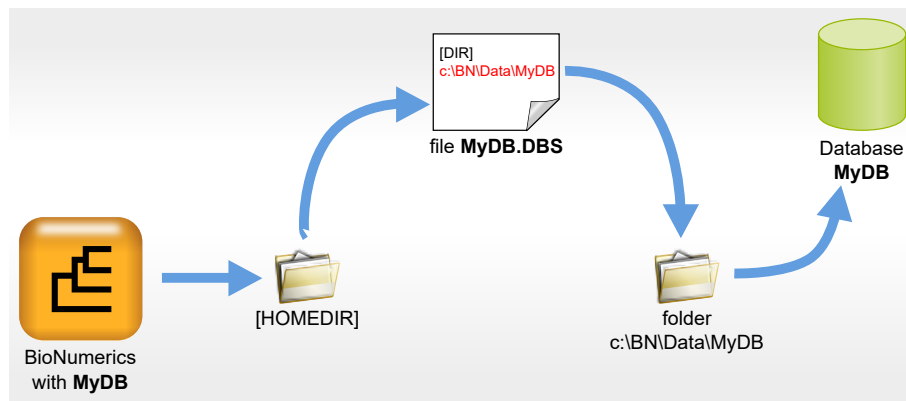


Figure 3.1.2: Steps in opening a database "MyDB": (1) BioNumerics looks in the home directory for file MyDB.dbs. (2) File MyDB.dbs is opened to obtain the database path. (3) The database is opened in the database path found.

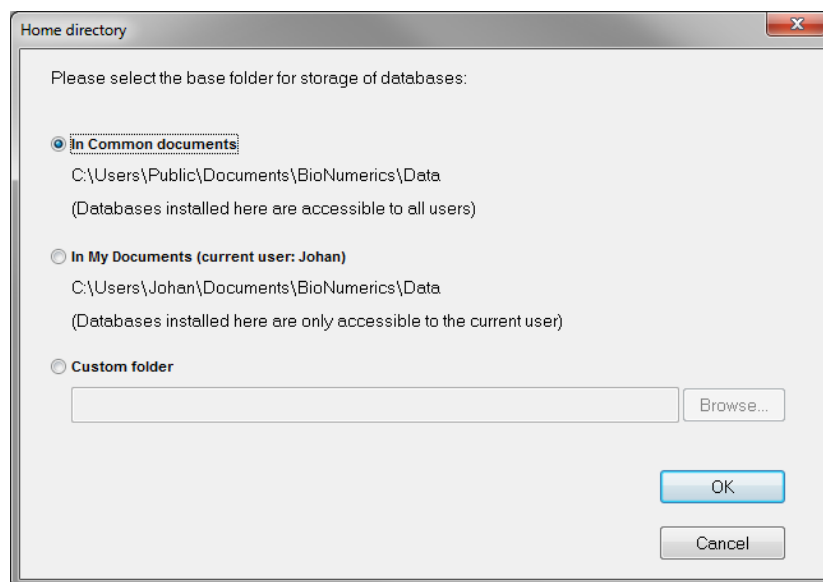


Figure 3.1.3: The *Home directory* dialog box.

The program offers two default options for the location of the home directory: ***In Common Documents*** and ***In My Documents***. The option ***In Common Documents*** makes the BioNumerics databases available for all Windows users on that computer. The option ***In My Documents*** makes the databases only accessible to the user currently logged on. The third option, ***Custom folder***, allows the user to specify any desired directory. You can even specify a directory on a network drive, on condition that this drive is permanently available with write-access.

Press **<OK>** to start using the newly selected home directory. After confirmation, the program updates the list of available databases in the new directory.

By default, BioNumerics will install any new database under the home directory (see 3.1.3).



The actual data are stored in a relational database and/or source files directory (see 3.7), while the database directory merely contains some settings files. Depending on the database settings used, the relational database and source files directory might be located **outside** of the BioNumerics home directory. When creating database backups (see 3.1.5), always verify that all data (database directory, source files directory and relational database) are included in the backup schema!

The *Database descriptor file* has the name of the database with the extension ".dbs". The *Database*.dbs

file is a simple text file which can be edited in Notepad or any other text editor.

The line after [BACKCOL] contains the RGB values for the window background color, and the line after [SAVELOGFILE] indicates whether log files are saved or not (see 3.7.2).

For databases created in a BioNumerics version prior to version 5.0, the line after the tag [DIR] always indicates the full path where the database is located (see Figure 3.1.4 a). In databases created using default settings in version 5.0 or higher, this absolute path is replaced by a [HOMEDIR] tag (see Figure 3.1.4 b). This tag points to the home directory as defined in the *BioNumerics Startup* window. Because the database paths are stored relatively with respect to the home directory, databases can easily be copied to other locations or computers. After copying the database(s) (and their .dbs files) to another location, you only need to change the home directory (see above).

[DIR]	[DIR]	
C:\Users\Public\Documents\BioNumerics\Data\DemoBase	[HOMEDIR]\DemoBase	
[BACKCOL]	[BACKCOL]	
195 200 200	195 200 200	
[SAVELOGFILE]	[SAVELOGFILE]	
0	0	(a) 0 (b)

Figure 3.1.4: Database descriptor file, with the full path of the database directory indicated (a) or with a path indication relative to the home directory (b).



If you are working in version 5.0 or higher with databases created in a prior version, it may be useful to replace the paths in the .dbs files by a [HOMEDIR] tag. A script is available to store the paths relatively with respect to the home directory. Contact Applied Maths to obtain this script.



In case a database has been physically removed (or moved) from a computer, the *Database*.dbs file may still be present in the home directory, which causes the *BioNumerics Startup* window to list the database. When attempts are made to open or edit such a removed database, BioNumerics will produce an error. The only remedy is to delete the *Database*.dbs file.

3.1.3 Creating a new database

A new BioNumerics database is created from the *BioNumerics Startup* window (see 3.1.1) by pressing the



button. This will start the *New database* wizard (see Figure 3.1.5).

A **Database name** should be entered in the corresponding text box.

Press <Next> to go to the second step of the *New database* wizard (see Figure 3.1.6).

Two options exist:

- **Create new:** Select this option to automatically create a new relational database. Note that BioNumerics can only create new databases in SQLite, the BioNumerics MS SQL Server Express instance (if installed earlier), or MS Access.
- **Use existing:** The main usage of this option is to set up databases that are shared between different users and/or when you have set up your own database management software (DBMS) such as MS SQL Server, Oracle or MySQL for data storage. Prerequisite is that the relational database itself already exists. It can contain the necessary BioNumerics database tables, but this is not required since the tables will be created automatically if not present.

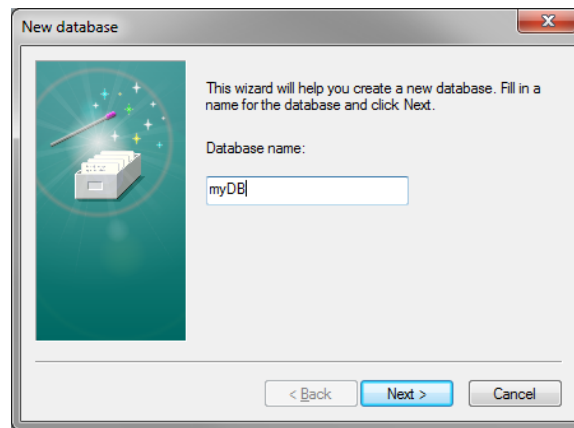


Figure 3.1.5: The first step of the *New database* wizard.

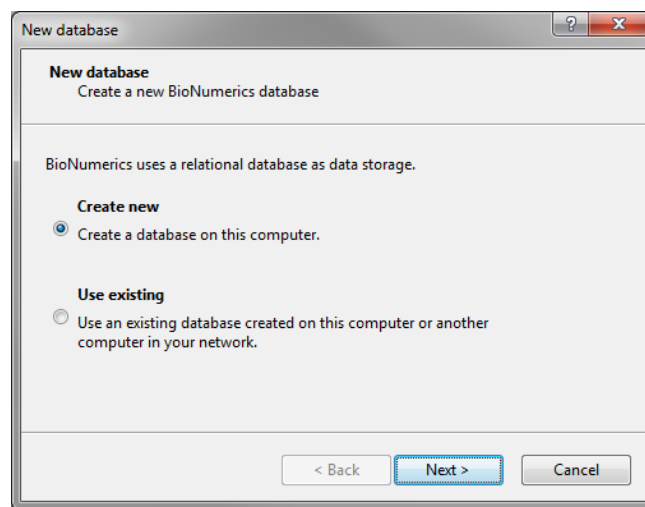


Figure 3.1.6: The *New database* wizard page.

With the option *Create new* selected, pressing *<Next>* will open the *Database engine* wizard page (see Figure 3.1.7).

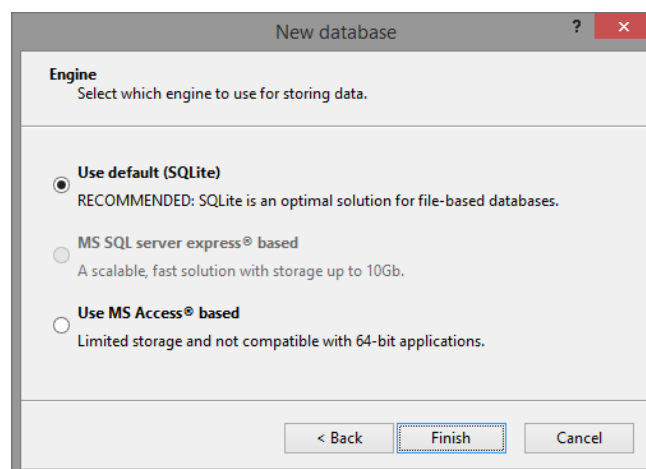


Figure 3.1.7: The *Database engine* wizard page, prompting about the database engine to use.

Following options are available in this dialog box:

- **Use default SQLite:** The default SQLite option is a file-based database system with a theoretical storage limit of 32 TB.
- **MS SQL Server Express[®] based:** Uses the MS SQL Server Express BioNumerics instance, which has a storage limit of 10 GB. This option is only available if the MS SQL Server Express database engine was installed previously with an earlier (7.0 - 7.5) BioNumerics version.
- **Use MS Access[®] based:** Uses the Microsoft Jet Engine (shipped with all Windows operating systems), which has a storage limit of 2 GB. This option is not compatible with 64-bit versions of BioNumerics (see 1.1).

When the **<Finish>** button is pressed, BioNumerics will automatically create the relational database and necessary tables therein.

When the creation of the data model has finished, the *Plugins* dialog box will open, as discussed in 2.1.5. Press **<Proceed>** to start working with the newly created database without installing any plugins.

When the option **Use existing** was selected in the *New database* wizard page (see Figure 3.1.6), the *Locate database* wizard page will appear (see Figure 3.1.8).

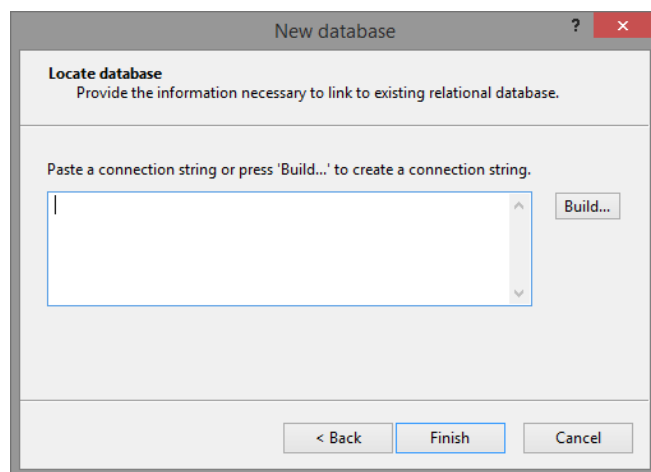


Figure 3.1.8: The *Locate database* wizard page.

To connect BioNumerics to the existing relational database, advanced users can paste a *connection string* that was created earlier. Alternatively, a new connection string can be built by pressing the **<Build>** button. This action will open the *Connect to existing database* dialog box (see Figure 3.1.9).



Advanced users can edit the connection string via the text box, e.g. for adding so-called pragmas when using SQLite.

In the *Connect to existing database* dialog box, the **Type** of database needs to be selected first. Four options are available:

- **My databases:** Check this option if the database is located on the BioNumerics SQL Server Express instance on your computer. The database can be selected from the drop-down list that appears.
- **SQLite:** Use this to connect to an existing SQLite database. Via the **<Browse>** button, you can browse for the SQLite data file (with file extension `.sqlite`) on your computer or on a network drive.

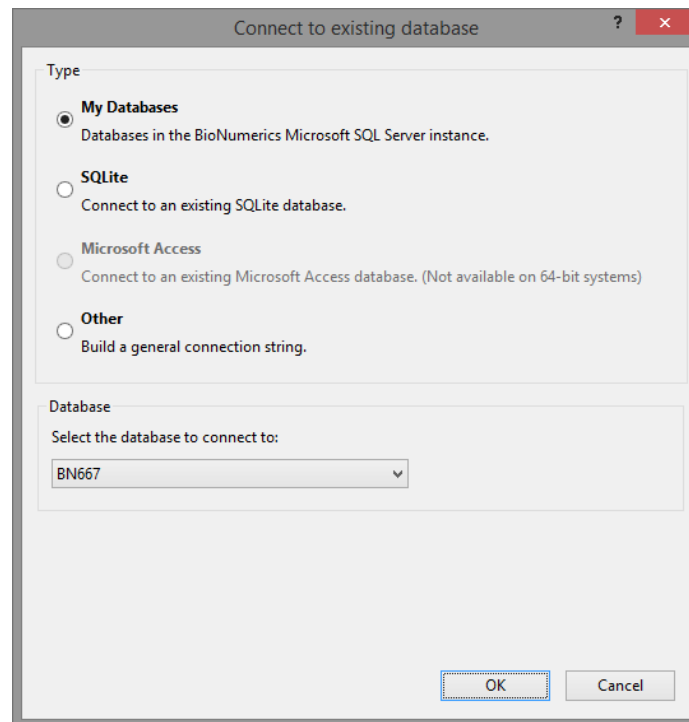


Figure 3.1.9: The *Connect to existing database* dialog box.

- **Microsoft Access:** Check this option to connect to an existing MS Access database. Via the **<Browse>** button, you can browse for the MS Access data file (with file extension `.mdb` or `.accdb`) on your computer or on a network drive.
- **Other:** This option should be used for any other ODBC-compatible relational database. Pressing the **<Build>** button allows you to create an ODBC connection via the general functionality provided by your Windows operating system. As the dialog box and options available may differ depending on the Windows version installed, we refer to the Windows documentation for instructions on how to select or create a DSN file (ODBC Data Source) that specifies the ODBC driver and to set up a connection to the database. In addition, the Structured Query language (SQL) dialect needs to be selected from the **SQL dialect** drop-down list. Currently, four DBMS types – and hence four different SQL dialects – are supported: **SQL Server[®]**, **Access[®]**, **Oracle[®]** and **MySQL[®]**.




File-based databases such as MS Access or SQLite are **not** recommended in a simultaneous multi-user setup. For this purpose, a DBMS such as MS SQL Server, MySQL or Oracle is much better suited.

Once the database connection is properly configured, you can press **<OK>** to return to the *Locate database* wizard page.

3.1.4 Removing databases

A BioNumerics database is removed as follows:

In the *BioNumerics Startup* window, select the database from the list, press the  button and select **Delete database**. The program will ask for confirmation before deleting the database.

If you answer **<Yes>** to the question "Do you want to remove the corresponding directories and database?", all data and settings of the database, stored in files or in the relational database, will be deleted.



Deleting a database – more specifically removing the corresponding directories and relational database – cannot be undone and all data will be irrevocably lost!

3.1.5 Backing up and restoring databases

3.1.5.1 Backups to protect against data loss

In many cases, BioNumerics will be used to construct large databases of information that has been collected over a long time span. Obviously, the user should pay attention to protect such databases from accidental data losses, e.g. due to hard disk crashes, power interruptions, etc.. and take backups on regular intervals.

In case the database was automatically created by BioNumerics (see 3.1.3), an easy procedure to create a backup is available in the *BioNumerics Startup* window (see 3.1.5.2). This procedure cannot be followed when BioNumerics was connected to an existing relational database. As this is typically the case for large network databases that are accessible by multiple users, such backups are generally part of a centralized backup routine. 3.1.5.3 describes where all data can be found in this case.

3.1.5.2 Databases created by BioNumerics


To create a backup from an automatically created BioNumerics database (the option **Create new** in the *New database* wizard page was checked when creating a database, see 3.1.3), select the database in the list in

the *BioNumerics Startup* window and press the  button. From the menu that appears, select **Backup database...** The software will ask for a **Destination folder**. The default destination folder is the home directory. The backup procedure will start when <OK> is pressed.



Depending on the amount of data stored in the database, the backup procedure might take a few seconds up to several minutes.

The backup will be stored as a single binary file named after the original database with today's date appended and with the extension .bnbk.

To restore a database from a backup, select the database in the list in the *BioNumerics Startup* window and press the  button. From the menu that appears, select **Restore database...** to call the *Restore database dialog box* (see Figure 3.1.10).

Press <Browse> to browse for a previously created .BNBK backup file.

Two options exist for the **Destination** where the backup will be restored:

- **Overwrite:** Overwrites the selected database in the *BioNumerics Startup* window with the data from the backup file.
- **Create a copy:** Restores the database under a different name, which should be provided in the *New database name* text box.

As soon as the database is restored, you will be prompted to open it.

3.1.5.3 Databases not created by BioNumerics

When BioNumerics was linked to an existing relational database (the option **Use existing** in the *New database* wizard page was checked when creating a database, see 3.1.3), the **Backup database...** command in the *BioNumerics Startup* window will not work and the BioNumerics database should be backed up manually or with third-party backup tools.

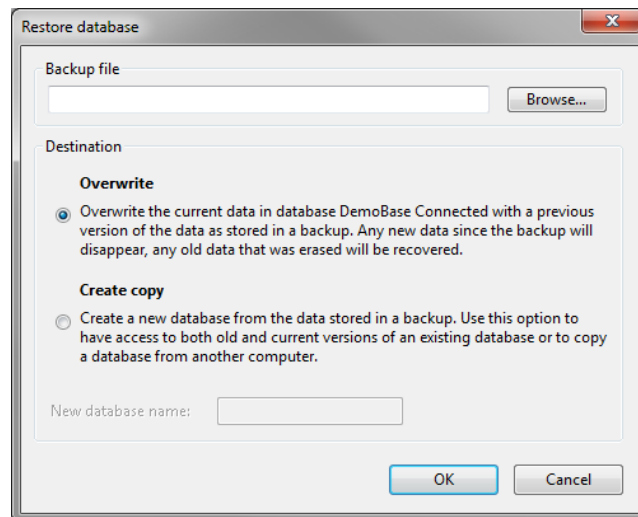


Figure 3.1.10: The *Restore database* dialog box.

The bulk of the actual data in a BioNumerics database will be stored in the relational database (see 3.7). For performance reasons, next-generation sequence data are stored as files in the source files directory (see 3.7.2). In addition, imported files such as gel TIFF images or sequencer trace files can optionally also be stored in the source files directory. Window and viewing settings are stored as files in the database directory.

The location of the relational database and associated source files can be found from the *Database settings* dialog box (see 3.7.2). The ODBC connection string in the *Connection tab* contains the database name and location. The location of the source files is shown in the *Files tab*.

To ensure completeness, the database directory, the relational database and the source files directory should be backed up.

Professional DBMS such as SQL Server, Oracle, MySQL, etc. can be configured to take automatic backups on regular time intervals. We refer to the DBMS documentation for the setup of such automatic backups.

Chapter 3.2

Database objects

3.2.1 Introduction

In a BioNumerics database, all major data classes are seen as *objects*. A database object can be seen as a piece of information in the database that is:

- Referable (both by the software and by the user)
- Interpretable by the user
- Self-contained (all information required to interpret is available)

In many cases, an object corresponds to a single record in a single table (see [21.1](#) for the database table structure). Examples of such objects are database entries, fingerprint files, comparisons, etc.. However, an object can also correspond to a set of records, even from different tables. Examples are character experiments and sequences in case the sequences were assembled from trace files.

A further differentiation can be made between *top-level* objects and *lower-level* objects. Top-level objects correspond to "real" objects in a BioNumerics database, such as entries, character experiments, sequences, etc.. Lower-level objects are subcomponents of top-level objects, such as individual character values, sequence trace files, etc..

An object can have one or more *parent objects*, i.e. objects to which the information in the object is linked to. For example, a fingerprint object has three parent objects: fingerprint files, experiment types and entries. Parent objects are mainly important in *object queries* (see [3.2.14](#)), where parent object fields can be displayed and queried in addition to the object's own fields.

The concept of objects has important consequences for users working in a certified environment, especially when compliance with the Title 21 Code of Federal Regulations (*21 CFR Part 11*), as set forth by the U.S. Food and Drug Administration (FDA), is required. In 21 CFR Part 11, which deals with replacing paper records by electronic records, an electronic record is described as "...any combination of text, graphics, data, audio, pictorial, or other information representation in digital form that is created, modified, maintained, archived, retrieved, or distributed by a computer system." The term "object" in BioNumerics therefore maps to "electronic record" in the FDA terminology. As such, objects form the atomic component for user privilege control ([3.5](#)), the audit trails and versioning tool ([3.6](#)), digital signatures ([3.6.4](#)), etc..

Lists of BioNumerics database objects are often displayed in *object grid panels*. The grid consists of information fields organized in columns and database objects of a certain type in rows. The configuration options of an object grid panel are numerous: The user can control which information fields to display and also the order in which they are displayed (see [3.2.7](#)). Via views (see [3.2.2](#)), any selection of objects can be made from all objects of a certain type that are available in the database.



All actions will be executed on the *active* panel, i.e. the panel that currently has the focus.

3.2.2 Object views

An object grid panel does not necessarily displays all available database objects of a certain type. Instead, it provides a dynamical *view* on the objects.

One can easily switch from one view to the other via the Views drop-down list (see Figure 3.2.1), displayed in the toolbar of the object grid panel.

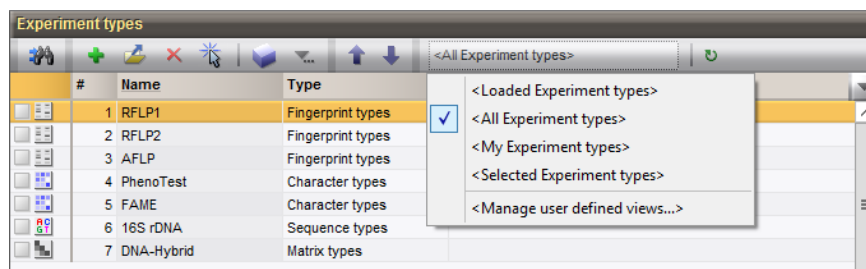


Figure 3.2.1: An example of a Views drop-down list, here from the *Experiment types* panel.

To manage the views that are displayed from this drop-down list, select "<Manage user defined views. . . >". This will display the *Manage user views* dialog box (see Figure 3.2.2).

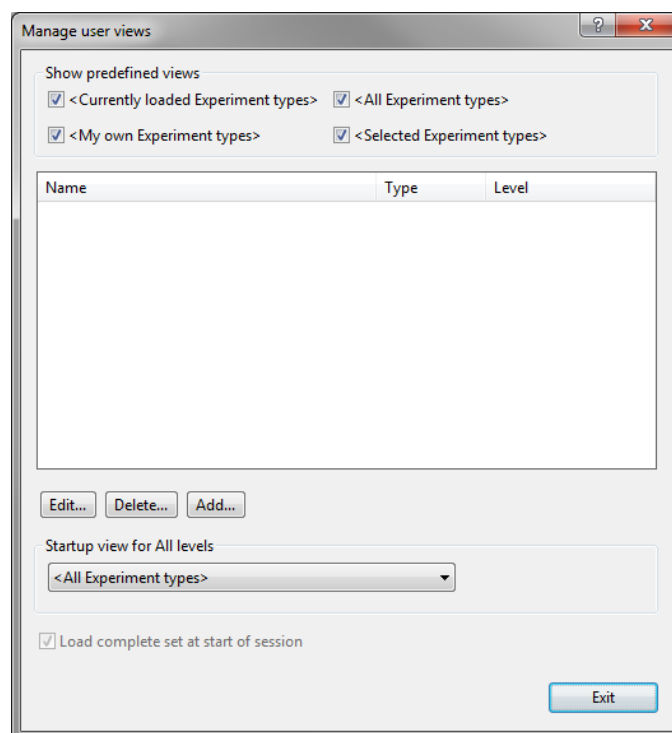


Figure 3.2.2: The *Manage user views* dialog box.

Under *Show predefined views*, a number of predefined views are listed:

- **<Loaded objects>**: All objects that are currently loaded into memory (see further).
- **<All objects>**: All objects that are available in the database

- **<My objects>**: All objects of which the currently logged-in user is the owner. For more information about ownership, see 3.2.
- **<Selected objects>**: Objects that are currently selected (see 3.2.4).

The predefined views will only be listed in the Views drop-down list (Figure 3.2.1) when the corresponding check box is checked.

A custom view can be created by pressing the **<Add>** button, which calls the *New object view* dialog box (see Figure 3.2.3).

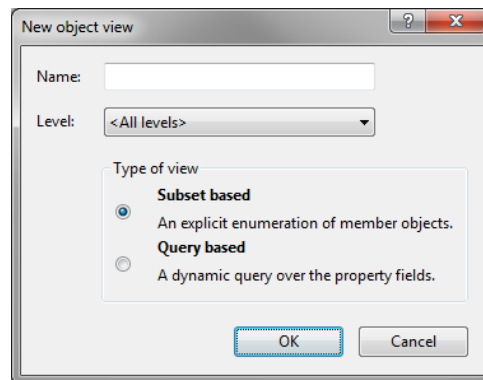


Figure 3.2.3: The *New object view* dialog box.

The dialog box prompts for the *Name* that will be used to refer to the custom view.

Level is the database level (see 3.3.10) that the view is defined for. The drop-down list allows you to select any level that is defined in the database.

Two distinct types of custom views can be created:

- A **Subset based** view is created based on a selection of objects. Although the user can add or remove objects from the selection at any time, this is essentially a *static* view, since always the currently defined selection of objects will be returned by the view. Newly created entries will never be part of a **Subset based** view, unless they are explicitly assigned to the view after creation.
- A **Query based** view is a *dynamical* view. The database query is created in the same way as object grid panels are searched (see 3.2.10). When newly created objects fulfill the query criteria of a **Query based** view, they will be returned by the view.

Pressing **<OK>** will create the view in case **Subset based** was checked or will display the *Query view editor* dialog box (see Figure 3.2.4).

This dialog box helps to build a query that will return a set of database objects (i.e. entries, experiment types, comparisons, characters, ...).

On the left hand side, all information fields from the current view (see 3.2.2) are listed in the initial view. For each of these fields, a *select statement* can be prepared. A select statement consists of following components: an information field, an operator and a comparison value. The latter can be any text, date, number or one of the tokens [CurrentDate], [CurrentYear], [CurrentMonth], [CurrentDay] or [CurrentDateTime] (see 3.7.2 for a description). As soon as a comparison value is entered, the operator appears. By default this is **Equals**, but it can be changed into any of the following: **Contains**, **Begins with**, **Ends with**, **Smaller than**, **Larger than**, **Not larger than**, **Not smaller than**, **Differs from**, and **Contains no data**.

Pressing **<Add new>** will add a new select statement to the list, which initially appears as a question mark. Clicking on the question mark will display a list from which an information field can be selected.

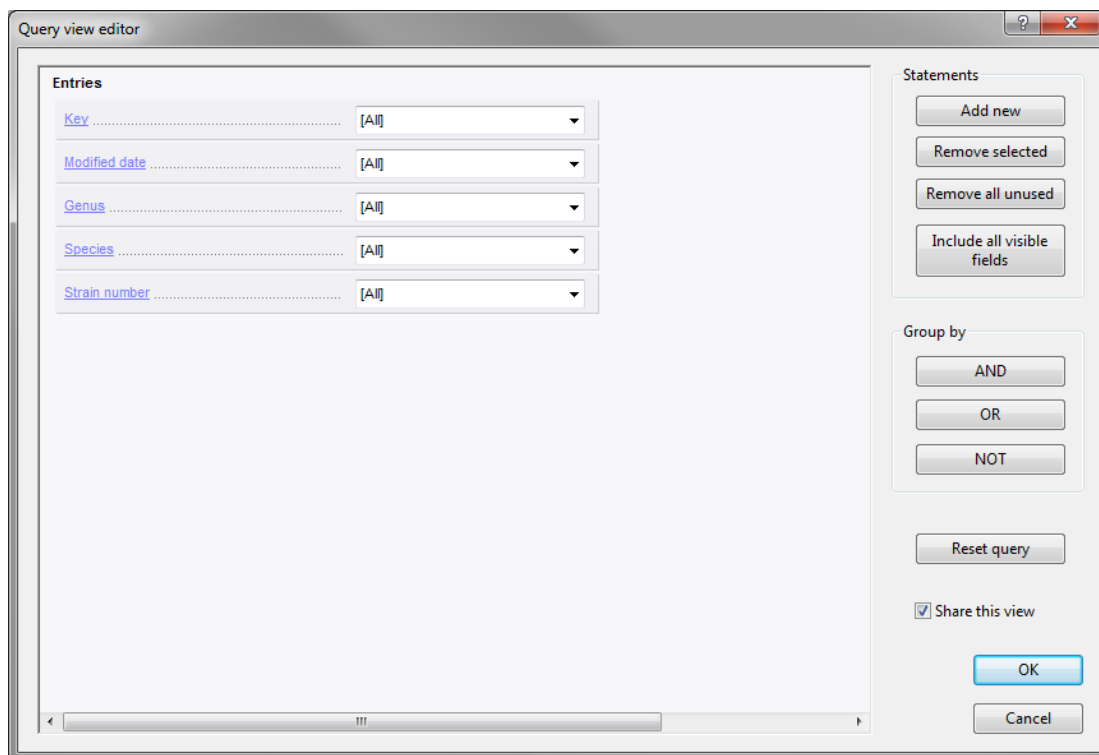


Figure 3.2.4: The *Query view editor* dialog box.

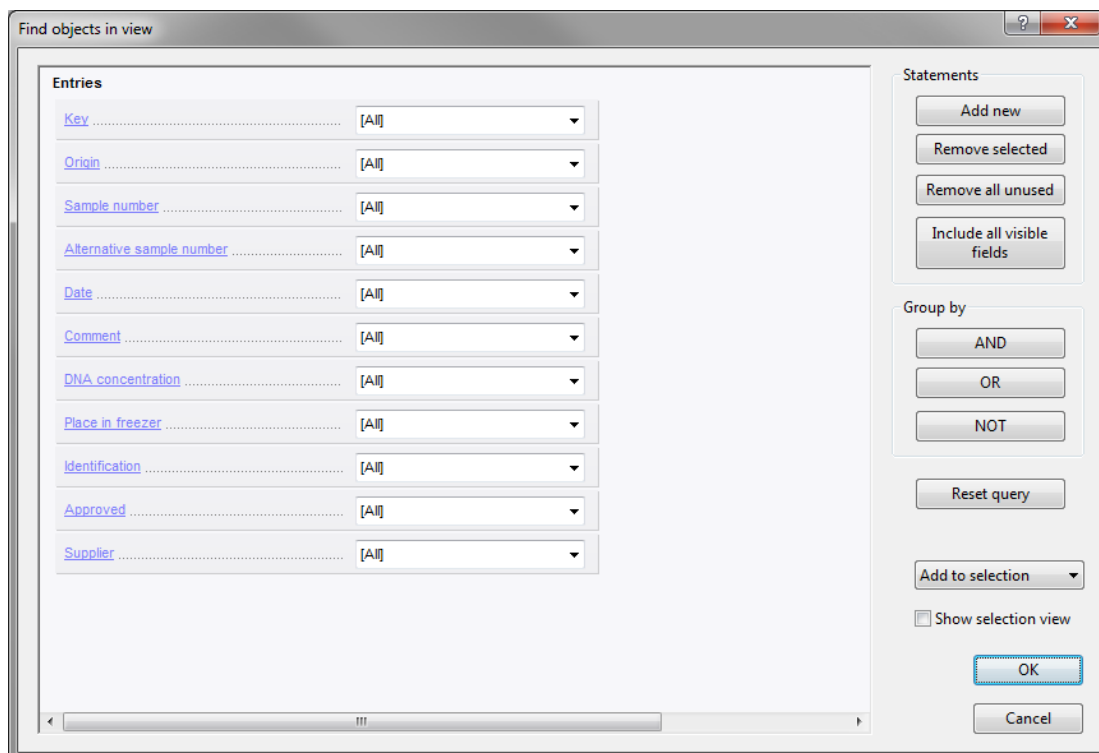


Figure 3.2.5: The *Find objects in view* dialog box, displayed for the *Database entries* panel.

Statements can be selected using **Ctrl+click**, as indicated with a colored border. Selected statements can be removed with **<Remove selected>**. Pressing **<Remove all unused>** will remove all statements for which no search value was entered.

Pressing **<Include all visible fields>** will prepare a statement for each of the information fields as displayed in the current view.

By default, all statements are combined with an AND logical operator. Selected statements can be combined with AND, OR, and NOT logical operators using the corresponding buttons. This allows for the construction of *compound statements* with the required level of sophistication.

To restore the initial view of the *Find objects in view* dialog box, press **<Reset query>**.

Checking **Share this view** will make the view available for other database users.

Any custom view will be listed, with its 'Name', 'Type' (Subset or Query based) and the 'Level' it belongs to (empty when the view was assigned to all levels).

With the **<Edit>** button, query based views can be edited via the *Find objects in view* dialog box. Subset based views cannot be edited this way. Instead, they are modified by cutting selections from or pasting selections into the view (see 3.2.4).

A highlighted view in the list can be deleted with **<Delete>**. The software will ask for confirmation before actually deleting the view from the database.

The drop-down list **Startup view for <Database Level>** displays a list of all available views (predefined and custom views), of which one can be selected as the startup view, i.e. the view that will be used when the database is opened.

The option **Load complete set at start of session** can be very useful in case of large databases: when unchecked, only those objects that are retrieved by the view are loaded in computer memory. In some circumstances, this can significantly reduce the startup time of large databases.

With **Edit > Views > Switch to Selected Objects view (Ctrl+Shift+S)**, one can easily switch from the current view to a view that only contains the currently selected objects. Using **Edit > Views > Switch to default view (Ctrl+Shift+D)**, the current view is switched to the default or startup view as defined in the *Manage user views* dialog box. The same actions can be performed by selecting **<Selected objects>** or the default view, respectively, from the Views drop-down list in the object grid panel.

3.2.3 Object access settings

Any object in the database can be locked or unlocked, and has ownership and sharing settings. These settings are referred to as *object access settings*. For objects that have their own specific window, the object access settings can be called from that window. Object access settings can be changed for individual objects or in bulk for a whole set of objects using object queries (see 3.2.14).

For example, to edit the object access settings of an entry, highlight the entry in the *Database entries* panel and select **Edit > Object properties > Object access status....** This action displays the *Object access* dialog box (see Figure 3.2.6).

Under **Access privileges**, all possible actions on an object are listed and whether or not this action is allowed for the user that is currently logged in. If an action is not allowed, the reason why is displayed. The **Access privileges** cannot be modified in the *Object access* dialog box (the list is read-only) and should be set at the level of user management (see 3.5 for more information).

The **Object access status** of an object can be either **Unlocked** or **Locked**. If an object is locked, it will not be possible to overwrite the object in the database, until the lock is reset by a user with this privilege.

Each object in the database has an *owner*, which is by default the user who created the object. The current user can be made owner of the object by pressing **<Take ownership>**, if he/she has the privilege to do so. An object can furthermore be made **Shared** or not by checking the corresponding check box. Objects that are not shared can only be edited by their respective owners; shared objects can be edited by any user who

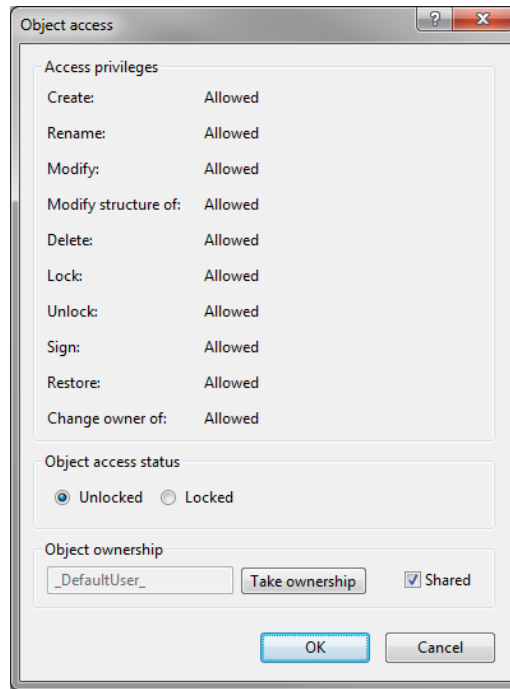


Figure 3.2.6: The *Object access* dialog box.

has the *Access privileges* to do so.

The *Object access* dialog box allows to change the *Object access status* and *Object ownership* for an individual object.

When the object is locked by checking **Locked** followed by <OK>, a "padlock" icon (🔒) will be displayed left of the object. If now the object information is modified and an attempt is made to save the changes to the database, the save action will be prevented and an "Invalid action" error message generated.

3.2.4 Object selections

Several actions on objects, such as deleting, cutting, copying and pasting, are executed on *selected* objects in the active object grid panel. The *selection state* (whether or not an object is selected) is indicated with a check box: ☒ (object selected) or ☐ (object unselected). To select or unselect an object in the list, simply click on the check box with the mouse (or **Ctrl+click** the object) to toggle the selection state. A range of objects can be selected by clicking the first one (this will become the *active* object, i.e. highlighted in the list) and – whilst holding the **Shift**-key on the keyboard – clicking on the last object to be selected.

If you wish to select objects using exclusively the keyboard, you can scroll through the object grid panel using the Up/Down arrow keys and select or unselect objects using the **space bar**.

To invert the selection, i.e. to select all objects in the list except the currently selected ones, use **Edit > Invert selection**.

With **Edit > Clear selection** (**Ctrl+Shift+A**), the selection is cleared, meaning that all objects in the list will be unselected. All objects in an object panel can be selected at once with **Edit > Select all** (**Ctrl+A**).

3.2.5 Adding and removing object information fields

All top-level objects have the fields 'Owner', 'Shared', and 'Locked', to hold the information of the object access settings (see Figure 3.2.3). Most top-level objects also have 'CreationDate' and 'ModifiedDate', corresponding to the date when the object was created and when the object was last modified, respectively. Dates are indicated following the ISO 8601 notation (YYYY-MM-DD hh:mm:ss -mm), optionally with indication of the time zone (see 3.7.2).

In addition, custom information fields can be added by selecting **Edit > Information fields > Add information field...** in the object grid panel. This pops up the *Add field* dialog box (see Figure 3.2.7).

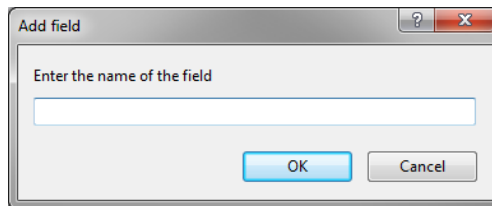


Figure 3.2.7: The *Add field* dialog box.

This dialog box prompts for the name of the custom field that will be created.

Names of custom fields are underlined in the information fields header, to indicate that their information can be edited directly in the panel (see 3.2.6).

To remove an information field from the database object, highlight the field in the information fields header and select **Edit > Information fields > Delete information field...**. A warning message will appear, stating that all information in this field will be deleted from the database. Press <OK> to remove the information fields and its content.

For objects that cannot be displayed in an object grid panel, custom information fields can also be defined when storing additional (meta) information might be relevant. This provides a very generic and powerful tool for making annotations at various levels.

The *Custom object fields* dialog box (see Figure 3.2.8) allows custom fields to be created and managed. It is called from the *Main* window by selecting **Database > Custom object fields...**

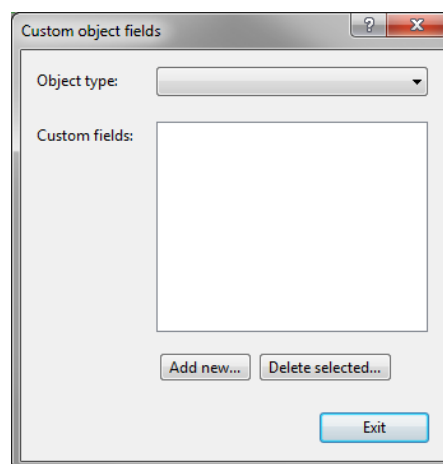


Figure 3.2.8: The *Custom object fields* dialog box.

From the **Object type** drop-down list, any object type that can have custom fields can be selected. For a selected object, the defined custom fields appear in the **Custom fields** list. A new custom field can be

defined with **<Add new>**. A selected (highlighted) custom field in the **Custom fields** list can be deleted with **<Delete selected>**.

Any custom object fields can be displayed and the information therein edited via *object queries* (see 3.2.14).

3.2.6 Editing object information

Information in custom information fields can be edited by clicking twice (not double-click) on an item, or by pressing **Ctrl+Enter** on the keyboard. The information then appears selected blue against the colored background and can be modified. This is illustrated in Figure 3.2.9, where information about gel staining is added to the *Fingerprint files* panel.

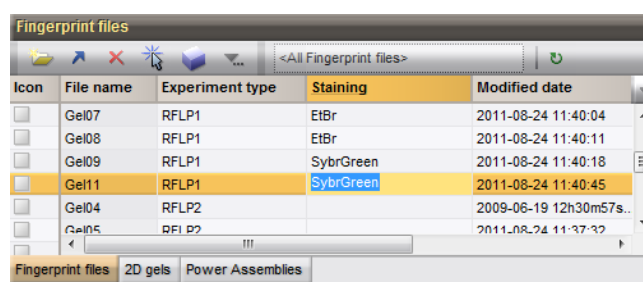


Figure 3.2.9: Editing information in non-default information fields, illustrated for the *Fingerprint files* panel.

If desired, Single-click field editing can be enabled in the *Preferences* window (see 2.3.3).

Information fields can be edited in bulk, i.e. for the selected objects, with **Edit > Information fields > Edit field in selection...** (**Ctrl+M**). This pops up the *Modify field for selection* dialog box (see Figure 3.2.10).

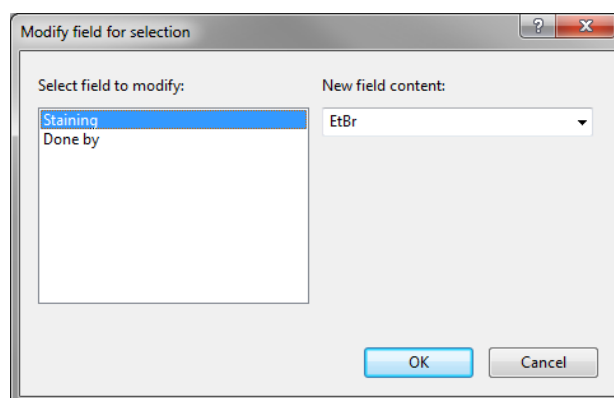



Figure 3.2.10: The *Modify field for selection* dialog box.

The **Select field to modify** list displays all editable information fields. The highlighted field can be updated with the **New field content** for the selected objects after pressing **<OK>**. The software will ask for confirmation before actually updating the information.

3.2.7 Displaying information fields

Object grid panels can be customized up to a high degree by the user. The width of columns can be changed by moving the separator line in the column heading left or right. Other column properties can be accessed

via the column properties button , located on the right hand side in the information fields header. As an example, the column properties of the *Fingerprint files* panel are illustrated in Figure 3.2.11.

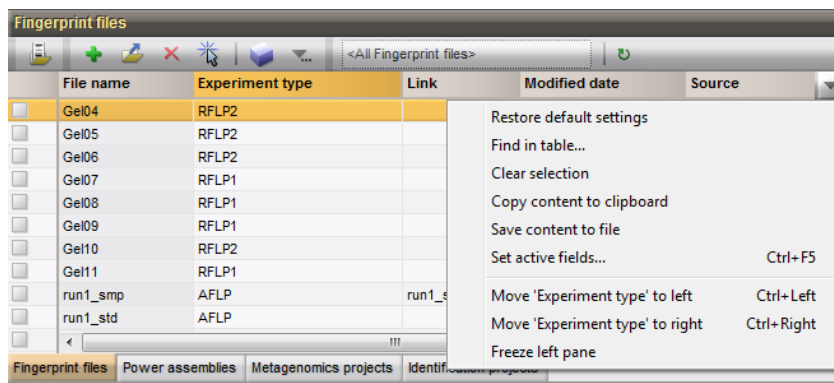



Figure 3.2.11: Column properties of the *Fingerprint files* panel.

To determine which information fields should be displayed, click on the column properties button  and select **Set active fields**. This pops up the *Set active fields* dialog box as shown in Figure 3.2.12.

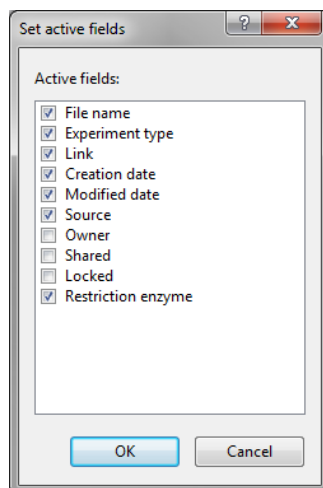


Figure 3.2.12: The *Set active fields* dialog box; checked field names will be displayed in the corresponding grid panel (here illustrated for the *Fingerprint files* panel).

This dialog box lists all available information fields for the database objects. The fields for display can be enabled and disabled with the check boxes next to the field names. The display of the corresponding object grid panel will be updated after pressing **<OK>**.

The relative position of a selected column within the panel can be changed using the menu items **Move 'column name' to left** and **Move 'column name' to right**. The shortcut keys **Ctrl+left arrow** and **Ctrl+right arrow** can be used for the same purposes.

The option **Freeze left pane** allows the user to freeze one or more information fields so that they always remain visible left from the scrollable area.

Similar as for the window configuration (see 2.3.4), it is possible to revert to the default column properties settings for the active panel by selecting **Restore default settings**. This will disable all introduced changes to the column properties of the object grid panel.



Any information field can be used as *display field* for the objects by highlighting an information field and selecting **Edit > Information fields > Use as display field**. The content of the display field will be used to indicate the object in every window where reference to the object is made. The current display field is

visually distinguishable from other fields in the object grid panel by its darker shade.

3.2.8 Sorting object grid panels


Objects in object grid panels can be sorted based on the content of the highlighted information field in several ways:


- **Edit > Information fields > Sort by field** arranges information alphabetically (a to z).
- **Edit > Information fields > Sort by field (reverse)** sorts information in reverse alphabetical order (z to a).
- **Edit > Information fields > Sort by field (Numerical)** sorts numerical information in ascending order.
- **Edit > Information fields > Sort by field (Numerical, reverse)** sorts numerical information in descending order.


Some, but not all, object grid panels can be sorted manually. In panels where this is allowed, select **Edit > Move up** () to move the highlighted object towards the top of the list or **Edit > Move down** () to move the objects towards the bottom of the list.

3.2.9 Actions on objects

Objects can be created, opened or deleted from an object grid panel.

To create a new object, select **Edit > Create new object...** (). This will open a dialog box or wizard that will prompt for the necessary information for creating the object. Examples are the *Create a new experiment type* dialog box for creating experiment types (see 2.3.6) and the *Add new database entries* dialog box for creating database entries (see 3.3.2).

To open a highlighted object, select **Edit > Open highlighted object...** (, **Enter**) or simply double-click the object. This will open a dedicated window in which the object can be edited or its settings modified. Examples are the *Fingerprint* window for fingerprint files (see 4.1.3.1) and the *Entry* window for database entries (see 3.3.4).

Selections of objects (see 3.2.4) can be deleted with **Edit > Delete selected objects...** (). Note that, if no selection is present, the *highlighted* object will be deleted. The software will ask for confirmation before actually deleting the object(s) from the database.

Selected objects can be locked and unlocked in bulk with **Edit > Object properties > Lock selected objects...** and **Edit > Object properties > Unlock selected objects...**, respectively. For more information about object access settings, see 3.2.3.

3.2.10 Searching for objects using statements

Object grid panels can be searched for objects that fulfill a certain criterium or a specific combination of criteria.

Edit > Search... (**Ctrl+F**) calls the *Find objects in view* dialog box.

This dialog box helps to build a query that will return a set of database objects (i.e. entries, experiment types, comparisons, characters, ...).

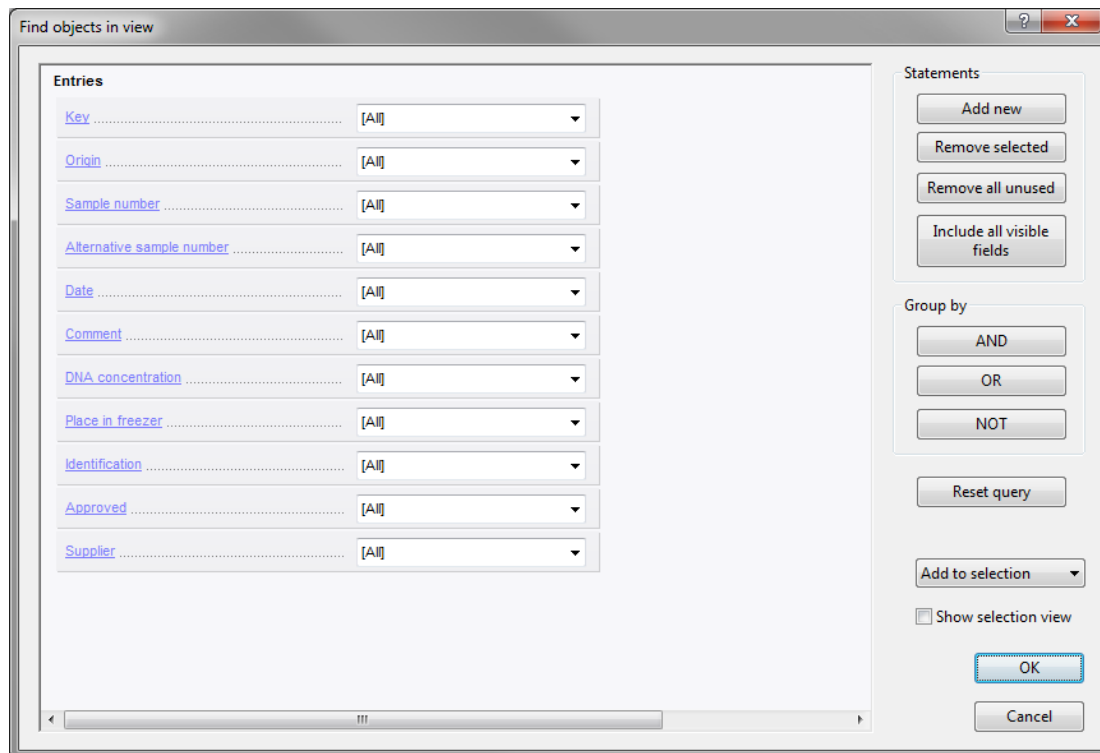


Figure 3.2.13: The *Find objects in view* dialog box, displayed for the *Database entries* panel.

On the left hand side, all information fields from the current view (see 3.2.2) are listed in the initial view. For each of these fields, a *select statement* can be prepared. A select statement consists of following components: an information field, an operator and a comparison value. The latter can be any text, date, number or one of the tokens [CurrentDate], [CurrentYear], [CurrentMonth], [CurrentDay] or [CurrentDateTime] (see 3.7.2 for a description). As soon as a comparison value is entered, the operator appears. By default this is *Equals*, but it can be changed into any of the following: *Contains*, *Begins with*, *Ends with*, *Smaller than*, *Larger than*, *Not larger than*, *Not smaller than*, *Differs from*, and *Contains no data*.

Pressing <Add new> will add a new select statement to the list, which initially appears as a question mark. Clicking on the question mark will display a list from which an information field can be selected.

Statements can be selected using **Ctrl+click**, as indicated with a colored border. Selected statements can be removed with <Remove selected>. Pressing <Remove all unused> will remove all statements for which no search value was entered.

Pressing <Include all visible fields> will prepare a statement for each of the information fields as displayed in the current view.

By default, all statements are combined with an AND logical operator. Selected statements can be combined with AND, OR, and NOT logical operators using the corresponding buttons. This allows for the construction of *compound statements* with the required level of sophistication.

To restore the initial view of the *Find objects in view* dialog box, press <Reset query>.

In case some objects were already selected prior to calling this dialog, three possibilities exist to deal with this:

- **Add to selection:** The previous selection is kept and objects that fulfill the current search criteria are added to this selection.
- **Replace selection:** All objects are first unselected and only objects that fulfill the current search criteria are selected.

- **Find in selection:** The current search criteria will be applied only to the already selected objects.

Checking **Show selection view** will display the selected objects in the panel after **<OK>** is pressed.

3.2.11 Finding an object in an object grid panel

Object grid panels potentially hold information from large numbers of objects. For example, the number of entries, gel files and comparisons all can be quite substantial in a database that has been used over a number of years. To quickly find back information about one or a few objects in such a long list, the command **Edit > Find object in list...** (🔍, **Ctrl+Shift+F**) is available. This functionality is fundamentally different from the search function described in 3.2.10 and which allows to select objects based on a combination of clearly specified criteria. Selecting **Edit > Find object in list...** (🔍, **Ctrl+Shift+F**) calls the *Find* dialog box (see Figure 3.2.14).

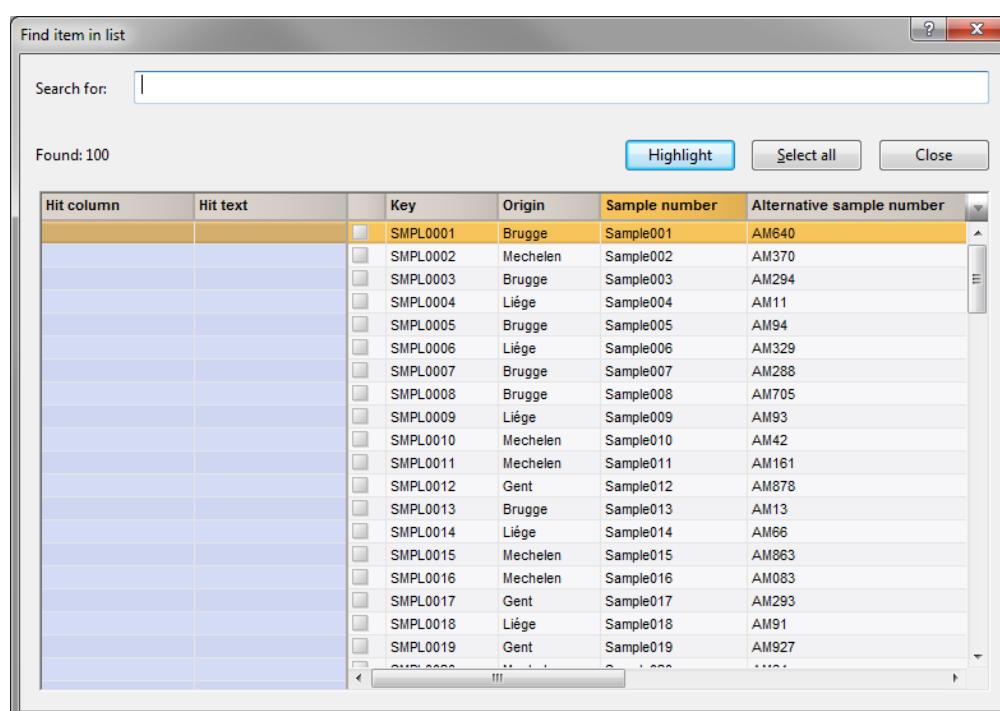


Figure 3.2.14: The *Find* dialog box, displayed for the *Database entries* panel.

In the text box next to **Search for:**, you can type some text that will be searched for in all currently displayed information fields. With every stroke on the keyboard, the display gets updated with those database objects that fulfill the search criteria (= "hits"). The number of hits is displayed as well (**Found:**). When one or more hits are found, the column 'Hit column' displays the name of the information field in which the hit was found and the column 'Hit text' shows the matched string.

Hits are sorted so that the most relevant ones are shown on top of the list:


- Hits that occur at the beginning of an information string are considered more important than hits in the middle of a string.
- Complete matches are ranked higher than partial matches.
- If two search strings are entered (separated with a space), objects that match both searches will appear higher on the list than objects that have a hit for only one of the search strings.

Further typing should narrow the search down to the specific object that you were looking for. When this object is the only hit, it will be automatically be highlighted. If a short list of objects is shown that all fulfill the search criteria, the object can be highlighted by clicking on it. When pressing the **<Highlight>** button or by hitting **Enter** on the keyboard, the highlighted object in the *Find* dialog box will become highlighted in the object grid panel from which the search was launched.

Alternatively, when looking for a group of objects, a selection of objects can be made manually (see 3.2.4) or all objects that fulfill the search criteria can be selected at once by pressing the **<Select all>** button. The objects will also be selected in the object grid panel.

Pressing the **<Close>** button closes the *Find* dialog box without changing the active object in the object grid panel.

3.2.12 Exporting object information

The currently displayed information in an object grid panel can be exported to the Windows clipboard for use in other programs by clicking the column properties button () and selecting **Copy content to clipboard**. The content of the clipboard can then be pasted in applications such as Notepad or MS Excel.

An alternative is to click the column properties button and use **Save content to file**, which will save the content of the object grid panel as an `export.csv` file, which will be opened in the default CSV editor (by default MS Excel). With the preference **Export table files in CSV format** (see 2.3.3.2), the software can be configured to export a tab-delimited `export.txt` file instead, which will open in the default text editor.

3.2.13 Object attachments


In addition to information fields, any database object can also have *attachments*. Attachments can be text (plain or formatted) or any kind of document. Since there is no hard-coded restriction on size, attachments can be used to store long descriptions that cannot be contained in information fields or to maintain links to e.g. images, PDF documents, text processor or spreadsheet files that in one way or another relate to the database object.

To add an attachment to an object that is displayed in an object grid panel, highlight the object and select **Edit > Object properties > Attachments window...**. This opens the *Attachments* window as shown in Figure 3.2.15.



Object attachments can also be accessed from the object's own dedicated window. For example, attachments of database entries via the *Entry* window, experiment type attachments in their respective experiment type windows (*Fingerprint type* window, *Character type* window, *Sequence type* window, etc.).

This window lists all attachments (if any) of the object with their 'Content type', 'Name' and 'Description'. Note that the latter two columns can be edited. In the left margin, an icon is displayed that corresponds to the default application for opening the attachment. The window furthermore provides the tools to add, delete and open attachments.

To add an attachment, select **File > Attachments > Add new attachment** () . This opens the *Create new attachment* dialog box as shown in Figure 3.2.16.

Following attachment types are supported:

- **Flat text:** Plain, unformatted text of unlimited length.
- **Formatted text:** Text of unlimited length in the Rich Text Format, supporting different font types, sizes, colors, etc. and the insertion of bitmap images.

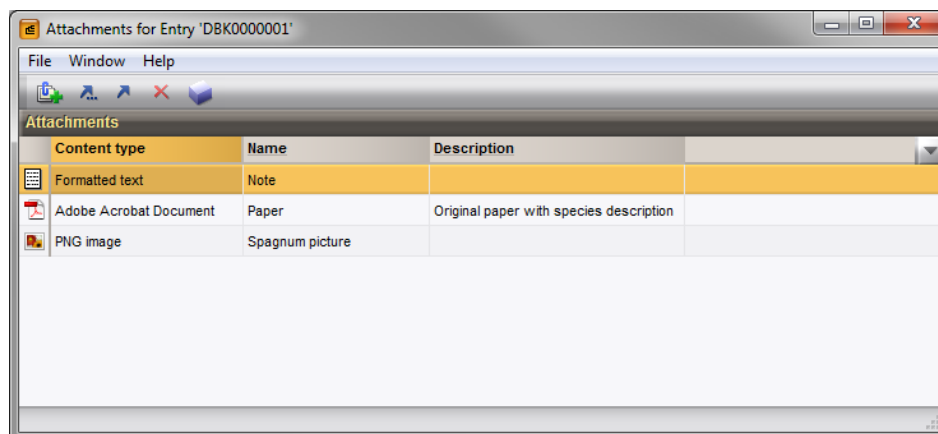


Figure 3.2.15: The *Attachments* window, shown for a database entry.

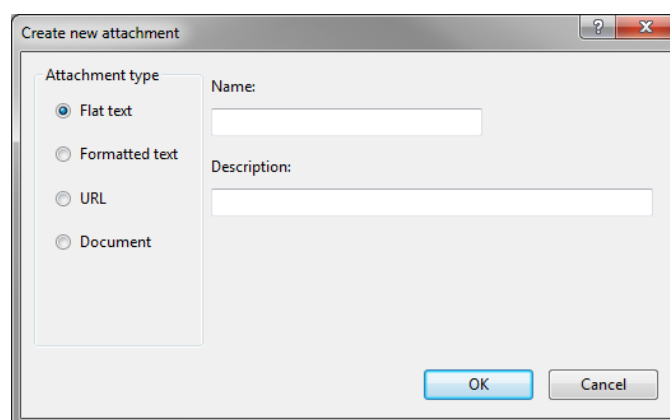


Figure 3.2.16: The *Create new attachment* dialog box.

- **URL:** An Uniform Resource Locator (URL) for a web page.
- **Document:** Any file on your computer or on a network drive.

A **Name** for the attachment and an optional **Description** can be entered in the corresponding text boxes. For text attachments, a name is required. For documents, the file name will be used in case no name was provided.

If **URL** was specified as **Attachment type**, the *Specify URL* dialog box opens.

In this dialog, the Uniform Resource Locator (URL) for the attached web page can be entered.

When **Document** was specified as **Attachment type**, the full path of the file that you want to have attached (**Source file**) can be entered or browsed for with the **<Browse>** button.

When the option **Store in database** is unchecked (the default setting), a copy of the document will be stored in the *Sourcefiles* subdirectory of the database folder (see 3.7.2). With the **Store in database** option checked, the document will be stored as a record in the "OBJATTCH" table (see 21.1). In case different database users need access to the attachment, storing the attachment in the database is to be preferred. Text attachments are always stored in the database.

When **<OK>** is pressed, the attachment will be added to the list in the *Attachments* window. For text attachments (plain or formatted), the *Text attachment* window appears (see Figure 3.2.17).

In this window, text attachments can be edited. Note that the formatting options are inactivated for flat text attachments.

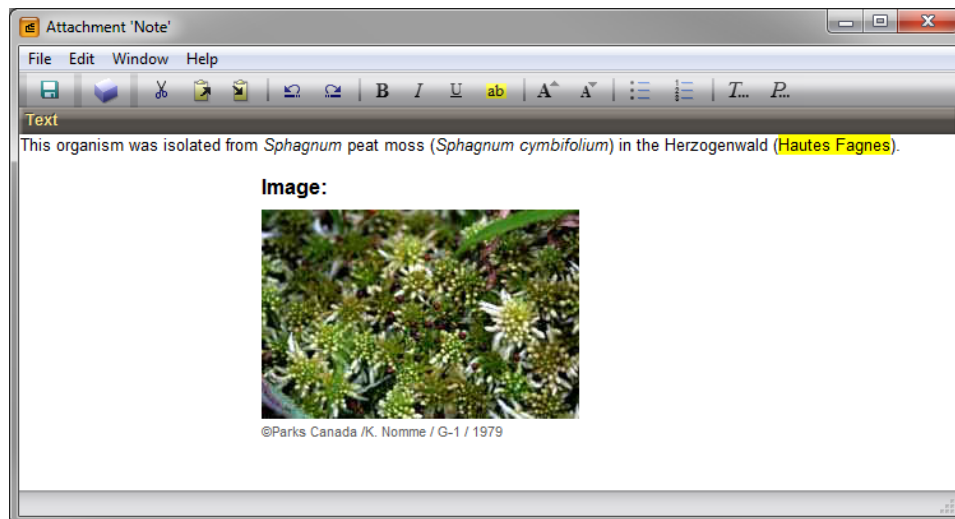


Figure 3.2.17: The *Text attachment* window for a formatted text attachment.

In the *Text* panel, text can be entered via the keyboard or pasted with **Edit > Paste** (📄). Highlighted text can be copied or cut with **Edit > Copy** (📄) or **Edit > Cut** (✂️), respectively. Editing actions can be undone with **Edit > Undo** (↶). To redo an undone action, select **Edit > Redo** (↷). Selecting **File > Save** (💾) will save the text attachment.



In case a flat text attachment type was opened, formatted text and bitmaps can be pasted from other applications and will be displayed in the current session. However, when the attachment is closed and opened again, only the plain text will be retained.

For formatted text attachments, a number of formatting options are available.

To make text bold, italic, underlined or highlighted, use **Edit > Formatting > Bold** (B), **Edit > Formatting > Italic** (I), **Edit > Formatting > Underline** (U) or **Edit > Formatting > Highlight** (ab), respectively. To increase the font size, use **Edit > Formatting > Larger font** (A⁺). Conversely, **Edit > Formatting > Smaller font** (A⁻) will decrease the font size.

More font options are available when **Edit > Formatting > Set font...** (T...) is selected. This pops up the *Set font* dialog box.

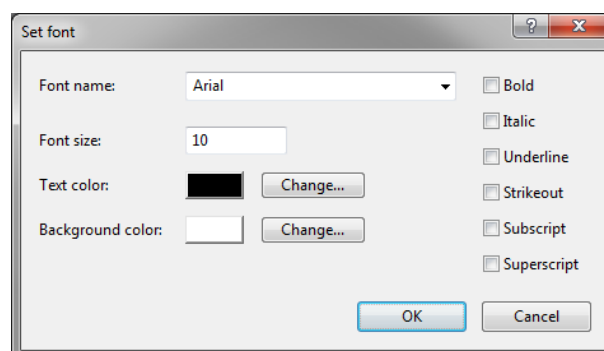


Figure 3.2.18: The *Set font* dialog box.

The font type can be selected from the **Font name** drop-down list and the **Font size** entered in the corresponding text box. The **Text color** as well as the **Background color** can be set by pressing the corresponding **<Change...>** button. This action will display the *Color* dialog box.

Any desired color can be picked from this dialog using (a combination of) any of the methods below:

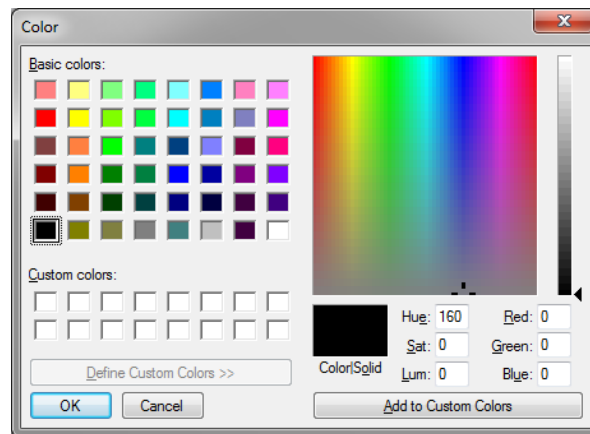


Figure 3.2.19: The *Color* dialog box.

- By clicking any of the **Basic colors** in the upper left-hand side of the dialog box.
- By clicking any of the **Custom colors** (if defined) in the lower left-hand side of the dialog box.
- By entering **Red**, **Green** and **Blue** values in the corresponding text boxes.
- By entering **Hue**, Saturation (**Sat**) and Luminosity (**Lum**) values directly in the corresponding text boxes.
- By picking a point in the hue-saturation plot on the right-hand side of the dialog box and selecting a luminosity value using the slider on the far right.

When a color is selected, it can be added to the **Custom colors** by clicking on one of the custom color cells and pressing the **<Add to Custom Colors>** button.

Press **<OK>** to use the selected color in the *Color* dialog box. Conversely, press **<Cancel>** to keep the original color.

The check boxes on the left hand side of the *Set font* dialog box allow text to be displayed as **Bold**, **Italic**, **Underline**, **Strikeout**, **Subscript**, and **Superscript**, respectively.

To format text as a bulleted or numbered list, use **Edit > Formatting > Bulleted list** (☐) or **Edit > Formatting > Numbered list** (☐), respectively.

Additional paragraph formatting options become available with **Edit > Formatting > Set paragraph style...** (☐). This action displays the *Paragraph style* dialog box (see Figure 3.2.20).

This dialog box offers some basic paragraph formatting options. The text **Alignment** can be set from the drop-down list. Options are **Left**, **Right**, **Center**, and **Justified**.

A **Left indent** and a **Right indent** can be specified. In addition, a **Left offset** for all but the first line of the paragraph can be entered as a positive or negative number.

The (vertical) **Space before paragraph** and **Space after paragraph** can be specified as well.

When done editing the text attachment, select **File > Save** (☐) to save the modifications to the database. If you try to close the *Text attachment* window (e.g. via **File > Exit**) when there are some unsaved changes to the text, the software will warn you for this.

Since object attachments are database objects as well, they also have object access settings as discussed in 3.2.3 and they can be versioned and included in the audit trail as discussed in 3.6.

Attachments listed in the *Attachments* window can be opened for editing by selecting **File > Attachments > Edit attachment** (☐) or by simply double-clicking on the attachment. For text attachments, the *Text at-*

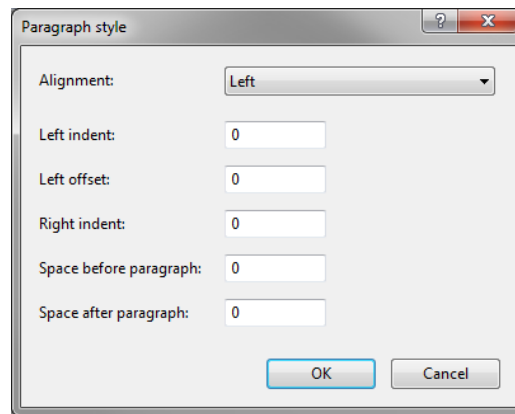


Figure 3.2.20: The *Paragraph style* dialog box.

tachment window will appear. For documents, the default application to edit the document type (as specified in the Windows operating system) will open with the document loaded. While the document is being edited, BioNumerics will be locked. One can only resume working in BioNumerics after closing the application in which the attachment is edited.

Attachments can be viewed (i.e. read-only) by highlighting the attachment and selecting **File** > **Attachments** > **View attachment** (🔍). For text attachments, the *Text attachment* window will open in read-only mode. For documents, the default application to view the document type will open with the document loaded. Obviously, it is possible to continue working with BioNumerics while an attachment is being viewed.

To delete a highlighted attachment, select **File** > **Attachments** > **Delete attachment** (🗑️) and confirm the action.

3.2.14 Object queries

The object query tool is a flexible and powerful way to query any object type in the database, even if the object does not have a dedicated object grid panel. A user-friendly query builder makes it possible to design queries on any combination of information fields for the object and its parent object. Custom queries can be saved in the database and are immediately available.



Since object queries are in fact SQL statements, they only work on the relational database and not e.g. comparisons stored as files.

To create a new object query and/or to access any saved queries, select **Database** > **Object queries...** (📄) in the *Main* window. This will display the *Select object query* dialog box (see Figure 3.2.21).

This dialog box lists previously made object queries (if any) and an additional item <Create new ...>. Queries listed here are the queries of the currently logged on BioNumerics user and any shared queries.

Selecting <Create new ...>, followed by <OK> will create a new object query via the *Object query* dialog box (see Figure 3.2.22).

From the **Object to report** drop-down list, any object type can be selected. When an object type is selected, the available fields are displayed in the **Include fields** list. By default, all available fields are highlighted and will be included in the object query. However, the user can select any combination of fields to be included. The parent objects of the object to report are available from the **Include parent object** drop-down list and can be included in the object query as well. Again, any combination of parent object fields can be selected.

A field indicated with "(*)" is an **Object identifier**, i.e. a unique "name" that is used to identify an object. All

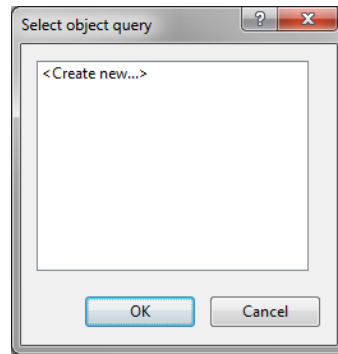


Figure 3.2.21: The *Select object query* dialog box.

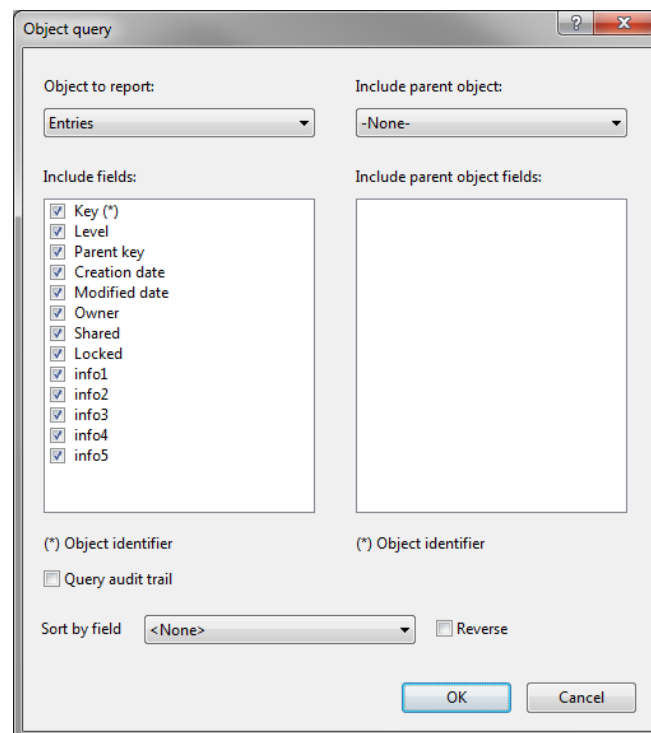


Figure 3.2.22: The *Object query* dialog box.

objects have such an object identifier. In some cases, a compound identifier is used; the object is uniquely defined by a combination of two fields (both indicated with "(*)").

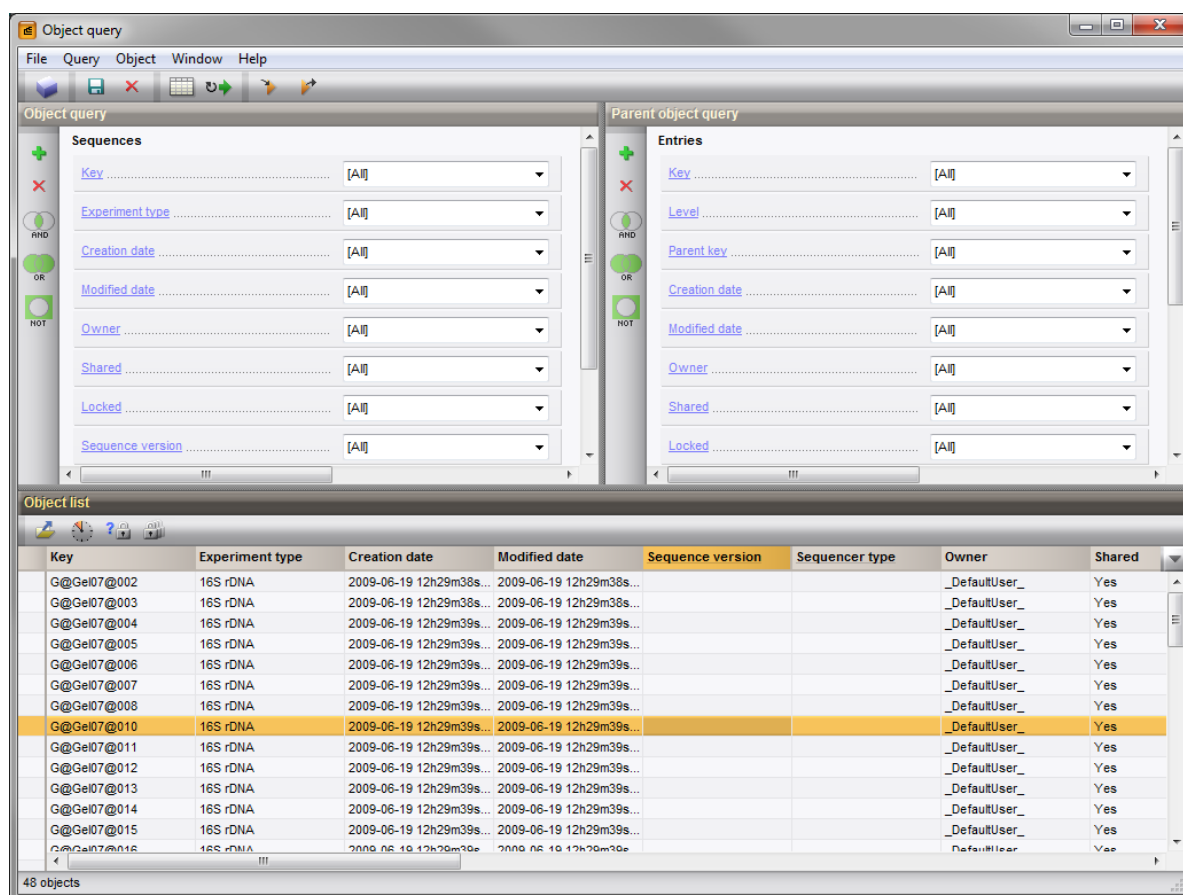
When **Query audit trail** is checked, the query will not be limited to the current version of objects, but historic versions of the objects, as present in the audit trail, will be queried as well (see 3.6 for more information about the audit trail).

From the **Sort by field** drop-down list, a field to sort the query results on can be selected. Checking **Reverse** will sort the query results in descending order.

Pressing **<OK>** will open the *Object query* window with the query results (see Figure 3.2.23).

The *Object query* window consists of three dockable panels: the *Object query* panel and the *Parent object query* panel, each containing a graphical query builder for the respective queries, and the *Object list* panel, displaying the query result.

The graphical query builder works in a similar way as described for the *Find objects in view* dialog box (see 3.2.10). Once a query is created, it can be run with **Query > Run query** (🏃🏻♂️).

Figure 3.2.23: The *Object query* window, initial view.

Note that the *Object list* panel is a grid panel, for which certain fields can be displayed or hidden and moved left or right. Furthermore, objects can be sorted according to any field and information in editable fields (those where the field name is underlined) can be edited, including the information in custom object fields (see 3.2.5).



The query builder recognizes the special tokens [CurrentUser], to denote the database user that is currently logged in, and [CurrentDate]-X, to specify a moment in the past, relative to the current date. An example of the latter would be "Modified date Larger than [CurrentDate]-7" to find objects that were modified in the last seven days.




When querying fields, a difference exists between the expression "Equals empty string" and "Contains no data" (DBNULL). When an information field is created in BioNumerics, it gets initialized. Therefore, empty information fields will match the "Equals empty string" expression and not "Contains no data".

A selection of *objects* can be made from the *Object list* panel (see 3.2.4) and exported as a selection of *entries* in the database. Obviously, this mapping of objects to entries can only be done if the objects actually relate to entries, i.e. if a 'Key' field is available for either the object itself or for the parent object that is included in the query.

Use **Object > Select all (Ctrl+A)** to select all objects and **Object > Clear selection** to unselect all objects. Selected objects can be brought to the top of the *Object list* panel with **Object > Bring selection to top**.

Once an object selection is made, it can be exported as a selection of entries with **Object > Export entry selection** (📁). Conversely, a selection of entries can be mapped to a selection of objects with **Object > Import entry selection** (📁).

Object queries can be saved and added to the list of saved object queries (see Figure 3.2.21) with **File > Save** (). This will display the *Save object query* dialog box.


In this dialog box, the name for the object query should be entered.




When the *Object query* window with an unsaved query is closed (e.g. with **File > Exit**), you will NOT be prompted to save the query!





An object query is an object itself and therefore also has object access settings. Object queries that are specified as 'not shared' will not be seen by users other than the object query's owner.


An object query can be deleted with **File > Delete this query** ().


To change the objects that are being queried, select **Query > Edit query source...** (). This will display the *Object query* dialog box again, as shown in Figure 3.2.22.

In the *Object query* window, the access privileges for the selected objects (see 3.2.3) can simultaneously be modified. This is a very powerful tool, because it allows one to e.g. lock in bulk a whole set of objects, based on a query.

Editable fields (underlined) can be edited directly in the *Object query* window by clicking twice or pressing **Ctrl+Enter**. Objects can be opened in their corresponding window by double-clicking the object or by first highlighting the object and then selecting **Object > Open object** ().

The object access settings can be edited from the object's specific window as discussed in 3.2.3. For the highlighted object in the *Object query* window, they can also be edited with **Object > Object access status...** ().

For a selection of objects, the object access settings can also be changed with **Object > Lock**, **Object > Unlock**, **Object > Share**, **Object > Disable sharing**, or **Object > Take ownership**. These commands are also available from the  button.

Using **Object > Show audit trail...** (), the audit trail of a highlighted object can be displayed (see 3.6 for more information regarding the audit trails and versioning tool).



For character experiments that were created in an older version of the software, the experiment objects will not exist yet (the table *CharacterTypeName*EXPER in the connected database is not populated). A script is available that will create the experiment objects for existing character experiments. Please contact Applied Maths to obtain this script.

3.2.15 Cross-links between database objects

3.2.15.1 Introduction

In BioNumerics, *cross-links* can be made with the purpose of documenting an existing relation of some kind between any two database objects (see 3.2.1) or *aspects* thereof. Cross-links are directional, meaning that the description of the forward relation can be different from the reverse relation. For example, if cross-links would be used to describe family ties in a hospital's patient database, the cross-link between a father and his daughter would be described by a forward relation name of "Is parent of" and a reverse relation name of "Is child of".

As already mentioned above, *aspects* of objects can also be targeted by cross-links. A typical example of this could be in PCR-DGGE analysis, where certain bands are excised from the DGGE gels to be re-amplified and sequenced. To document the fact that a certain sequence (a database object) is originating from a specific band in a PCR-DGGE profile, a cross-link can be created between the sequence and the band. Although an individual band is not a database object, this can be done because it is regarded as an *aspect* of a fingerprint experiment, which is indeed a database object.

Cross-links fundamentally differ from entry dependencies (see 3.3.10) in the sense that cross-links can be made between any two database objects, while dependencies are implicit one-to-many relations that occur between entries at different database levels.

3.2.15.2 Creating object cross-links

To create a cross-link between two database objects, the objects should either be open in their dedicated window or highlighted in the *Main* window.

Cross-links are created in the *Crosslink* window, which can be opened from many different places in BioNumerics. In general, the window should be called from the object-specific window of the *source component*, i.e. the object from which the cross-link should start from.

For example, to create a cross-link that starts from the highlighted entry in the *Database entries* panel of the *Main* window, select **Edit** > **Object properties** > **Object crosslinks...** to open the *Crosslink* window (see Figure 3.2.24).

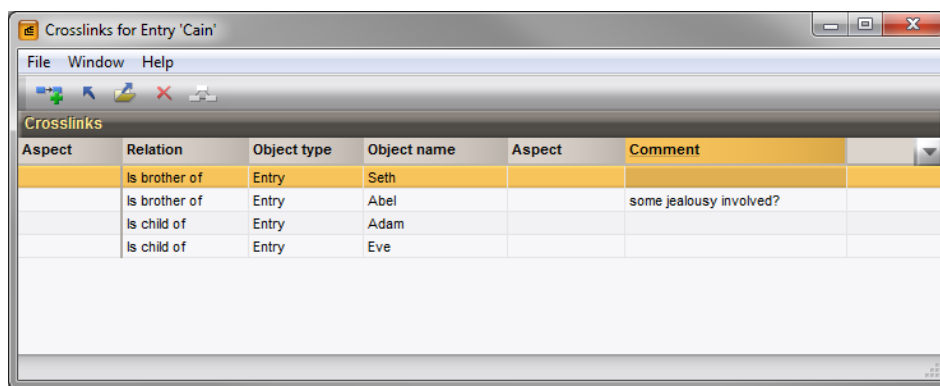


Figure 3.2.24: The *Crosslink* window, here showing the cross-links for a database entry.

The *Crosslink* window lists all cross-links for the object that the window was called for. The grid panel shows for each cross-link the Aspect of the source object (empty if the object itself was referenced), the Relation name, the Object type of the destination object, the Object name of the destination object and the Aspect (if any) that was referenced. In addition, each cross-link has a Comment, which can be directly edited in the grid panel.

To create a cross-link, select **File** > **Crosslinks** > **Add new crosslink** (🔗). This action will open the *Link source* wizard page (see Figure 3.2.25).



Since entries are the central entities in a BioNumerics database and cross-links often have entries as source component, the functionality of the *Crosslink* window is integrated in the *Crosslinks* panel of the *Entry* window. Therefore, the same action as described above can also be done via **File** > **Crosslinks** > **Add new crosslink** (🔗) in the *Entry* window.

In this page of the wizard, the **Link source**, i.e. the object from where the cross-link should start, needs to be selected from the list. This list might contain several objects, but often just a single object is displayed.

Pressing <Next> will display the *Link destination* wizard page (see Figure 3.2.26).

In this page, the **Link destination** should be selected. The tree control on the left shows an overview of all currently open windows that provide objects (or object aspects) to link to. If the object that you want to link to is not displayed in this tree, you should close the *Create new crosslink* wizard and open the dedicated window of that object.

Pressing <Next> will open the *Link type* wizard page (see Figure 3.2.27).

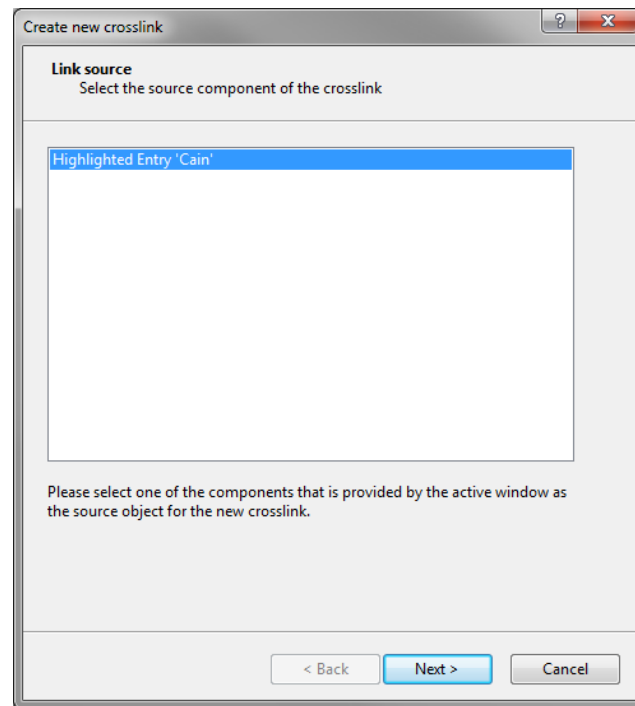


Figure 3.2.25: The *Link source* wizard page.

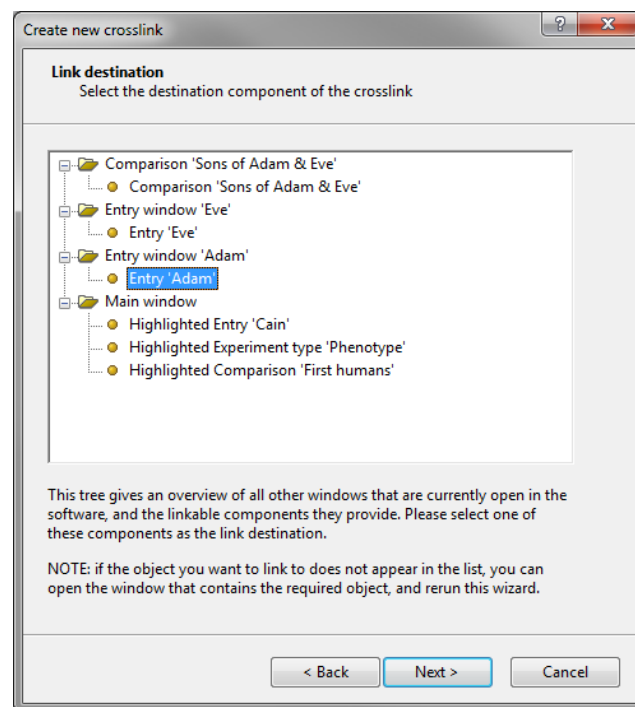


Figure 3.2.26: The *Link destination* wizard page.

Here, the *Link type* (or *Relation type* should be specified. Optionally, a *Link comment* can be specified to further annotate the cross-link.

There is one very general catch-all *Relation type* specified by default: "Relates to". Typically, one wants to be more descriptive when documenting a cross-link and to achieve this, a new *Relation type* can be created by pressing the *<Create new>* button. This action will open the *Edit relation type* dialog box (see Figure 3.2.28).

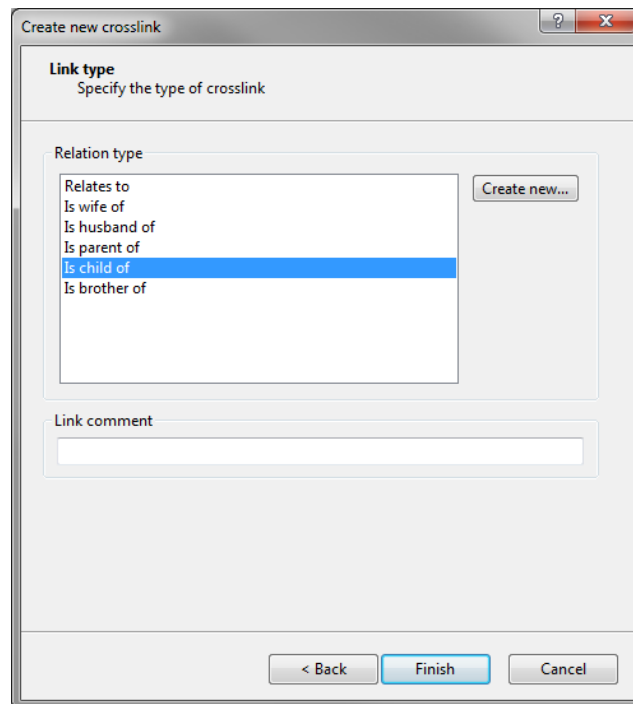


Figure 3.2.27: The *Link type* wizard page.

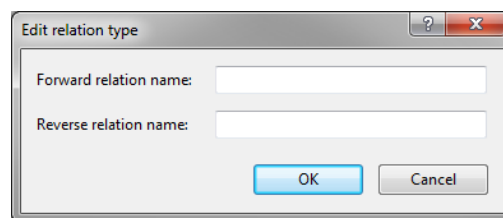


Figure 3.2.28: The *Edit relation type* dialog box.

The dialog box prompts for a **Forward relation name**, i.e. a sentence describing the relation that the source component has with the destination component (e.g., “Is child of”), and for a **Reverse relation name**, i.e. a sentence describing the relation that the destination component has with the source component (e.g., “Is parent of”).

When **<OK>** is pressed, the new relation type will be listed in the *Link type* wizard page. Only relation names (forward and/or reverse) that are compatible with the source object will be displayed in the list.

With a **Relation type** selected from the list, pressing **<Finish>** will create the cross-link, which will be listed in the *Crosslink* window.

3.2.15.3 Managing object cross-links

From the *Crosslink* window, the source object of a cross-link can be opened in its own dedicated window with **File > Crosslinks > Open link source** (🔗). Conversely, the destination object can be opened with **File > Crosslinks > Open link destination** (📁) or by double-clicking the cross-link.

A highlighted cross-link in the *Crosslink* window can be deleted with **File > Crosslinks > Delete crosslink** (✖). The software will ask for confirmation before actually deleting the cross-link.

3.2.15.4 Visualizing object cross-links

When using cross-links extensively, it can be very useful to have a graphical overview of all cross-links between objects. Such a graphical overview can be displayed in the *Crosslink map* window by selecting **File > Crosslinks > Show map** (🗺️) (see Figure 3.2.29).

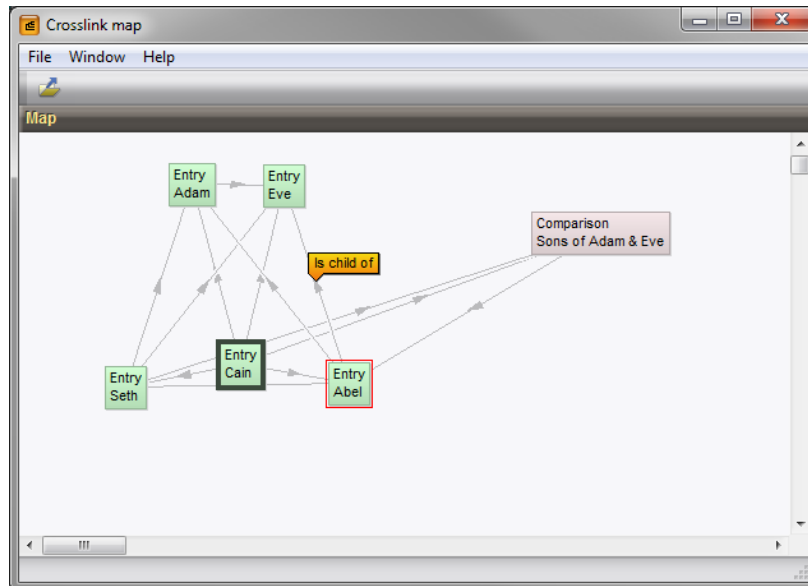


Figure 3.2.29: The *Crosslink map* window.

The *Crosslink map* window provides a graphical overview of all objects or object aspects that are linked directly or indirectly to the source component. The latter is indicated with a thick border. Different objects types are shown in differentiating colors. An object on the map can be opened in its dedicated window by highlighting it and selecting **File > Open link** (🔗) or simply by double-clicking the object.

Forward or reverse relations are indicated by the direction of the arrows on the cross-links. When hovering over a cross-link, the forward or reverse relation name and the optional comment pops up.

To optimize the map's display, the objects can be re-positioned on the map by dragging them with the mouse.

Chapter 3.3

Database entries

3.3.1 Introduction

The core unit of a BioNumerics database is the *entry*. Entries represent the biological entities for which data is sampled, digitized and imported, to be further compared and analyzed. Each entry is identified by a unique *key*, through which various pieces of information are linked to the appropriate entries: information fields, attachments, fingerprints, contig projects and sequences, character data, etc.. (see Figure 3.3.1).

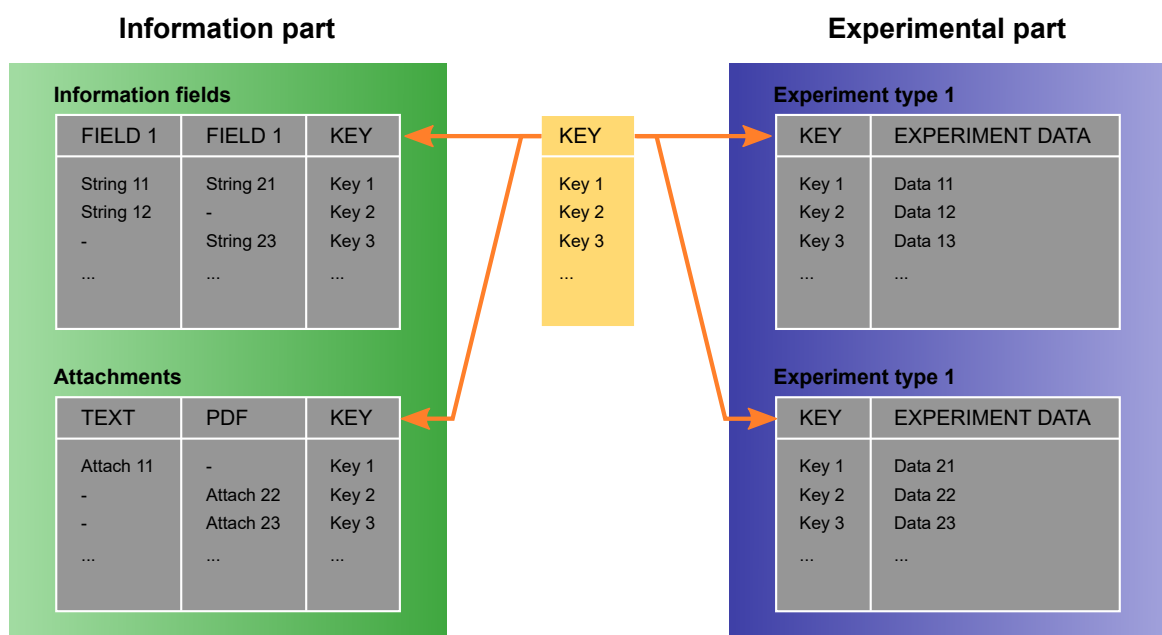


Figure 3.3.1: Linking various sorts of information to database entries through unique keys.

3.3.2 Adding and removing database entries

Adding entries to the database can be done in two ways:

- You can add one or more entries directly to the database. Initially, these entries will not have any information and no experiments will be linked to them. When you import experiment data later on, you can link the data to the entries.

- When you import entry information or experimental data (see 3.3.5), the software will automatically create the corresponding database entries.

To create new database entries without importing experiment data, click on the *Database entries* panel to highlight this panel in the *Main* window and select **Edit > Create new object...** (+). The *Add new database entries* dialog box appears (see Figure 3.3.2).

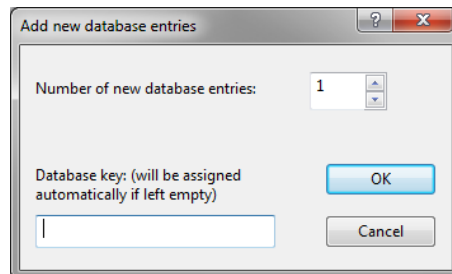


Figure 3.3.2: The *Add new database entries* dialog box for adding new entries to a database.

The dialog box prompts for the *Number of new database entries* to create. The *Database key* input field in the bottom of the window allows a key to be entered by the user. This input field is only accessible when a single entry is added. As soon as the number of entries is specified to be more than one, the field is disabled and the keys will be generated automatically by the software.

The way that BioNumerics generates unique keys can be modified in the *Database settings* dialog box (see 3.7.2).

The key is a critical identifier of the database entries (see 3.3.1), and if you already have *unique* labels that identify your organisms under study, you can use these labels as keys in BioNumerics. In the latter case, they can be effectively used as a database field. The key is also an important component in automatically linking experiments to existing database entries.

To remove one or more entries from the database, they first need to be selected in the *Database entries* panel. See 3.3.8 for more information on the selection of entries. Next, select **Edit > Delete selected objects...** (X).



There is no undo function for this action and removed entries are irrevocably lost, together with any experiment information linked to them! A removed entry can *only* be restored when the audit trail is enabled (see 3.6).

To remove all entries that have no experiments linked to them, you can select **Database > Entries > Remove all unlinked entries...**

3.3.3 Creating information fields

A number of predefined information fields are automatically created when creating a new database (see 3.1.3). Additional information fields can be added with **Edit > Information fields > Add information field...** or by highlighting the *Entry fields* panel and selecting **Edit > Create new object...** (+). In both cases, the *Create new entry information field* dialog box pops up (see Figure 3.3.3).

This dialog prompts for a *Name*, i.e. the name for the information field as being displayed in the BioNumerics user interface.

Optionally, a "machine name" or *Field ID* can be specified by pressing the <ID> button. This is only needed if you want complete control over the column name in the relational database in case of fixed fields (see further). If not specified, BioNumerics will automatically create a "safe" *Field ID* based on the *Name*.

Two possibilities exist for *Origin*:

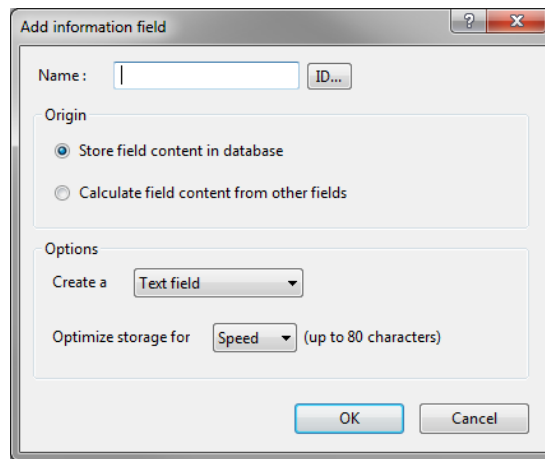


Figure 3.3.3: The *Create new entry information field* dialog box.

- **Store field content in database:** Any information that will be entered in this field, will be stored in the relational database. This is by far the most common option.
- **Calculate field content from other fields:** A calculated field is not stored in the database, but derived from other information fields. Only the "recipe" of how to calculate the field is stored.

Depending on which option is checked, the field *Options* will be different.

Following *Options* are available if *Store field content in database* is checked:

The type of content that the field will contain (*Create a ...*). Following options are available from the drop-down list:

- **Text field:** any text is allowed.
- **Number field:** only numerical values are allowed, a dot is used as decimal separator.
- **Date field:** a date in the format YYYY-MM-DD.



For existing information fields, the type of content that the field should contain can be changed in the *Database field properties* dialog box (see 3.3.6).

Optimize storage for either *Speed* or *Space*. Note that the maximum length of the field will be 80 characters with the first option and 150 characters if the second option is chosen.

This option will only appear if your database does not contain levels or when "All levels" is the active database level. For more information about database levels, see 3.3.10.



With the option **Optimize storage for Speed** selected, an information field corresponds to a column in the ENTRYTABLE table of the relational database (see 21.1). Its creation therefore requires "alter table" rights on the relational database. This type of field is more efficient in terms of database performance, since the SQL statements involve only a single table.

With the option **Optimize storage for Space** selected, an information field is stored in a normalized way in the relational database. Such "flexible" fields are easier to create and remove than fixed fields, because these actions do not involve a change to the relational database table structure. However, to retrieve information from flexible fields, several database tables need to be joined together, resulting in longer execution times of the SQL queries.

If *Calculate field content from other fields* is checked, the *Composition* of the calculated field should be specified. This can be done by pressing the *<Edit...>* button, which calls the *Calculated database field settings* dialog box as discussed in 3.3.6.

Pressing *<OK>* will create the new entry information field. Note that an unlimited number of fields can be defined in a database, but only 500 fields can be displayed at the same time.

3.3.4 Entering information in entry fields

Similar as for any database object in an object grid panel (see 3.2.6), information in non-default information fields can be edited directly by clicking twice on an information field in the database. The information will appear highlighted and can be edited as if it was in a spreadsheet program. When field states are defined (see 3.3.6), they now become available as a drop-down list.

When selecting *Edit > Open highlighted object...* (🔍, **Enter**), the *Entry* window appears for the currently highlighted entry (see Figure 3.3.4). Double-clicking an entry or right-clicking on an entry and selecting *Open highlighted object* also works.

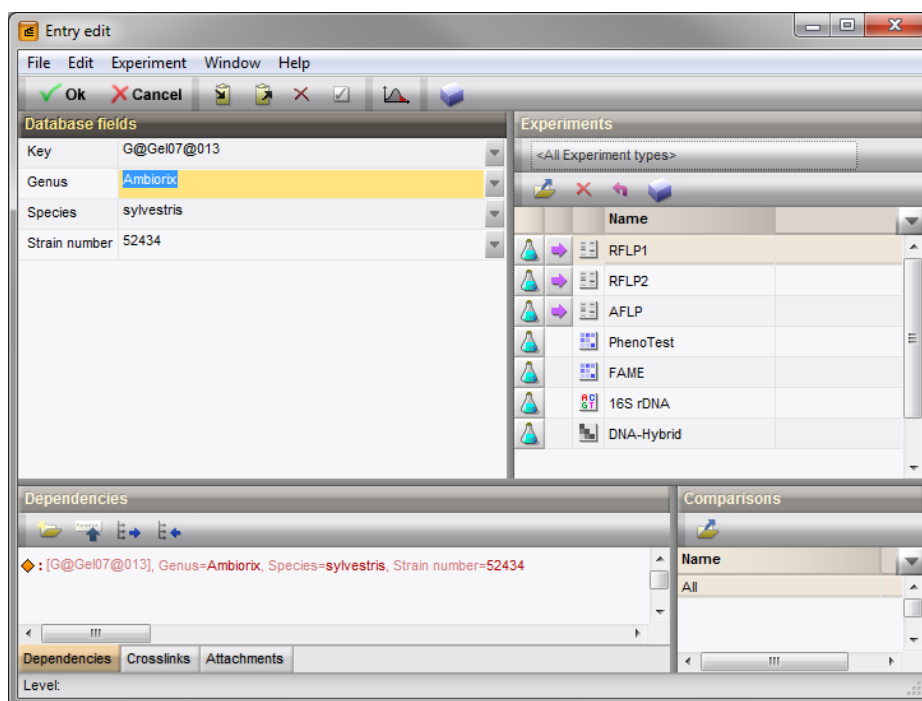


Figure 3.3.4: The *Entry* window, for editing information of a specific entry.


The *Entry* window contains six dockable panels:

- The *Database fields* panel (top left in default configuration) shows the information fields and the *Experiments* panel (top right) shows the available experiments for the entry.
- The *Dependencies* panel displays the entry's dependencies on entries in other database levels (for more information about levels and dependencies, see 3.3.10).
- The *Crosslinks* panel, that becomes available by clicking on its tab, shows the cross-links of this entry with other database objects (see 3.2.15 for more information about cross-links).
- The *Attachments* panel allows attachments to be added and viewed for the entry (see 3.2.13).

- The *Comparisons* panel shows all comparisons in which the entry is used.

The *Entry* window can be rescaled to see more and/or longer information fields. The relative size of the panels can also be modified by dragging the separator line between the panels. The display of the panels in the *Entry* window can be customized as described in 2.3.4.


Entry information for each of the available database fields can be entered in the corresponding text boxes.



If a number of entries have mostly the same fields, you can copy the complete entry information to the clipboard using *Edit > Copy to clipboard* ()



To clear the complete information of the entry, use *Edit > Clear all database fields* ()

To paste the information from the clipboard, use *Edit > Paste from clipboard* ()

If some of the information fields are the same as entered for previous entries (for example genus and species name), you can drop down a history list for each information field. The history lists can contain up to 10 previously entered strings for the information field. Using the history lists is recommended (1) to save time and work and (2) to avoid typographical errors.

Drop down a history list by clicking the  button on the right hand side from the information field. An information string can be picked from the list.


Using *Edit > Select / unselect this entry* () , you can select or unselect the opened entry in the database (see Figure 3.3.4), for the construction of comparisons. When the entry is selected, the button shows as .

Select *File > Save changes and close* () to close the *Entry* window and store the information, or select *File > Cancel changes and close* () to close the window without changing any information.

In order to quickly enter the same information for many entries, the use of the keyboard is recommended: use the **Arrow Up** and **Arrow Down** keys to move through the entries in the database, use the **Enter** key to edit an entry, use the **Ctrl+C** and **Ctrl+V** keys to copy and paste information, and use the **Enter** key again to close the *Entry* window.

3.3.5 Importing information from external sources

3.3.5.1 Introduction

Selecting *File > Import...* (, **Ctrl+I**) in the *Main* window calls the *Import* dialog box (see Figure 3.3.5). The import tree options are organized in groups based upon the type of data. By default, all groups are collapsed. A group can be expanded by clicking on the "+" sign next to the name of the group. When a particular import option is selected in the tree, a short description appears in the right panel.

The *Import* dialog box allows the following data types to be imported:

- **Entry information data:** Database information can be imported from text, Excel, and other ODBC-compatible files and linked to new or existing entries using the *Import fields (text file)*, *Import fields (Excel file)* and *Import fields (ODBC)* options, respectively. These import options share most steps in the *Import wizard* and are covered in the same section, i.e. in 3.3.5.2. With the *Import mapping* option, data mappings can be imported from external text or Excel files and applied on existing entries (3.3.5.3).
- **Fingerprint type data:** Gel files can be imported in the database using the *Import gel file* option. In an optional step, the *Fingerprint image import* window can be called to perform some preprocessing actions (see 4.1.3.1). Automated sequencer curves can be imported with the option *Import curves*. Supported formats are Applied Biosystems and Beckman. Text files containing a listing of peaks can be imported with the option *Import peak table*. This import option comes with two predefined

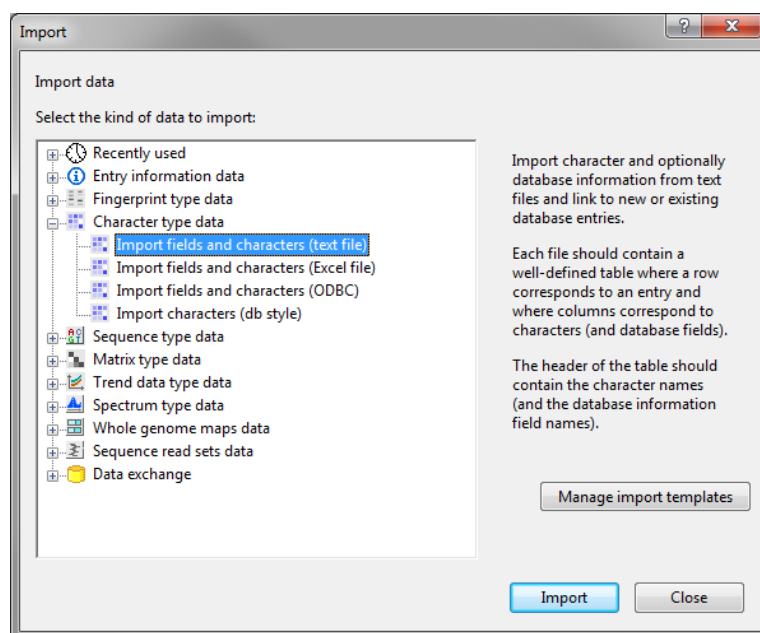


Figure 3.3.5: The *Import* dialog box.

formats to make commonly used AB GeneMapper and Beckman peak files easy to import. Optionally, a custom format can be created and saved in the database. Detailed information about the import of peak table files and sequencer curve files can be found in 4.1.4.1.

- Character type data:** Character type data can be imported from text, Excel, or other ODBC-compatible files and linked to new or existing entries using one of the character import routines in the import tree. Depending on the format of the data and file type, the user can choose between following import options: *Import fields and characters (text file)*, *Import fields and characters (Excel file)*, *Import fields and characters (ODBC)* and *Import characters (db style)*. The first three options share most steps in the *Import wizard* and are covered in the same section, i.e. in 6.1.3.2. The *Import characters (db style)* option is discussed in 6.1.3.3.
- Sequence type data:** Sequences in FASTA format and EMBL/GenBank format can be imported from text formatted files and linked to new or existing entries in the database using the import options *Import FASTA sequences from text files* (see 8.1.3.4) and *Import EMBL/GenBank sequences from text files* (see 8.1.3.5), respectively. With the import routine *Download sequences from internet*, sequences can be fetched from online repositories and linked to new or existing entries in the database (see 8.1.3.6). Sequences can be imported and assembled with the options *Import and assemble trace files* and *Import and assemble traces from FASTA text files*. Binary chromatogram files from Applied Biosystems, Beckman, and Amersham automated sequencers are accepted, and FASTA formatted sequences stored in text files. Detailed information can be found in 8.1.3.2.
- Matrix type data:** Similarity and distance matrices can be imported from text files using the *Import similarity matrix* option. During import, the similarity or distance values can be assigned to new or existing keys in the database.
- Trend data type data:** With the *Import trend data* option, files containing a set of measurements taken under a series of conditions can be imported and linked to new or existing database entries.
- Spectrum type data:** Spectrum type data can be imported from files and linked to new or existing entries using one of the spectrum import routines in the import tree (see 5.1.3).
- Whole genome maps data:** XML files generated by the OpGen[®] Argus[™] Optical Mapping System can be imported and linked to new or existing database entries (see 12.2.2).

- **Sequence read sets data:** With the *Import sequence read set files* option, a multitude of different file types can be imported. Depending on the file extension and the content of the file, the software will automatically detect which fill type is imported and how the import should be processed. Using this import functionality, sequence read sets can be imported from the following formatted files: Roche/454 (.fna, .qual), FASTA file (.fasta, .fna, .ffn, .faa, .frn or .txt), FASTQ files (.fq, .fastq or .txt).
- **Data exchange:** With the options listed under this topic, XML files containing entry, database field, and/or experiment type information can be imported and analyzed. See 3.4.3 for more information.



The entry information import options can be used in combination with any possible BioNumerics configuration. All other import options require the presence of the corresponding data type module (e.g. for the import of character type data, the Character data module needs to be present in the software configuration; for the import of sequence data, the Sequence data module needs to be present, etc.).

All recently used import options are grouped under **Recently used** in the *Import* dialog box. This group appears at the top of the tree. The last used import option is automatically selected when the *Import* dialog box is called.

The way the information should be imported in the database should be specified with an *import template*. An import template can be created and edited during import of the data. Import templates can also be managed by pressing the **<Manage import templates>** button. This calls the *Manage import templates* dialog box. More information about this dialog can be found in 3.3.5.4.

3.3.5.2 Importing entry information

3.3.5.2.1 Importing fields from a text file

With the *Import fields (text file)* option, listed under the topic *Entry information data* in the *Import* dialog box (see Figure 3.3.6), entry information can be imported from text files in the database and linked to new or existing database entries.

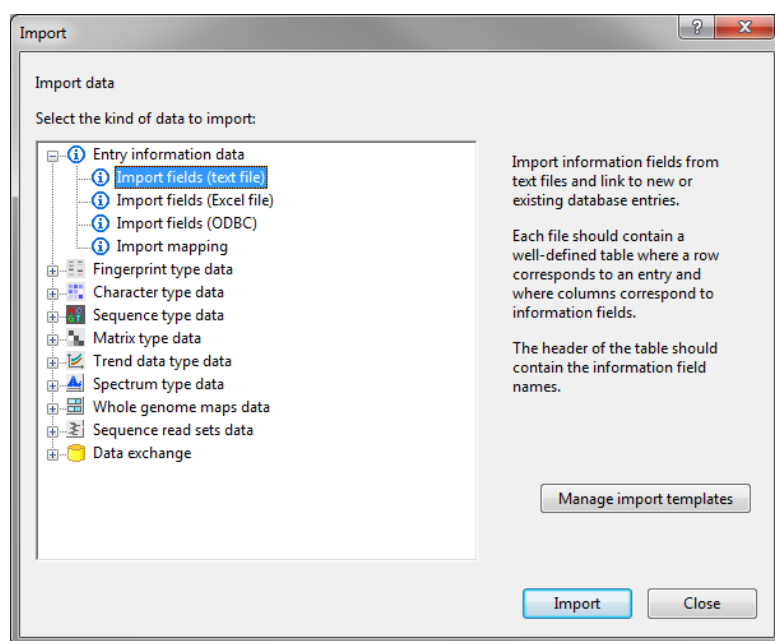
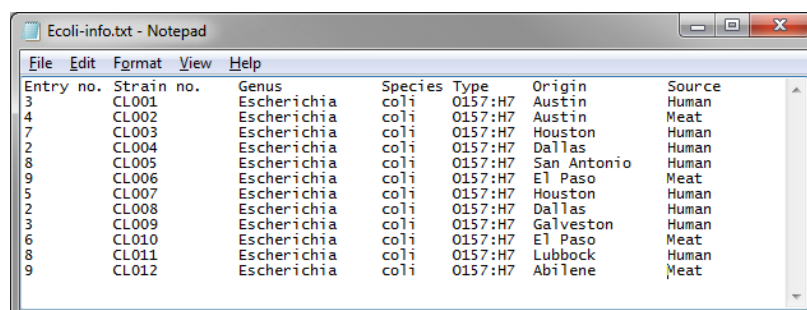


Figure 3.3.6: Import fields option.

Each file should contain a well-defined table with rows corresponding to entries and columns corresponding to entry information fields. The header of the table should contain the entry information field names (see Figure 3.3.7 for an example). There should be no extra rows or columns besides the table.



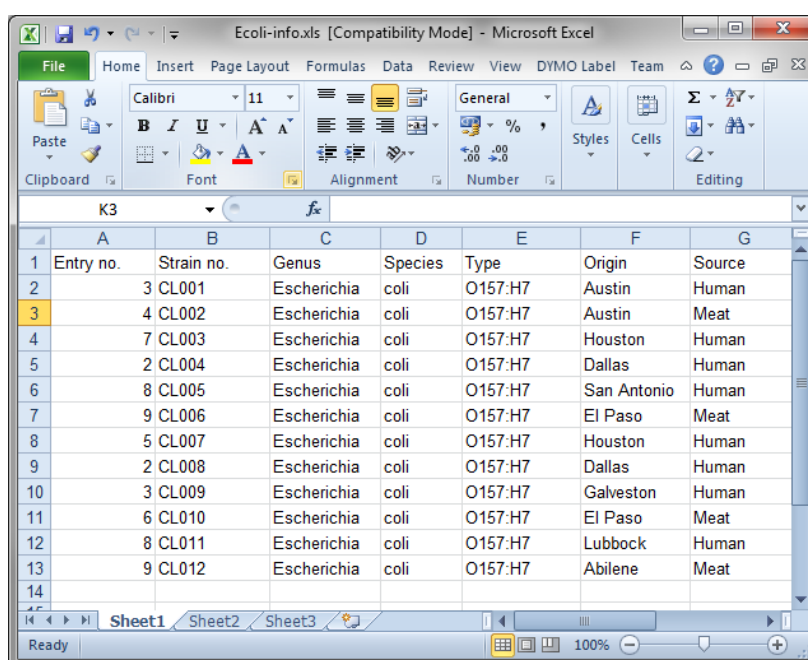
Entry no.	Strain no.	Genus	Species	Type	Origin	Source
3	CL001	Escherichia	coli	O157:H7	Austin	Human
4	CL002	Escherichia	coli	O157:H7	Austin	Meat
7	CL003	Escherichia	coli	O157:H7	Houston	Human
2	CL004	Escherichia	coli	O157:H7	Dallas	Human
8	CL005	Escherichia	coli	O157:H7	San Antonio	Human
9	CL006	Escherichia	coli	O157:H7	El Paso	Meat
5	CL007	Escherichia	coli	O157:H7	Houston	Human
3	CL008	Escherichia	coli	O157:H7	Dallas	Human
2	CL009	Escherichia	coli	O157:H7	Galveston	Human
6	CL010	Escherichia	coli	O157:H7	El Paso	Meat
8	CL011	Escherichia	coli	O157:H7	Lubbock	Human
9	CL012	Escherichia	coli	O157:H7	Abilene	Meat

Figure 3.3.7: Import fields from a text file.

3.3.5.2.2 Importing fields from an Excel file

With the **Import fields (Excel file)** option, listed under the topic **Entry information data** in the **Import** dialog box, entry information can be imported from Excel files in the database and linked to new or existing database entries (see Figure 3.3.6).

The Excel file should contain a well-defined table with rows corresponding to entries and columns corresponding to entry information fields. The header of the table should contain the entry information field names (see Figure 3.3.8 for an example). All information or a subset of information present in a particular sheet can be imported.



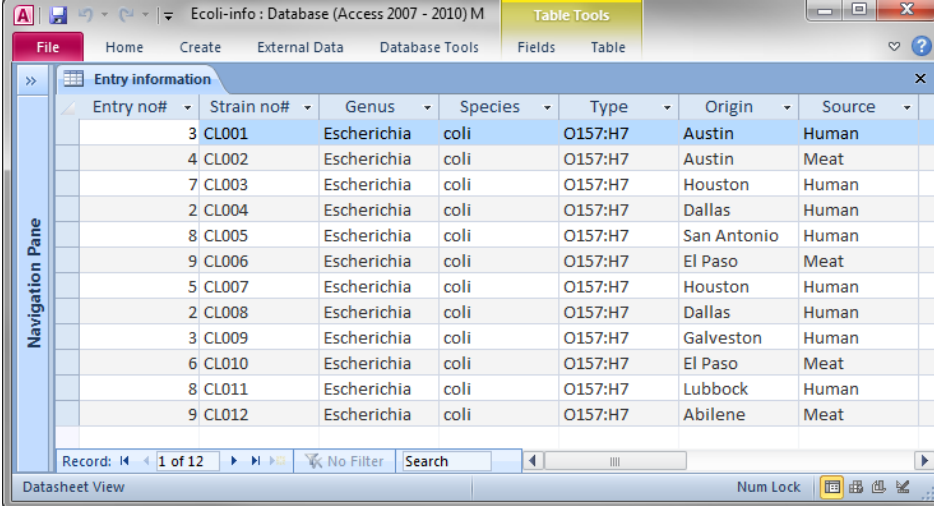
	A	B	C	D	E	F	G
1	Entry no.	Strain no.	Genus	Species	Type	Origin	Source
2		3 CL001	Escherichia	coli	O157:H7	Austin	Human
3		4 CL002	Escherichia	coli	O157:H7	Austin	Meat
4		7 CL003	Escherichia	coli	O157:H7	Houston	Human
5		2 CL004	Escherichia	coli	O157:H7	Dallas	Human
6		8 CL005	Escherichia	coli	O157:H7	San Antonio	Human
7		9 CL006	Escherichia	coli	O157:H7	El Paso	Meat
8		5 CL007	Escherichia	coli	O157:H7	Houston	Human
9		2 CL008	Escherichia	coli	O157:H7	Dallas	Human
10		3 CL009	Escherichia	coli	O157:H7	Galveston	Human
11		6 CL010	Escherichia	coli	O157:H7	El Paso	Meat
12		8 CL011	Escherichia	coli	O157:H7	Lubbock	Human
13		9 CL012	Escherichia	coli	O157:H7	Abilene	Meat
14							

Figure 3.3.8: Import fields from an Excel file.

3.3.5.2.3 Importing fields from an ODBC-compatible data source

With the **Import fields (ODBC)** option, listed under the topic **Entry information data** in the **Import** dialog box, entry information can be imported from ODBC-compatible files in the database and linked to new or existing database entries.

The file should contain a well-defined table with rows corresponding to entries and columns corresponding to entry information fields. The header of the table should contain the entry information field names (see Figure 3.3.9 for an example).

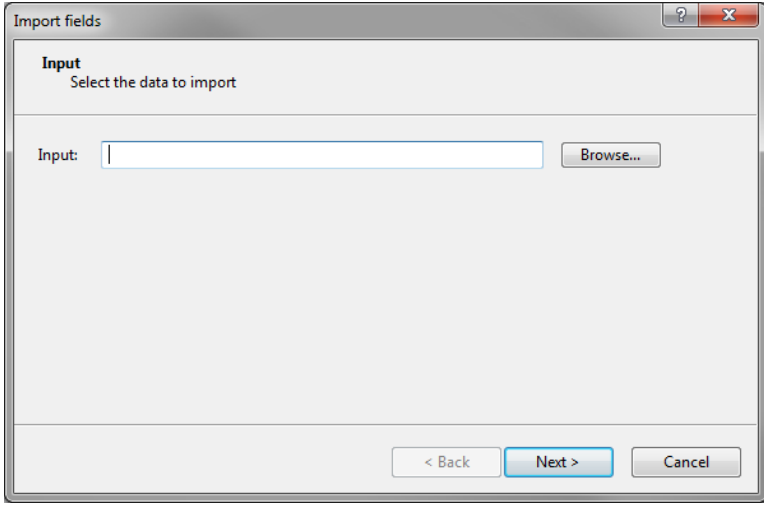


Entry no#	Strain no#	Genus	Species	Type	Origin	Source
3	CL001	Escherichia	coli	O157:H7	Austin	Human
4	CL002	Escherichia	coli	O157:H7	Austin	Meat
7	CL003	Escherichia	coli	O157:H7	Houston	Human
2	CL004	Escherichia	coli	O157:H7	Dallas	Human
8	CL005	Escherichia	coli	O157:H7	San Antonio	Human
9	CL006	Escherichia	coli	O157:H7	El Paso	Meat
5	CL007	Escherichia	coli	O157:H7	Houston	Human
2	CL008	Escherichia	coli	O157:H7	Dallas	Human
3	CL009	Escherichia	coli	O157:H7	Galveston	Human
6	CL010	Escherichia	coli	O157:H7	El Paso	Meat
8	CL011	Escherichia	coli	O157:H7	Lubbock	Human
9	CL012	Escherichia	coli	O157:H7	Abilene	Meat

Figure 3.3.9: Import fields (ODBC).

3.3.5.2.4 The import wizard

Selecting one of the **Import fields** options under **Entry information data** in the **Import** dialog box and pressing <**Import**> starts the import wizard.



Import fields

Input
Select the data to import

Input: Browse...

< Back Next > Cancel

Figure 3.3.10: The *Input* wizard page.

If the **Import fields (text file)** option was selected in the **Import** tree, the first step of the wizard prompts for the text file (see Figure 3.3.10). Pressing the <**Browse**> button allows you to select the file that you

want to import, located on your computer, external drive or on a network location. Three different text file separators are currently supported: "TAB", "Comma", and "Semicolon".

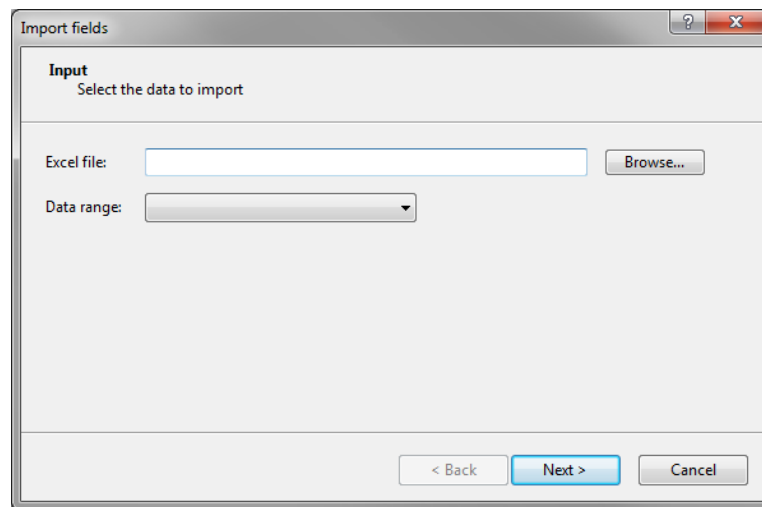


Figure 3.3.11: The *Input* wizard page.

If the *Import fields (Excel file)* option was selected in the Import tree, the first step of the wizard prompts for the Excel file and the table name (see Figure 3.3.11):

- Pressing the *<Browse>* button allows you to select the Excel file that you want to import, located on your computer, external drive or on a network location.
- All information present in a particular sheet can be imported by selecting the name of the sheet from the *Data range* drop down list. If a range of information has been saved in the Excel file and has been assigned a name (i.e. a so-called *named range*), the name of this selection can also be picked from the *Data range* list. If a named range is selected, the import action will only import the information that is present in the selection of the named range.

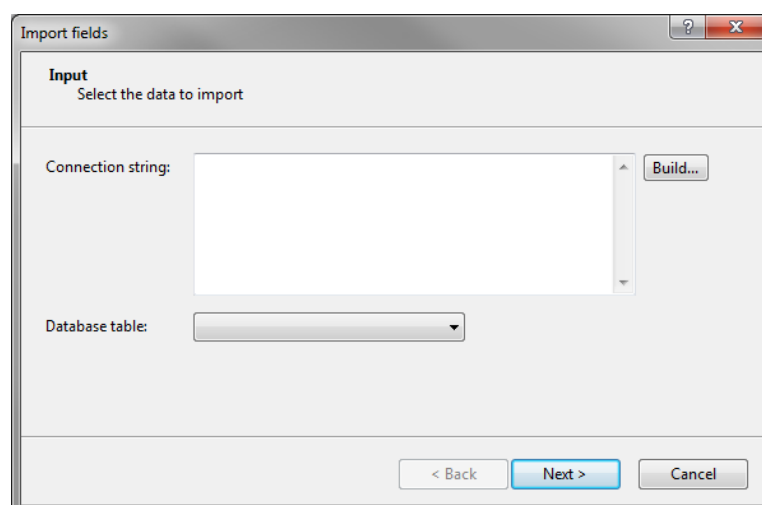


Figure 3.3.12: Import fields (ODBC): step 1.

If the *Import fields (ODBC)* option was selected in the Import tree, the first step of the wizard prompts for the ODBC connection string (i.e. the string that defines the database) and the table name (see Figure 3.3.12):

- Pressing the **<Build>** button allows you to create the ODBC connection string. The dialog box that pops up is generated by your Windows operating system and may differ depending on the Windows version installed. Select the correct data source from the list (e.g. **MS Access Database** to import information from an Access database). If the data source is not listed, create a new data source. Navigate to the correct path and select the "database" which can be located on your computer, external drive or on a network location. The ODBC string is updated in the **Connection string** input box.
- All information present in a particular **Database table** can be imported by selecting the table name from the drop down list.

Only when all settings have correctly been specified in the first step of the wizard, pressing **<Next>** will display the next step.

When importing entry data for the first time in the database, the *Import rules* dialog box will pop up (see Figure 3.3.14), otherwise the *Import template* wizard page (see Figure 3.3.13) will open.

If the import routine is unable to open the selected file, an error is generated.

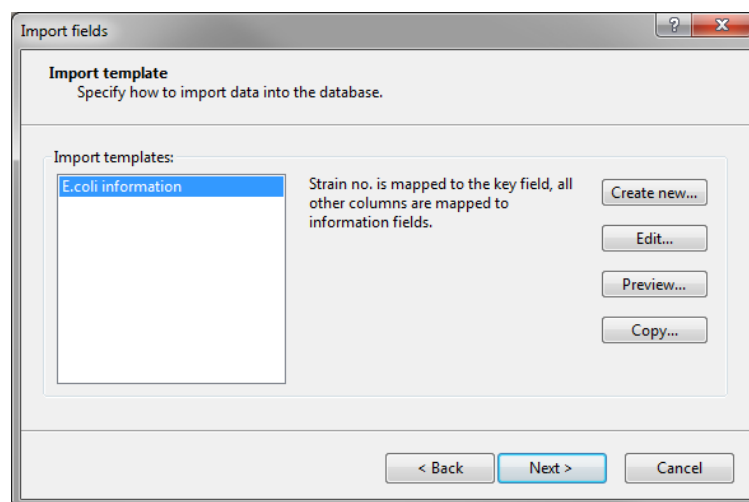


Figure 3.3.13: The *Import template* wizard page.

The way the information should be imported in the database can be specified with an import template. The *Import templates panel* lists all import fields templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up a new dialog box, allowing you to define a new import template (see Figure 3.3.14).

Each column in the selected file corresponds to a row in the grid (column 1 in the file corresponds to row 1 in the grid, column 2 corresponds to row 2, etc.). The text **File field** is specified in the **Source type** column and the column names are displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields. Initially the rows are not linked to any information in the database (the **Destination type** and **Destination** for all rows is set to **<None>**). Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

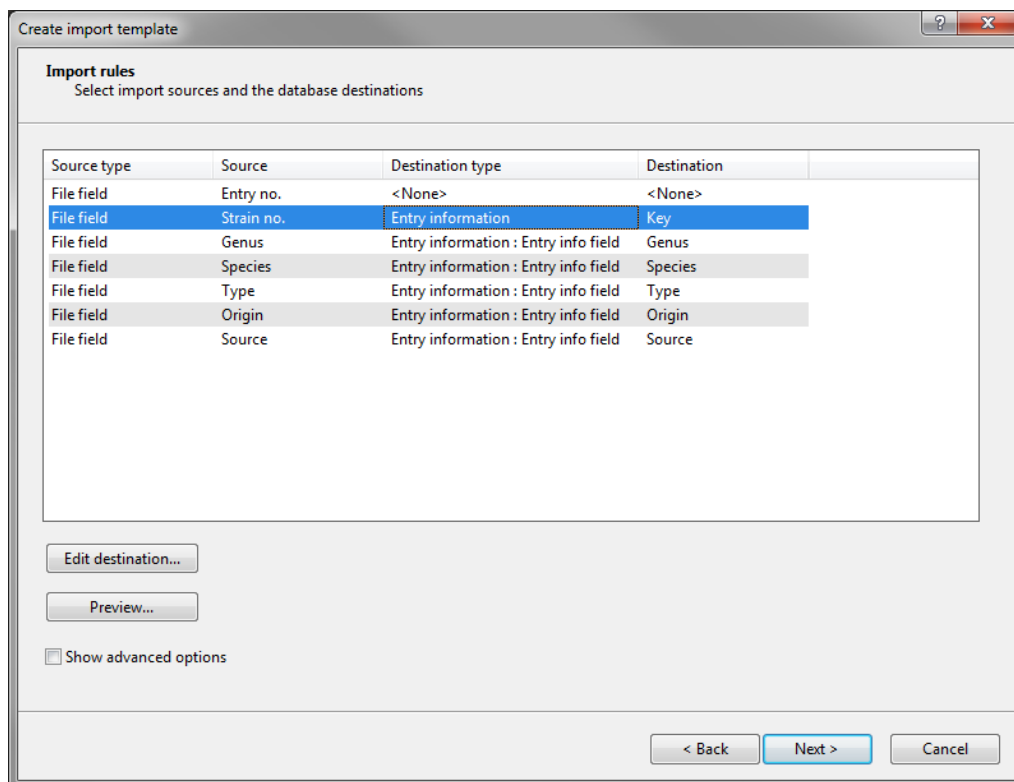


Figure 3.3.14: The *Import rules* dialog box.

When only one row is selected in the grid, the information of this row can be linked to (see Figure 3.3.15):

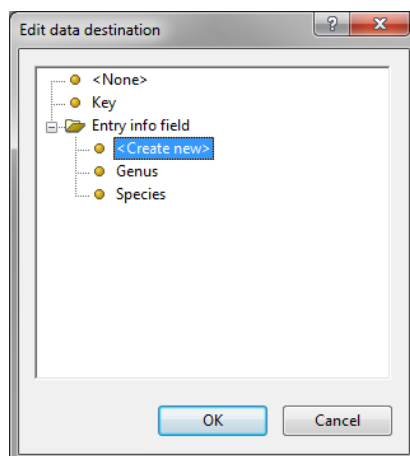


Figure 3.3.15: Edit data destination for a single selected row entry.

- The default information field **Key**.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).

If a row is linked to a new entry information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the entry information field name.

When multiple rows are selected in the grid, the information of these rows can be linked to non-default entry information fields in the database (select the **Entry info field** option, see Figure 3.3.16).

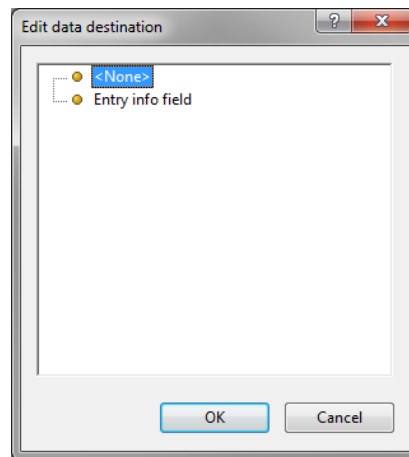


Figure 3.3.16: Edit data destination for multiple selected row entries.

When pressing the **<OK>** button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields in the database. If no fields exist with the same name, a new dialog box pops up prompting for the entry information field names.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. Rows in the grid that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **Show advanced options** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

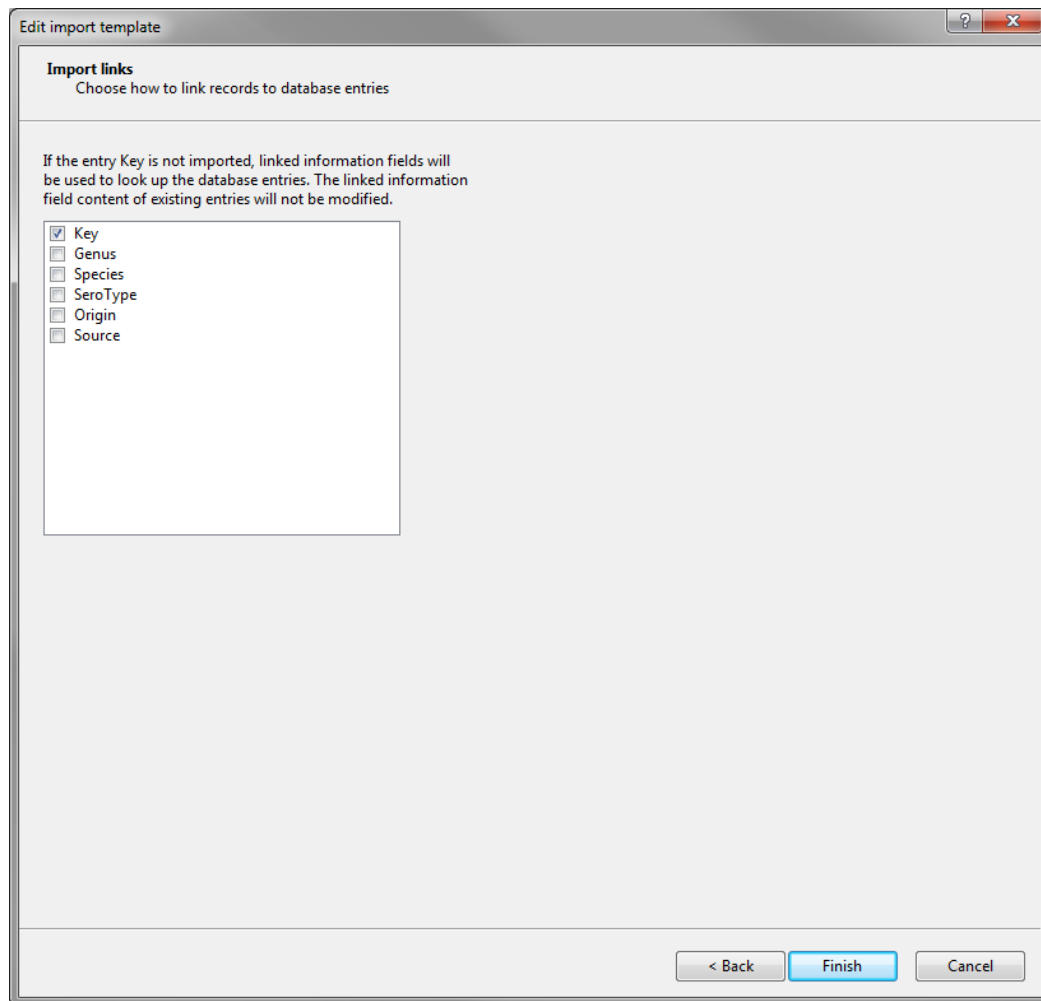


Figure 3.3.17: Specify the entry link field.

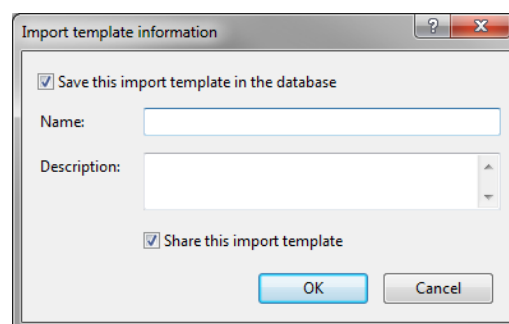


Figure 3.3.18: The *Import template information* dialog box.

The import template is saved to the database when the option *Save this import template in the database* is checked.

Check or uncheck the option *Share this import template* when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template *Name* is shown in the *Import templates panel* and is

automatically selected. The template **Description** is shown in the right panel.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

Pressing **<Next>** opens the last step of the wizard, prompting for some final settings.

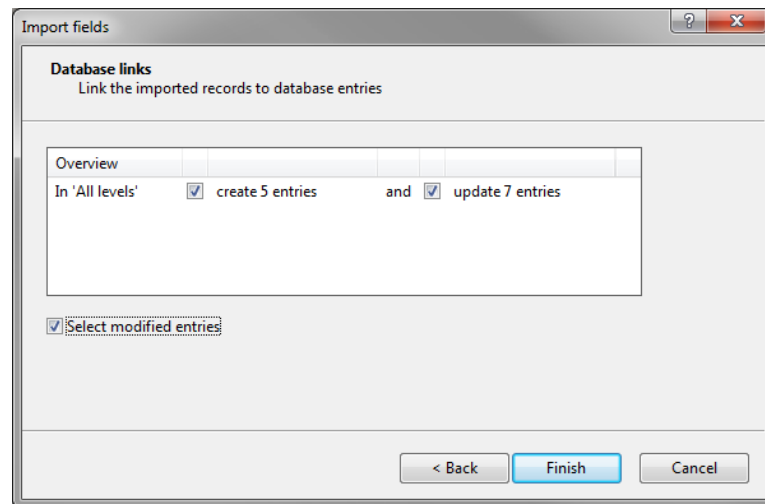


Figure 3.3.19: The *Database links* wizard page.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the entry information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing **<Next>** will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

3.3.5.3 Importing entry mapping information

With the **Import mapping** option, listed under the topic **Entry information data** in the *Import* dialog box (see Figure 3.3.20), mapping information stored in a text or Excel file can be imported in the database and linked to existing database entries.

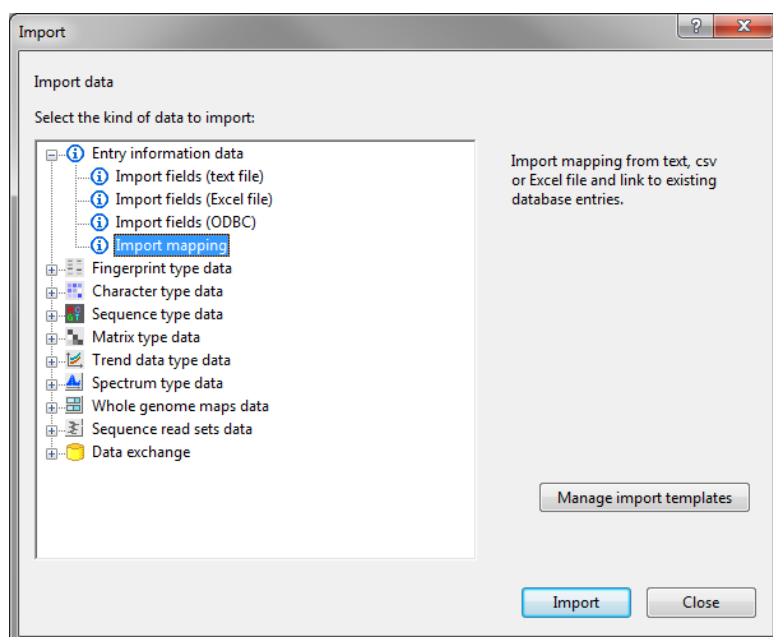


Figure 3.3.20: The *Import* dialog box: Import mapping option.

The data should be organized in two columns and the header of each column should contain the column names (see Figure 3.3.21 for an example).

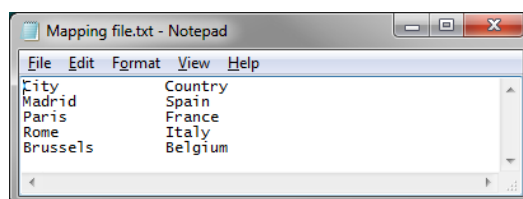


Figure 3.3.21: Import mapping example.

Selecting the **Import mapping** option under **Entry information data** in the *Import* dialog box and pressing <Import> starts the import wizard (see Figure 3.3.23).

The import routine will be applied on the selected entries in the *Main* window. If no selection is present, a warning message pops up, asking to confirm the selection of all entries in the *Main* window (see Figure 3.3.22).

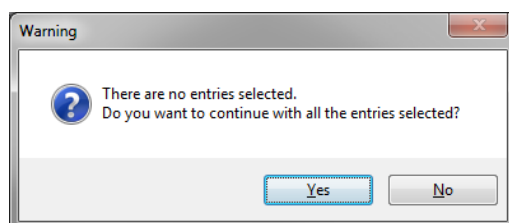


Figure 3.3.22: Warning message.

Pressing the <**Browse**> button allows you to select the file that you want to import, located on your computer, external drive or on a network location.

When importing information from an Excel file, all information present in a particular sheet can be imported by selecting the name of the sheet from the **Data range** drop down list. If a range of information has been

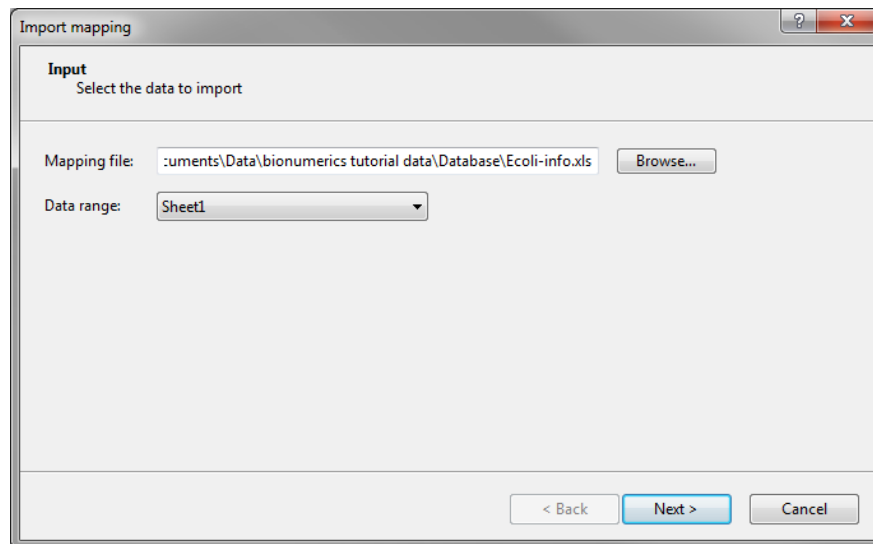


Figure 3.3.23: The *Mapping* dialog box.

saved in the Excel file and has been assigned a name (i.e. a so-called *named range*), the name of this selection can also be picked from the **Data range** list. If a named range is selected, the import action will only import the information that is present in the selection of the named range.

Only when all settings have correctly been specified in the first step of the wizard, pressing **<Next>** will display the next step (see Figure 3.3.24).

If the import routine is unable to open the selected file, an error is generated.

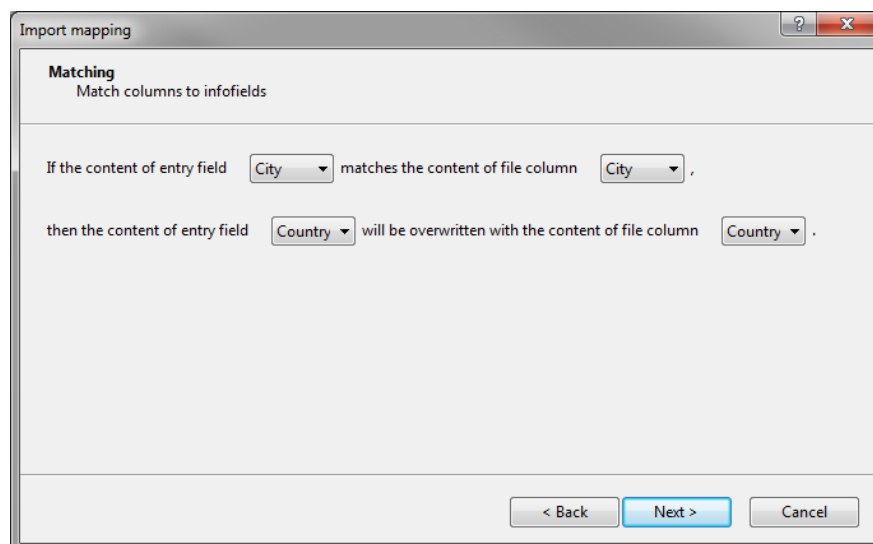


Figure 3.3.24: The *Match columns to infofields* dialog box.

In the *Match columns to infofields* dialog box the columns in the external file need to be matched to entry information fields in the database.

Pressing **<Next>** will display the third and final step of the import wizard (see Figure 3.3.25).

The *Overview* dialog box shows the effects of the defined mapping:

- The first column displays the matched information.

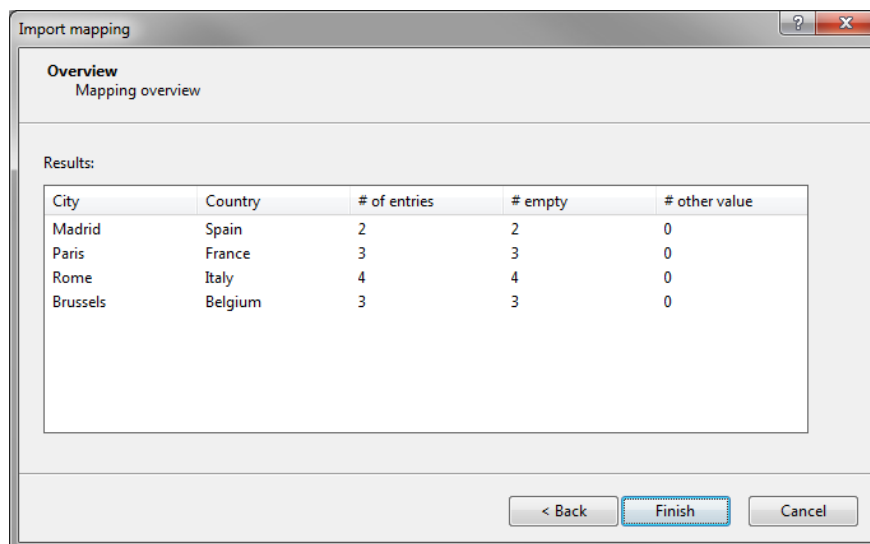


Figure 3.3.25: The *Overview* dialog box.

- The second column displays the content that will be added to the database for each match.
- The third column displays the number of entries found in the database with the matched information.
- The last two columns indicate if the information will be added to an empty or non-empty field. In the latter case the data will be overwritten.

Pressing **<Finish>** will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

3.3.5.4 Managing import templates

BioNumerics uses a unified concept of *import templates* to determine how external information should be imported into a BioNumerics database.

Import templates for all available import routines can be managed by pressing the **<Manage import templates>** button in the *Import* dialog box (see Figure 3.3.5). This calls the *Manage import templates* dialog box (see Figure 3.3.26).

The import templates are displayed in a tree-like structure, grouped by the type of import. An import template can be highlighted by clicking on it. The description of a highlighted template is displayed on the right-hand side of the dialog box.

To edit import template properties such as their name, description and whether or not the template is shared with other database users, highlight the import template and press **<Edit>** to pop up the *Import template information* dialog box.

Each import template has its own unique *Name*.

Optionally, a descriptive text string can be entered in the *Description* input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option *Save this import template in the database* is checked.

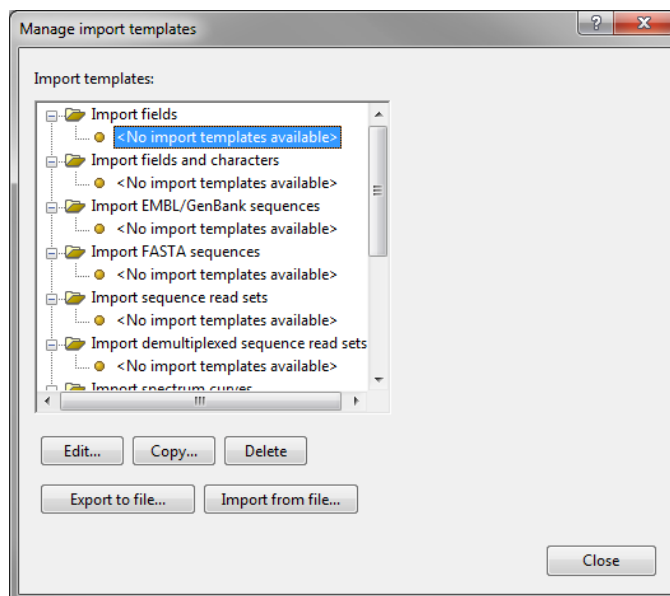


Figure 3.3.26: The *Manage import templates* dialog box.

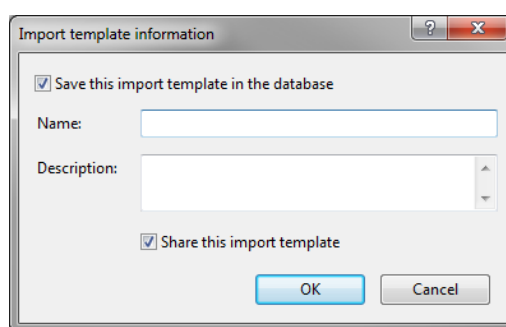


Figure 3.3.27: The *Import template information* dialog box.

Check or uncheck the option *Share this import template* when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

To remove an import template, highlight it and press **<Delete>**. The software will ask for confirmation before the template is deleted from the database.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

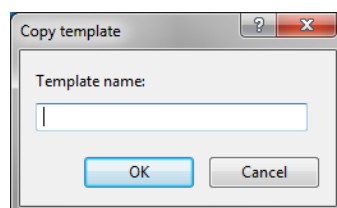


Figure 3.3.28: The *Copy template* dialog box.

A dialog box will prompt for a new template name. If the name entered corresponds to an already existing template name, the software will ask whether or not you want to overwrite the existing import template.

Import templates can be exported as XML files and imported again. This makes it possible to exchange import templates between databases. To export a highlighted template, press **<Export to file>**. This calls the *Export import template* dialog box.

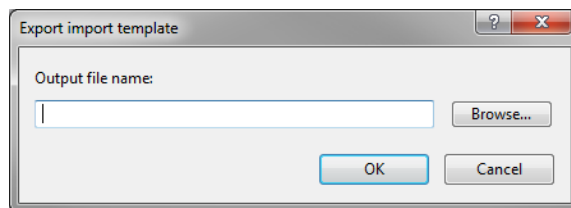


Figure 3.3.29: The *Export import template* dialog box.

You are prompted for and **Output file name**, which can be entered in the text box or browsed for with **<Browse>**.

To import an import template that has been saved as an XML file, press **<Import from file>**. This calls the *Import import template* dialog box.

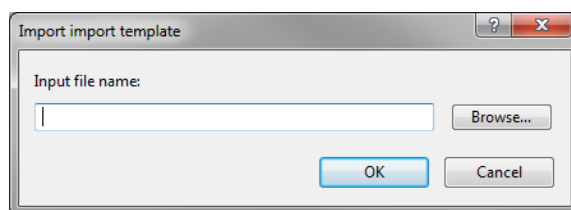


Figure 3.3.30: The *Import import template* dialog box.

Browse for the XML file (**<Browse>**). The *Import template information* dialog box will pop up prompting for the **Name** and a **Description**.

3.3.5.5 Advanced template options

3.3.5.5.1 Introduction

The way information should be imported in the database using the import routines described in 3.3.5.2, 6.1.3.2, 8.1.3.4, 8.1.3.5, and 8.1.3.6 needs to be specified with an import template. The template settings need to be defined in the *Import rules* dialog box (see Figure 3.3.14 for an example).

When the **<Show advanced options>** check box is enabled in the *Import rules* dialog box, three more columns appear inside the grid and eight extra buttons appear below the grid (see Figure 3.3.31 for an example). These advanced options are explained below.

3.3.5.5.2 Parse settings

Pressing the **<Edit parsing>** button pops up the *Edit data parsing* dialog box prompting for the parsing strategy for the selected row entry or entries in the grid (see Figure 3.3.32).

- When the **Parse component** option is checked, the mapped information is parsed using the parsing strategy as specified by the **Data parsing string**. This string should at least contain the [DATA] component, in which case the complete information is retained. The asterisk (*) can be used as a wildcard to omit characters from the information. In the parsing string ***_[DATA]-*** for example, any characters before the underscore and after the hyphen will be ignored.

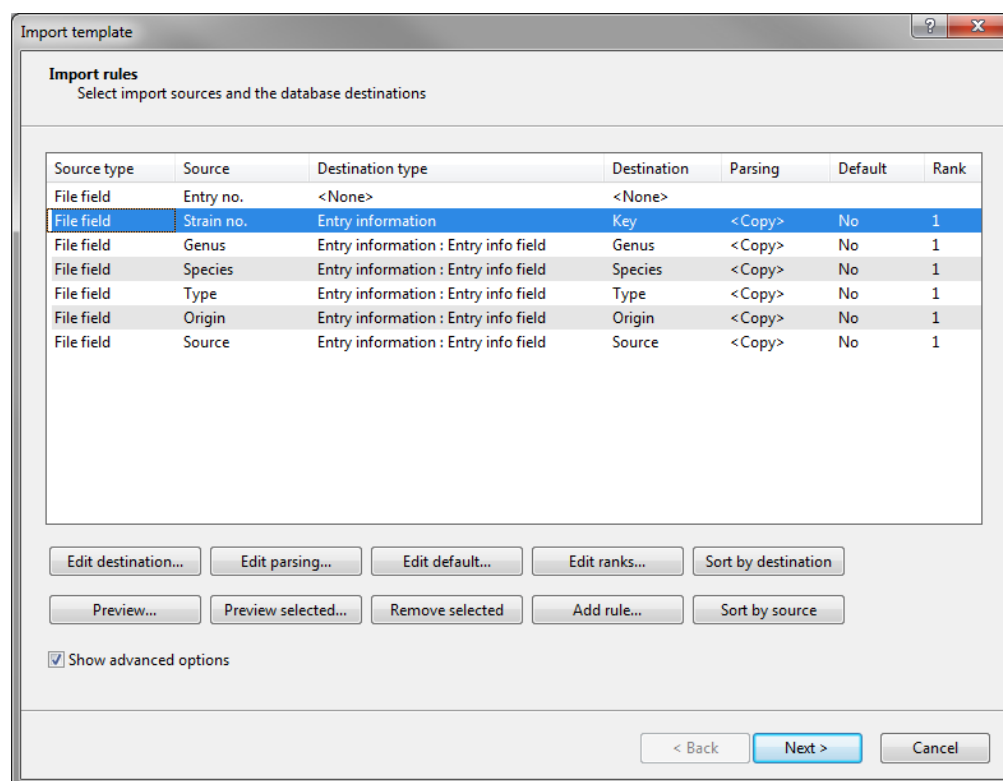


Figure 3.3.31: Advanced options in the *Import rules* dialog box, here depicted for the import of information fields.

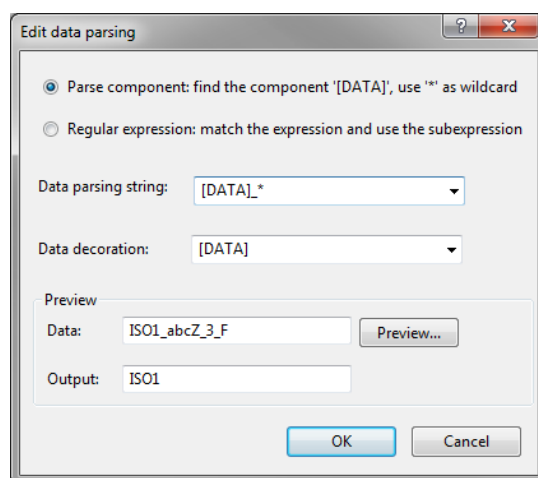


Figure 3.3.32: The *Edit data parsing* dialog box.

- When the **Regular expression** option is checked, the mapped information is parsed using the regular expression as specified in the **Data parsing string** input box. The default regular expression can be changed to any other valid regular expression using the correct syntax. The use and syntax of regular expression will not be covered in here. More detailed information about this topic can be found in the literature.
- Using a **Data decoration string**, information can be added to the parsed information. The **Data decoration string** recognizes the [DATA] tag and can be combined with plain text.

The mapped information is shown in the **Data** text box. Pressing the **<Preview>** button will display the

retained parsed information in the **Output** text box.

The default suggested parsing settings will not alter the mapped information (the text **<Copy>** is displayed in the **Parsing** column in the grid). If the parsing settings of a row are different from the default settings, the text **<Parse>** is displayed in the **Parsing** column.



The *Edit data parsing* dialog box can only be called when the selected row entry/entries in the grid is/are linked to a **Destination type**.

3.3.5.5.3 Default information settings

When during import the parsing fails, the information will not be imported in the database using the default settings (the information in the **Default** column is default set to **No**).

Pressing the **<Edit default>** button pops up the *Edit default value* dialog box allowing you to specify a default value for the selected row(s) in the grid (see Figure 3.3.33).

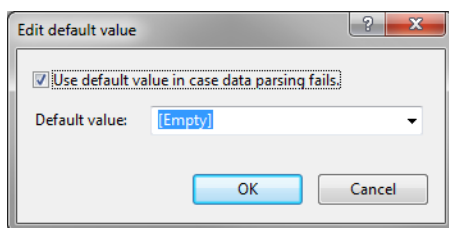


Figure 3.3.33: The *Edit default value* dialog box.

When the option **Use default value in case data parsing fails** is checked, a default value can be specified in the **Default value** input box. When parsing fails during import, the default value is imported in the database. The text **Yes** is shown in the **Default** column if rows are assigned a default value.



The *Edit default value* dialog box can only be called when the selected row entry/entries in the grid is/are linked to a **Destination type**.



Default values that were last entered are saved with the template and can be called again by pressing the arrow on the right hand side of the **Default value** edit box.

3.3.5.5.4 Rank settings

Only when the **Shown advanced options** is checked, multiple row entries in the grid can be linked to the same destination in the grid. The number of rows in the grid that are linked to the same destination is displayed in the **Rank** column. A row entry that is linked to a destination which is not used in any other mapping in the grid is assigned position **1** in the **Rank** column. When linking a row to a destination that is already the destination of another row in the grid, the row is assigned position **2** in the **Rank** column. The next row entry that is linked to the same destination is assigned position **3**, etc.

During import, the import tool checks if multiple rows are assigned to the same destination. If multiple rows are assigned to the same destination, the import plugin tool first tries to import the information of the row that is assigned to rank 1 for this destination. If the parsing fails and a default value has been specified for this row entry (see Figure 3.3.33), the default value is imported in the database. If the parsing fails and no default value has been specified for the row entry, the import plugin tool tries to import the information of the row that is assigned to rank 2 for this destination, etc.

The ranking of the rows that are assigned to the same destination can be changed by pressing the **<Edit ranks>** button.

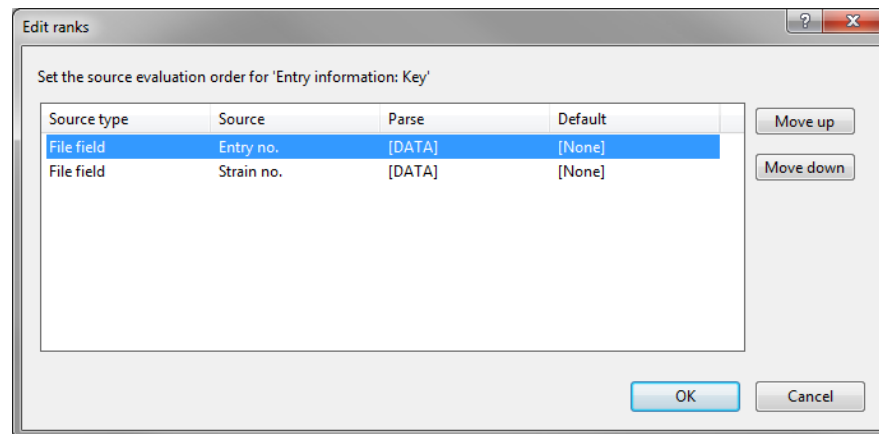


Figure 3.3.34: The *Edit ranks* dialog box.

The *Edit ranks* dialog box lists all row entries that are assigned to the same selected **Destination**. The row appearing on top of the list is assigned position 1 in the list, the second row entry in the list is assigned position 2, etc. With the **<Move up>** and **<Move down>** buttons, the order in the list can be changed. The information in the **Rank** column is updated when pressing the **<OK>** button.



In case only one row entry is selected in the grid, the *Edit ranks* dialog box can only be called when at least a second row entry in the grid is linked to the same **Destination**.



When multiple row entries are selected in the grid, the *Edit ranks* dialog box can only be called when all selected row entries in the grid are linked to the same **Destination**.

3.3.5.5.5 New rule

A new row can be added to the grid with the **<Add rule>** option.

The first step of the wizard prompts for the **Source** (see Figure 3.3.35 for an example).

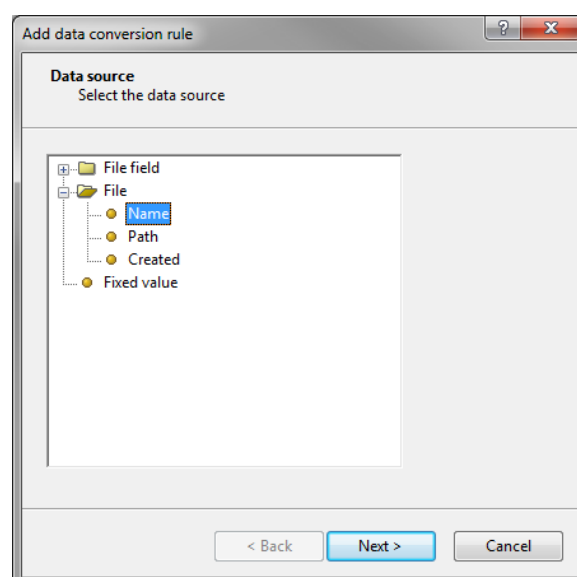


Figure 3.3.35: The *Data source* dialog box.

The choice is offered between one of these options:

1) *File field/Sequence header/FASTA field*:

- **File field**: The **File field** option is listed when running the import routines that are listed under the topics **Entry information data** and **Character type data** in the **Import** dialog box. All column names that are present in the selected file/database are listed under the topic **File field** in the tree. When a column name is selected from the tree, the information contained in this column can be linked to a destination in the database and optionally be parsed.
- **Sequence header**: The **Sequence header** option is listed when importing sequences in EMBL or GenBank format from the online repositories or from text files. All header tags are listed under the topic **Sequence header** in the tree. When a header tag is selected from the tree, the information contained in this tag can be linked to a destination in the database and optionally be parsed.
- **FASTA field**: The **FASTA** option is listed when importing FASTA sequences from text files in the database. All FASTA tags are listed under the topic **FASTA field** in the tree. When a FASTA tag is selected from the tree, the information contained in this tag can be linked to a destination in the database and optionally be parsed.

2) **File**: File information such as the name of the file (**Name**), the path of the file (**Path**) and the creation date of the file (**Created**) can be linked to a destination in the database and optionally be parsed.

3) **Fixed value**: A **Fixed value** can be linked to a destination in the database.

The second step of the wizard prompts for the **Destination** (see Figure 3.3.36 for an example).

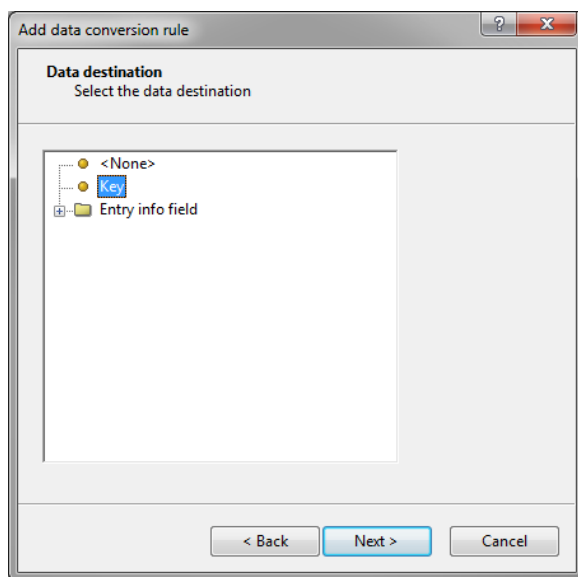


Figure 3.3.36: Select a data destination.

In the third step of the wizard, the parsing of the information needs to be specified (see Figure 3.3.37).

- When the **Parse component** option is checked, the mapped information is parsed using the parsing strategy as specified by the **Data parsing string**. This string should at least contain the [DATA] component, in which case the complete information is retained. The asterisk (*) can be used as a wildcard to omit characters from the information. In the parsing string ***_[DATA]-*** for example, any characters before the underscore and after the hyphen will be ignored.
- When the **Regular expression** option is checked, the mapped information is parsed using the regular expression as specified in the **Data parsing string** input box. The default regular expression can be

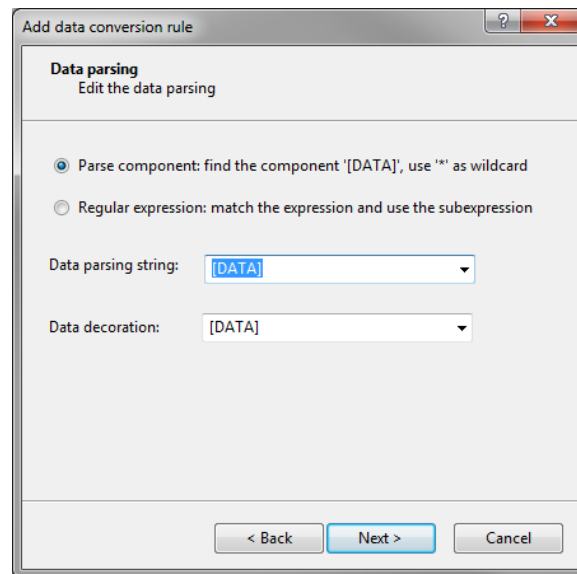


Figure 3.3.37: The *Data parsing* dialog box.

changed to any other valid regular expression using the correct syntax. The use and syntax of regular expression will not be covered in here. More detailed information about this topic can be found in the literature.

- Using a *Data decoration string*, information can be added to the parsed information. The *Data decoration string* recognizes the [DATA] tag and can be combined with plain text.

In the fourth and last step, a default value can be specified in case the parsing should fail (see Figure 3.3.38).

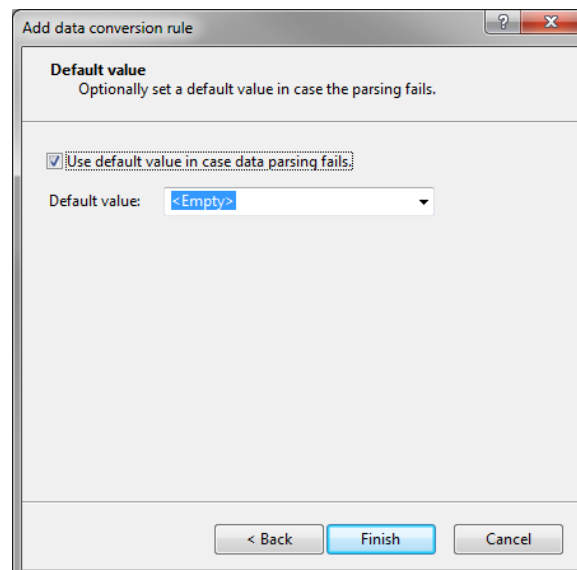


Figure 3.3.38: Edit the data parsing.

When the option *Use default value in case data parsing fails* is checked, a default value can be specified in the *Default value* input box. When parsing fails during import, the default value is imported in the database. The text *Yes* is shown in the *Default* column if rows are assigned a default value.

After having specified all settings in the wizard, the new row is added to the grid.

3.3.5.5.6 Sorting options

The **<Sort by destination>** button sorts the row entries based upon the sorting in the destination tree. All unlinked rows are listed at the bottom of the list.

The **<Sort by source>** option sorts the rows based upon the original sorting.

3.3.5.5.7 Preview options

Pressing **<Preview selected>** opens the *Preview* dialog box with the parsed information based upon the selected rules in the grid. The preview can be closed with the **<Close>** button.

Pressing the **<Preview>** button opens the *Preview* dialog box displaying the parsed information using all rules defined in the grid. The preview can be closed with the **<Close>** button.

3.3.5.5.8 Remove rules

Pressing the **<Remove selected>** button removes the selected row(s) in the grid.

3.3.6 Information field properties

In BioNumerics, properties can be assigned to any non-default entry information field. These properties can be used to ensure correct inputting, updating, formatting and sorting of information. They also offer a quick visual discrimination of *field states* using colors. Furthermore, if levels are defined in the database (see 3.3.10), the level assignment and the way that the information field is summarized over the database levels can be specified here. Finally, the "formula" by which calculated fields are derived from other fields can be changed here.

To set the properties of an entry information field, first display *Entry fields* panel by clicking on its tab in the *Main* window. Next, click on the field and select **Edit > Open highlighted object...** (🖱️, Enter). The *Database field properties* dialog box appears (see Figure 3.3.39).

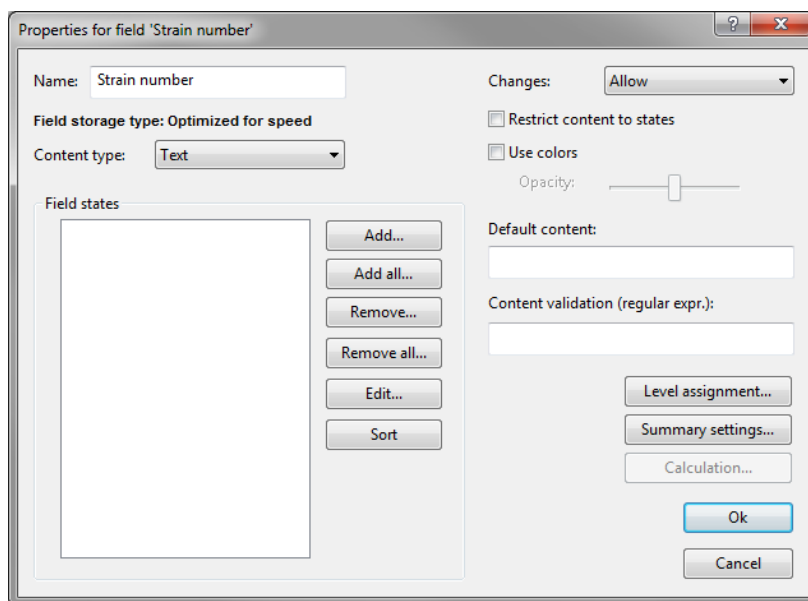


Figure 3.3.39: The *Database field properties* dialog box.

Each entry information field in BioNumerics has a display name that is shown in the user interface and an internal identifier or machine name (see 3.3.3). In the caption of the *Database field properties* dialog box, the ID is displayed to unequivocally indicate the information field for which the properties are being modified. The display name can be edited in the *Name* text field.

The *Field storage type* is indicated (either "Optimized for speed", "Optimized for space" or "Calculated"), but this property cannot be changed once a field has been created.

The *Content type* as set in the *Create new entry information field* dialog box can be modified if needed. Options are "Text", "Date" or "Number". Setting a *Content type* forces database users to provide the right content type, as no other content than the one specified by the *Content type* will be accepted for the information field. The *Content type* is also taken into account when displaying ("Text" and "Date" types are left-aligned, "Number" is right-aligned) and sorting information ("Text", "Date" and "Number" types are sorted alphabetically, by date or numerically, respectively).



The format supported by BioNumerics for the *Content type* "Date" is YYYY-MM-DD. Most commonly used data formats can be converted to this standard format with a special script, available on the Applied Maths website. Launch the script using *Scripts > Browse internet...* and selecting the option "Database related tools" > "Convert dates to standard format".

Making changes to the information contained in the field is allowed by default, but fields that are stored in the database (i.e. not calculated) can also be made read only. This feature is available in two "flavors":


- **Read only in UI:** Information cannot be updated via the user interface, but can be modified via BioNumerics scripts or plugins.
- **Read only:** Information in these fields will never be updated by BioNumerics. This feature can be used to display content that is automatically generated by the database management software (e.g. MS Access, SQL Server, Oracle or MySQL) or by scripts acting directly on the relational database.

Other information field properties are *Field states*, i.e. possible variants that can be contained within the information field. Individual states can each be displayed against a differently colored background, for an improved display in entry grid panels. *Field states* also provide an additional display option in e.g. a *Advanced cluster analysis* window (see 16) or *Coordinate space* window (see 17.4.2 and 17.4.3). Finally, when field states are defined, they become available as a drop-down list in entry grid panels, facilitating and harmonizing the input of data through direct field editing. See Figure 3.3.40 for an example.

Database entries		
Key	Genus	Species
G@Gel04@009	STANDARD	
G@Gel07@002	Ambiorix	sylvestris
G@Gel07@003	Ambiorix	aberrans
G@Gel07@004	Vercingetorix	palustris
G@Gel07@010	Vercingetorix	nemorosum
G@Gel07@011	Ambiorix	sylvestris

Figure 3.3.40: Detail of the *Database entries* panel, showing the drop-down list with field states for 'Genus'.



The field states drop-down list that appears after clicking the  button in the *Database entries* panel is different from the history date drop-down list in the *Entry* window (see 3.3.4). The latter automatically remembers the 10 last values entered for the field, while the former is not updated when a new state is entered by typing; the field states list needs to be updated via the *Database field properties* dialog box.

To add a new field state or to edit an existing one, press **<Add>** or **<Edit>**, respectively. In both cases, the *Variant properties* dialog box pops up (see Figure 3.3.41).

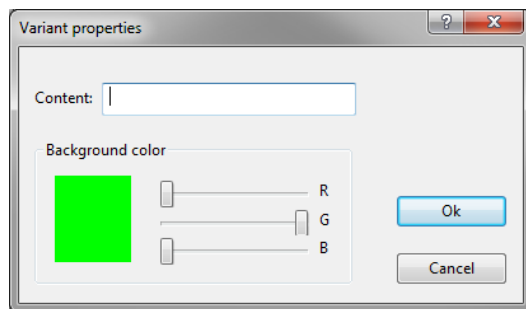


Figure 3.3.41: The *Variant properties* dialog box.

In this dialog, a possible variant (a *state*) for the information field can be entered or edited in the **Content** text box. The **Background color** can be set using the Red, Green and Blue sliders, only if *Use colors* was checked in the *Database field properties* dialog box.

If you want to remove a field state, select it from the list and press **<Remove>**. All field states can be removed at once by pressing **<Remove all>**. The program will ask for confirmation.

If the information field already contains information, a list of all existing **Field states** can be automatically created by pressing **<Add all>**.

Pressing **<Sort>** will sort the list of field states alphabetically.

Check *Use colors* to display a specific color code for each field state. Color codes will be automatically generated for the first 30 field states, but can be edited if desired using the **<Edit>** button. The **Opacity slider** sets the applied color intensity. The extent of the background coloring can also be modified (see 2.3.3).

Check **Restrict content to states** if the **Fields states** list contains all possible states for this information field. Turning on this feature forces database users to provide consistent information, as no other values than the ones specified in the **Fields states** list will be accepted for the information field.



If **Restrict content to states** is checked for an information field, the information cannot be edited by typing. Only the states available from a drop-down list can be selected as content.

A **Default content** can be specified. This default content will be filled in for new entries when they are created. Following special tokens are available: [CurrentDate], [CurrentDateTime], [CurrentYear], [CurrentMonth], [CurrentDay], [DbUser], [OSUser], [UniqueNr], [UniqueNrFx], [Guid], and [GuidShort]. For a detailed description of these tokens, see 3.7.2. Tokens can be combined with other tokens and/or with plain text.

In the **Content validation** text box, a regular expression can be entered to validate any user input. When the user enters information that does not pass the validation, an error message will be generated and the information will not be added to the database. See 21.2 for more information about regular expressions.

Pressing **<Level assignment...>** will display the *Level assignment* dialog box, as shown in Figure 3.3.42.

This dialog box is only relevant in case levels are defined in the BioNumerics database (see 3.3.10). If levels are present, following options are available for the information field or experiment at each level:

- **Not used:** The information field or experiment will not be available for the level, i.e. it will not be displayed and content cannot be edited. Use this option when the information is not relevant at the level specified.
- **Defined:** The information or experiment will be displayed and can be edited in this level.

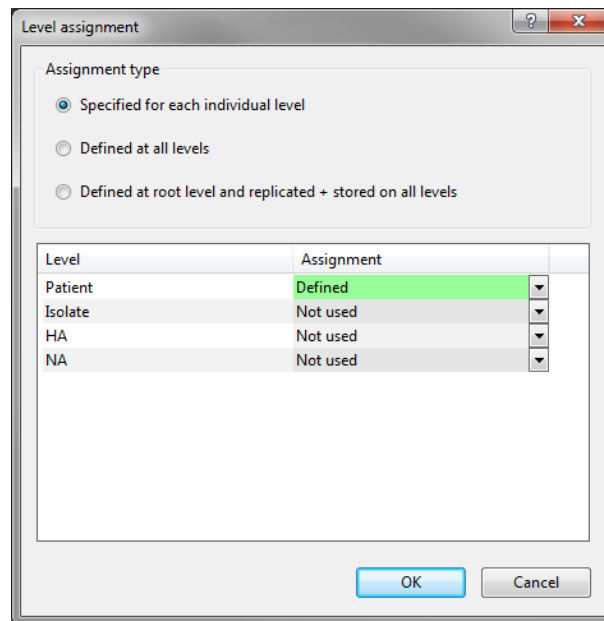


Figure 3.3.42: The *Level assignment* dialog box.

- **Replicated:** The information or experiment will be displayed read-only in this level. It is not stored in the database, but taken in real time from dependent entries at the level where the information field or experiment is defined. Since the replicated information is not stored in the database, no query-based views (see 3.2.2) can be defined.
- **Replicated and stored** (only available for entry information fields): Similar to the above option, information will be read-only and replicated from the level at which it is defined. However, the replicated information will be stored in the database to allow queries on this field.

When the option *Specified for each individual level* is checked, the grid in the lower part of the dialog box becomes active, where the type of assignment can be specified for each level in the database. The two other options for *Assignment type* correspond to specific usage scenarios and will render the grid inactive:

- **Defined at all levels:** The information will be displayed and can be edited individually at each level.
- **Defined at root level + replicated and stored on all levels** (only available for entry information fields): The information can only be edited at the root level (i.e. left in the *Database design* panel), but will be displayed at all levels. In addition, the information is stored in the database to enable query-based views.

Pressing <*Summary settings...*> will open the *Field summary method* dialog box (see Figure 3.3.43).

The summary settings are only relevant in case levels are defined in the BioNumerics database (see 3.3.10). They determine the summarization of information from higher to lower levels (i.e. from right to left in the *Database design* panel). Since several entries from a higher level can be dependent from the same entry in a lower level, the information can be summarized in different ways. Following summary methods are available:

- **Combine and count:** Displays all distinct states of the field for the dependent entries. Each state is followed by the number of times it occurs (in between brackets). The text in the *Separator* field separates each state.

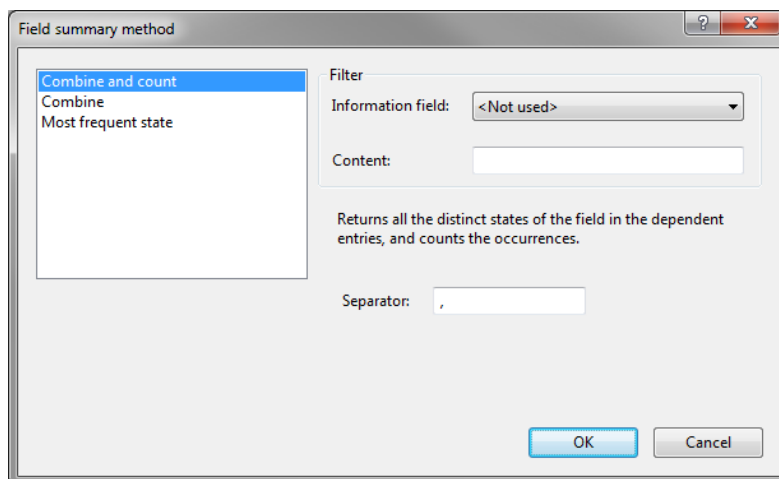


Figure 3.3.43: The *Field summary method* dialog box.

- **Combine:** Displays all distinct states of the field for the dependent entries in the order as they are encountered. The text in the **Separator** field separates each state.
- **Most frequent state:** Displays the most frequent state found in the dependent entries. This method has following options:
 - **Threshold (% of total):** The minimum occurrence of the state, expressed as a percentage of the total number of dependent entries.
 - **Threshold (absolute):** The minimum occurrence of the state, expressed as an absolute count.
 - **Alternative text:** The text that will be displayed in case not all of the above conditions are met.

Pressing **<Calculation...>** will open the *Calculated database field settings* dialog box (see Figure 3.3.44).

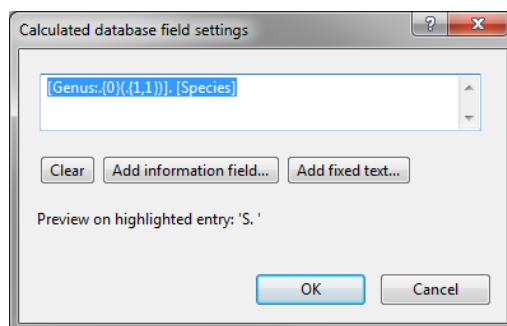


Figure 3.3.44: The *Calculated database field settings* dialog box.

The "formula" to derive a calculated field from other entry information fields can consist of tokens for (substrings of) existing fields and fixed text and is displayed in the text box in the upper part of the dialog box.

Pressing **<Add information field...>** will display the *Add information field* dialog box (see Figure 3.3.45).

Entry information fields can be picked from the list on the left. All entry fields are displayed by default, but if desired the list can be limited by selecting a view (see 3.2.2) from the drop-down list in the top left of the dialog box.

Once an information field is selected, the information herein can be further parsed, either by taking a substring or – more sophisticatedly – by using a regular expression.

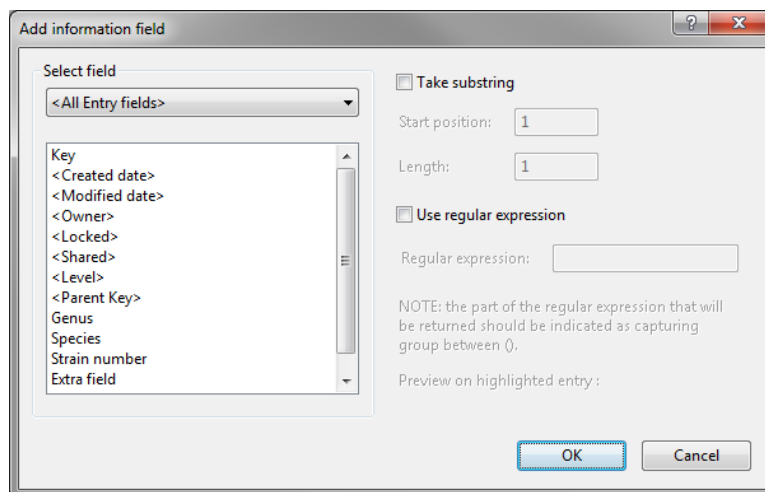


Figure 3.3.45: The *Add information field* dialog box.

- **Take substring:** This will simply take only a part of the information, starting at the *Start position* and with a specific *Length*.
- **Use regular expression:** The regular expression can be entered in the text box. The part that needs to be returned should be enclosed in round brackets. See 21.2 for more information about regular expression.

Pressing <OK> will add a token to the text box in the *Calculated database field settings* dialog box.

Press <Add fixed string...> to add static text to the formula.

In the *Add fixed text* dialog box, simply type the fixed text that should appear at the cursor position in the *Calculated database field settings* dialog box.

3.3.7 Configuring the database entries panel

Since the *Database entries* panel is an object grid panel, all display and customizing features discussed under 3.2 are valid for this panel as well. Some features that are particularly useful in the context of database layout will be discussed here in detail.

Entries in the database can be ordered by any of the information fields by clicking on the database field name and selecting **Edit > Information fields > Sort by field**.


Information present in non-default information fields with a content type *Number* or *Date* are sorted according to increasing numbers and dates respectively. Information present in the default information fields and in non-default information fields with a field type *String* (see 3.3.6) are ordered alphabetically.

Select **Edit > Information fields > Sort by field (reverse)** to sort the entries in the reverse order.

When two or more entries have identical *strings* in a field used to rearrange the order, the existing order of the entries is preserved. As such it is possible to categorize entries according to fields that contain information of different hierarchical rank, for example 'Genus' and 'Species'. In this case, first arrange the entries based upon the field with the lowest hierarchical rank, i.e. 'Species', and then upon the higher rank, i.e. 'Genus'.

In case "numbers" are combined numerically and alphabetically, for example entry numbers [213, 126c, 126a, 126c], you can first arrange the entries alphabetically using **Edit > Information fields > Sort by field** or **Edit > Information fields > Sort by field (reverse)** and then numerically using **Edit > Information**

fields > *Sort by field (Numerical)* or *Edit* > *Information fields* > *Sort by field (Numerical, reverse)*. The result will be [126a, 126b, 126c, 213].

The user can determine which information fields are displayed and the order in which they are shown by creating a view for the entry information fields as described in 3.2.2. Next, click on the column properties button  in the header of the *Database entries* panel and select *Set active fields*. The *Visible entry fields* dialog box will pop up (see Figure 3.3.46).

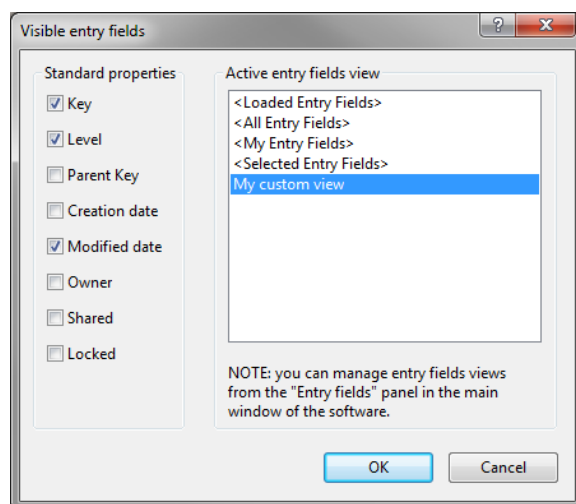





Figure 3.3.46: The *Visible entry fields* dialog box.

The display of default information fields or *Standard properties* in entry grid panels can be toggled on or off using the corresponding check boxes on the left hand side of the dialog box.


The *Active entry fields view* can be selected from the list on the right. This list contains all predefined views that were enabled in the *Manage user views* dialog box in addition to any user-defined view (see 3.2.2).

To quickly change the displayed information fields in entry grid panels without having to create a view, select the required information fields in the *Entry fields* panel and pick "<Selected entry fields>" from the drop-down list in the same panel.

To change the order of information fields in entry grid panels, highlight a field in the *Entry fields* panel and select *Edit* > *Move up* () or *Edit* > *Move down* (). Alternatively, in the *Database entries* panel click on the header of the field you want to move and then on the column properties button . Select *Move "Field Name" to left* or *Move "Field Name" to right*. Shortcut keys for these actions are **Ctrl+left arrow** and **Ctrl+right arrow**, respectively. The order of the entry fields will be saved with the view.



The default information fields have a fixed order. If you want to display this information in a different order, create a calculated field for each default information field you wish to display (see 3.3.3) and hide the default information fields. Calculated fields can be rearranged as described above.

It is possible to freeze one or more information fields, so that they always remain visible left from the scrollable area. For example, if you want to freeze the 'Key' field, select the field right from the 'Key' in the field header, and select *Freeze left pane* from the column properties button . This feature, combined with the possibility to change the order of information fields makes it possible to freeze any subset of fields.

The width of the *Database entries* panel as a whole can be changed by dragging the separator lines between the *Database entries* panel, the *Experiment presence* panel and the remaining panels to the left or to the right. All panel sizes and positions are stored when you exit the software and are specific for each database.

When a new comparison is created or when an existing comparison is opened (see 13.2), the same layout as applied to the *Database entries* panel of the *Main* window (which fields to display/hide, column width and

order of information fields) is used for the *Information fields* panel of the *Comparison* window.


3.3.8 Selections of database entries

Selections in BioNumerics provide a tool to perform an action on an arbitrarily defined subset of database entries, instead of on the database as a whole. As such, selections form the basis for the creation of comparisons (see 13), enabling a host of analysis tools to be applied to the selected entries. Selections can be cut, copied, pasted or deleted and furthermore allow the user to choose entries to be identified (see 15.5), run decision networks (see 15.6) on specific entries, include entries in an alignment (see 8.4), chromosome comparison (see 8.6 and 8.7) or genome annotation project (see 8.8) and to share well-defined information with other users via the creation of bundles (see 3.4.2) or XML files (see 3.4.3).

Entries are selected in the same way as any other object in the database (see 3.2.4), i.e. by clicking their check box (☐) , by holding the **Ctrl**-key and clicking anywhere on the entry, by pressing the **space bar** or by using the **Shift**-key for selecting a range of entries. Check boxes for selected entries are indicated as ☒ . For a complete overview of the manual selection tools, see 3.2.4.

A single entry can also be selected or unselected from its *Entry* window (see 3.3.4) using **Edit > Select / unselect this entry** (☒). When the entry is selected, the corresponding button shows as ☒ .

Selected entries can be brought to the top of the list in the *Database entries* panel with **Database > Entries > Move selection to top** (**Ctrl+T**).

To clear any selection of database entries, use **Database > Entries > Unselect all entries (all levels)** (, **F4**). The keyboard shortcut for this command (**F4**) can be used in any BioNumerics window.



A very convenient command in combination with the manual selection functions is **Edit > Information fields > Sort by field**, which allows the database to be sorted according to the selected information field (see 3.3.7 for a detailed description).

Complementary to the manual selection tools, entries can also be searched and selected automatically like any other database object (see 3.2.10 and 3.2.11). Specifically for searching entries based on entry information, attachments and experimental information an advanced query tool is available (see 3.3.9).

It is possible to search for all entries that contain a certain experiment:

- **Database > Entries > Select entries with experiment**: will select all entries that have an experiment for the highlighted experiment type in the *Experiment types* panel, hereby replacing the current selection (if any).
- **Database > Entries > Select entries with experiment (add)**: will add all entries that have an experiment for the highlighted experiment type in the *Experiment types* panel to the current selection.
- **Database > Entries > Select entries with experiment (within selection)**: will search for entries that have an experiment for the highlighted experiment type in the *Experiment types* panel within the current selection.

These commands are also available from the floating menu in the *Experiment presence* panel.

To make it easier to see which entries are selected in a large database, only the selected entries can be displayed with **Edit > Views > Switch to Selected Objects view** (**Ctrl+Shift+S**).

3.3.9 Entry searches based on information and experiment data

3.3.9.1 Introduction

BioNumerics contains an *advanced query tool* that allows searches of any complexity to be made within the database based on information fields, attachments and experiment data. The *Find entries (advanced)* window is called by selecting **Database** > **Entries** > **Find entries (advanced)...** (**Alt+F**) (see Figure 3.3.47).

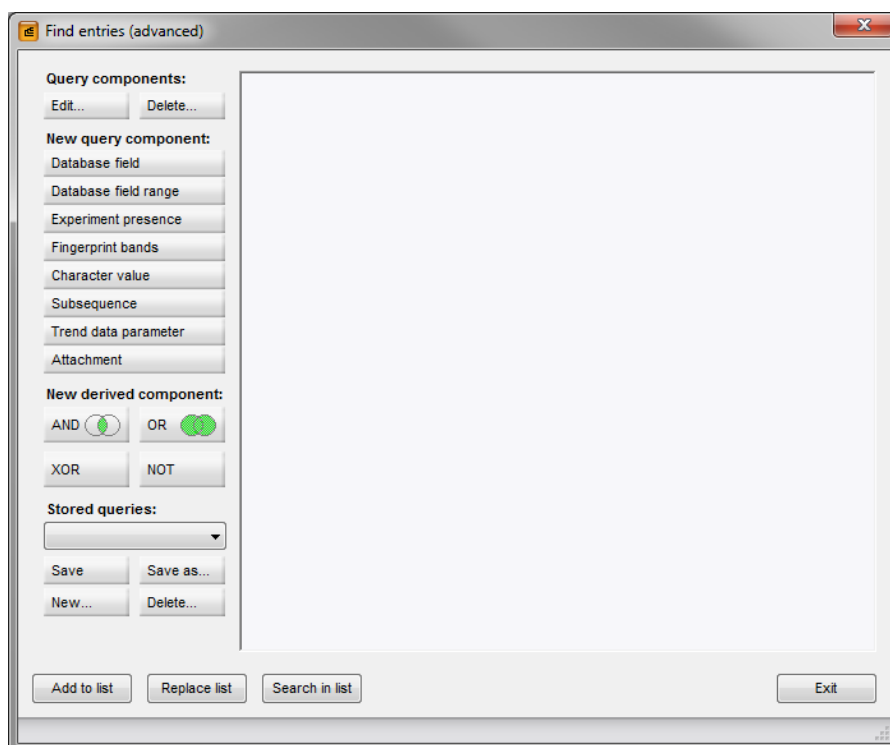


Figure 3.3.47: The *Find entries (advanced)* window.

The advanced query tool allows you to create individual *query components*, which can be combined with *logical operators*. The available targets for query components are **Database field**, **Database field range**, **Experiment presence**, **Fingerprint bands**, **Character value**, **Subsequence**, **Trend data parameter**, and **Attachment**.

3.3.9.2 Database field

Pressing the <**Database field**> button calls the *Database field search* dialog box (see Figure 3.3.48).

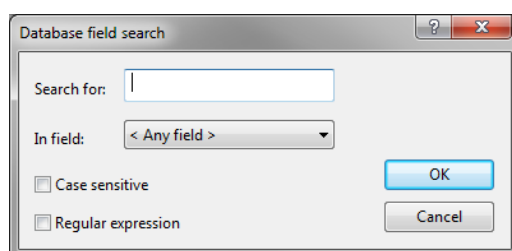


Figure 3.3.48: The *Database field search* dialog box.

In the **Search for** text field, you can enter a (sub)string to find in any database field ("<Any field>") or in any specific field that exists in the database by selecting it from the drop-down list.

The search component can be specified to be **Case sensitive** or not. In addition, a search string can be entered as a **Regular expression** (see 21.2).

3.3.9.3 Database field range

Pressing the <**Database field range**> button calls the *Database field range* dialog box (see Figure 3.3.49).

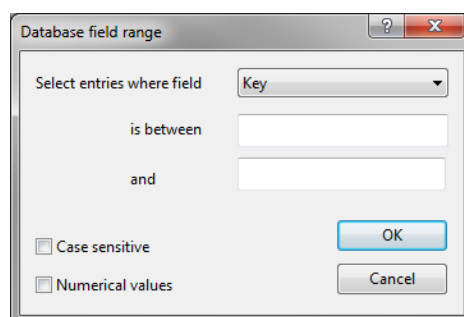


Figure 3.3.49: The *Database field range* dialog box.

Using this component button, you can search for database field data within a specific range, which can be alphabetical or numerical. Specify a database field and enter the start and the end of the range in the respective input boxes. A range should be specified with the lower string or value first. Note that, when only one of both limits is entered, the program will accept all strings above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit of the range is entered and the upper limit is left blank, all strings (values) *above* the specified string (value) will be accepted.

The search component can be specified to be **Case sensitive** or not. When **Numerical values** is checked, the search component will look only for numerical values and ignore any other characters.

3.3.9.4 Experiment presence

Pressing the <**Experiment presence**> button calls the *Experiment presence* dialog box (see Figure 3.3.50).

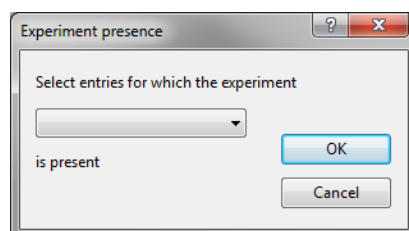


Figure 3.3.50: The *Experiment presence* dialog box.

From the drop-down list, you can specify an experiment to be present in order for entries to be selected.

3.3.9.5 Fingerprint bands

Pressing the <**Fingerprint band presence**> button calls the *Fingerprint band presence* dialog box (see Figure 3.3.51).

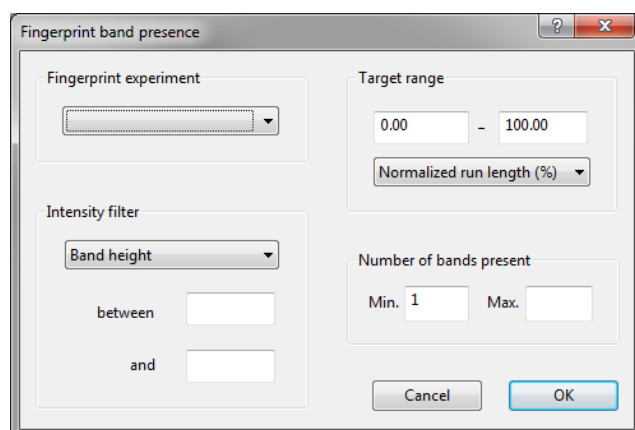


Figure 3.3.51: The *Fingerprint band presence* dialog box.

This search component allows specific combinations of bands to be found in the database entries. The drop-down list allows you to select a ***Fingerprint experiment***, present in the database. Furthermore, an ***Intensity filter***, a ***Target range***, and the ***Number of bands present*** can be specified.

Under ***Intensity filter***, you can choose which intensity parameter to be used: ***Band height***, ***Band surface*** or ***Relative band surface***. When a 2D quantification analysis is done, you can also choose ***Volume***, ***Relative volume*** or ***Concentration***. A range should always be specified with the lower value first. Note that, when only one of both limits is entered, the program will consider all bands above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit is entered and the upper limit is left blank, all bands above the specified intensity will be accepted. When both fields are left blank, no intensity range will be looked for, i.e. all bands will be considered.

Under ***Target range***, you can search for bands with specific sizes, either entered as ***Normalized run length (%)*** or as ***Metric values***. A target range should always be entered with the lower value first. Note that, when only one of both limits is entered, the program will consider all bands above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit is entered and the upper limit is left blank, all bands above the specified size will be accepted. When both fields are left blank, no size range will be looked for, i.e. all bands will be considered.

Under ***Number of bands*** present, you can enter a minimum and a maximum number of bands the patterns should contain. Note that, when only one of both limits is entered, the program will consider all patterns with band numbers above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit is entered and the upper limit is left blank, all patterns having *at least* the specified number of bands will be accepted. At least one of both limits must be entered.

3.3.9.6 Character value

Pressing the **<Character value>** button calls the *Character value* dialog box (see Figure 3.3.52).

With the character value component, you can search for characters within certain ranges. You should select a character type from the ***Experiment*** drop-down list, specify a ***Character*** or select "<All>" characters, and enter a minimum and maximum value. A range should always be specified with the lower value first. Note that, when only one of both limits is entered, the program will consider all characters above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit is entered and the upper limit is left blank, all characters with values above the specified value will be accepted.

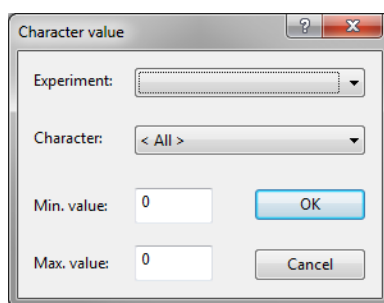


Figure 3.3.52: The *Character value* dialog box.

3.3.9.7 Subsequence

Pressing the *<Subsequence>* button calls the *Subsequence* dialog box (see Figure 3.3.53).

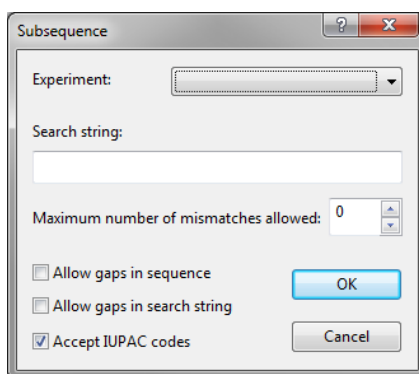


Figure 3.3.53: The *Subsequence* dialog box.

With the Subsequence component you can perform a search for a specific subsequence in a sequence type experiment. The sequence type should be chosen from the *Experiment* drop-down list and a subsequence entered. A mismatch tolerance can be specified with *Maximum number of mismatches allowed*. The program can also search for sequences that have one or more gaps as compared to the search sequence, with the option *Allow gaps in sequence*. Similarly, the program can also find subsequences that match the search string with one or more gaps introduced with *Allow gaps in search string*. The gaps are counted with the mismatches, and the total number of mismatches and gaps together is defined by the parameter *Maximum number of mismatches allowed*. Unknown or partially unknown positions can also be entered according to the IUPAC code, when *Accept IUPAC codes* is enabled.

3.3.9.8 Trend data parameter

Pressing the *<Trend data parameter>* button calls the *Trend curve parameter* dialog box (see Figure 3.3.54).

With the Trend data parameter component, you can search for trend data parameters within a specific range. You need to specify a trend data type from the *Experiment* drop-down list, a curve that belongs to it and a parameter defined for this experiment. Enter the start and the end of the range in the *Min. value* and *Max. value* text fields, respectively. A range should be specified with the lower value first. Note that, when only one of both limits is entered, the program will accept all values above or below that limit, depending on which limit was entered. For example, when only the first (lower) limit of the range is entered and the upper limit is left blank, all values *above* the specified value will be accepted.

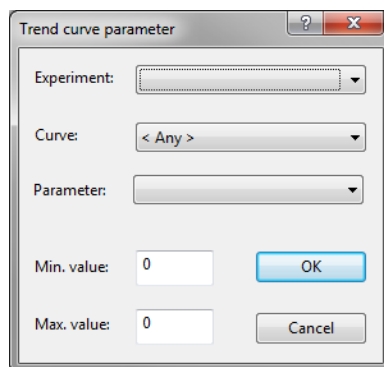


Figure 3.3.54: The *Trend curve parameter* dialog box.

3.3.9.9 Attachment

Pressing the **<Attachment>** button calls the *Attachment search* dialog box (see Figure 3.3.55).

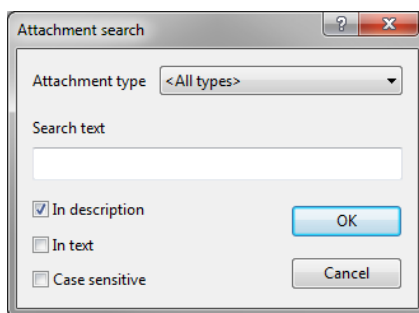






Figure 3.3.55: The *Attachment search* dialog box.

With the Attachment component, one can perform a search in attachments (see 3.2.13) that are linked to database entries. With the **Attachment type** drop-down list you can choose the type of attachments to search in. One of the possibilities is "<All>", i.e. to search within all attachment types. For all types of attachments it is possible to search in the **Description field**, and for text type attachments, it is also possible to search within the **Text**. The **Text** option does not apply to document attachment types.

3.3.9.10 Logical operators

Individual query components in the *Find entries (advanced)* window can be selected using **Ctrl+Click** (or by dragging the mouse over the components) and combined with logical operators. The buttons for the logical operators contain a helpful Venn diagram icon that clearly explains the function of the operator:

- **NOT**  **NOT**, operates on one component. When a component is combined with NOT, the condition of the component will be inverted.
- **AND**  **AND**, combines two or more components. *All* conditions of the combined components should be fulfilled at the same time for an entry to be selected.
- **OR**  **OR**, combines two or more components. The condition implied by *at least one* of the combined components should be fulfilled for an entry to be selected.
- **XOR**  **XOR**, combines two or more components. Exactly one condition from the combined components should be fulfilled for an entry to be selected.

In order to speed up the search function in case of large databases, it is important to know that searching through the database fields is extremely quick, while searching through sequences or large character sets can be much slower. Using the AND operator, it is always recommended to define the quickest search component as the first, since the searching algorithm will first screen this first component and subsequently screen for the second component on the subset that match the first component. When combined with a logical operator, query components contain a small node at the place where they are connected to the logical operator box (AND, OR, XOR). By dragging this node up or down, you can switch the order of the query components, thus making it possible to move the most efficient component to the top in AND combinations, as explained above.

Individual components can be re-edited at any time by double-clicking on the component or by selecting them and pressing **<Edit>**. Selected components can be deleted with **<Delete>**.

Queries can be saved with **<Save>** or **<Save as>**. Saved queries can be loaded using the drop-down list under **Stored queries**. Existing queries can be removed by loading them first and pressing **<Delete>**.



The Decision Network functionality in BioNumerics (see 15.6) provides an alternative to the advanced query tool: any selection that can be made in the advanced query tool can also be created using Decision Networks.

3.3.10 Levels and dependencies

3.3.10.1 Introduction

In a default BioNumerics database, all entries belong to the same level or category. Some applications, however, require a more advanced database structure, where entries belong to different hierarchies or levels. BioNumerics offers this possibility by introducing *levels*. Levels are hierarchical layers in the database, with the purpose of storing and representing entries of different categories in a better organized way.

The meaning and utility of levels can probably best be explained with the following example: In a clinical lab, samples are regularly obtained from patients (e.g. blood, skin, ...). From these samples, fingerprints (profiles) are generated using MALDI as technique. In this context, there are three levels of entries to which information fields and data can be assigned: the *patients*, the *samples*, and the *profiles*. In a flat BioNumerics database setup, one would create a new entry for each profile generated and enter patient and sample-specific information in dedicated information fields. For example, the patient could be described in a set of fields "Patient name", "Patient age", "Patient gender", etc.. The same can be done for sample-specific information. It is clear that, for a number of reasons, this is not the most elegant approach for building a leveled database.

- Patient information is unnecessarily duplicated over all samples/profiles;
- If patient information is to be added or changed, it has to be added for all samples/profiles;
- There is no formal way of linking profiles to patients/samples, except by filling in an information field. In case of a typing error, the link is lost.
- There is no framework to deal with duplicate runs (e.g. averaging, standard deviations).

In addition to the concept of *levels*, BioNumerics also introduces the concept of *dependencies*. A dependency is the relation between an entry on a parent level and child level. This relation is always of the type one-to-many: one parent entry can have several dependent child entries, but a child entry can only have a single parent entry. Using the same example of patients, samples and profiles, the interaction between levels and dependencies is illustrated in Figure 3.3.56. The database consists of 3 levels, *Patients*, *Samples* and *Profiles*. Each level has specific information fields associated, e.g. for Patients: "Name", "Gender", "Birth date". Multiple samples can be obtained from one patient, as illustrated in Figure 3.3.56: samples 11 and

12 are both obtained from patient Abc (red). The link between the two samples and the patient is provided by a *dependency*. Similarly, 3 profiles were obtained from sample 11 (red) in the figure.

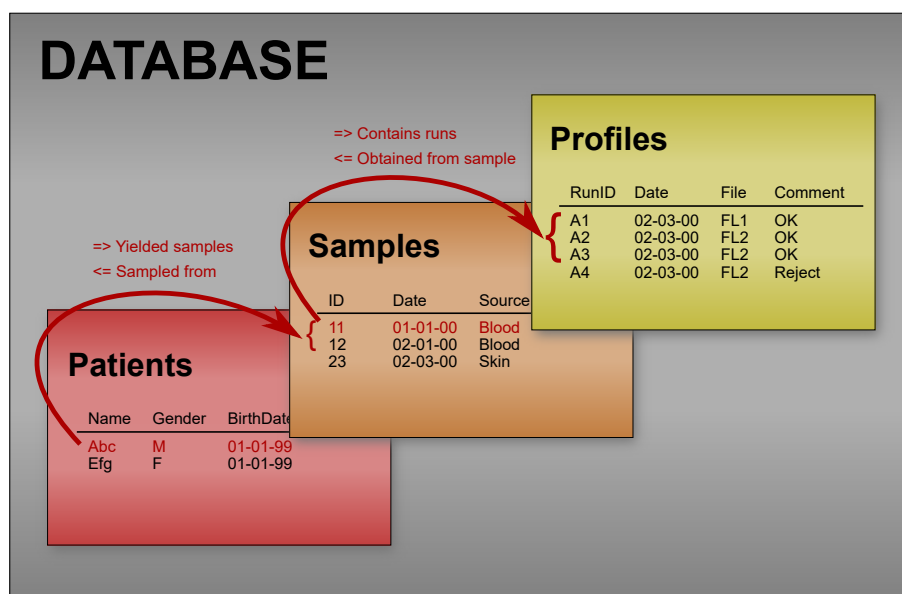


Figure 3.3.56: Scheme illustrating the use of levels and dependencies in a BioNumerics database.

In this way, samples are unambiguously assigned to patients, and profiles to samples, without information being duplicated. However, there is more to be achieved with the levels/dependencies construction. It becomes for example possible to calculate average MALDI profiles based upon all MALDI profiles that belong to the same sample. Or, one can also calculate and fill in the experiment type for MALDI profiles at the level of Samples, also by averaging the profiles obtained for each sample.

Database levels are managed from within the *Database design* panel in the *Main* window (see Figure 3.3.57).

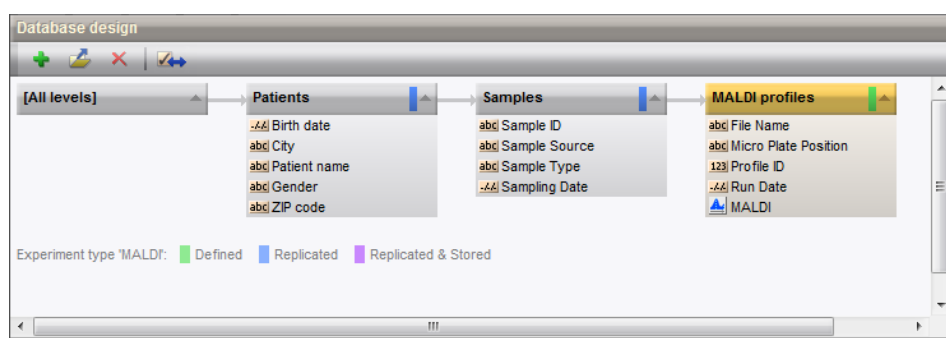


Figure 3.3.57: The *Database design* panel, from which database levels are managed.

Database levels are schematically visualized in the *Database design* panel. This panel appears by default as a tab behind the *Database entries* panel. However, when working in a leveled database, it might be more convenient to dock the panel to a different position (see 2.3.4 for instructions).

The levels are represented in hierarchical order: "All levels" on the left, followed by the highest database level(s) and proceeding to the deepest level on the far right. Dependencies between the levels are indicated with a connecting arrow. Entry information or experimental data (see 3.3.10.3) that is available at each level, is listed below the level name. With a small icon, the information field content type (text, date or number; see 3.3.6) and the experiment type is indicated. The list can be collapsed by clicking on the arrow icon on the right-hand side of the level name.

When clicking on a specific level in the *Database design* panel, this level becomes the *active* level in the BioNumerics database. As a result, the *Database entries* panel and the *Entry fields* panel are updated with the default view (see 3.2.2) for that level. By default, this means that all entries and all entry information fields from the active level are displayed in their corresponding panel.

When an entry information field or an experiment is highlighted (in resp. the *Entry fields* panel or *Experiment types* panel), its level assignment (see 3.3.10.3) is indicated using a color code. See Figure 3.3.57 for an example.



Obviously, in a default unleveled database, only "All levels" will be shown in the *Database design* panel.



Using the concept of levels and dependencies, "branching" database designs can be created as well, as long as each level has either "All levels" or another level as parent level. Several dependent levels can share the same parent level.

3.3.10.2 Creating database levels

It is important to note that levels should be added in order of hierarchy, with the highest level first. Therefore, it good practice to first make a sketch of the desired database layout and to verify if this design meets all requirements, before actually structuring the database.

Proceed as follows to create a new database level. In the *Database design* panel, click on the level that will serve as parent level for the new database level (or on "All levels" to create a top level) and select **Database > Levels > Add new level...** (+). This will show the *Level information* dialog box (see Figure 3.3.58).



Structuring a database with levels and dependencies is typically something to do *before* any data are added to the database. Existing data can be organized in levels as well, but there are currently no tools available in the user interface to do so. Please contact Applied Maths for assistance with this issue.

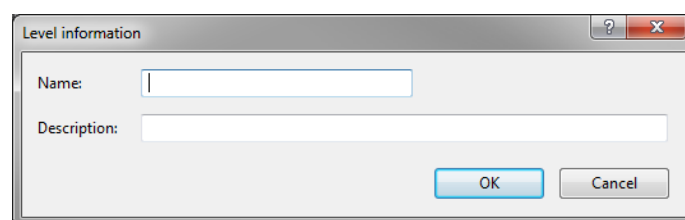


Figure 3.3.58: The *Level information* dialog box.

This dialog box prompts for a level *Name*, i.e. the display name that will be used to indicate the level in the software and an optional *Description*. This description will appear as a tool tip when hovering over the graphical visualization of the level in the *Database design* panel.

After pressing <OK>, the new level will appear in the *Database design* panel and will be linked to its parent level or to "All levels", depending on the choice made earlier.

After creation, a level can be renamed by double-clicking on the level in the *Database design* panel. This opens the *Level information* dialog box again, as discussed above.

An "empty" level (i.e., not containing any entries) can be deleted with **Database > Levels > Delete level...** (X). The software will ask for confirmation before actually removing the level.

3.3.10.3 Summarizing information and experiments

Whether or not entry information and experimental data that is available at dependent levels is displayed at their parent level, is determined by the *level assignment* of the entry fields and experiment types. When entry information fields (see 3.3.3) or experiments are created, they are *defined* at the *active* database level (i.e. the highlighted level in the *Database design* panel). When there is no level active (i.e. "All levels" is highlighted in the *Database design* panel), they are defined for all database levels. After creation, the level assignment of an entry information field or an experiment can still be changed from the *Level assignment* dialog box.

For entry information fields, this dialog is called from the *Database field properties* dialog box. For experiments, select **Settings > Level assignment...** in the corresponding experiment type window. The *Level assignment* dialog box is discussed in 3.3.6.

The way entry information is *summarized* from dependent entries to their parent entries is determined in the *Field summary method* dialog box (see 3.3.6).

Similarly, experiments can be summarized from dependent entries to their parent entries. The way that is done can be determined by selecting **Settings > Summary replication settings...** in any of the experiment type windows. This action will show the *Experiment summary method* dialog box (see Figure 3.3.59).

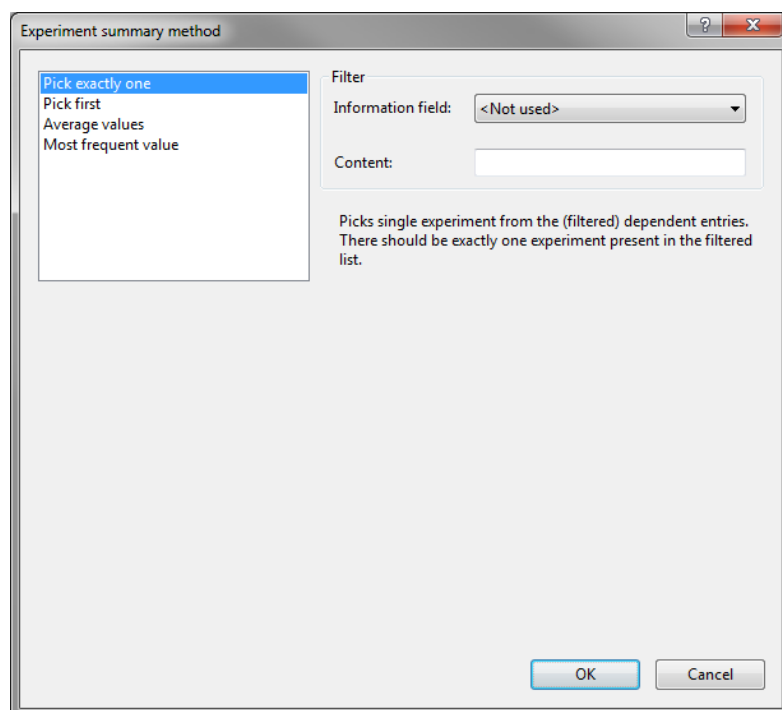


Figure 3.3.59: The *Experiment summary method* dialog box.

Depending on the experiment type, different summary methods are available:

- **Pick exactly one** (all experiment types): Takes a single experiment from the dependent entries. In case the list of dependent entries contains more than one experiment, no experiment will be summarized to the parent entry. To avoid this, use a filter (see below).
- **Pick first** (all experiment types): Takes the first experiment that is encountered in the list of dependent entries.
- **Average values** (only for character types): Takes the average of the character values of all dependent experiments. This method also allows to specify the *Minimum number of experiments*. When

the number of (filtered) dependent experiments is lower than this value, the experiment will not be summarized.

- **Most frequent value** (only for character types): Takes the most frequently occurring value among all dependent experiments. The *Minimum number of experiments* can be specified in addition to a relative (*Threshold (% of total)*) and an absolute (*Threshold (absolute)*) threshold. All three conditions should be fulfilled before the experiment is summarized.
- **IUPAC consensus** (only for sequences): The consensus sequence determined on all dependent sequences. IUPAC nomenclature for ambiguous bases will be used. Since no alignment is performed, this method should only be used for trimmed sequences. In addition to the *Minimum number of experiments*, the consensus determination parameters *Unique base calling consensus*, *2-fold degeneracy consensus* and *3-fold degeneracy consensus* can be specified, which are the same as explained in 8.1.3.7.7. When *Do not allow undetermined bases* is checked, sequences that contain undetermined bases (N) will not be summarized.

In addition, to determine which experiments will be summarized, a **Filter** can be used based on the content of an information field. The *Information field* can be picked from the drop-down list and the *Content* entered in the corresponding text box. Experiments will only be summarized from entries that contain the specified content in the specified entry information field. For example, one could create an entry information field 'Approved', specify two information field states "Yes" and "No" for this field (see) and require that the experiment is approved (= "Yes" filled out for the field 'Approved') before it is summarized to a higher level.

Experiments that are defined at the active level are indicated with a green dot in the *Experiment presence* panel, summarized experiments are shown with a blue dot. The latter are read-only: when a blue dot in the *Experiment presence* panel is clicked, the message "This experiment is replicated and cannot be edited" appears.

3.3.10.4 Editing entry dependencies

The hierarchical structure of entries depending on others is achieved by setting a *parent entry* for each dependent entry. Multiple dependent entries in the same database level can have the same parent entry, but any given entry can only have a single parent entry. This principle can be explained with the "Patient" – "Sample" example that we used earlier: several samples can be taken from the same patient (e.g. different sample types like blood, sputum, urine or the same type of sample taken at different points in time), but an individual sample always originates from just a single patient.

The parent entry can be set for individual entries, for a selection of entries (see below) or – most conveniently – directly during data import (see 3.3.10.5).



The predefined entry information field 'ParentKey' contains the key of the parent entry and can be displayed in any entry grid panel (see 3.3.7).

All dependencies of a single entry can be visualized in the *Dependencies* panel of the *Entry* window, as shown in Figure 3.3.60.

In this panel, the position of the entry in the hierarchy is shown. The entry for which the *Entry* window was opened is displayed in red and indicated with an orange rhomb. Parent entries are shown above this entry and the dependency indicated with a thick grey arrow pointing towards the parent entry. Entries that are dependent on the current entry are shown below and the dependencies illustrated by thin grey arrows. A plus sign (+) indicates the presence of deeper levels with more dependent entries.

To display more dependent entries at a deeper level, select *Edit > Display more dependent levels* (↗). To display fewer dependent entries, use *Edit > Display less dependent levels* (↖).

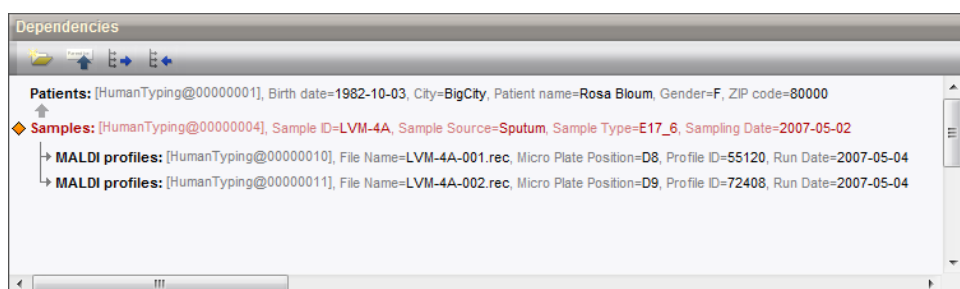


Figure 3.3.60: The *Dependencies* panel.

To set the parent entry for a single entry, select **Edit > Set parent entry...** (📁). This action will display the *Select entry* dialog box, as discussed in 3.2.11. In case a parent entry was already defined, this action will overwrite the parent entry.

To create a new entry that is dependent on the current entry, select **Edit > Add new dependent entry...** (📁). The *New dependent entry* dialog box will pop up (see Figure 3.3.61).

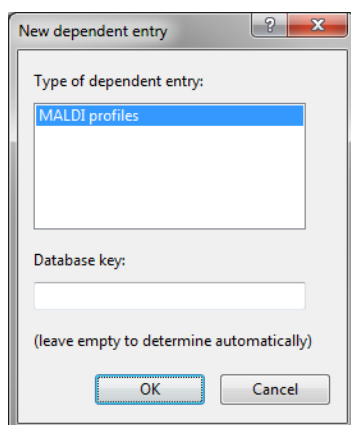


Figure 3.3.61: The *New dependent entry* dialog box.

A database level should be specified in the *Type of dependent entry* list.

The **Database key** can either be entered manually or automatically determined according to the key template, as specified in the *Database settings* dialog box (see 3.7.2).

The parent key can be set for a selection of entries after installing the *Database tools plugin* (for more information, see the separate *Database tools plugin* manual).

To find all dependent entries from a selection of entries, select **Database > Levels > Transfer selection to other levels** (↔). All parent entries and dependent entries will be selected in the corresponding database levels.

3.3.10.5 Importing data in a leveled database

Importing data in a database that has been structured with levels is largely the same as importing data in a "flat" database (e.g., see 3.3.5). However, it is convenient and good practice to set the parent key of any new entries directly during import.

The import wizard can import information at the active database level and in any parent level thereof. Therefore, prior to starting the import, click on the lowest (i.e., left-most) level where the external source contains data for to make this the active level.



Import templates (see 3.3.5.4) are defined at a certain database level. Therefore, only the import templates that were defined at the active database level will appear in the *Import template* wizard page.

In case the database contains levels, the *Edit data destination* dialog box will show the keys and information fields of the active level and all parent levels, with the information fields grouped by level (see Figure 3.3.62 for an example).

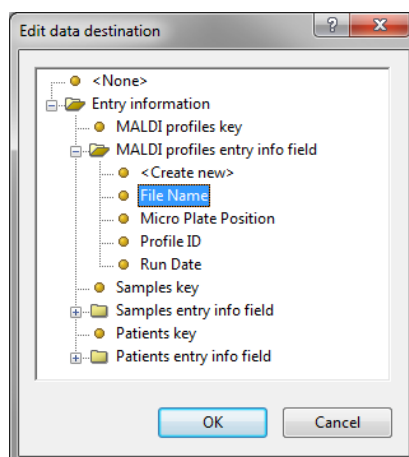


Figure 3.3.62: The *Edit data destination* dialog box, when importing data in a database containing levels.

Chapter 3.4

Database exchange tools

3.4.1 Solutions for data exchange: bundles and XML files

BioNumerics offers two simple and powerful solutions to exchange database information between research sites on a peer-to-peer basis: via *bundles* or *XML files*.

A *bundle* contains selected information (e.g. experiment types, information fields) for a selection of database entries and is the original tool for exchanging BioNumerics database information. It is a compact data package contained in a single file, which can be sent to other research sites over the internet. The receiver can open the bundle directly in BioNumerics and compare the entries contained in it with the own database. However, the information in a bundle is "as is", and cannot be modified or re-analyzed by the receiver.

Exporting BioNumerics database information as *XML files* and importing these again in another database is another available exchange tool. Like bundles, selected information can be included for a selection of database entries. When the XML files are imported in a database, the database entries that were contained in the XML files behave just like other database entries.

Which database exchange tool is to be preferred (bundles or XML files), depends on the specific case and will be a trade-off between compactness and flexibility of analysis.

3.4.2 Database exchange using Bundles

Make a selection of entries in the *Main* window (see 3.3.8 on how to select database entries).

A bundle is created with **File > Create new bundle...** (). This action opens the *New bundle* window (see Figure 3.4.1).

The *New bundle* window lists the available database information fields in the left panel and all available experiment types in the right panel.

You can check each of the database information fields and experiment types to be incorporated in the bundle. For fingerprint types, the fingerprint images, band information, and densitometric curves can be incorporated separately.

A **File name** needs to be specified of the bundle.

Pressing <OK> creates the bundle. A bundle file *filename*.BDL is created in the **Bundles** directory of database.

In case of fingerprint types, the bundle holds the complete information about the reference system used and the molecular weight regression, so that BioNumerics can automatically remap the bundle fingerprints to be compatible with the database fingerprints.

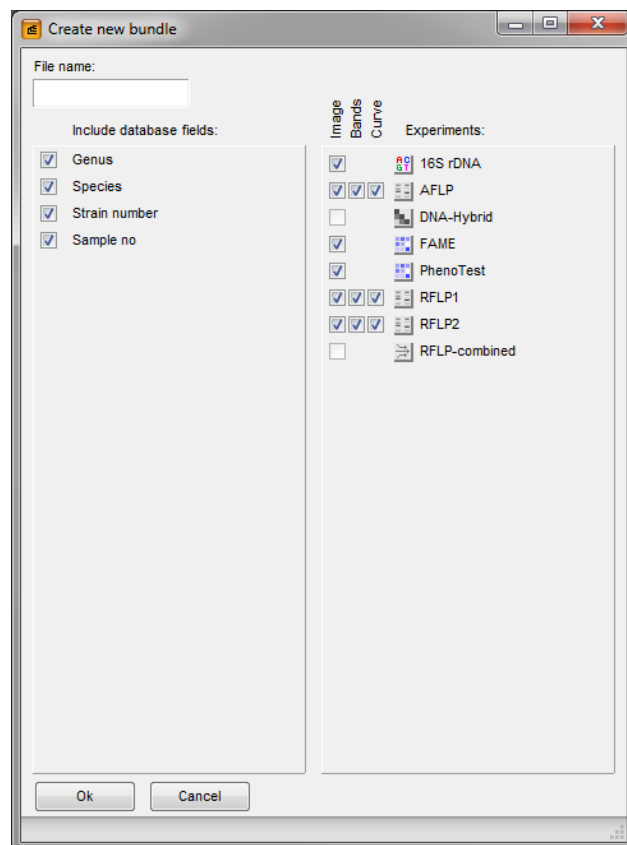


Figure 3.4.1: The *New bundle* window.

To open/close a bundle in a database select **File > Open bundle...** (🔗). This action calls the *Open/close bundles* dialog box (see Figure 3.4.2).

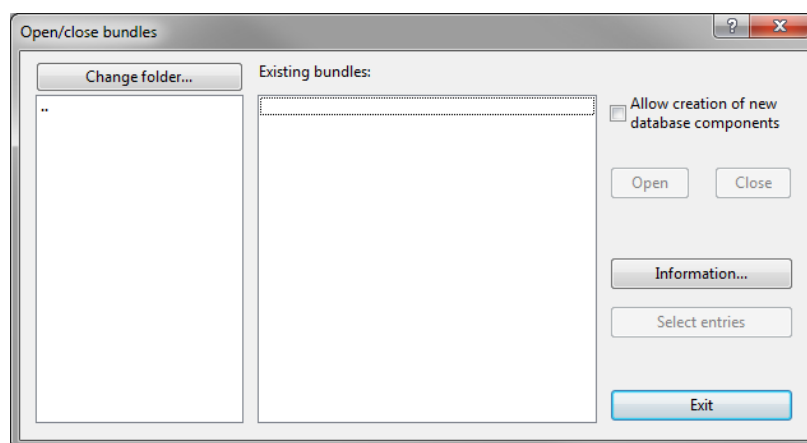


Figure 3.4.2: The *Open/close bundles* dialog box.

In the *Open/close bundles* dialog box, you can browse to the local or network path where the bundle files can be found with the **<Change folder>** button.

In the right panel, you can select a bundle in the list of available bundles in the specified path.

Pressing the **<Open>** button, loads the information of the selected bundle into the database. If the option **Allow creation of new database components** is checked, new experiments will be added to the database if present in the bundle. If this option is unchecked, new experiments will not be added to the database.

A loaded bundle is marked with a "+" in the list.

All entries from an opened bundle can be selected by press the **<Select entries>** button.

Information from a loaded bundle can be removed again from the database with the **<Close>** button.

Pressing the **<Information>** button opens the *Bundle information* dialog box for the selected bundle (Figure 3.4.3).

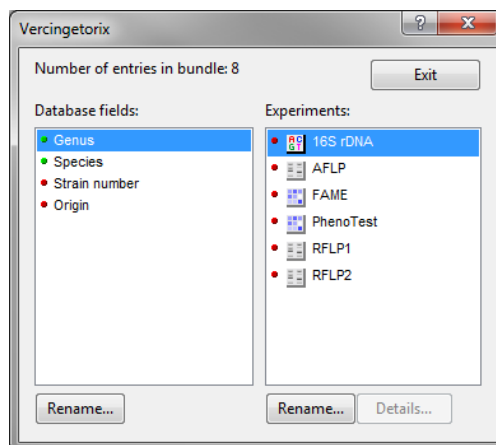


Figure 3.4.3: The *Bundle information* dialog box.

It shows the available information fields in the bundle, as well as the experiment types contained in it.

If an information field or an experiment type is recognized as one of the fields or experiment types in the database, a green dot is shown left from it. If not, a red dot is shown left from it.

As soon as the bundle is opened, the missing information fields and experiment types are automatically added to the database.

Another user may have given a different name to the same database field, and this would BioNumerics cause to consider the fields as different. If you know a field in a bundle is the same as a field defined in the database, you can rename it using the **<Rename>** button below the *Database fields* list.

Another user may have given a different name to the same technique, and this would BioNumerics cause to consider the techniques as different experiment types. If you know a technique in a bundle is the same as one of the experiment types defined in the database, you can rename it using the **<Rename>** button below the *Experiments* list.

In addition, in character types, the characters may have received different names from other users. For example, institution 1 may have named a character "Alpha-Glucosidase", and institution 2 "a-Glucosidase". Obviously, BioNumerics will consider these different names as different tests. To avoid this, you can select the character type and press the **<Details>** button. This calls the *Bundle information* dialog box (see Figure 3.4.4).

A list of all characters in the experiment type is shown, and those corresponding to characters in the database's experiment type are marked with a green dot; the characters not recognized in the database's experiment type are marked with a red dot. You can rename such characters with the **<Rename>** button.

For all functions, entries from a bundle behave like normal database entries.

If you exit the software, they are not automatically loaded when you run the software again. If you know a saved comparison contains bundle entries, you should load the bundles before opening the comparison, in order to avoid an error message.

If you want a bundle to be always opened with the database when BioNumerics is started up, you should rename it to contain the prefix "@_" before its name and the .bdl file should be placed in the **Bundles** folder

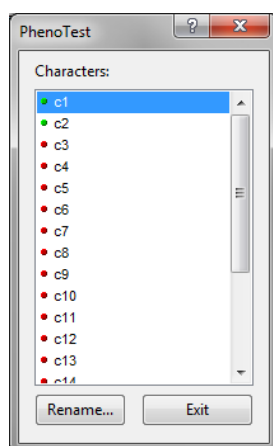


Figure 3.4.4: A list of characters.

of the corresponding database.

3.4.3 Database exchange using XML files

3.4.3.1 Introduction

With the XML-file based database exchange tools in BioNumerics, database entries, database entry fields, and/or experiment type data can very easily be exchanged between different databases.

3.4.3.2 Export data exchange data

Selecting the *Data exchange* option under *Data exchange* in the *Export* dialog box (see Figure 3.4.5) and pressing <Export> brings up the *Export database exchange* dialog box (see Figure 3.4.6).

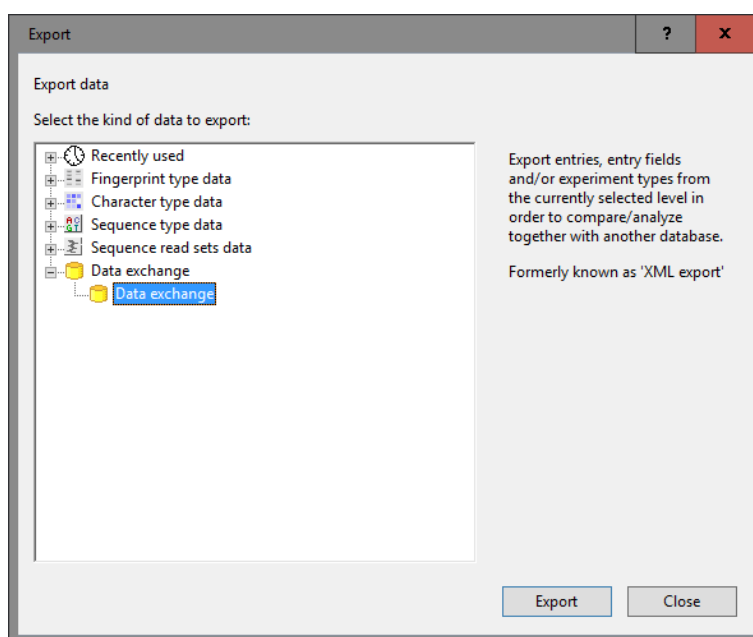


Figure 3.4.5: The *Export* dialog box.



When working in a leveled database (see 3.3.10), the data of the currently selected level will be exported. Make sure to select the correct level before launching the *Export* dialog box.

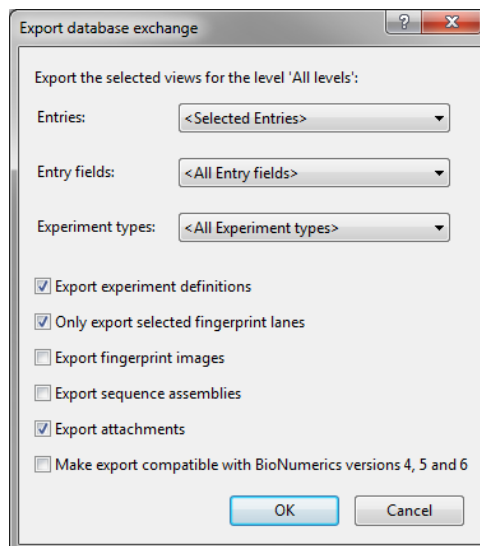


Figure 3.4.6: The *Export database exchange* dialog box.

In the *Export database exchange* dialog box one can choose between following views for the export of the database entries, database fields, and experiment types: ***Selected***, ***Loaded***, ***All***, and ***My***.

Checking ***Export experiment definitions*** will export the general and comparison settings of the experiment types contained in the selected ***Experiment types*** view.

When the option ***Only export selected fingerprint lanes*** is checked, BioNumerics will only export the lanes that are linked to entries that are contained in the selected ***Entries***.

To transfer the fingerprint images, check the option ***Export fingerprint images***.

To export the assembly projects, including the individual trace files, check the option ***Export sequence assemblies***. Unchecking this option will only export the consensus sequences.

Checking the option ***Export attachments***, exports text attachments and links to attachments (Bitmap, HTML, Word, Excel, or PDF documents) that are present in the database for the entries contained in the selected ***Entries*** view.

If you want to import the data into a BioNumerics database version 6 or lower, check the last option for compatibility reasons.

Pressing <OK> will start with the creation of the XML files.

When the option ***Make export compatible with version 4,5 or 6*** was disabled, the XML files are zipped and the zipped folder is stored in the sub-folder **Export** of the database folder. Information contained in the zipped folder can be imported in a BioNumerics 7 database using the ***Import data exchange*** option (see 3.4.3.3).

When the option ***Make export compatible with version 4, 5 or 6*** was checked, the xml files are stored in the sub-folder **Export** of the database folder. The files can be imported in a BioNumerics version 4, 5 or 6 database using the ***XML tools plugin*** import options (see the plugin manual for more information).

3.4.3.3 Import data exchange data

3.4.3.3.1 Import data from BioNumerics version 7

Files created with the **Data exchange** option in the *Export* dialog box (see 3.4.3.2) can be imported in a BioNumerics database with the **Import data exchange data** option under **Data exchange** in the *Import* dialog box (see Figure 3.4.7).

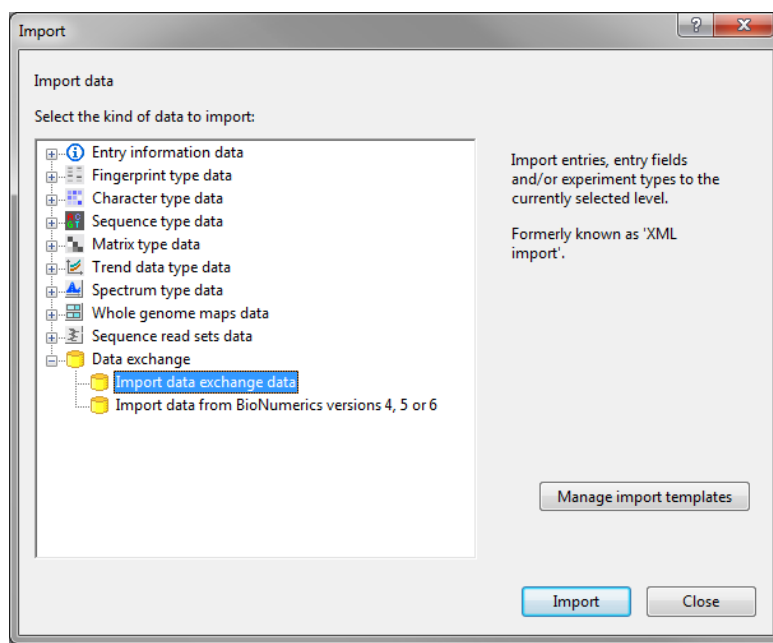


Figure 3.4.7: The *Import* dialog box.

Selecting the **Import data exchange data** option under **Data exchange** in the *Import* dialog box and pressing **<Import>** brings up the *Input* dialog box of the *Import database exchange* dialog box (see Figure 3.4.8).



When working in a leveled database (see 3.3.10), the data will be imported in the currently selected level. Make sure to select the correct level before launching the *Import* dialog box.

Browse for the zipped folder containing the XML files, created with the **Data exchange** option in the *Export* dialog box. The folder is stored in the sub-folder **Export** of the database folder of the database the data was exported from.

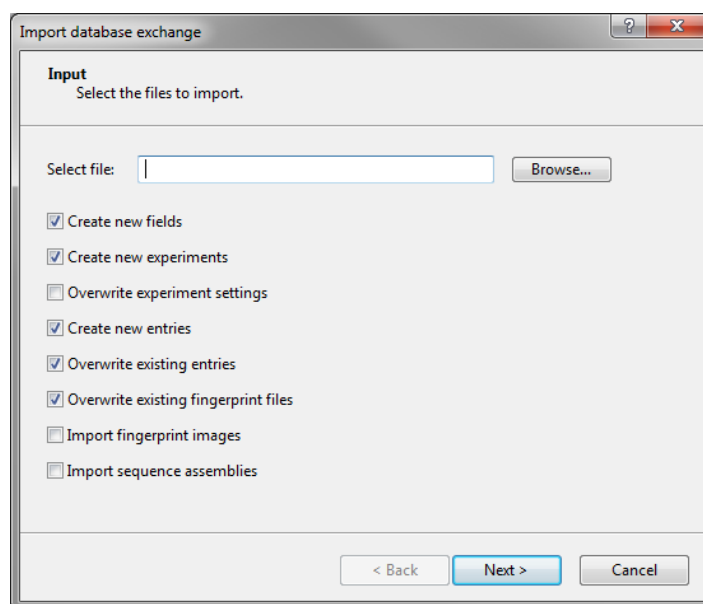
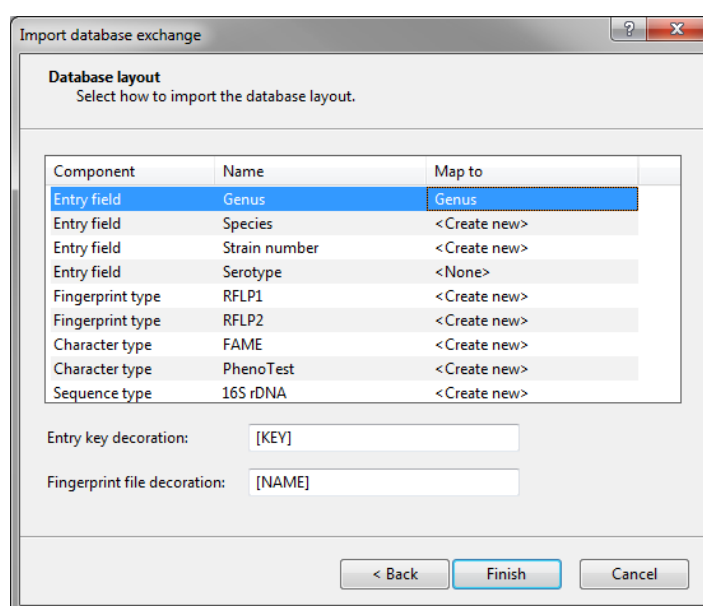
Check the **Create new** options if you want to create new fields, experiments, and entries in the database.

Uncheck the options **Overwrite experiment settings**, **Overwrite existing entries**, and/or **Overwrite existing fingerprint files** if you do not want BioNumerics to overwrite existing experiment settings, entries and fingerprints files in the database.

When the option **Import fingerprint images** is checked, the raw TIFF images are imported in the database (if present in the zipped export folder).

When the option **Import sequence assemblies** is checked, the assembly projects, including the individual trace files, are imported in the database (if present in the zipped export folder). The trace files will be stored *inside the relational database* (irrespective of the state of the **Store trace files in database** option in the *Database settings* dialog box), so they might quickly fill up the database when large numbers of trace files are contained in the assembly projects. Unchecking the option **Import sequence assemblies** will only import the consensus sequences.

Pressing **<Next>** brings up the *Database layout* dialog box (see Figure 3.4.9).

Figure 3.4.8: The *Input* dialog box.Figure 3.4.9: The *Database layout* dialog box.

All entry fields and experiment types found in the zipped folder are displayed as rows in the grid. The names of the entry fields and experiment types contained in the selected folder are displayed in the **Name** column, and the type of information is displayed in the **Component** column.

In the **Map to** column the link with the database needs to be provided. Entry fields and experiment types contained in the zipped folder can be associated with existing, new (<Create new>), or no (<None>) entry fields and experiment types.

When an entry field or experiment type name contained in the selected folder is the same as an existing field or type in the database, the database field/type is automatically selected in the **Map to** column. If the names cannot automatically be mapped to existing fields/types in the database, the <Create new> option is selected.

Changing the destination of a row is done by double-clicking on the row in the grid. This action calls the

Database mapping dialog box (see Figure 3.4.10).

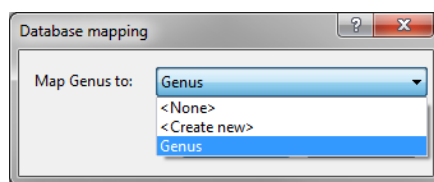


Figure 3.4.10: The *Database mapping* dialog box.

In the *Database mapping* dialog box the selected field/type can be mapped to an existing or new (<*Create new*>) database field/type. Selecting <*None*> will not import the information in the database.

Using an *Entry key decoration* string, information can be added to the imported keys. The *Entry key decoration* string recognizes the [KEY] tag and can be combined with plain text, inserted before or after the [KEY] tag.

Using a *Fingerprint file decoration* string, information can be added to the imported fingerprint file names. The *Fingerprint file decoration* string recognizes the [NAME] tag and can be combined with plain text, inserted before or after the [NAME] tag.

Pressing <*Finish*> will start with the import of the data.

When information is linked in the *Database layout* dialog box to new entry fields or experiment types (<*Create new*>), the *Create database components* dialog box will prompt for the entry field and experiment type names (see Figure 3.4.11).

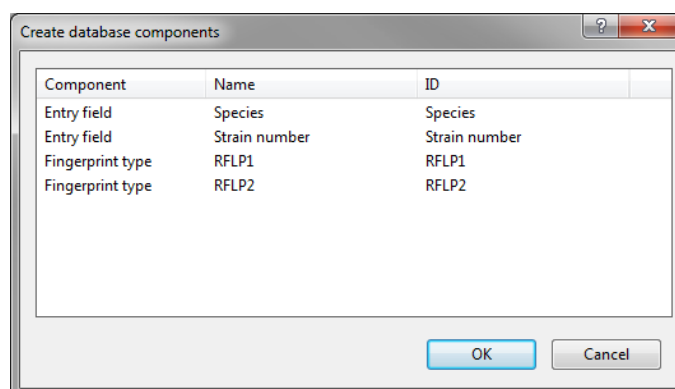


Figure 3.4.11: The *Create database components* dialog box.

All entry fields and experiments that are linked to new fields/experiments in the databases are listed in the *Create database components* dialog box. The names found in the zipped folder are suggested by the software (*ID* column), but can be edited by clicking twice on the name in the *ID* column. The name will appear highlighted and can be edited. Pressing <*OK*> will create the specified database components in the database.

After import BioNumerics needs to be restarted to activate all changes.

3.4.3.3.2 Import data from BioNumerics versions 4, 5 or 6

Files created with the *XML tools plugin* in BioNumerics version 6 or lower can be imported in a BioNumerics database with the *Import data from BioNumerics version 4, 5, or 6* option under *Data exchange* in the *Import* dialog box (see Figure 3.4.12).

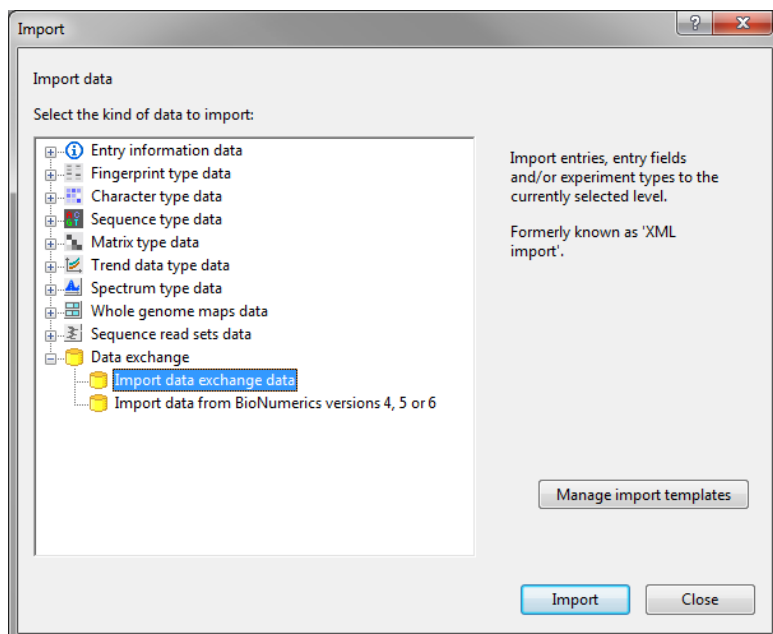


Figure 3.4.12: The *Import* dialog box.

Selecting the *Import data from BioNumerics version 4, 5, or 6* option under *Data exchange* in the *Import* dialog box and pressing <Import> brings up the *Import database exchange* dialog box (see Figure 3.4.13).



When working in a leveled database (see 3.3.10), the data will be imported in the currently selected level. Make sure to select the correct level before launching the *Import* dialog box.

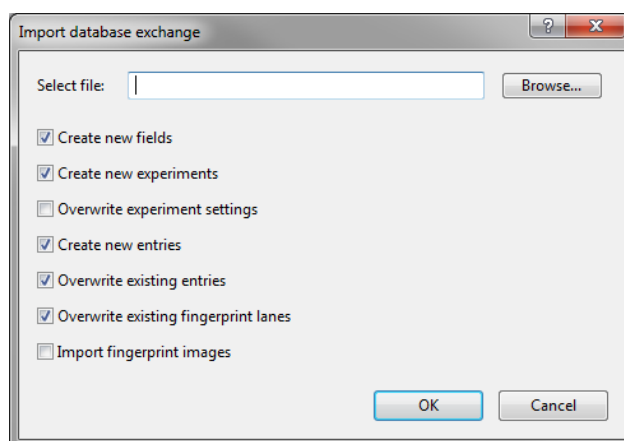


Figure 3.4.13: The *Import database exchange* dialog box.

Browse for the individual file(s), created with the *XML tools plugin* in BioNumerics version 6 or lower.

Check the *Create new* options if you want to create new fields, experiments, and entries in the database.

Uncheck the options *Overwrite experiment settings*, *Overwrite existing entries*, and/or *Overwrite existing fingerprint files* if you do not want BioNumerics to overwrite existing experiment settings, entries and fingerprints files in the database.

When the option *Import fingerprint images* is checked, the raw TIFF images are imported in the database (if available in the selected files).

Pressing <OK> will start with the import of the data.

After import BioNumerics needs to be restarted to activate all changes.

Chapter 3.5

User management

3.5.1 Introduction

When working with a database that can be accessed by multiple persons, an effective database user management becomes important, to ensure both data confidentiality and integrity. The larger an organization, the more vital and elaborate this task will become. BioNumerics uses three key concepts for user management: users, user groups and privileges. The relation between these concepts is explained in Figure 3.5.1.

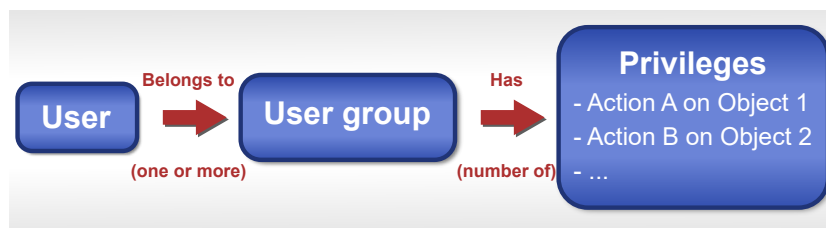


Figure 3.5.1: Relations between the three key concepts in user management: users, user groups and privileges.

A *user* thereby corresponds to a physical person who accesses the BioNumerics database. Each user is authenticated via his/her own unique user ID and password. A strict user authentication policy can enforce confidential data only being viewed by persons qualified to do so. Restrictions regarding data modification are set using *privileges*. A privilege can be defined as a certain type of action that is allowed on a certain type of object. Rather than granting privileges directly to a user, a system which would become quickly very complex with a large number of users, a much more surveyable system is employed: users are assigned to one or more *user groups*, to each of which a specific set of privileges is granted. Privilege management is accomplished by setting up an ordered list of allow/deny rules for each user group. A user group should be thought of as a number of users (i.e. persons), all fulfilling the same or a very similar role. To protect against accidental modification of data, a user group should ideally be granted only the necessary privileges to fulfill this role and not more (rule of least privilege). Furthermore, users should be assigned to a user group (or to a number of user groups) that best fits their role. As indicated above, privileges only apply for *modification* of data: for users that belong to a user group to which no privileges are granted, the BioNumerics database will be read-only.



In addition to the BioNumerics' own user management tools, described in this section, the relational database itself should be protected at the level of the database management software (see 3.7.5) in order to prevent direct or SQL access to the relational database.

3.5.2 Users and user groups

When a new database is created or when a database is upgraded from an older version, three *user groups* are automatically created by the software (**Administrators**, **Powerusers** and **Users**), as is one *user* labeled **_DefaultUser_**. This default user is assigned to the **Administrators** group. Each new user added to the database, can be assigned to one or more user groups. A user is granted the permissions of the user group(s) it is assigned to. New user groups can be added to the database, and permissions can be assigned to the new groups, depending on which operations the users assigned to these groups are to perform on the database.

Selecting **Database > User management...** brings up the *User management* window (see Figure 3.5.2).

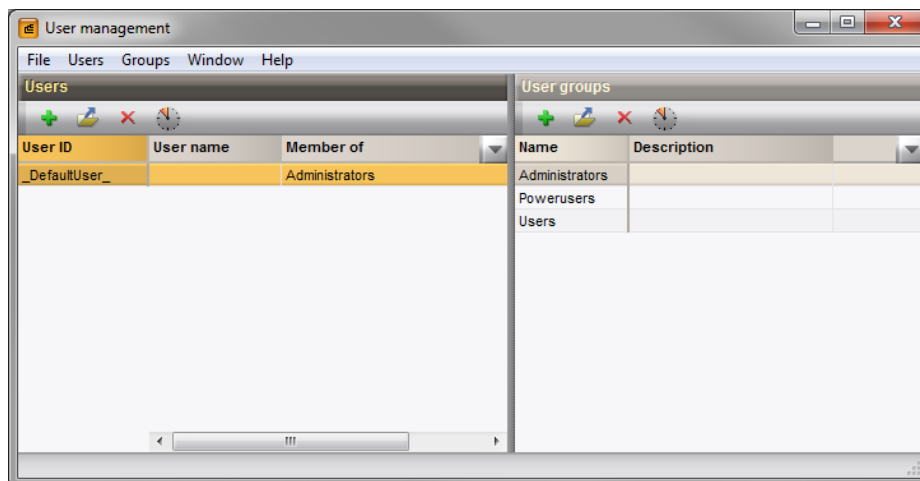


Figure 3.5.2: The *User management* window: default view.

In the *User management* window, the database users are listed in the *Users* panel (left panel in default configuration) and the user groups are shown in the *User groups* panel (right panel). The three default user groups are **Administrators**, **Powerusers**, and **Users**.

To edit the privileges for a user group, highlight the user group from the list and select **Groups > Edit...** (🔧). Alternatively, simply double-click on the user group in the *User groups* panel. This action will open the *Account group settings* dialog box (see Figure 3.5.3).

The *Account group settings* dialog box consists of three tabs, of which the *General tab* is displayed by default.

The name of the user group is listed in the left upper corner (**Group ID**) of the *General tab*. Once a user group is added to the database, the Group ID cannot be changed anymore. Optionally a **Description** for the user group can be specified.

The **Allowed components for BNServer client** text box can contain a comma-separated list of privileges that are taken into account when a client accesses a database on a BioNumerics Server directly.

The *privileges* are represented in an expandable tree under **Privilege management** (left list). The privileges are categorized in different groups, and are in most cases a combination of an *object type* (see 3.2) and an *action type*:

- **Administration** privileges:
 - Users assigned to at least one user group that is allowed to **Modify database system settings**, can access the *Database settings* dialog box and subsequently encrypt the connection string and/or the complete .XDB file (see 3.7.2) or change database system settings (see 3.5.3).
 - **Modify audit trail settings** is the privilege to determine which object(s) will be included in the audit trail (see 3.6.2).

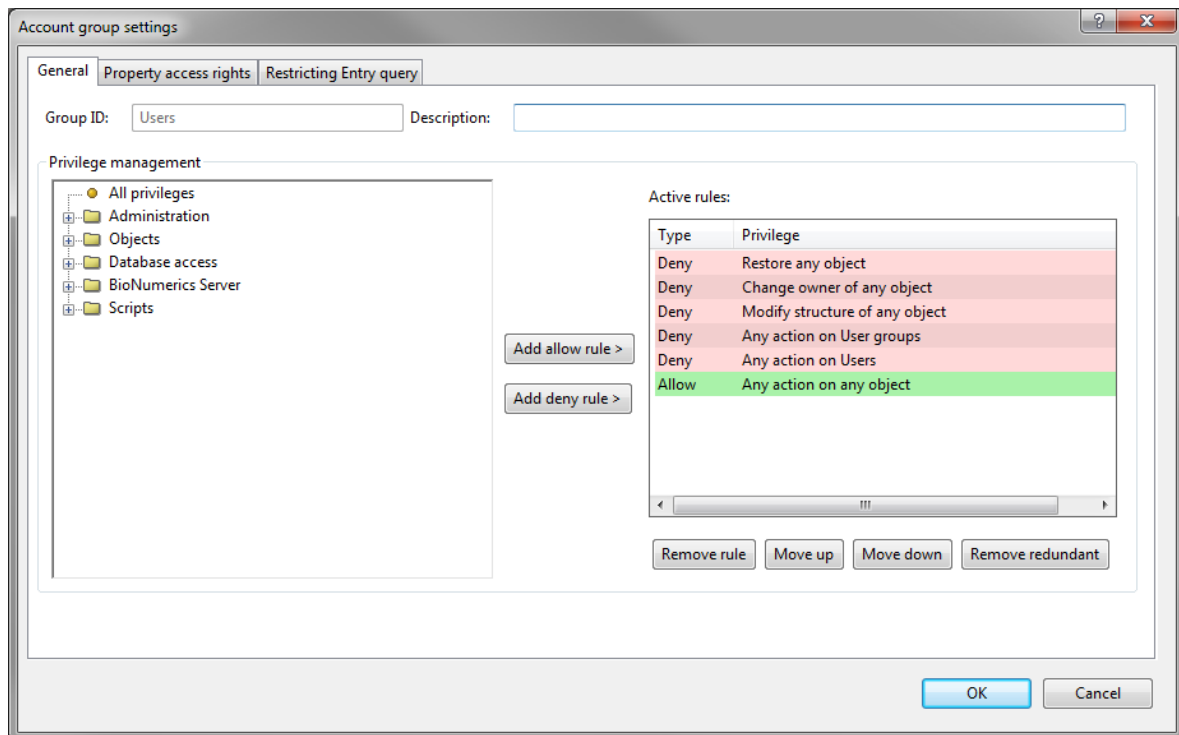


Figure 3.5.3: The *Account group settings* dialog box, showing the general settings for the default user group Users.

- A separate *Open user management window* privilege exists to open the *User management* window, from which the current dialog is called.
- **Objects** privileges:
 - All object privileges can be specified for each individual object (e.g. *Create Entries*, *Create Entry fields settings*,...), or all objects in the database (e.g. *Create any object*, *Rename any object*,...).
 - Use the object privilege *Any action on any object* to apply all object actions on all objects in the database.
- **Database access** privileges: Whether the database can be opened in an interactive session or as a BioNumerics Server client.
- **BioNumerics Server** privileges: These privileges are only applicable when connecting as client to a BioNumerics Server.
- **Scripts** privileges: This option contains the *Run arbitrary scripts* privilege. When this privilege is denied, the user cannot open the script editor or execute scripts from a file or command line. Scripts saved in the database directory or home directory (see 2.1.6) cannot be executed and will not appear as menu items. Only scripts that are stored in the relational database via the *Plugins* dialog box (see 2.1.5) can be executed.
- **All privileges**: This option groups the administration, objects, database access and BioNumerics Server privileges into one single privilege.

Each privilege in the left list can be combined with "allow" or "deny" into a *rule*. To add a new rule to the list, select a privilege from the left list and press <Add allow rule> to grant the selected privilege to the

user group. Press the **<Add deny rule>** button to refuse the action(s) of the selected privilege for the user group. The rule is added to the **Active rules** list.

The **Active rules** appear in an ordered list, with the most general rule at the bottom of this list. Towards the top of the list, the rules get more and more specific, i.e. they form exceptions to the more general rules below them. When a user that is assigned to the user group performs an action on the database, the **Active rules** will be checked from top to bottom. If the privilege of the first (i.e. most specific) rule covers the action, the rule will be applied and the action is either allowed or denied. If the privilege does not cover the action, the second rule is checked, etc. (see Figure 3.5.4 a for an example). When a user belongs to more than one user group, the action is allowed if the **Active rules** of at least one group allow the action.

Type	Privilege
Deny	Any action on Users
Deny	Any action on User groups
Allow	Any action on any object

Type	Privilege
Allow	Any action on any object
Deny	Any action on Users
Deny	Any action on User groups

Figure 3.5.4: (a) Users assigned to this group can perform all object actions in the database, except user management alterations; (b) Users assigned to this group can do all object actions in the database (based on first allowed rule). The two deny rules are redundant, and are removed from the list when checking for redundancy.

If more than one rule is specified for a user group, the order of the rules in the **Active rules** list can be changed with the **<Move Up>** and **<Move Down>** buttons.

Pressing the **<Remove redundant>** button, removes all redundant rules from the list, if any (see Figure 3.5.4 b for an example).

A rule is removed from the list with the **<Remove>** button.

Users assigned to the default user group **Administrators** can by default do everything (**Allow All privileges** is default the only active rule for this group). **Powerusers** can by default do everything except database system changes (e.g. change the audit trail settings) and user management alterations (see Figure 3.5.4 a for all default active rules for this group). Users assigned to the **Users** group (see Figure 3.5.3) can by default only do the normal day-to-day work in the database (e.g. they cannot make new information fields or remove them, or cannot change the ownership of objects, etc.).

The **Property access rights tab** of the **Account group settings** dialog box is shown in Figure 3.5.5.

All entry information fields and experiment types present in the database as well as custom scripts and plugins stored in the database through **Plugins** dialog box are listed in the grid, with their **Name** and two access rights:

- **Can read:** If checked, the information will be visible to members of this user group. If unchecked, the information will not be displayed (appears as hatched areas). Note that unchecking this option will also uncheck the **Can modify** access right.
- **Can modify:** If checked, members of this user group can make modifications to the information. Checking this option will also uncheck the **Can read** access right. If unchecked, information will not be editable (entry fields) or it will not be possible to save any changes to the database (experiments).

Entry fields and/or experiments can be selected (highlighted) in the grid using **Ctrl+click** or **Shift+click** for

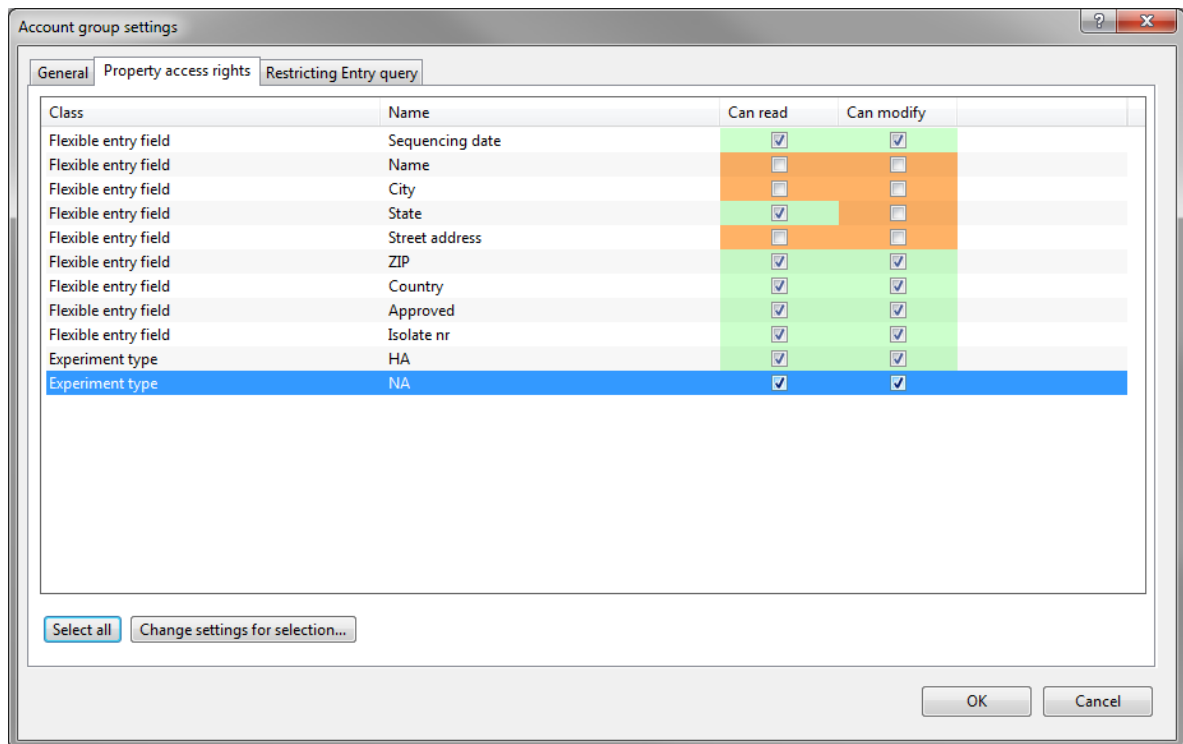


Figure 3.5.5: The *Account group settings* dialog box, *Property access rights* tab.

a complete range and the access rights can be changed for the selection by pressing **<Change settings for selection>**. This action calls the *Property access right* dialog box (see Figure 3.5.6).

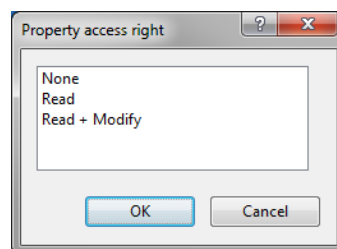


Figure 3.5.6: The *Property access right* dialog box.

From this dialog, the access rights can be set for the selection made in the *Account group settings* dialog box.

All entry information fields and experiment types can be selected at once with **<Select all>**. In combination with **<Change settings for selection>**, this can be used to reset all access rights.

The *Restricting entry query* tab of the *Account group settings* dialog box is shown in Figure 3.5.7.

All query-based entry Views (see 3.2.2) that are available in the database will be listed. A view can be selected that will act as a *restricting query*: only those entries that are included in the selected view will be visible for the current user group. When "All entries" is selected, no restriction will be applied.

To add a new user group to the database, select **Groups > Add new...** (+) in the *User management* window. The *Account group settings* dialog box will pop up again. You can specify a unique **Group ID**, optionally add a **Description**, and assign rules to the user group. Pressing **<OK>** will add the group to the list of user groups.

Use **Groups > Remove highlighted...** (X) to remove a user group from the database.

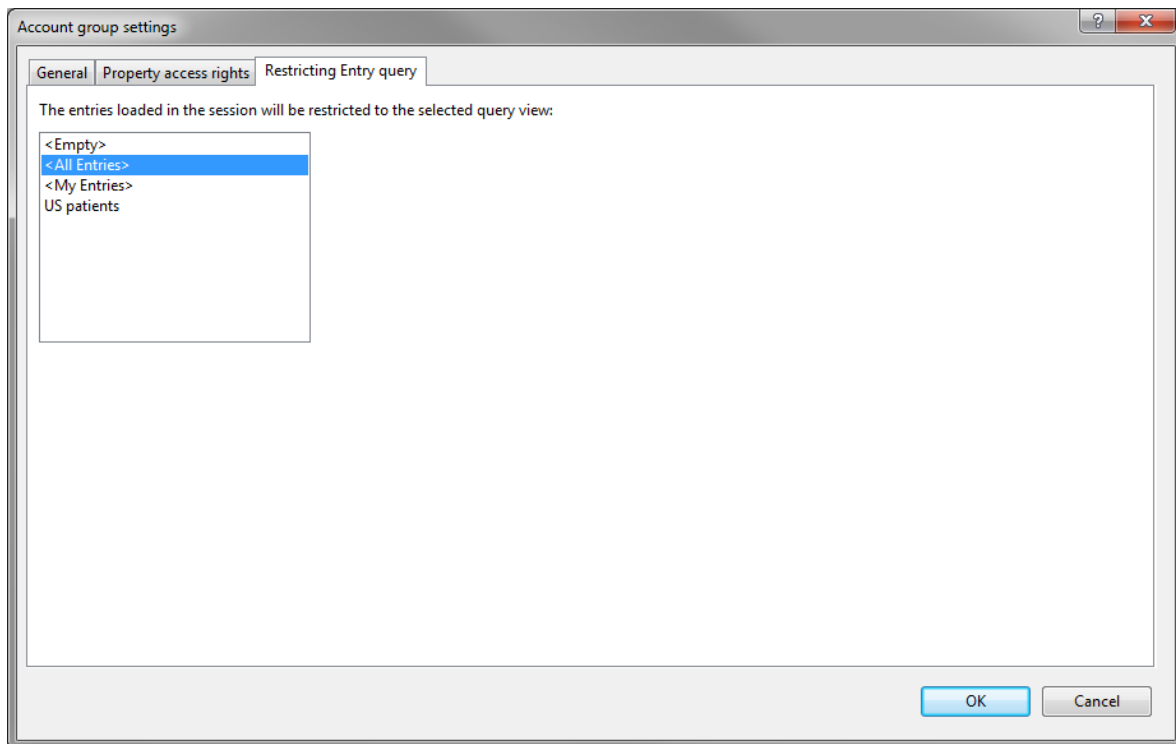


Figure 3.5.7: The *Account group settings* dialog box, *Restricting query* tab.



User groups can only be removed if no users are linked to the group.

The *Users* panel lists the users that are assigned to the database (left panel in Figure 3.5.2). By default, only one user is listed in the left panel, the **DefaultUser**. This user is assigned to the **Administrators** group. Each user that is added to the database, can be assigned to one or more user groups. If a user is assigned to more than one user group, the highest level privileges of the user groups will apply for this user. New users can be added to the user list, and existing users can be removed from the list, or re-assigned to other user groups.

To add a new user to the list, select *Users* > *Add new...* (+). This brings up the *User settings* dialog box (see Figure 3.5.8).

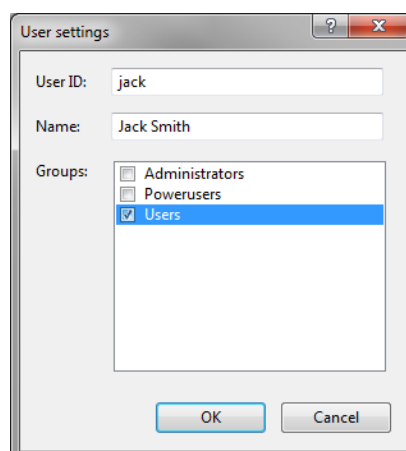


Figure 3.5.8: The *User settings* dialog box.

This dialog box allows you to specify a unique **User ID** for a new user, which cannot be changed later on.

Optionally, a **Name** can be entered for the user, which will be listed as 'User name' in the *User management* window. The bottom part of the dialog lists all user groups that are defined in the database. Via the check boxes, one or more groups that the user should belong to, can be selected. Pressing <OK> will add the user to the *Users* panel of the *User management* window.

The **User ID** is used to uniquely identify a user. As soon as a user is added to the database, the User ID cannot be changed anymore. The User ID of the user that is currently logged on, is displayed in the status bar of the *Main* window (see Figure 3.5.9).

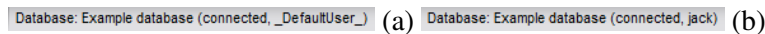


Figure 3.5.9: User ID of the user that is currently logged on, is displayed in the status bar of the *Main* window: (a) _DefaultUser_, (b) Jack.

To recall the *User settings* dialog box for a user, highlight the user in the list and select **Users > Edit...** (🔧). Alternatively, simply double-click on the user in the *Users* panel.

Use **Users > Remove highlighted...** (✖) to remove a user from the database.



The *User management tools plugin* allows you to copy users, user groups and privileges from one database to another.

When there is more than one user defined in the database, one can switch between different users by closing and reopening the database. The *User* dialog box pops up (see Figure 3.5.10).

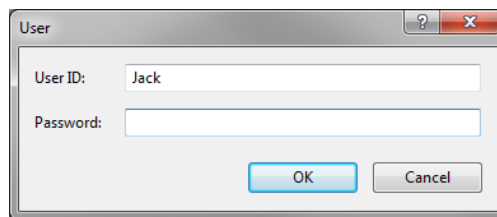


Figure 3.5.10: The *User* dialog box.

This dialog box prompts for the **User ID** and **Password**. By default, the name of the Windows user is displayed in the **User ID** text box.

Specify the correct User ID and - if specified - enter the correct password (see 3.5.3) for this user to log on.

Note that a number of **safety rules** concerning user and user group management are implemented to prevent users from locking themselves out of a database unwillingly:

- The **Administrators** user group cannot be removed.
- A warning message appears when the privileges associated with the **Administrators** user group are edited.
- The last user belonging to the **Administrators** cannot be removed.
- For a user, the **Administrators** user group membership cannot be removed if that user is the only one belonging to this group.

3.5.3 User authentication

User authentication is done through a combination of a unique User ID and a password. Assigning passwords to users is optional in BioNumerics, but it is recommended for security reasons.

To specify a (new) password for the user that is currently logged on to the database, select **Database > Current user > Change password...** in the *Main* window and confirm the action. The *Change password* dialog box pops up (see Figure 3.5.11).

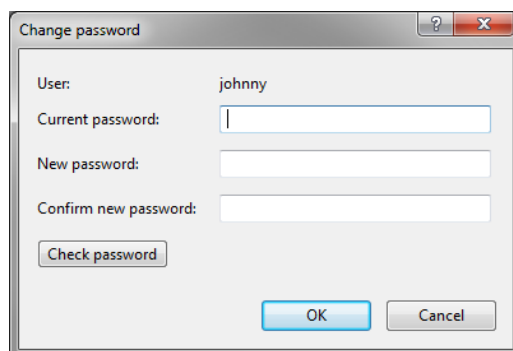


Figure 3.5.11: The *Change password* dialog box, to change the password for the current user.

Type the current password of the user next to **Current password**. If no password has been assigned to the user, leave this field blank. Type the **New password** and re-enter the new password for confirmation (**Confirm new password**).

Use the **<Check password>** button to check if the new password meets the password criteria. Standard user passwords should be at least 4 characters long. Improved security can be achieved when *strong* user passwords are enforced in the database, by checking the option **Require strong passwords** in the *Database settings* dialog box (see 3.7.2).

Press **<OK>** to save the new password of the current user in the database.

BioNumerics encrypts the password in such a way that the original password can not be recovered. The encrypted passwords are stored for each user in the column "PSWRD" in the default table "USERS" of the database. This protects the original password from everyone that can access the relational database. When a user tries to log on to a database, the password that is entered by the user is encrypted and compared with the stored encrypted password in the database.

Passwords assigned to the users in the database are by default *permanent* passwords. This implies that the passwords do not have to be changed after a certain period of time. However, some institutes might want to force users to change their passwords when they log in, if they have not been changed for an extended period (e.g., 60 or 90 days). In BioNumerics it is possible to set an expiration date for passwords in the *Database settings* dialog box (see 3.7.2).

When the validity time of a user's password has expired, the software will prompt for a new password when logging on to the database (see Figure 3.5.12).

Sometimes users forget the password they need to log into the database. When this happens, users must request assistance of another user that belongs to a user group that has the privilege to reset a password in the database (**Allow Modify Users** rule). Users assigned to the default user group **Administrators** are able to reset passwords in the database.

To reset a password for a user, highlight the user in the *Users* panel of the *User management* window, select **Users > Reset password...** and confirm the action. When a password is reset, it will effectively be blank.

A new password should be specified by first logging as the user whose password was reset in the previous step. The **User ID** should be entered in the *User* dialog box and the password field left blank. Next, press **<OK>**. To actually change the password, select **Database > Current user > Change password...**, leave the **Current password** field blank, type the **New password** and re-enter the new password for confirmation (**Confirm new password**). Press **<OK>** to set the new password.

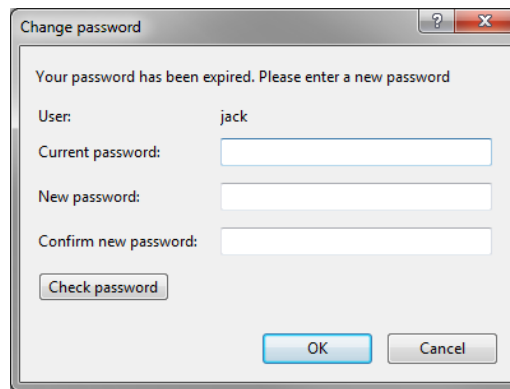


Figure 3.5.12: The *Change password* dialog box pops up when a password has expired.

3.5.4 User activity log

Sometimes institutes might want to keep track of the different BioNumerics users that are logging onto a database. In addition, failed authentication attempts, failed attempts to use signatures (see 3.6.4) and changes made to the database system parameters might be of interest. User activity is logged when the **Log user activity** check box in the *Database settings* dialog box is checked (see 3.7.2): all *user activity actions* will be saved in the database.

Each user logged on to the database can consult the logged user activity actions by selecting **Database > Current user > Show user activity log...** in the *Main* window. This action will load the user log events in the *User activity log* window (see Figure 3.5.13).

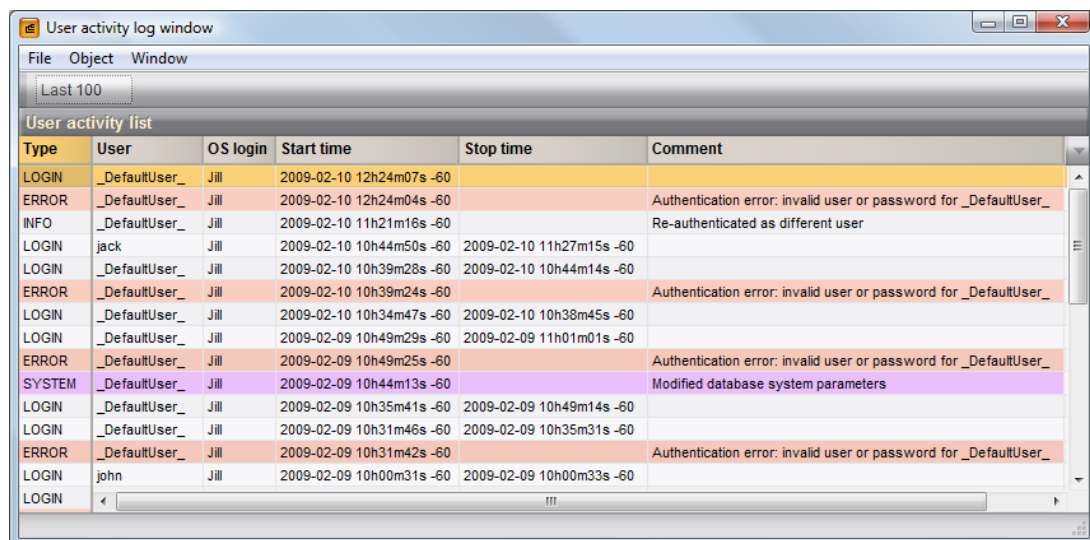


Figure 3.5.13: The *User activity log* window, listing a number of user actions.

By default, only the 100 most recent actions are shown when the *User activity log* window is called. To display more actions, click on the **Last 100** drop-down list and select **Last 1000** or **All** actions.

The type of activity that is recorded is displayed in the 'Type' column (LOGIN, INFO, ERROR or SYSTEM). Each type of activity is displayed in a different color.

- A LOGIN type activity is created each time the database has successfully been loaded by one of the users. The user is displayed in the 'User' column, and the time the database was opened and closed

is shown in the 'Start time' and 'Stop time' columns respectively. The Windows user is shown in the 'OS login' column.


- Every change of current user in the database is reported as an INFO action in the list. The name of the newly logged on user is shown in the 'User' column, and the time the change occurred is displayed in the 'Start time' column. Timestamps are indicated following the ISO 8601 notation, optionally with indication of the time zone (see [3.7.2](#)). The windows user is shown in the 'OS login' column.
- Changing settings in the *Database settings* dialog box is recorded as a SYSTEM type activity. The user who has made the changes in the database is displayed in the 'User' column, the Windows user is shown in the 'OS login' column. The 'Start time' reflects the time the database system parameters were saved in the database.
- Every unsuccessful login and failed attempt to sign an object (see [3.6.4](#)) is reported as an ERROR action. The User ID is displayed in the 'Comment' field and the time the error occurred is displayed in the 'Start time' column. The Windows user is shown in the 'OS login' column.

To close the *User activity log* window, select **File > Exit**.

Chapter 3.6

Audit trails and versioning

3.6.1 Introduction

The functionality described in this section requires the Audit trails module () to be present in your BioNumerics configuration.

The audit trails and versioning tool is an extremely powerful logging tool, especially for users working in a certified environment. When used in combination with users and user privileges, the tool allows you to record for any object in the database who has made which change at which moment in time. The principle is relatively easy: for every change performed on every object, a new version of this object is created, and all previous (historic) versions are kept in the audit trail. Four approaches are available for inspecting previous object versions:

- XML export of the full content
- Generate a readable report
- Open the historic version in its corresponding window
- Restore the previous version of an object as the present state

Restoring previous versions is an action that is audit trailed by itself, i.e. the present state becomes the previous state. The audit trail should be regarded as an ever-progressing time line, a history list to which actions can be added, but no actions can be removed from the list. All functionality for checking the history list and inspecting historic object versions is bundled in the *Audit trail* window (see 3.6.3), which can be accessed from a number of other windows in BioNumerics.



Since the audit trail continuously builds on previous states, it is fundamentally different from an undo chain. The audit trails and versioning tool cannot be readily used to restore the entire database to a previous version, because the history is tracked per individual object and because the creation of new objects cannot be undone. Furthermore, since all data are stored in the same database, the audit trail should by no means be regarded as a backup tool. For the latter two purposes, we recommend to use the backup tools and rollback mechanisms provided by the database management software (DBMS).

An administrator (or any user with sufficient privileges; see 3.5.2) can specify which objects should be included in the audit trail via the *audit trail settings* (see 3.6.3). The audit trail becomes active from the moment at least one object type is specified to be audit trailed. Obviously, activating the audit trail has a number of consequences:

- Objects cannot be renamed, since the object name is the unique identifier in the audit trail.

- Information fields cannot be added, renamed or removed.
- Experiments cannot be removed.

Audit trails will typically be used in a production environment, not in a research setting. The above-mentioned actions are in fact quite fundamental changes to the database layout and will therefore be uncommon in a production environment anyhow.



Only objects that are stored in the relational database will be audit trailed. Therefore, when using the audit trail, it is recommended to store any object by default in the database. This preference can be set in the *Preferences* window (see 2.3.3).

In conjunction with the audit trail, *digital signatures* (see 3.6.4) can be used to approve object versions, similar to the use of handwritten signatures to approve documents. To ensure a similar level of trustworthiness as for their handwritten counterparts, state-of-the-art cryptographic methods are used in BioNumerics to link the signature to the content of the signed object version.

3.6.2 Audit trail settings

By default, audit trails are switched off. A user with sufficient privileges (see 3.5.2) can determine which objects will be included in the audit trail.

To access the audit trail settings, select **Database > Database settings...** in the *Main* window and press the **<Audit trail settings...>** button in the *Tables* tab of the *Database settings* dialog box. The *Audit trail settings* dialog box appears (Figure 3.6.1).

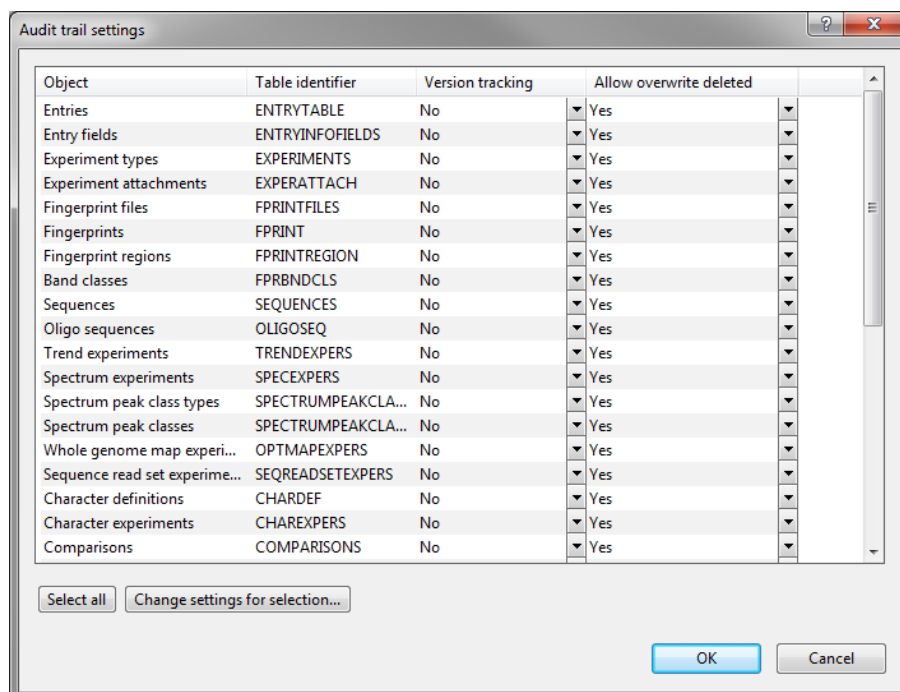


Figure 3.6.1: The *Audit trail settings* dialog box, from which the objects to be included in the audit trail can be selected.

The dialog box list all objects in the database which can be audit trailed ("Objects") and the table name from the connected database ("Table identifier") where they are stored. By default, no objects are included in the audit trail ("Version tracking") and "Allow overwrite deleted" is enabled for all objects.

A number of individual objects can be selected using **Ctrl+click** or a consecutive range of objects selected by clicking the first object to be selected and, while holding the **Shift**-key, clicking the last object in the range. All objects in the list can be selected at once by pressing the **<Select all>** button.

The audit trail settings can be modified for the selected objects by pressing the **<Change settings for selection>** button.

At this stage, it is important to determine which objects should be audit trailed. Obviously, including objects in the audit trails takes up additional storage space in the database. If the database is to be kept compact, only relevant objects should be included.



Because of dependencies between object types, not every combination of audit trailed and non-audit trailed object types can be made. The software will warn for inconsistencies and will make suggestions to obtain feasible combinations.

Select the objects to be audit trailed from the list and press **<Change settings for selection>**. The *Audit trail detail settings* dialog box that appears shows the audit trail settings for the selected objects (see Figure 3.6.2).

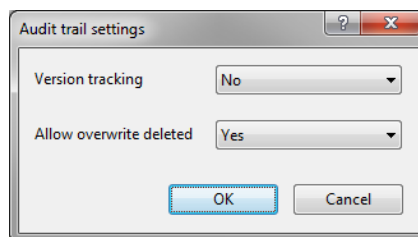


Figure 3.6.2: The *Audit trail detail settings* dialog box, showing the audit trail settings for selected objects.

For **Version tracking**, three different options are available from the drop-down list: "No" will exclude the selected objects from the audit trail. "Log only" will log any modification to the selected objects in the history list, but historic versions will not be saved and it will therefore not be possible to restore historic versions of the object to the present state. "Full tracking" means that all options will be available for the selected objects.

When **Allow overwrite deleted** is enabled ("Yes"), new objects can be created with the same name as a deleted object. In a strict audit trailed environment, this option should be disabled ("No"), to make sure each object can be uniquely identified by its name.

The audit trail settings for an individual object can be changed directly in the *Audit trail settings* dialog box: select an object and click on the cell in the "Version tracking" or "Allow overwrite deleted" column to display a drop-down list from which the corresponding option can be set.

Press **<OK>** in the *Audit trail settings* dialog box to accept the new settings. A confirmation dialog box appears, warning for the consequences of changing the audit trail settings. Press **<OK>** to close the confirmation dialog box.



In case of pre-existing data, it is recommended to create a new empty database, create the necessary information fields, enable audit trails and import the existing data using the XML tools. This procedure ensures that the current version of each object in the database will be written to the audit trail.

3.6.3 The Audit trail window

The *Audit trail* window is a generic tool that provides an overview of the history of any database object that is included in the audit trail (see 3.6.2). When the *Audit trail* window is called from the *Main* window, the history of all database objects is displayed. The *Audit trail* window can also be called from a number of more specific windows (e.g. the *Entry* window, *Fingerprint processing* window, etc.), in which case only the history of the relevant object(s) is displayed. It offers an interface that allows users to:

- Display the history of each object, including its child and parent objects.
- Show deleted objects.
- Restore older versions of an object (an action that is audit trailed by itself).
- Obtain a readable report regarding the previous state of an object.
- Open historic versions of objects in their corresponding window (where applicable).

The latter two functions generally are complementary to each other: objects for which the readable report is not optimal can be opened in their own analysis window, while for objects that are already fully described in the readable report, the option to open them in their corresponding window will not be available.

In the *Main* window, select **File > View audit trail...** (🕒). The *Audit trail* window appears (see Figure 3.6.3).

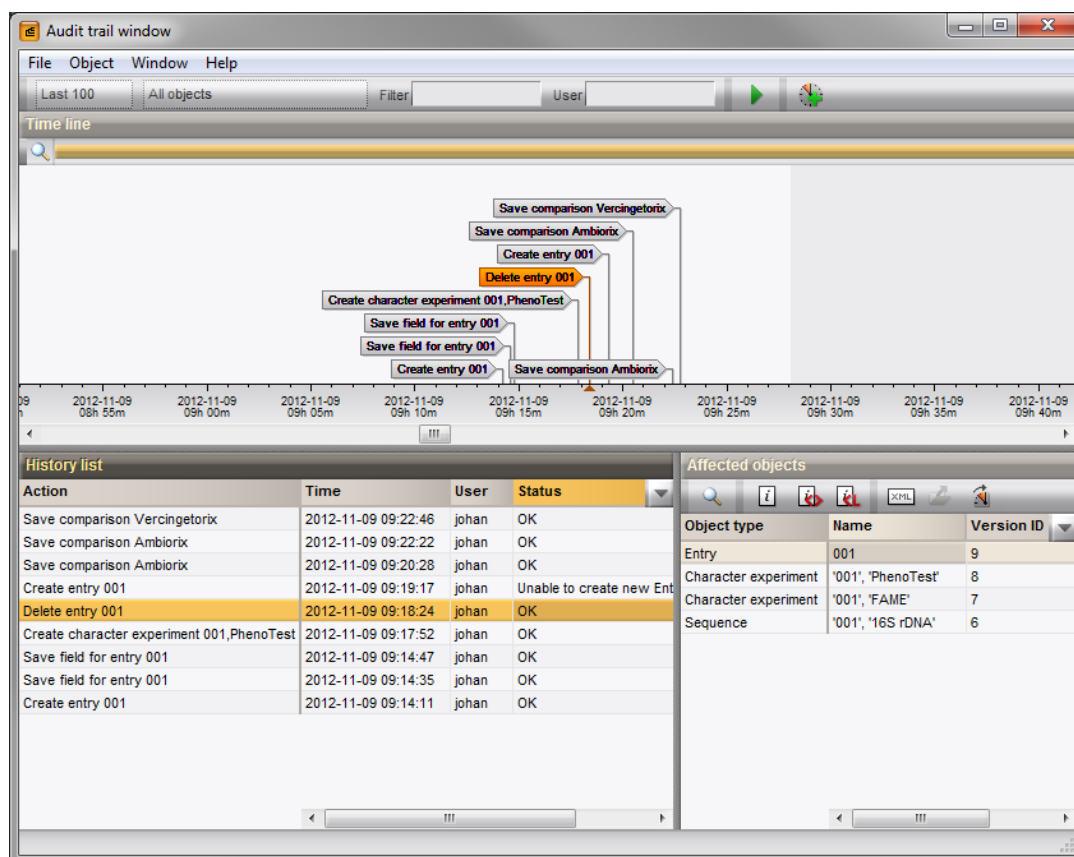


Figure 3.6.3: The *Audit trail* window, when called from the *Main* window.

The *Audit trail* window consists of three dockable panels: the *Time line* panel, *History list* panel and *Affected objects* panel.

In the *Time line* panel, the history of the database or of specific object(s) is visually represented on a time line based on logged actions. Using the zoom slider, it is possible to zoom in and out on the time line. When any of the flags in the time line is clicked on, the corresponding action becomes highlighted in the *History list* panel.


The *History list* panel shows the actions in the order in which they occurred, with the most recent actions on top. The top action in the history list therefore corresponds to the current state. In the "Action" column, a brief description of the action is given. Note that some user actions in fact consist of a sequence of actions. Furthermore, the time at which the action occurred ("Time") and the user who performed the action ("User") is logged. The "Status" column shows "OK" for actions that were performed without problems. In case an action was attempted, but could not be completed because of a database error, the corresponding error message is displayed.


An action can affect either no object (the action failed), a single object or a number of objects simultaneously. For the highlighted action in the *History list* panel, the affected objects are listed in the *Affected objects* panel (see Figure 3.6.3). For each object, the object type, name and version ID are displayed.







Obviously, the *Audit trail* window will be empty directly after switching on the audit trails. It is only after performing some actions in the audit trailed database, that the logged actions will appear in the *Audit trail* window.


When a database is audit trailed over a long period of time, the history list can grow very large. Therefore, a number of filtering options are available from the toolbar of the *Audit trail* window to filter out the history list according to different criteria.

By default, only the 100 most recent actions are shown when the *Audit trail* window is called. To display more actions, click on  and select **Last 1000** or **All** actions.


The history list can also be filtered according to affected object type using the  drop-down list, which lists all available object types.

In the **Filter** text box of the toolbar, a string can be entered that the program will search for in the "Action" field. When the window is refreshed with **File > Refresh** (, **Ctrl+R**), only those actions for which the search string occurs in the "Action" field will be displayed. To obtain a complete list of actions again, clear the **Filter** text box and select **File > Refresh** (, **Ctrl+R**).

To filter out the history list for actions performed by a certain user, enter the user name in the **User** text box and select **File > Refresh** (, **Ctrl+R**). In contrast to search terms entered in the **Filter** text box, substrings will not be matched: the entered user name needs to match exactly with the database user name. To obtain a complete list of actions again, clear the **User** text box and select **File > Refresh** (, **Ctrl+R**).

In addition to filtering actions based on affected *object types* (see above), a filtering can also be applied to affected *objects*. To do so, click on an object in the *Affected objects* panel and select **Object > Show object specific actions** (). Only the two actions on the comparison **Ambiorix** will be listed. To obtain a complete list of actions again, select **All objects** from the corresponding drop-down list in the toolbar of the *Audit trail* window.

Note that, when the *Audit trail* window is called from an object-specific window, the history list will be filtered by default and will only display actions specific for that object. The history of an object shows also actions undertaken on related objects, when the object was affected by these actions.

When audit trails are enabled in a database that already contains data, the original object versions are not included in the audit trail. The current version of a selected object can be written to the audit trail by selecting **Object > Write current version to audit trail...** (). The *Write to audit trail* dialog box appears.

In this dialog box, a comment can be entered. This comment will be displayed on the time line and in the "Action" field of the *History list* panel of the *Audit trail* window.

<OK> will write the current version of the object in the audit trail.

Several options are available to examine historic versions of objects.

An object version can be exported as XML file. This option is available for all object types when the object version is present in the audit trail. Select any object in the *Affected objects* panel and use **Object > Export object version as xml** (XML). A confirmation message appears, saying that the XML file is exported to the BioNumerics home directory. Press <OK> to open the XML file in its default program (e.g. Internet Explorer), as defined on your computer.

Furthermore, a readable report can be generated for an object version with **Object > Object version report > Full report** (F). An *Report* window pops up (see Figure 3.6.4).

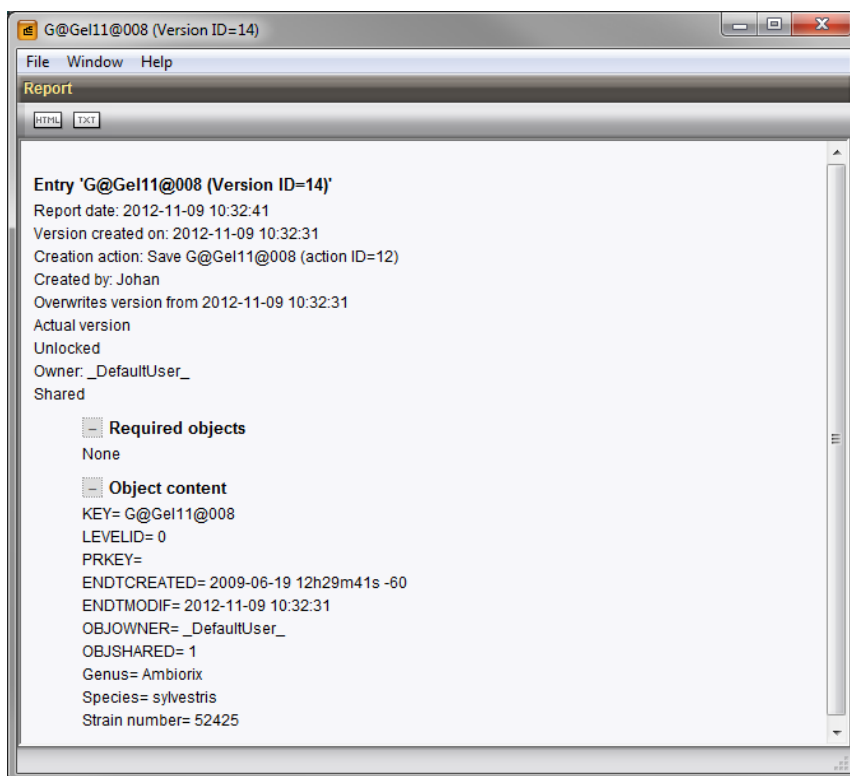


Figure 3.6.4: The *Report* window, displaying a full audit trail report for a historic version of a database entry.


From the *Report* window, the report can be exported as HTML with **File > Save as html** (HTML) or exported as text with **File > Save as text** (TEXT). In both cases, the exported file is saved in the BioNumerics home directory.

For different versions of the same object, full reports could be displayed alongside on the screen and compared with each other. However, other tools are available that make it easier to spot the differences between an object version and its previous version or the current version.

Click on an object in the *Affected objects* panel and select **Object > Object version report > Report differences with previous version** (P). In the *Report* window that pops up, only the differences with the previous version are listed. The information in this version of the object is indicated in green, the information from the previous version in red.

Similarly, select **Object > Object version report > Report differences with last version** (L) to pop up a *Report* window that reports the differences between this version and the current version of the object.

Although the report options discussed above are available for all object types, the generated reports might not be very informative for object types such as comparisons, gel files, sequences, etc.. These object types are easier to examine when they are opened in their own analysis window with **Object > Open object version** (O). The caption of the window shows the name of the object and the version ID.

Historic versions of an object can be restored as the actual working version in the database. This option can be used to undo unwanted changes to an object. Click on the object that you want to restore and select **Object > Restore object version...** . A message appears, warning for the consequences of this action and for the need to restart the software after restoring the object. Note that restoring an historic version of an object is an action that is audit trailed by itself. Historic child objects from the audit trail cannot be restored when they preceded the action on a parent object.



Objects cannot be restored in batch; each object should be restored individually.



The creation of objects cannot be undone. Once an object is created, the actions performed on the object remain logged in the audit trail.

3.6.4 Digital signatures

Digital signatures are an important component of the 21 CFR Part 11 regulations and are defined by the U.S. Food and Drug Administration (FDA) as "...electronic signatures based upon cryptographic methods of originator authentication, computed by using a set of rules and a set of parameters such that the identity of the signer and the integrity of the data can be verified", with an electronic signature meaning "...a computer data compilation of any symbol or series of symbols executed, adopted, or authorized by an individual to be the legally binding equivalent of the individual's handwritten signature". Similar to handwritten signatures that are used to approve paper documents, digital signatures can be used in BioNumerics to approve objects. It is important to realize that always the *current version* of an object is signed, not the object itself. Therefore, only objects that are audit trailed (and for which the current version is present in the audit trail) can be signed. When an attempt is made to sign an object that is not audit trailed, an error message will be generated.

To make sure that digital signatures cannot be removed from the signed object, transferred to other objects or repudiated, sophisticated cryptographic methods are used in BioNumerics to link a signature to the object content: an MD5 digest is made of the object content, encrypted with the signer's private key (can be decrypted with a public key for verification). BioNumerics maintains a "key ring", since it stores all private keys in the database, AES encrypted with the user password. Before a user can digitally sign objects, a signature key pair needs to be generated.

In the *Main* window, select **Database > Current user > Create signature key...** and confirm the action. A confirmation message appears, stating that the signature key pair has been created. The user can now sign objects.



Only users with the appropriate user privileges can sign objects.



When a user password is reset by an administrator, the signature key is automatically revoked and needs to be generated again.

To sign an object (e.g. a comparison), open it in its dedicated window and select **File > Object signatures...** to open the *Signature* window (see Figure 3.6.5).

This window shows the digital signatures for an object (if any). The signature contains the Version ID of the object, the User ID and User name of the signer, the Time of signing and the Comment that was entered. When the Version ID corresponds to the current version of the object, this is indicated with "=current". Similar to handwritten signatures, digital signatures cannot be removed once an object has been signed: they will stay listed in the *Signature* window of the object. Note that objects need to be signed individually; signing objects in batch is not possible.

Select **Signatures > Sign object...** . This pops up the *Sign object* dialog box (see Figure 3.6.6).

To avoid accidental signing, the user needs to re-enter his/her **User ID** and **Password** (see 3.5.3) each time an object version is signed. It is not possible for a user to sign under a different user name. When this is

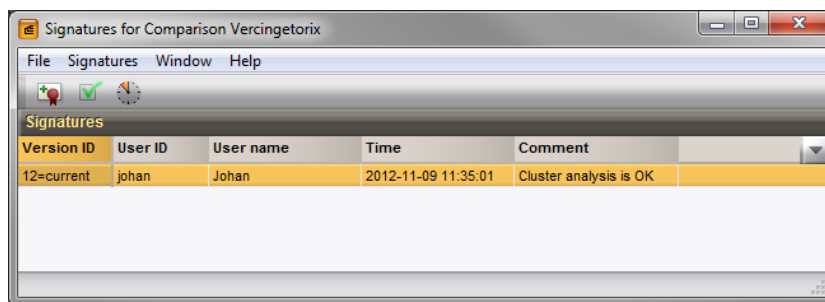


Figure 3.6.5: The *Signature* window for a comparison.

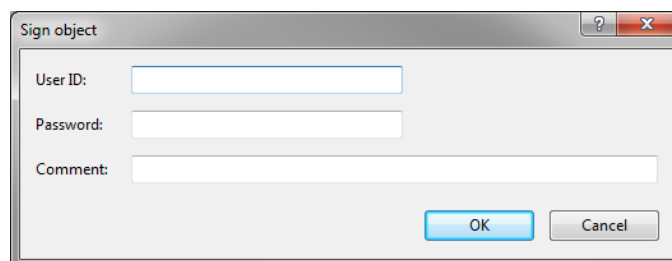


Figure 3.6.6: The *Sign object* dialog box.

attempted (or when the **User ID** is accidentally misspelled), the error message "This user is not consistent with the user that is currently logged in" will be generated. Optionally, a **Comment** can be entered with the signature.

Press <**OK**> to sign the object. The signature will now listed in the *Signature* window.



Only the current version of an object can be signed. When an historic version of an object is opened in its own analysis window (via the *Audit trail* window) and an attempt is made to sign this historic version, an error message will be generated.

Digital signatures can be verified by unencrypting them with the public key: Highlight the signature and select **Signatures > Verify selected signature...** (✓). This pops up the *Verify signature* dialog box (see Figure 3.6.7).

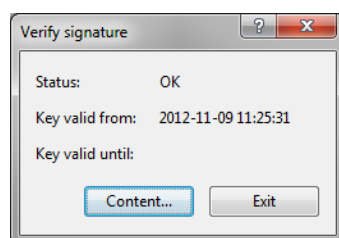


Figure 3.6.7: The *Verify signature* dialog box.

In this dialog box, **Status** displays either "OK" or a description of the problem with the signature. **Key valid from** shows the time when a signature key was generated for the user. Under normal circumstances, **Key valid until** will be blank. Only when the user's password has been reset, the time at which the signature key has been revoked will be displayed.

Press the <**Content**> button to export the object version's content to an XML file. A confirmation message appears, stating that the XML file is exported to the BioNumerics home directory.

Press <**OK**> to open the XML file in its default program (e.g. Internet Explorer), as defined on your com-

puter.

Press <**Exit**> to close the *Verify signature* dialog box.

Chapter 3.7

The BioNumerics relational database

3.7.1 Principles

Almost all data in a BioNumerics database is stored in a relational database. Options are available to store some of the data outside the relational database, in the *source files directory* (see 3.7.2 for more information). The *database directory*, typically located inside the BioNumerics home directory (see 3.1.2) contains in principle only settings files.

The connection to the relational database is made via an Open Database Connectivity (ODBC) driver (see Figure 3.7.1). The only exception is SQLite, to which BioNumerics connects directly i.e. without using any external program or driver.

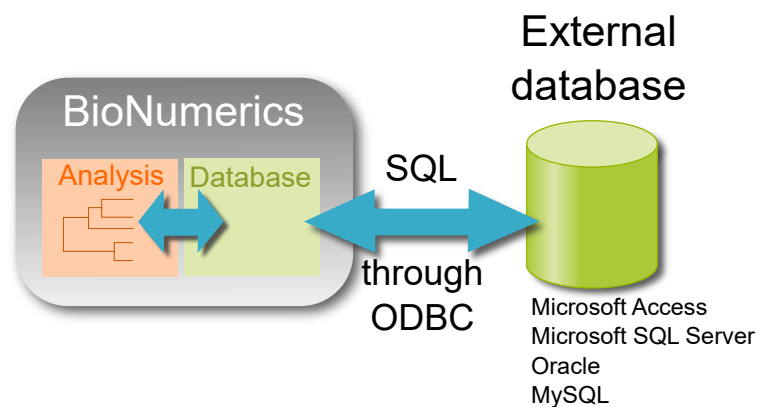


Figure 3.7.1: Database setup: all data is passed on to the relational database through ODBC.



Please note that BioNumerics 64-bit installations (see 1.1) require 64-bit versions of the ODBC drivers. BioNumerics 32-bit installations (e.g. all versions prior to version 7.5) require 32-bit ODBC drivers.

Currently supported relational database engines are Microsoft Access, SQLite (version 3.8.0 and higher), Microsoft SQL Server (version 2005, 2008 and 2012), Oracle (version 9i, 10g and 11g), and MySQL (version 5.x). Mentioned version numbers are under the condition that these products are still covered by vendor support. Other database engines than those listed above may work as well, but are not guaranteed to be fully compatible in a standard setup.

3.7.2 Configuring the relational database

In the *Main* window, you can set up a connection to a relational database, or configure an existing connection. In case the program reports database linkage problems when opening the database, you will need to use this configuration to create the required tables in the database.

Select **Database > Database settings...** to open the *Database settings* dialog box (see Figure 3.7.2).



Only users assigned to a user group that is allowed to **Modify database system settings**, can access the *Database settings* dialog box (see 3.5.2).

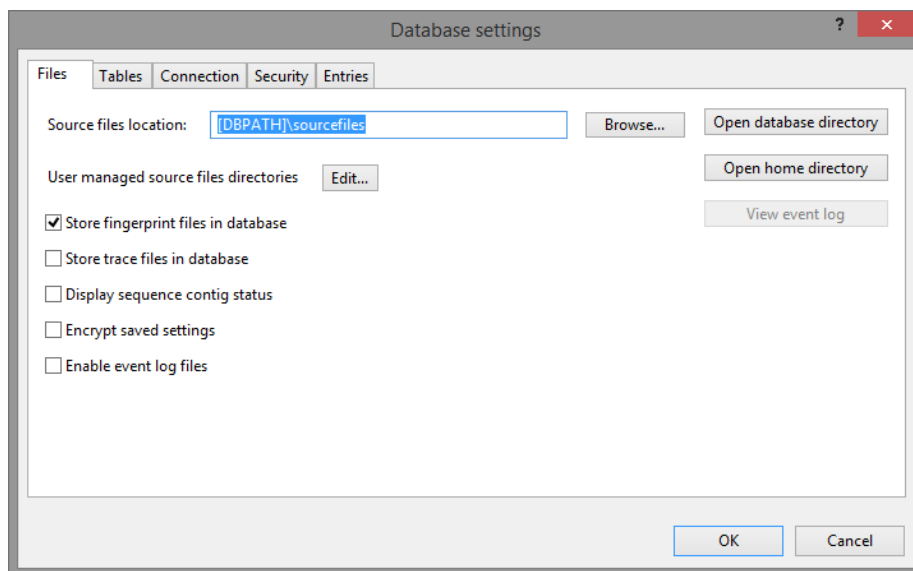


Figure 3.7.2: The *Database settings* dialog box, *Files* tab.

The *Database settings* dialog box consists of five different tabs, of which the *Files* tab is displayed by default.

In **Source files location**, the path for storing source files (e.g. TIFF images, .CRV curve files and/or sequence trace files) is entered. This location is only used when the information is not stored in the relational database itself. The default directory is denoted in a relative way as [DBPATH] \sourcefiles. [DBPATH] hereby refers to the database folder in the BioNumerics home directory (see 3.1.2). However, the path can be any local directory or network path, for example on a server computer. To change the path, press the **<Browse>** button to browse through your computer or the network. The database directory and the home directory can be opened in Windows explorer with the **<Open database directory>** and **<Open home directory>** buttons, respectively.

When the option **Store fingerprint files in database** is checked (the default setting), fingerprint files (TIFF files and .CRV files) are always saved as base64 encoded text in the relational database. When this option is unchecked, all newly imported fingerprint files will be stored in the source files directory.

With **Store trace files in database**, the user has the choice to store trace files from automated sequencers (four-channel sequence chromatogram files) either inside the relational database (option checked) or as files in the sourcefiles directory (option unchecked). In the former case, the trace files are stored as base64 encoded text in column DATA of table SEQTRACEFILES (see 21.1.44). In the latter case, a copy of the original trace file will be made automatically and the column DATA will hold a relative link to this file. The option **Store trace files in database** is by default disabled for newly created MS Access databases (because of its 2 Gigabyte size limit) and enabled for all other database management systems.

For contigs associated with sequences, it is possible to display the contig status by checking **Display sequence contig status**. When checked, the program shows the presence of a contig file as well as an "Ap-

proved” flag.

All the settings specified in .XDB file can be encrypted by checking the option *Encrypt saved settings*. The software needs to be restarted to encrypt the file.

With *Enable event log files*, it is possible to log events that alter following information:

- The database: the log file lists any changes in names of database fields, any entries that are added or deleted, and keys of entries that are changed. It also reports if new experiment types are created, if experiment types have been renamed or removed.
- The settings of the experiment types: for every change made, the kind of change is indicated. All settings are recorded in the log file, so that the user may restore the previous settings based upon the log file, if enabled.
- The data for the experiment types: if data for entries are changed, the log file lists these entries. It also mentions the creation of new experiments and the deletion of experiments.

For every BioNumerics session, the log files show the Windows user who has last made the changes together with the kind of changes and the date and hour. The log files can be viewed with the *<View event log>* button.



The audit trails and versioning tool (see 3.6) provides a more extensive mechanism for logging user actions and even for restoring of database information.

The *Tables* tab of the *Database settings* dialog box is shown in Figure 3.7.3.

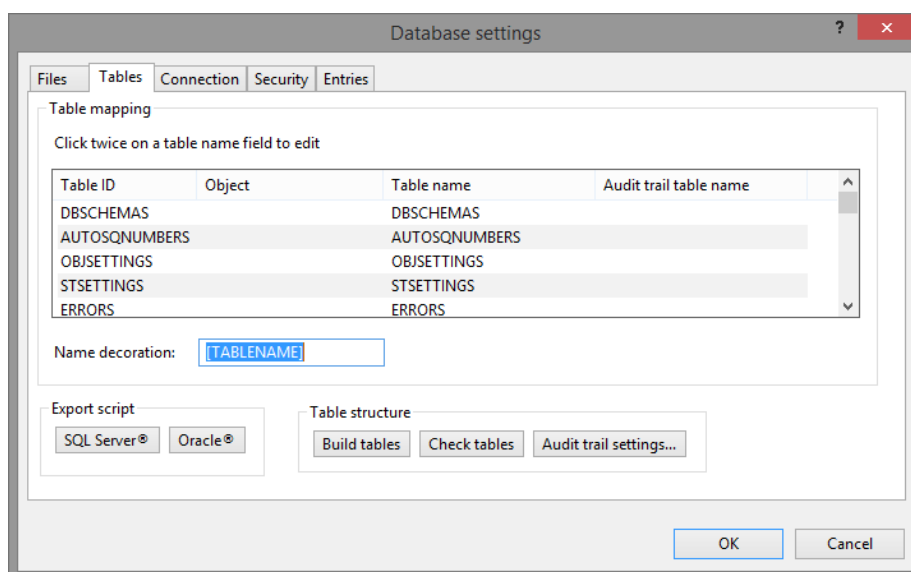


Figure 3.7.3: The *Database settings* dialog box, *Tables* tab.

In the *Tables* tab, all internal BioNumerics table identifiers are listed in the 'Table ID' column. The objects are listed in the 'Object' column (see 3.2). Each 'Table ID' is mapped to a default table ('Table name'). The default mapped audit trail tables are shown in the last column ('Audit trail table name'). Since BioNumerics assumes a certain table structure to store its different kinds of information, all mapped tables should contain a set of columns with fixed names. This table structure is described in detail in 21.1. In a database setup where BioNumerics is connected to an existing relational database, tables might be present with table names that do not correspond to the suggested BioNumerics tables names. In this case, it is necessary to edit the table name fields, by clicking twice on the field in the 'Table name' or 'Audit trail table name' column. The table names will appear highlighted and can be edited.

In a typical case, a number of information fields and/or experiment fields from the relational database will need to be linked to BioNumerics. However, these fields will occur in different tables having different field names. The obvious method in this case is to create *views* (referred to as *queries* in Microsoft Access) in the database.

- For those BioNumerics tables for which the relational database contains fields to be used, a view should be constructed in the database. Within that view, those database fields that contain information to be used by BioNumerics should be linked to the appropriate field.
- Finally, the database should be configured in such a way that the BioNumerics tables that contain fields already present in the database, be present either as table or as view, with all the recognized field names as outlined in 21.1. The names for the tables or views, however, can be freely chosen.
- Additional tables required by BioNumerics for which there are no fields available in the database can be created automatically by BioNumerics.



When views are created in the database to match the required BioNumerics tables, it is recommended to name the views using the standard BioNumerics names for the required tables. This will allow new users to log on to an existing connected database in the easiest way, by just defining the connection in the Startup program. By using different names, new users will have to specify the table/view names manually.



When entry information fields are obtained from a view (query) in the relational database, it will not be possible to define new entry information fields directly from BioNumerics. In that case, you will have to create the field in Oracle, SQL Server, MySQL, or Access, add it to the view, and reload the BioNumerics database. Note that the option **Automatically create entry fields for all columns in ENTRYTABLE** (see the description of the *Security tab*) should be checked in order for this procedure to work.



Views with joined columns may be read-only and it may not be possible to add new records to the database that are seen through these views (e.g. entries, experiments). It is possible to bypass this in Oracle or SQL Server using triggers.

If a *schema* is in use that is different from the default *schema* for the relational database user (dbo.[TABLENAME]), add the schema name to the **Table name decoration**, e.g. “myschema.[TABLENAME]”.

The buttons <**Check tables**> and <**Build tables**>, allow one to check if all required tables and fields are present in the relational database, and to automatically insert new tables and fields where necessary, respectively.



When pressing <**Build tables**>, BioNumerics will automatically create a new table for every required table that is not yet linked to an existing table in the database. For tables already linked, it will insert all required fields that do not yet exist in the database. In case you want to link BioNumerics to an existing database, this may cause a number of tables and fields to be created and cause irreversible database changes!

The button <**Audit trail settings...**> opens the *Audit trail settings* dialog box, which is discussed in 3.6.

The buttons <**SQL Server**[®]> and <**Oracle**[®]> under **Export script** can be used to generate an SQL script that will create the required tables and constraints in an SQL Server or Oracle database, respectively, when migrating from MS Access (see 3.7.3).

The *Connection tab* of the *Database settings* dialog box is shown in Figure 3.7.4.



When switching to the *Connection tab*, a warning message pops up to emphasize the importance of the settings listed here.

The **ODBC connection string** is defined in the text area on the left-hand side. The same string can be found in the *connection description file*, on the second line next to “ConnectionString”.

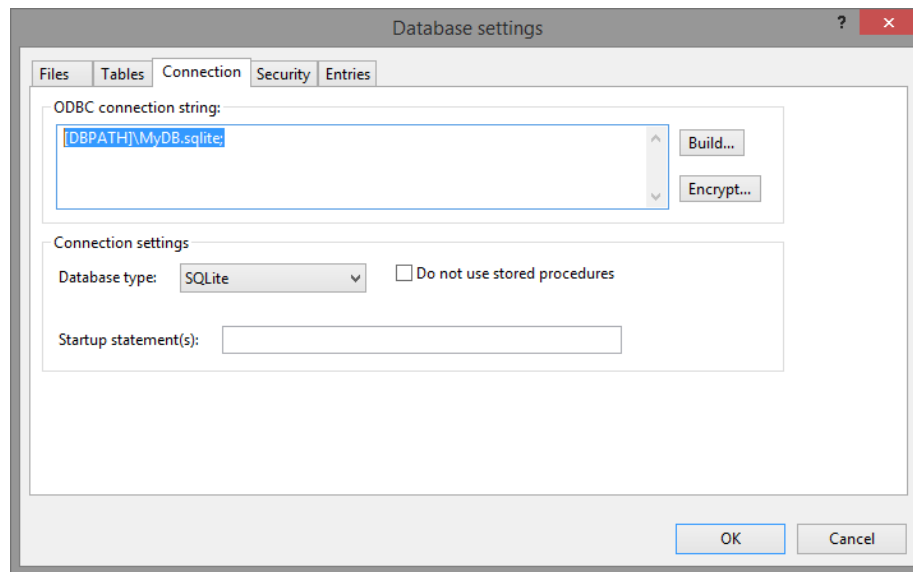


Figure 3.7.4: The *Database settings* dialog box, *Connection* tab.



When BioNumerics has created a new database via the *BioNumerics Startup* window, the *connection description file* is named **dbname*.XDB* by default. The default directory for the .XDB file is [HOMEDIR] *dbname*. The [HOMEDIR] tag thereby points to the home directory as defined in the *BioNumerics Startup* window (see 3.1.2).

The **<Build>** button allows a new connection string to be defined. This will call the *Connect to existing database* dialog box as discussed in 3.1.3.

With the **<Encrypt>** button, the ODBC connection string can be encrypted. Encrypting the connection string will make the connection string permanently invisible in the .XDB file and in the *Database settings* dialog box. The software will ask for confirmation before encrypting the connection string.

The database management software (DBMS) can be selected from the **Database type** drop-down list. Options are *SQL Server*[®], *SQLite*, *Access*[®], *Oracle*[®], and *MySQL*[®]. This information is written next to "DatabaseType" in the connection description file.

The option **Do not use stored procedures** should be checked in case the DBMS you are using does not support stored procedures, for example when connecting to a PostgreSQL database. Stored procedures (called "Sequences" in Oracle) are used in BioNumerics for auto-numbering of database objects.

In the **Startup statement(s)** text box, one or more SQL statement(s) can be entered that will be executed before the ODBC connection is opened. This can be used e.g. to set the SQL mode when working with a MySQL database.

The *Security* tab of the *Database settings* dialog box is shown in Figure 3.7.5.

Use the **Password validity time** text box to set an expiration date for passwords (see 3.5.3): passwords assigned to users in the database will only be valid for the period specified. The default value of "999" days means that passwords are valid indefinitely.

Security can be achieved when user passwords are enforced in the database (see 3.5.3). Check **Require passwords** to enforce a standard user password of at least 4 characters long. Users having no password in the database, are prompted to specify a password when logging on to the database. Improved security can be achieved with *strong* user passwords. To enforce the usage of strong passwords in the database, check **Require strong passwords**. A strong user password in BioNumerics should be at least 8 characters long, including at least two alphabetic characters, and should have a complexity score of 20. The complexity score is calculated using following scoring rules: new alphabetic character: **+2**; alphabetic character: **+1**; new

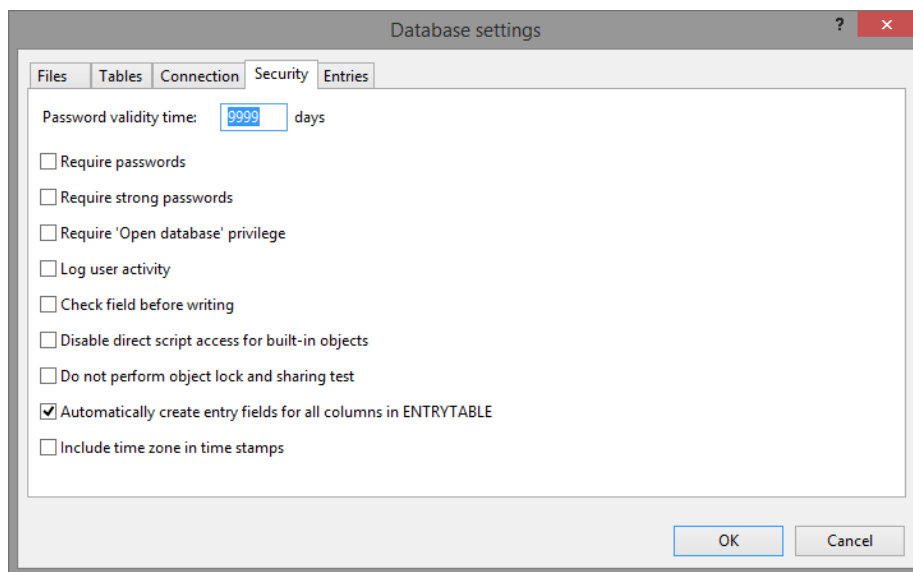


Figure 3.7.5: The *Database settings* dialog box, *Security* tab.

non-alphabetic character (e.g. numeric characters, %, ?): **+4** points; non-alphabetic character: **+2**. When the option is checked, all *new* passwords specified in the database, have to meet the strong password criteria. Passwords that were stored in the database *before* the option ***Require strong passwords*** was checked, and that do not meet the strong password criteria, can still be used in the database. Users having no password in the database, are prompted to specify a strong password when logging on to the database.

To start recording user activity actions in the database (see 3.5.4), check ***Log user activity***.

When ***Check field before writing*** is checked, the software will perform a column presence, data type and data type size check on the relational database before each writing operation. This option should rarely be used and is mainly intended for debugging purposes.

When ***Disable direct script access for built-in objects*** is checked, a number of general script functions ("Connected databases" > "Interface" > "Other tables") that act directly on the relational database are disallowed.

The option ***Do not perform object lock and sharing test*** can be (temporarily) checked to speed up large import tasks in the database.

When ***Automatically create entry fields for all columns in ENTRYTABLE*** is checked, columns that are added via the DBMS to the ENTRYTABLE table (or view) will be automatically picked up by BioNumerics as entry fields.

Check ***Include time zone in time stamps*** to add an indication of the time zone anywhere time stamps are used in the software (e.g. for the 'Created date' and 'Modified date' fields).

The *Entries* tab of the *Database settings* dialog box is shown in Figure 3.7.6.

This tab allows you to specify a *template* that will be used when entry keys are automatically generated. Plain text and dynamic components (tokens) can be combined in such a template, which is displayed in the upper text box. Following tokens are available:

- ***Database user*** ([DbUser]): The BioNumerics database user ID.
- ***OS user*** ([OSUser]): The user name of the currently logged-on Windows user.
- ***Database ID*** ([DbaseID]): The database ID. This defaults to the database name, but can be overridden via the ***Database ID*** text box.

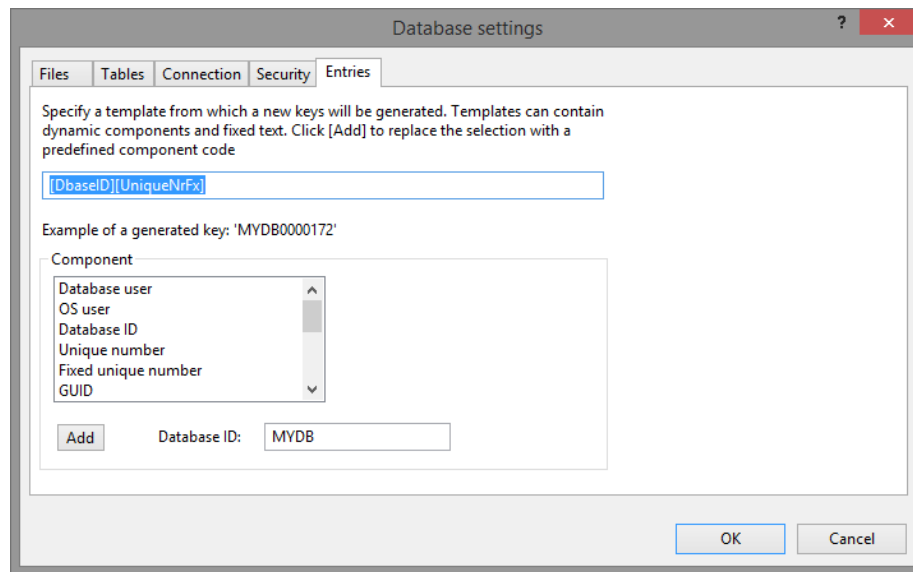


Figure 3.7.6: The *Database settings* dialog box, *Entries* tab.

- **Unique number** ([UniqueNr]): A unique consecutive number, starting at 1.
- **Fixed unique number** ([UniqueNrFx]): A unique number with a fixed length of 7 digits, i.e. ranging from 1 to 9,999,999.
- **GUID** ([Guid]): A globally unique identifier (ensures uniqueness between databases).
- **Short GUID** ([GuidShort]): A short globally unique identifier (ensures uniqueness between databases).
- **Current date** ([CurrentDate]): The current date in the format YYYY-MM-DD.
- **Current year** ([CurrentYear]): The current year in the format YYYY.
- **Current month** ([CurrentMonth]): The current month in the format MM.
- **Current day** ([CurrentDay]): The current day in the format DD.
- **Current DateTime** ([CurrentDateTime]): Time stamp in the format YYYY-MM-DD hh:mm:ss -mm. Whether the time zone is added, depends on the **Include time zone in time stamps** setting (see higher).



A preview with an example of a generated key is provided for convenience. However, this preview will not be automatically updated when the **Database ID** is changed.

Entry keys in a BioNumerics database should be unique and have a maximum length of 80 characters. The software will generate a warning message if the created key template generates keys that do not fulfill both criteria.



The tokens [UniqueNr] and [UniqueNrFx] generate sequential numbers, of which the last-used is stored in the relational database: it corresponds to the value in column "SQVALUE" for record "_AUTONKEY" in table "AUTOSQNUMBERS" (see 21.1 for more information about the relational database table structure). If you want the auto-numbering to start at a certain value, the value in this field should be adjusted accordingly.

3.7.3 Data migration

3.7.3.1 Introduction

When a new BioNumerics database is created (see 3.1.3), the relational database to which it connects (and in which the bulk of the data is stored) needs to be specified in the *New database* wizard. Obviously, this relational database engine will continued to be used under normal conditions. Due to technical reasons, it may occur that a different database management software (DBMS) needs to be employed at some point. To accommodate for this situation, a migration procedure is available to transfer the *complete* database to a new empty BioNumerics database, which could use a different DBMS.



A relatively common scenario in which data need to be migrated from one relational database to another is with MS Access databases. When the size of the .MDB (or .ACCDB) data file approaches its 2 Gigabyte size limit, it is recommended to migrate to a DBMS with a higher storage capacity.



For transfer of parts of the data stored in a BioNumerics database, we refer to 3.4.

The migration process consists of following steps:

1. Ensure that the connection description file (.XDB file) of the source database is up-to-date. This can be done by opening the *Database settings* dialog box for the source database (see 3.7.2) and pressing the **<OK>** button in this dialog.
2. Create a new BioNumerics database from the *BioNumerics Startup* window (see 3.1.3). This destination database needs to have the required BioNumerics table structure, but should **not** contain any data and no BioNumerics plugins should be installed.
3. Copy the content of the source database to the destination database, either via a script available from our website (see 3.7.3.2) or by using the `Migration.exe` command line tool (see 3.7.3.3).
4. Perform an auto build tables action (see 3.7.2) in the destination database to ensure that it is in a coherent state (for Oracle databases only).

3.7.3.2 Migrating data with the MigrationGUI script

A script is available from the Applied Maths website to transfer all data from a source database to a destination database, regardless of which DBMS is used to store the data in. The script should be run in the destination database and simply provides a user interface for the `migration.exe` command line tool (see 3.7.3.3).

Select **Scripts** > **Browse internet...** and browse for the **Miscellaneous tools** category.

Click on **Migrate all data from another database** to run the script. The *Select original database* dialog box appears (see Figure 3.7.7).

By default, **Home directory** shows the path to the current BioNumerics home directory. A different directory can be selected via the **<Browse>** button.

The original database, i.e. the source database containing the data to be migrated, should be selected from the **Database** list. This list contains all databases in the **Home directory**, as specified above.

Pressing **<OK>** will start the migration. Depending on the size of the database and speed of connection, this process might take several minutes.



The source database should be empty. If this is not the case, the error message "Please start with an empty database" is generated and the script stops.

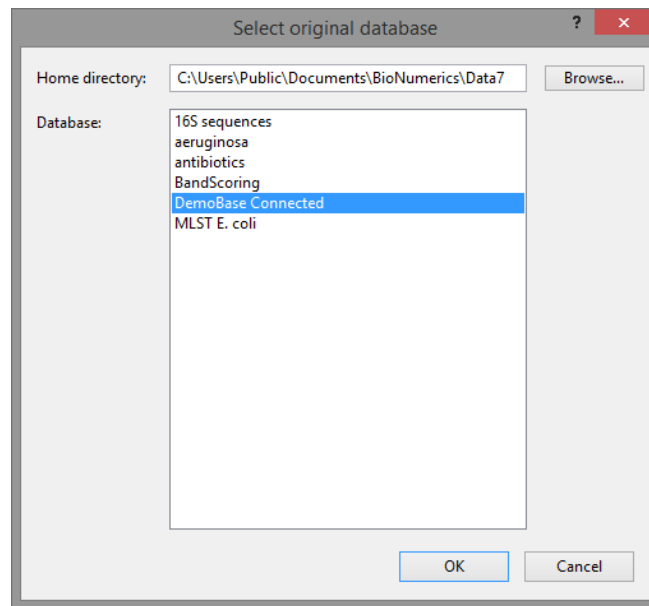


Figure 3.7.7: The *Select original database* dialog box.

When the migration is completed, the database will automatically close and re-open with the data that were migrated from the source database. A log file called `migration.log` is generated in the directory of the destination database.

3.7.3.3 The Migration.exe command line tool

The BioNumerics installation directory (see 1.3) contains a standalone command line tool called `Migration.exe`, which can be used to migrate all data from a source database to a destination database. Both source and destination database should use a supported DBMS, such as SQLite, MS SQL Server, Oracle or MySQL (see 3.7.1 for more information).

Table 3.7.1 lists all available options for the `Migration.exe` command line tool.

Option	Description
<code>-sourceHome</code>	[Required] The home directory (see 3.1.2) of the source database.
<code>-sourceDb</code>	[Required] The name of the source database, as shown in the <i>BioNumerics Startup</i> window.
<code>-destHome</code>	[Optional] The home directory of the destination database. If this option is not specified, then the home directory of the source database is used.
<code>-destDb</code>	[Required] The name of the destination database, as shown in the <i>BioNumerics Startup</i> window.
<code>-log</code>	[Optional] File to write logging messages to.
<code>-help</code>	[Optional] Shows a help message.

Table 3.7.1: Command line options for `Migration.exe`.

Example: the command below will migrate a BioNumerics database called “MyOldDB” that uses MS Access to a new database (called “MyNewDB”) that resides in the same home directory.

```
Migration.exe -sourceHome="C:\Users\Public\Documents\BioNumerics\Data" -sourceDb="MyOldDB"
-destDb="MyNewDB"
```



For Windows PowerShell, start any command line with `".\ "`. For example, “`Migration -help`” in a command prompt becomes `".\Migration -help`” in PowerShell.



In case a file path contains one or more spaces, it should be enclosed with double quotes in the Windows command prompt or PowerShell.

The `Migration.exe` tool will copy all database records via ODBC. Tables that are processed are those from the connection description file. If a table or column is not present in the destination database, it will be created. In addition, the tool will copy all files from the source database folder to the destination database folder (except for .MDB or .XDB files). `Migration.exe` can be interrupted and will restart at the point where it was stopped. To this end, it writes the migrated tables in a text file called `MigrationProgress.txt` in the `Conversion` sub-folder of the destination database directory.

3.7.4 BioNumerics 64-bit versions and MS Access

The 64-bit version of BioNumerics is *not* compatible with MS Access. The reason is that Microsoft does not support the 64-bit MS Access database engine on systems where a 32-bit MS Office version is installed (i.e. the majority of the Windows computers).

When creating a new database in BioNumerics 64-bit, the option *Use MS Access® based* will be grayed out in the *Database engine* wizard page (see 3.1.3). When an existing database that connects to an MS Access relational database (i.e. a database that was created earlier with a 32-bit version of BioNumerics) is opened in BioNumerics 64-bit, the *Confirm Access Migration* dialog box will appear (see Figure 3.7.8).

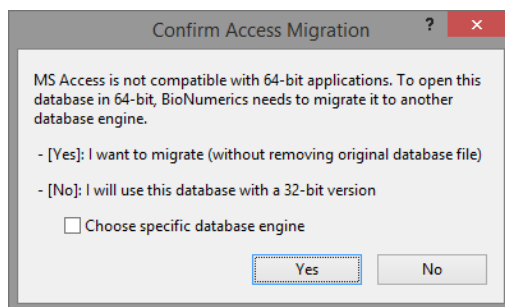


Figure 3.7.8: The *Confirm Access Migration* dialog box.

This dialog box indicates that the 64-bit version of BioNumerics (see 1.1) is not compatible with the MS Access database in which the data are stored. Two options are offered:

- Pressing **<No>** will abort the current action, i.e. the database will not be opened and no modifications of any kind will be made to the database.
- Pressing **<Yes>** will first migrate all data from the MS Access database to a new database (using `migration.exe`, see 3.7.3.3), create a new connection description file (see 3.7.2) that points to the new database and then open the database. When the option *Choose specific database engine* is not checked, a SQLite database will be created. If *Choose specific database engine* is checked, pressing **<Yes>** will open the *Choose migration settings* dialog box (see Figure 3.7.9).

This dialog allows you to select the type of relational database: either the default **SQLite** option (always available and recommended), or the **MS SQL Server Express** database engine (only available when installed on with an earlier BioNumerics version and when the home directory is on a local drive).

Pressing **<OK>** will migrate the data from MS Access to the selected database engine, adapt the connection description file and open the database in BioNumerics.

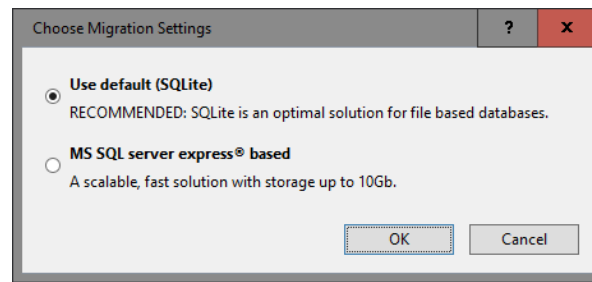


Figure 3.7.9: The *Choose migration settings* dialog box.

3.7.5 Protecting databases at the DBMS level

3.7.5.1 Introduction

To prevent users to open/edit a relational database with the BioNumerics software, passwords can be defined in BioNumerics and actions in the software can be restricted to certain users. Detailed information can be found in [3.5](#).



A database implementation that is fully compliant with the 21 CFR Part 11 regulations not only requires an appropriate setup of the available tools in BioNumerics, such as user management ([3.5](#)), the audit trail ([3.6](#)), digital signatures ([3.6.4](#)) and encryption of the connection string and connection description file (see [3.7.2](#)), but also adequate measures to prevent direct access to the relational database.

Another way of protecting information present in a relational database used by BioNumerics, depends on the protection and security measures provided by the DBMS. The procedure to follow depends on the DBMS used.

3.7.5.2 Access databases

To password-protect an MS Access database that is used by BioNumerics, proceed as follows:

Open MS Access and select the "Open" command in the menu of Access. In the *Open file dialog box*, navigate to the database. Click the arrow to the right of the Open button and choose the option "Open Exclusive" (see [Figure 3.7.10](#)).

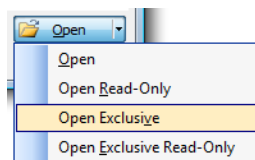


Figure 3.7.10: Open MS Access database for exclusive use.

If you are using MS Access 2000 or 2003, go to the "Tools" menu, and select **Security option > Set database password**. If MS Access 2007 is installed on your computer, select the "Database Tools" tab and select **Set Database Password**. In MS Access 2010, select the "File" tab and select **Set Database Password**. A dialog box pops up, asking you to enter and confirm your password (see [Figure 3.7.11](#)).

If you close the database in MS Access and open the database in BioNumerics, the program will prompt you for the specified password before loading the database.

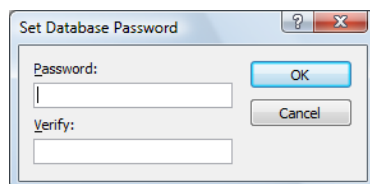


Figure 3.7.11: The *Set Database Password* dialog box in MS Access.

3.7.5.3 Other relational databases

For all other relational databases (SQL Server, MySQL, Oracle, . . .), a user name and password are required upon creation of the database. The reason why BioNumerics does not prompt for it when loading the database, is because the password is saved in the ODBC connection string.



SQL Server can also use integrated Windows authentication, which is the case when a SQL Server Express database is created. In this case, the user name and password is not stored in the connection string.


In the *Connection tab* of the *Database settings* dialog box (see 3.7.2), the line "PWD=*password*" holds the password. If you want BioNumerics to prompt for a user name and a password each time you open the database, delete the line "PWD=*password*" in the ODBC connection string. If you want a specific user name to be filled in the user name box, change the user name after "UID=".

Chapter 3.8

BioNumerics process templates

3.8.1 The Process data dialog

Processing workflows are available in the BioNumerics *Process data* dialog box, facilitating the analysis of data in BioNumerics by confronting the user with less dialogs.

Selecting **File > Process...** () in the *Main* window calls the *Process data* dialog box (see Figure 3.8.1). The process tree options are organized in groups based upon the type of data, with one exception, the **Comparison** option. By default, all groups are collapsed. A group can be expanded by clicking on the "+" sign next to the name of the group. When a particular process option is selected in the tree, a short description appears in the right panel.

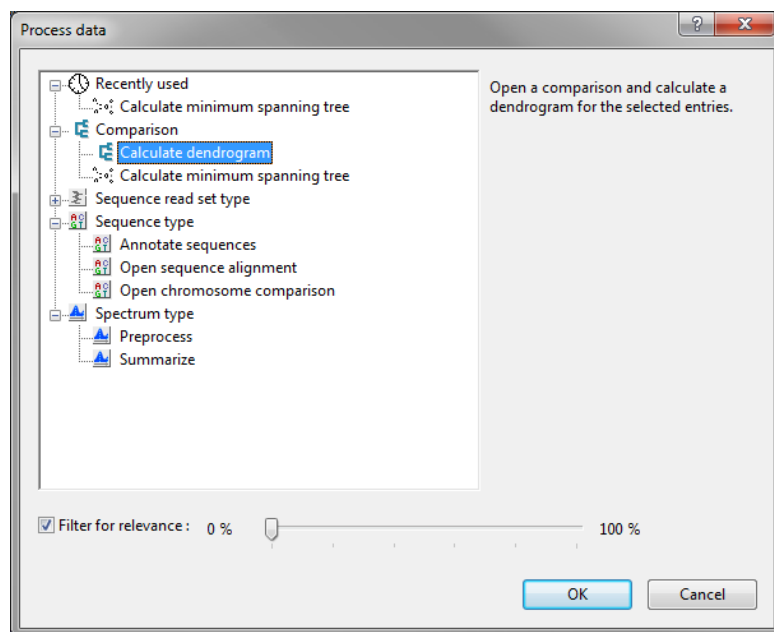


Figure 3.8.1: The *Process data* dialog box.

All recently used process options are grouped under **Recently used**. This group appears at the top of the tree. The last used process option is automatically selected when the *Process data* dialog box is called.



After installation of certain plugins (e.g. *HIV resistance plugin*, *MLVA plugin*, etc.), additional process options are injected in the *Process data* dialog box.

3.8.2 Process templates

3.8.2.1 Comparison

3.8.2.1.1 Calculate dendrogram

Calculate dendrogram: This option opens the *Comparison* window containing all selected entries and calculates a dendrogram using the comparison settings saved for the selected experiment. The **Experiment type** and optionally the **Groups** need to be selected by the user in the *Open comparison* dialog box (see Figure 3.8.2).

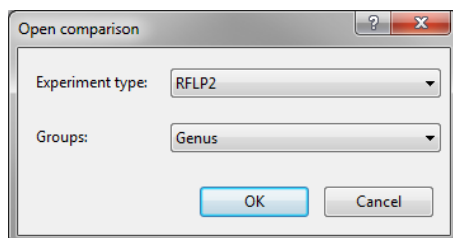


Figure 3.8.2: The *Open comparison* dialog box.

3.8.2.1.2 Calculate minimum spanning tree

Calculate minimum spanning tree: This option opens the *Comparison* window containing all selected entries and calculates a minimum spanning tree for the selected experiment. The input data is treated as categorical data and two priority rules (SLV and DLV) are applied. The **Experiment type** and optionally the **Groups** need to be selected by the user in the *Open comparison* dialog box (see Figure 3.8.3).

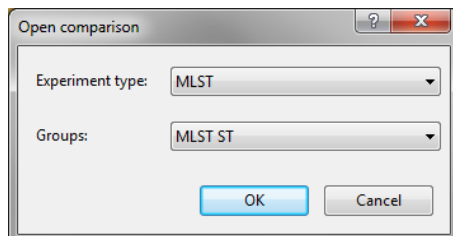


Figure 3.8.3: The *Open comparison* dialog box.

3.8.2.2 Sequence read set type

3.8.2.2.1 Chimera detection

Chimera detection: This option launches a chimera detection analysis, removing all chimeric sequences in the selected data set. The **Input** and **Output** experiment types and chimera detection settings need to be specified by the user.

3.8.2.2.2 Calculate keyword profile

Calculate keyword profile: This action calculates the key word profiles of the sequence read sets of the selected entries. The **Input** and **Output** experiment types and **Keyword length** need to be specified by the

user.


3.8.2.2.3 De novo assembly

De novo assembly: With this option, a de novo assembly is performed on the selected entries. The *De novo assembly* wizard is called where the de novo assembly settings can be specified in the different steps of the wizard.

3.8.2.2.4 Demultiplexing

Demultiplexing: This options calls the *Demultiplexing* dialog box where the settings can be specified for demultiplexing analysis. This analysis splits up the data set according to the multiplex identifiers contained in the reads of the selected entries.

3.8.2.2.5 Global statistics

Global statistics: This action calculates global statistics on the sequences in the selected sequence read data set. The experiment type containing the data needs to be selected by the user. The analysis is added to the list in the *Analyses* panel of the *Sequence read set experiment* window. The selected analysis can be opened in a separate window by selecting **File > Open selected analyses** () or double-clicking on its name.

3.8.2.2.6 Identify against taxonomic database

Identify against taxonomic database: This action identifies all sequences of a sample against a taxonomic database, and stores the abundances as characters in a character type experiment in the database. The *Identify against taxonomic database* wizard is called where the settings can be specified in the different steps of the wizard.

3.8.2.2.7 Primer removal

Primer removal: This action removes the primers of all selected sequences and only saves the trimmed reads to the output sequence read set in the database. The **Input** and **Output** experiment types and primers to be trimmed off need to be selected by the user.

3.8.2.2.8 Resequencing assembly

Resequencing assembly: With this option a resequencing assembly is performed on the selected entries. The *Resequencing assembly* wizard is called where the resequencing settings can be specified in the different steps of the wizard.

3.8.2.2.9 Map to reference

Map to reference: This action will map all the reads of the selected entries against a reference. The settings need to be specified by the user in the *Map to reference* dialog box.

3.8.2.2.10 Sequence selection

Sequence selection: This option performs sequence selection based on certain user defined criteria such as start and end position, and based on minimum and maximum length. The sequence selection functionality is based on the mothur command screen.seqs [35].

3.8.2.2.11 Single-sample diversity analysis

Single-sample diversity analysis: In this analysis, the alpha-diversity of a single sample is assessed. For the operational taxonomic units (OTUs) obtained, the within-sample diversity, the community evenness, the community richness and the community diversity indices are calculated.

For this analysis, BioNumerics makes use of the mothur [35] project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). BioNumerics uses the flexibility of the algorithms incorporated in mothur and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results.

3.8.2.2.12 Split paired-end reads

Split paired-end reads: This preprocessing action takes read sequences from Roche/454[®] paired-end runs that were imported as single-end reads, and creates paired-end read sequences. The split position is determined by aligning the adapter sequence to the read. When the adapter does not match any part of the read sequence, the sequence is left as a whole. If the adapter matches but is too close to the beginning and/or the end of the read sequence, only the parts that are long enough are retained, thus yielding a single-end read sequence, or no sequence at all. Sequence qualities are split accordingly.

3.8.2.2.13 Trimming

Trimming: This action filters out sequences based on criteria such as sequence content and read quality. The *Input* and *Output* experiment types and trimming settings need to be selected by the user.

3.8.2.3 Sequence type

3.8.2.3.1 Annotate sequences

Annotate sequences: Using this option, all selected sequences are annotated based on the annotations that are saved with the selected reference sequence(s). The sequence *Experiment* type and the *Reference(s)* need to be specified by the user in the *Annotate sequences* dialog box (see Figure 3.8.4).

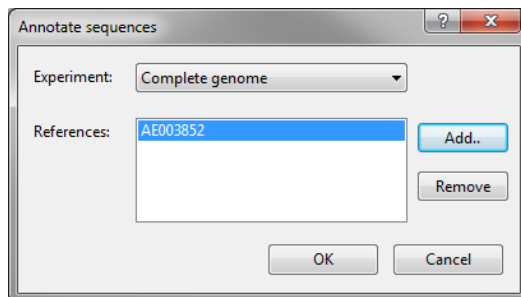


Figure 3.8.4: The *Annotate sequences* dialog box.

3.8.2.3.2 Open sequence alignment

Open sequence alignment: This option opens the *Sequence alignment* window containing all selected entries and calculates a sequence alignment for the selected experiment. The **Experiment type** needs to be selected by the user in the *Choose sequence experiment type* dialog box (see Figure 3.8.5).

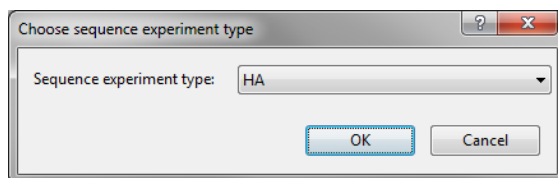


Figure 3.8.5: The *Choose sequence experiment type* dialog box.

3.8.2.3.3 Open chromosome comparison

Open chromosome comparison: This option opens the *Chromosome Comparison* window containing all selected entries. The *Experiment types* dialog box displays a list of available sequence types and the number of associated entries. From this list, the user needs to select the experiment type(s) that should be included in the chromosome comparison project.

3.8.2.4 Spectrum type

3.8.2.4.1 Preprocess

Preprocess: With this option, spectra for the selected entries can be preprocessed according to one of the three predefined templates, or based on a custom template (see Figure 3.8.6). The preprocessing workflow is performed in the background.

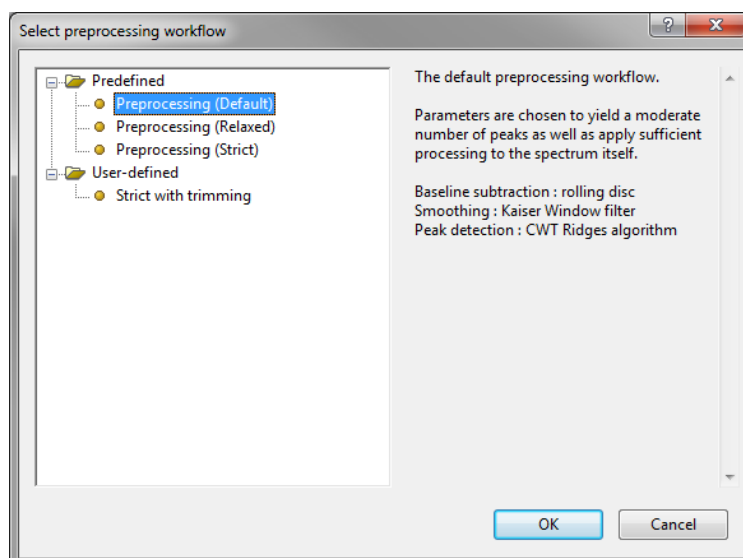


Figure 3.8.6: The *Select preprocessing workflow* dialog box.

3.8.2.4.2 Summarize

Summarize: With this option, spectra for the selected entries can be summarized. The settings need to be

specified in the *Create summary spectra* wizard wizard (see [5.3.1](#) for more information about this wizard).

Part 4

Fingerprint types

Chapter 4.1

Setting up fingerprint type experiments

4.1.1 Introduction

A fingerprint type in BioNumerics is in principle any type of experiment that generates a one-dimensional densitometric profile on which peaks or bands can be assigned (see 2.1.2). This includes "traditional" gel electrophoresis techniques (e.g. on agarose or polyacrylamide slab gels) as well as capillary electrophoresis (e.g. on genetic analyzers or similar equipment). While the experiment type is the same for both, the preprocessing steps are rather different. In gel electrophoresis, *lanes* need to be defined on 2-dimensional gel images. This is clearly not the case for capillary electrophoresis where *electropherograms* are the raw output generated by the equipment. Furthermore, the *normalization* process is different: Typically, gel electrophoresis relies on *external reference markers*, loaded in separate lanes on each gel. For capillary electrophoresis, *internal size markers* carrying a different fluorescent dye are being used. These differences make that each type has its own specific workflow and requires a dedicated processing environment for convenient handling of the data. In this manual, the processing of gel images is explained in 4.1.3, while the workflow for multi-dye capillary electrophoresis patterns is discussed in 4.1.4.

4.1.2 Creating a new fingerprint type

To create a new fingerprint type, highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** (📄). In the *Create a new experiment type* dialog box, click on **New fingerprint type** and press <OK>. This will display the first step of the *New fingerprint type* wizard (see Figure 4.1.1).



To be able to work with fingerprint types, the Fingerprint data module (FP) needs to be present in your BioNumerics configuration.

The wizard prompts you to enter a **Fingerprint type name**. Enter a name for the new fingerprint type and press <Next> to continue to the next step (see Figure 4.1.2).

The type of fingerprint data files needs to be specified, either **Two-dimensional TIFF files** (i.e. gel photographs) or **Densitometric curves** (typically profiles generated on an automated sequencer).

Additionally, the optical density (OD) can be specified as **8-bit (256 values)**, **12-bit (4096 values)**, **16-bit (65536 values)** or **Other**. In the latter case, a bit value should be entered (e.g. "10" will result in $2^{10} = 1024$ possible OD values). A value should be provided that corresponds to the files at hand.



Any of the parameters specified in the *New fingerprint type* wizard can be adjusted later on in the fingerprint type settings (see 4.1.5.3).

Pressing <Next> will display the third step of the *New fingerprint type* wizard (see Figure 4.1.3).

In this step, the wizard asks whether the fingerprints have inverted densitometric values. This is the case

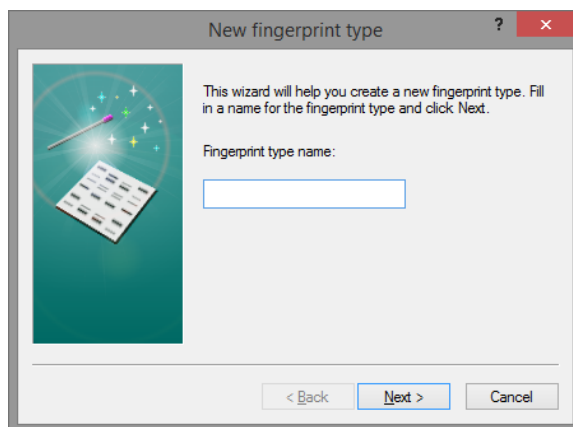


Figure 4.1.1: The first step of the *New fingerprint type* wizard.

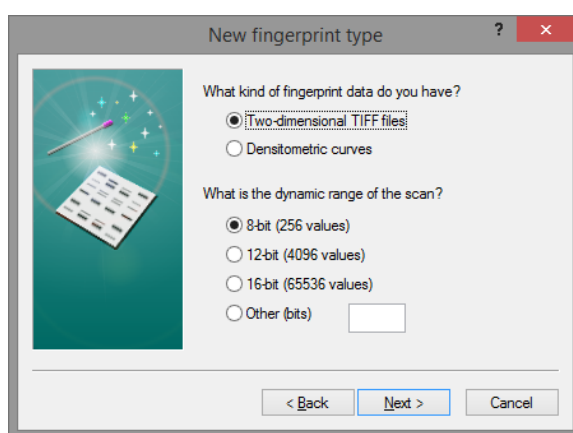


Figure 4.1.2: Step 2 of the *New fingerprint type* wizard.

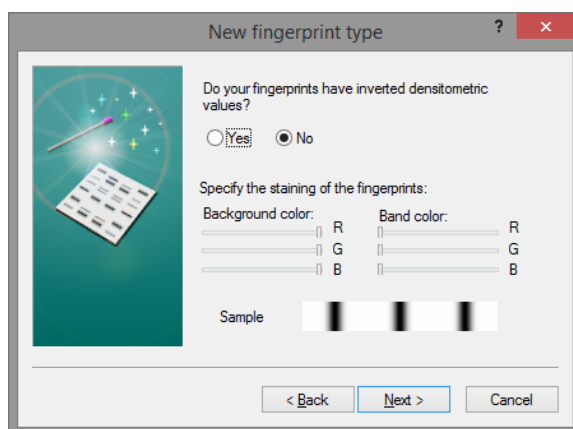


Figure 4.1.3: Step 3 of the *New fingerprint type* wizard.

when you are using e.g. ethidium bromide stained gels, photographed under UV light. The bands then appear as fluorescent lighting on a black background. Since BioNumerics recognizes the darkness as the intensity of a band, answer **<Yes>** to allow the program to automatically invert the values when converting the images to densitometric curves.

Furthermore, the wizard allows you to adjust the color of the background and the bands to match the reality. The red, green and blue components can be adjusted individually for both the background color and the

band color. Usually, you can leave the colors unaltered.

Pressing <*Next*> will display the fourth and final step of the *New fingerprint type* wizard (see Figure 4.1.4).

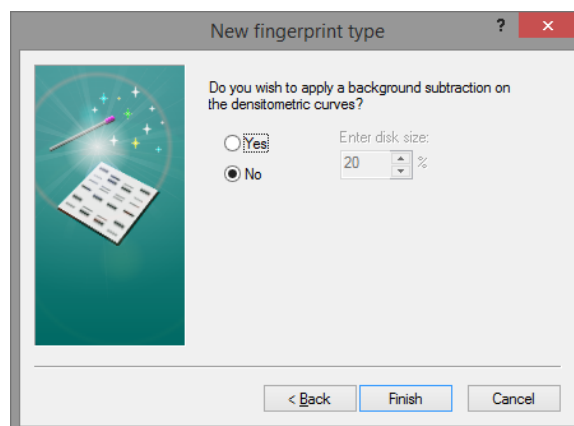


Figure 4.1.4: Step 4 of the *New fingerprint type* wizard.

In the next step, you are prompted to allow a **Background subtraction** and to enter the size of the disk, as a percentage of the track length. The default suggested disk size will suit most fingerprint types. For high resolution fingerprints (e.g. AFLP and sequencer-generated patterns) you can try a smaller disk size.

It is actually recommended *not* to perform a background subtraction at the raw data level and to subtract background from the densitometric curves. See 4.1.3.4 on how we can have the program propose the optimal background subtraction settings automatically.

Press <*Finish*> to complete the creation of the new fingerprint type, which will appear in the *Experiment types* panel.

4.1.3 Importing and processing gel images

4.1.3.1 Import and image pre-processing

To add a new fingerprint file to the database, highlight the *Fingerprint files* panel in the *Main* window and select *Edit > Create new object...* (+). Alternatively, select *File > Import...* (📁, Ctrl+I).

With the **Import gel file** option, listed in the *Import* dialog box under **Fingerprint type data** (see Figure 4.1.5), a gel file can be imported in the database. Optionally the *Fingerprint image import* window can be called to perform some preprocessing steps (rotating, flipping, ...) and/or to convert the gel file to an uncompressed gray scale TIFF file which is the standard format recognized by BioNumerics.

Selecting **Import gel file** under **Fingerprint type data** in the *Import* dialog box and pressing <*Import*> calls the *Select fingerprint file* dialog box (see Figure 4.1.6).



Before gel files can be imported in the database using this import option, a fingerprint type experiment must be defined in the database (see 4.1.2).

Pressing the <*Browse*> button allows you to select the gel file you want to import in the database. The gel file can be located on your computer, an external drive or on a network location. Note that gel files can only be imported one by one.

When **Open in image editor before import** is checked, the gel file will be opened in the *Fingerprint image import* window before the actual import of the gel file in the database. When importing a gel file directly in the database (**Open in image editor before import** unchecked) make sure the file is an uncompressed gray scale TIFF file, which is the standard format recognized by the database.

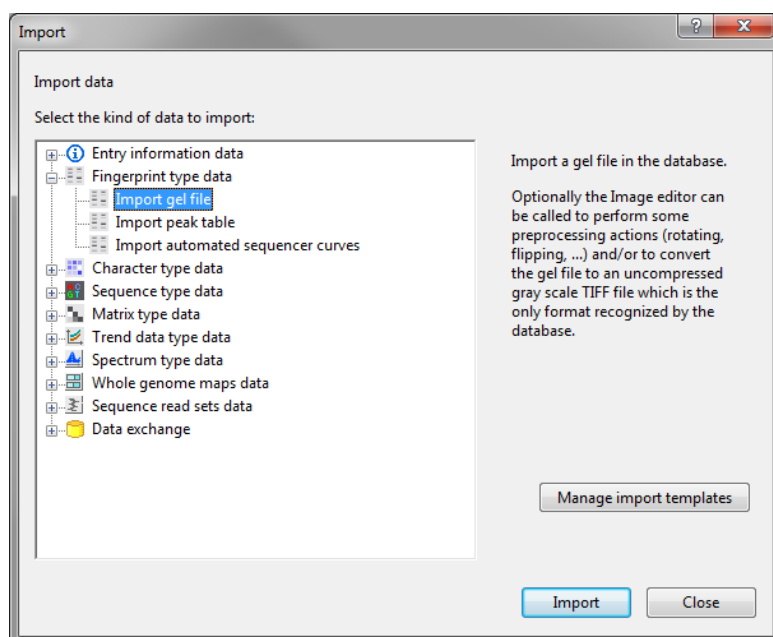


Figure 4.1.5: Importing a gel file via the *Import* dialog box.

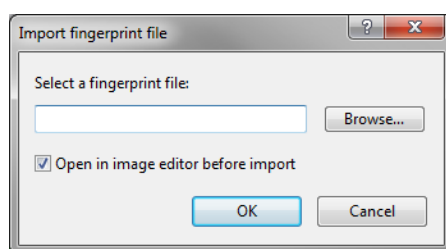


Figure 4.1.6: The *Select fingerprint file* dialog box.

The *Fingerprint image import* window (see Figure 4.1.7) is an image editor which allows the user to perform a number of preprocessing functions on the image. These functions include flipping, rotating and mirroring the image, inverting the image color, converting color images to gray scale, and cropping the image to defined areas. The image editor supports most known file types such as JPEG, GIF, PNG and compressed TIFF files in gray scale or RGB color.

The *Fingerprint image import* window consists of three tabs: *Original*, *Processed*, and *Cropped*.

In the *Original* tab, the unprocessed image is shown. In this tab, you can zoom in with **Edit > Zoom in** (🔍) or zoom out with **Edit > Zoom out** (🔍), and save the image to the database using **File > Add image to database...** (💾). The image can only be saved when it is in gray scale mode (see below).

In the *Processed* tab, the same options are available as in the *Original* tab, plus a number of image editing tools. These include:

- Inverting the color with **Image > Invert** (🔄) to invert images that have a black background, for example gels that were stained with ethidium bromide.
- Rotating the image 90° left with **Image > Rotate > 90 degrees left** (🔄), 90° right with **Image > Rotate > 90 degrees right** (🔄), or 180° with **Image > Rotate > 180 degrees** (🔄).
- Mirroring the image horizontally with **Image > Mirror > Horizontal** (🔄) or vertically with **Image > Mirror > Vertical** (🔄).

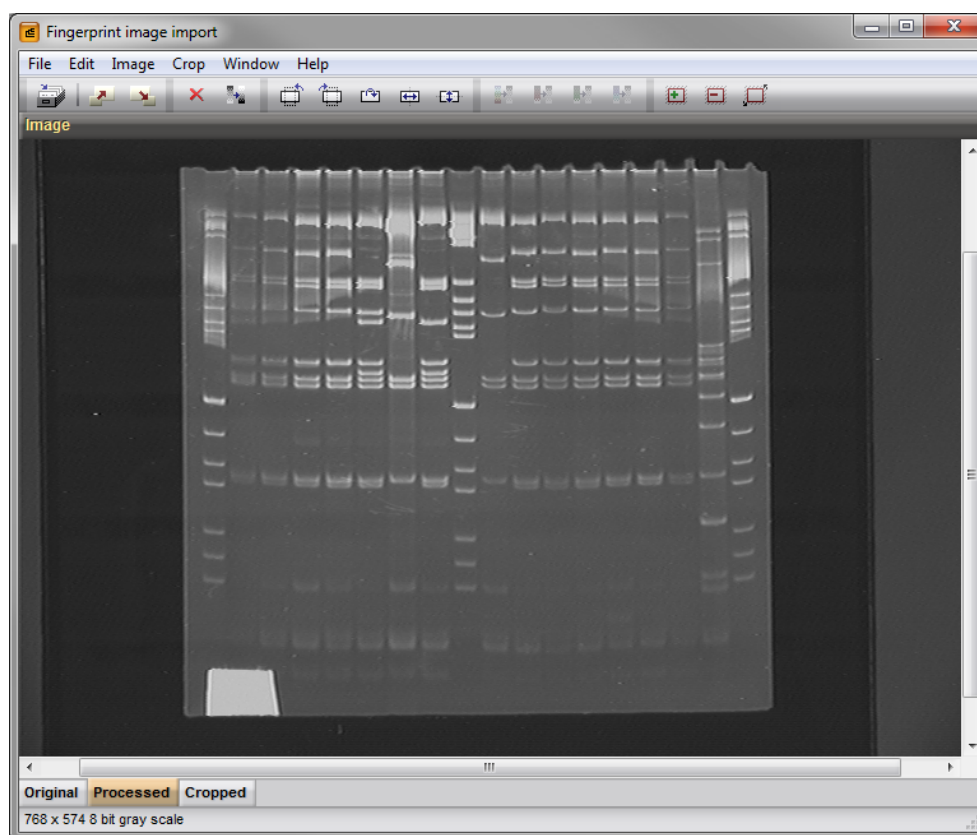


Figure 4.1.7: The *Fingerprint image import* window, *Processed* tab.

- Average RGB colors to gray scale with **Image > Convert to gray scale > Averaged** (🖼️), or convert a single channel to gray scale, either red (**Image > Convert to gray scale > Red channel** (🖼️)), green (**Image > Convert to gray scale > Green channel** (🖼️)) or blue (**Image > Convert to gray scale > Blue channel** (🖼️)).

The *Cropped* tab of the editor allows you to crop the image to a selected area, for which the following functions are available:

- **Crop > Add new crop** (🖼️), to add a new crop mask to the image. The crop mask can be moved by clicking anywhere inside the rectangle and dragging it to another position, or resized by clicking and dragging the bottom right corner of the rectangle.
- **Crop > Rotate selected crop...** (🖼️), to rotate the crop mask over a defined angle. Rotating the crop mask over an angle different from 90° or 180° will cause the program to recalculate densitometric values based upon interpolation, which means that the quality of the image may slightly decrease. This action is therefore not recommended unless it is inevitable.
- **Crop > Delete selected crop** (🖼️) can be used to delete the crop mask that is currently selected. Note in this respect that the program allows multiple crop masks to be defined for a single image. The final image that will be saved to the database, will be composed of all cropped areas aligned horizontally next to each other.

With **Image > Expand intensity range**, it is possible to recalculate the pixel values of the image so that they cover the entire range within the OD depth of the file, e.g. 8-bit = 256 gray levels, 16-bit = 65,536 gray levels.

To undo any previous action, the image can be reset to its original state with **Image > Load from original** (⏮).

When you are satisfied with the result of the preprocessing, the image should be saved to the database with **File > Add image to database...** (💾), which will open the *Add image to the database* dialog box.

The *Add image to the database* dialog box prompts for a name for the image. By default, the original file name is suggested. Pressing <OK> will add the gel image to the database.

The newly entered gel becomes available in the *Fingerprint files* panel. The file is marked with **N**, which means that it has not been edited yet (see Figure 4.1.8).

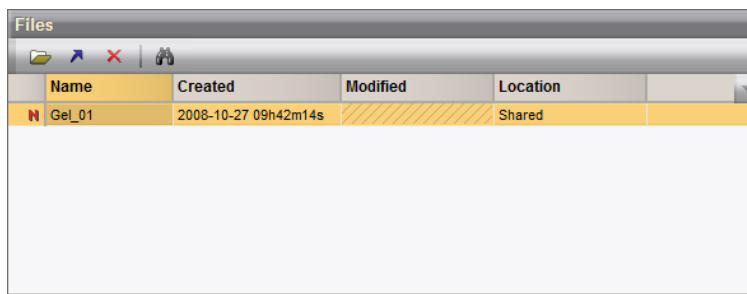


Figure 4.1.8: The *Fingerprint files* panel after import of a gel.

Any other gel TIFF file you want to process can be imported in the same way in the current database. The program will list these TIFF files in the *Fingerprint files* panel.

The *Fingerprint files* panel is an *object grid panel*, which means that all functionality as described under 3.2 is available for this panel. This allows you for example to search, sort and select fingerprint files in this panel and to add custom information fields. The latter feature is useful to store information such as gel processing parameters, the name of the person who ran the gel, etc..

4.1.3.2 Processing steps

To start the processing of a gel image, click on the gel name in the *Fingerprint files* panel (see Figure 4.1.8) and select **Edit > Open highlighted object...** (🔍, **Enter**) to open its *Fingerprint* window, which is initially empty. Next, select **File > Edit fingerprint data...** (📄).

When the gel is new (unprocessed), BioNumerics does not know what fingerprint type it belongs to. Therefore, the *Select experiment type* dialog box is first shown, listing all available fingerprint types, and allowing you to select one of them, or to create a new fingerprint type with <**Create new**>. Pressing <OK> confirms the selection of the fingerprint type.

The gel file is now loaded in the *Fingerprint processing* window (Figure 4.1.13). Depending on the size of the image, this may take some time.

The whole process of lane finding, normalization, band finding and band quantification is contained in a wizard, allowing the user to move back and forth through the process and make changes easily in whichever step of the process. The commands **File > Previous step** (⏮) and **File > Next step** (⏭) are to move back and forth, respectively. The complete process involves the following steps, shown in the tabs in the bottom left corner of the window:

1. **Strips:** defining lanes
2. **Curves:** defining densitometric curves
3. **Normalization**

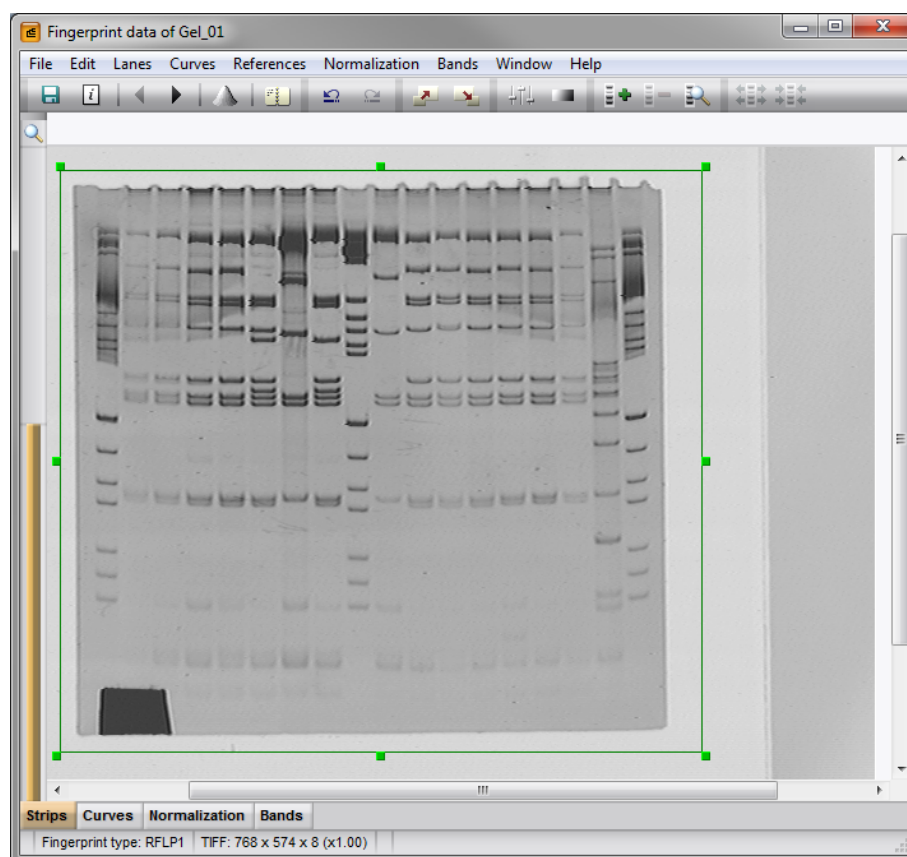


Figure 4.1.9: The *Fingerprint processing* window: initial view.

4. Bands: defining bands and quantification

The tabs themselves can be used for navigation between the different steps and allow you to "skip" steps, e.g. to return in one click from **Normalization** to **Strips** when it turns out a lane was not properly defined. When processing a new gel image, however, it is not recommended to skip any steps in the process.

Within each of these four steps, there is an undo/redo function. To undo one or more actions, you can use *Edit > Undo* (⌘, **Ctrl+Z**). To redo one or more actions, use *Edit > Redo* (⌘, **Ctrl+Y**). Once you have moved from one step to another, the undo/redo function within that step is lost.

4.1.3.3 Defining pattern strips on the gel

At the start, the image is shown in original size (x 1.00, as displayed in the status bar of the window).

You can zoom in and zoom out with *Edit > Zoom in* (⌘, **Ctrl+Page Up**) and *Edit > Zoom out* (⌘, **Ctrl+Page Down**), respectively. The zoom slider (left of the *Image* panel in default configuration) offers a convenient alternative for zooming in and out on the gel image. See 2.3.7 for a detailed description of the zoom slider functions.

When a large image is loaded, the *Fingerprint navigator* window can be popped up to focus on a region of the image. To call the *Fingerprint navigator* window, select *Lanes > Navigate* (**Alt+V**) or simply double-click on the image.

You can change the brightness and contrast of the image with *Edit > Change brightness & contrast...* (⌘). This pops up the *Image brightness & contrast* dialog box (see Figure 4.1.10).

Click *Dynamical preview* to have the image directly updated with the changes you make. Use the *Minimum*

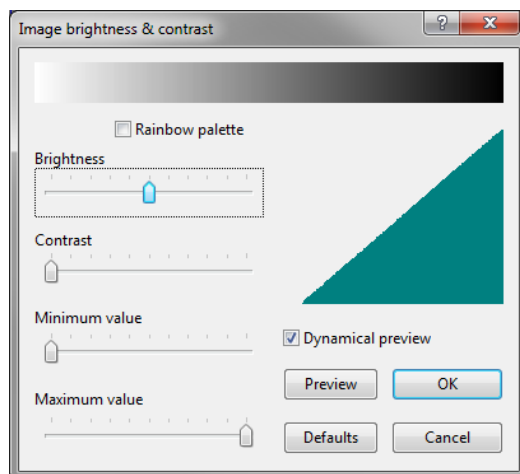


Figure 4.1.10: The *Image brightness & contrast* dialog box.

value slide bar to reduce background if the background of the whole image is too high. Use the **Maximum value** slide bar to darken the image if the darkest bands are too weak. The option **Rainbow palette** can be used to reveal even more visual information in areas of poor contrast (weak and over saturated areas) by using a palette that exists of multiple color transitions. When pressing <OK>, the changes made to the image appearance are saved along with the fingerprint type.



The brightness and contrast settings are saved along with the fingerprint type (see 4.1.5.4), but are not specific for a particular gel. The *Gel tone curve* window, as explained further, is a more powerful image enhancement tool for which the settings are saved for each particular gel.

With **File > Show 3D view...** (▲), a three dimensional view of the gel image can be obtained in a separate *Fingerprint 3D view* window (see Figure 4.1.11).

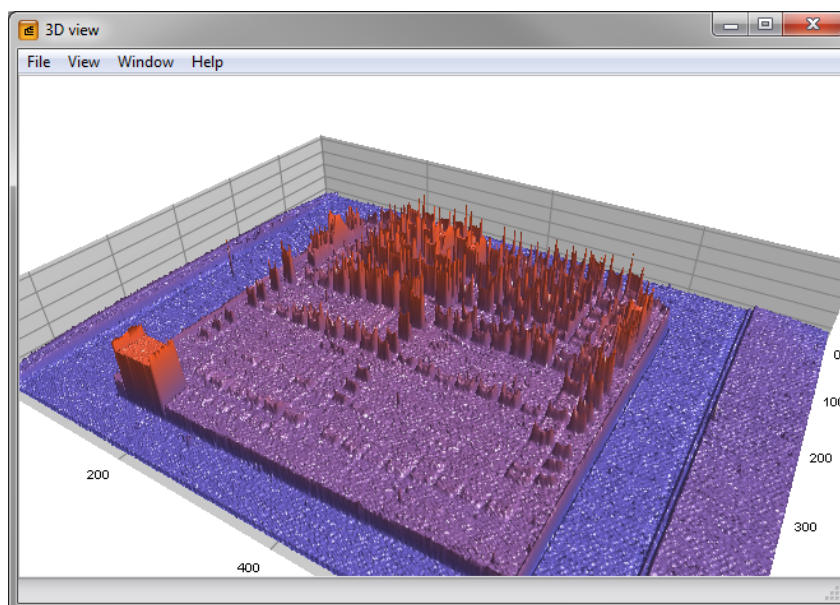


Figure 4.1.11: The *Fingerprint 3D view* window.

In the *Fingerprint 3D view* window, you can use the **Left**, **Right**, **Up** and **Down** arrows keys on the keyboard, to turn the position of the image in all directions. The image can also be rotated horizontally and vertically by dragging the image left/right or up/down using the mouse.

The zoom factor can be changed using **View > Zoom in (Pg Down)** or **View > Zoom out (Pg Up)**.

The vertical zoom, i.e. the Z-axis showing the peak height, can be changed with **View > Higher peaks (Insert)** or **View > Lower peaks (Delete)**.



The *Fingerprint 3D view* window can be called in all four steps of the image pre-processing. In the three last steps of the *Fingerprint processing* window (2. Curves, 3. Normalization, and 4. Bands), the *Fingerprint 3D view* window shows only the selected lane image rather than the entire gel image. In this case, any defined bands (see 4.1.3.6) on the lane can be displayed with **View > Show bands**.

Close the *Fingerprint 3D view* window with **File > Exit**.

To save the work done in the *Fingerprint processing* window at any stage of the process, select **File > Save** (📁, **Ctrl+S**). In case you work with complex gels, it is advisable to save your work at regular times.

When you save the gel file, the program may prompt you with the following question: "The resolution of this gel differs considerably from the normalized track resolution. Do you wish to update the normalized track resolution?". The gel resolution is explained further (see 4.1.3.5). If the question appears, answer **<Yes>**.

The green rectangle is the *bounding box*, which delimits the region of interest of the gel: tracks and gel strips will be extracted within the bounding box. To move the bounding box as a whole, hold down the **Ctrl**-key while dragging it in any of the green squares, situated at the edges of the box (*distortion nodes*). Adjust the box by dragging the distortion nodes as necessary: corner nodes can be used to resize the box in two directions, whereas inside nodes can only be used to resize one side of the box.

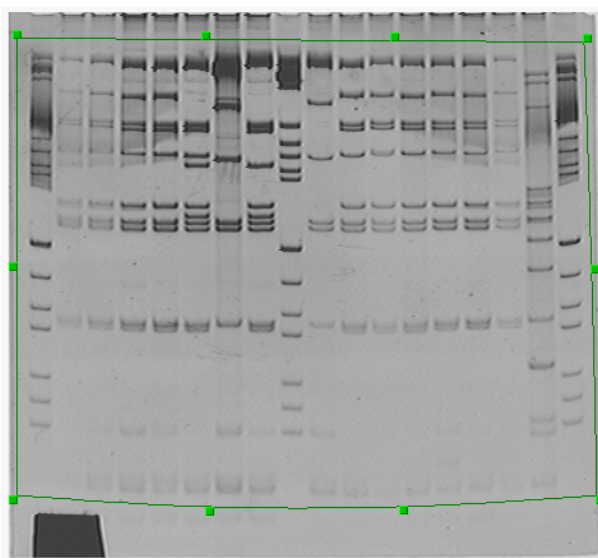


Figure 4.1.12: Defining the bounding box to follow contours of distorted gel.

- By using the **Shift**-key, one can even distort the sides of the rectangle. Holding the **Shift**-key while dragging the corner nodes will change the rectangle into a non-rectangular quadrangle (parallelepiped).
- A curvature can be assigned to the sides of the bounding box by holding the **Shift**-key while dragging one of the inside nodes in any direction (see Figure 4.1.12, top and bottom sides).
- On the top and bottom sides of the bounding box, more nodes can be added using **Lanes > Add bounding box node**. While holding down the **Shift**-key, a node can be dragged to the left or to the right using the mouse.
- A node can be deleted from the bounding box using **Lanes > Delete bounding box node**.

Following the curvature of a distorted gel is not crucial, as this is corrected in the normalization step (see 4.1.3.5) in case there are sufficient reference lanes on the gel. However, as it will provide a first rough normalization, it can aid the automatic or manual assignment of bands as explained in 4.1.3.5. Also, the software allows the bounding box curvature to be used for rectifying sloping or "smiling" lanes (e.g. Figure 4.1.12, outer lanes), if this option is enabled in the *Fingerprint processing settings* dialog box.

Select **Lanes > Auto search lanes...** (🔍) to let the program find the patterns automatically.

The *Search lanes* dialog box asks you to enter the approximate number of tracks on the gel.

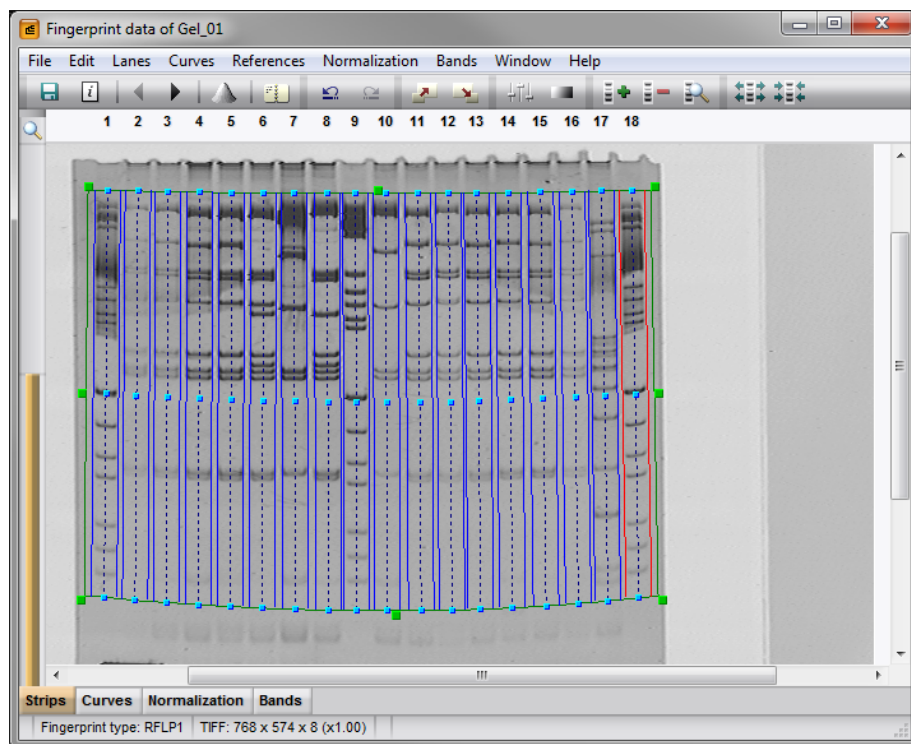


Figure 4.1.13: The *Fingerprint processing* window. Step 1: defining pattern strips.

Each lane found on the image is represented by a *strip*: a small image that is extracted from the complete file to represent a particular pattern. The borders of these strips are represented as blue lines, or red for the selected lane (see Figure 4.1.13). The default strip thickness is 31 points, which may be too wide or too narrow for the current gel.

Call the *Fingerprint processing settings* dialog box with **Edit > Edit settings...** (⚙️) (see Figure 4.1.14).

This dialog box consists of four tabs, of which the tab corresponding to the current stage of the processing is automatically selected.

The **Data source** is displayed here and refers to the fingerprint's data type (see 4.1.5.2), which can be altered in the *Select data type* dialog box.

The **Thickness** of the image strips should be adjusted so that the blue lines enclose the complete patterns (blue lines of neighboring patterns should nearly touch each other). See Figure 4.1.15 for an optimally adjusted example.

If necessary, increase the number of distortion **Nodes**. These nodes allow you to bend the strips locally. Usually, three nodes should be fine.

In the *Raw data* tab, two more options, **Background subtraction** and **Spot removal** allow gel scans with irregular background and spots or artifacts to be cleaned up to a certain extent. It should be emphasized that the options **Background subtraction** and **Spot removal** have an influence on gel strips in all further processes

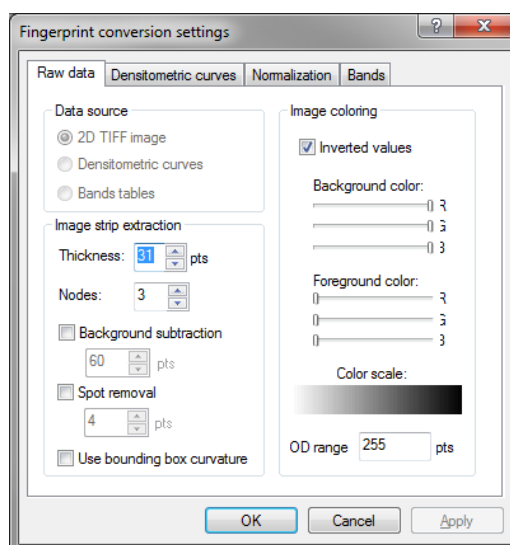


Figure 4.1.14: The *Fingerprint processing settings* dialog box, *Raw data* tab.

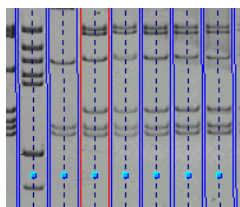


Figure 4.1.15: Optimal strip thickness settings, detail.

of the program: gel strips will always be shown with background subtracted and with spots removed. In addition, when two-dimensional quantification is done, the gel strips with background subtracted and spots removed are used. Hence, we recommend **not** to use these options unless (1) the image has a strong irregular background, for example by non-homogeneous illumination of the gel, so that the gel strips would not look appropriate for presentation or publication; (2) the gel contains numerous spots that would influence the densitometric curves extracted from the gel strips (spots on the image are seen as peaks on a densitometric curve, and hence have a strong impact on correlation coefficients, band searching, etc.).

The **Background subtraction** is based on the "rolling ball" principle, and the size of the ball in pixels of the image can be entered. The larger the size of the ball, the less background will be subtracted. Inevitably, the spot removal mechanism causes some distortion on the patterns. The smaller the size of the spot removal, the less the distortion.

The **Spot removal** is a similar mechanism as the rolling ball, but an ellipse is used instead, in order to separate bands from spots. The size of the ellipse can be entered in pixels. Unlike the background subtraction, the size of the ellipse should be kept as small as possible in order not to erase bands.



If background subtraction on the gel strips is applied, it is not necessary anymore to perform background subtraction on the densitometric curves, since this is doing exactly the same but on one-dimensional patterns.

The effect of background subtraction and spot removal on gel strips is only seen in the next step, when the gel strips are shown.

Using the option **Use bounding box curvature**, it is possible to have the program correct smiling or sloping bands due to distortion in the gel. The bands will be rectified according to the bounding box curvatures defined. An example is given in Figure 4.1.12, where the bounding box has been assigned a curvature to

follow the distortions in the outer lanes. The result of enabling the correction for bounding box curvature is shown in Figure 4.1.17, where it can be clearly seen that the bands of the outer lanes have been straightened.

BioNumerics recognizes the darkness as the intensity of a band. When processing a gel with bands that appear as fluorescent lighting on a black background, the check box **Inverted values** should be checked.

Adjust the position of each spline as necessary by grabbing the nodes using the mouse. Use the **Shift**-key to bend a spline locally in one node.

Lanes can be added one by one with **Lanes > Add new lane** (📄, Enter). A new lane is placed right from the selected one. A selected lane can be removed again with **Lanes > Delete selected lane** (🗑, Del) if necessary.

If one lane is more distorted than the current number of nodes can indicate, you can increase the number of nodes in that lane by selecting it and **Lanes > Strips > Increase number of nodes**.

If the lanes are not equally thick, you can increase or decrease the thickness of each individual strip with **Lanes > Strips > Make larger** (📏, F7) or **Lanes > Strips > Make smaller** (📏, F8), respectively.

Once the lanes are defined on the gel, select **Edit > Edit tone curve...** to display the *Gel tone curve* window as in Figure 4.1.16.

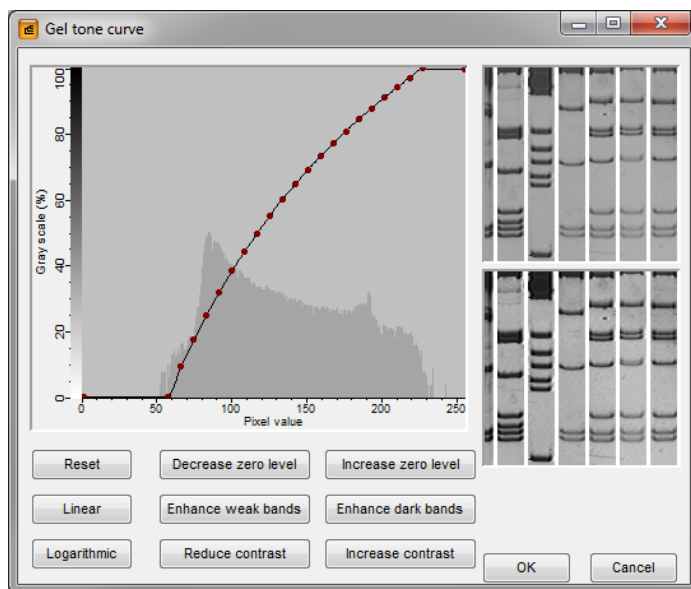


Figure 4.1.16: The *Gel tone curve* window.

The *Gel tone curve* window is a powerful tool to edit the appearance of the image. While the Image brightness and contrast settings act at the screen (monitor) level, i.e. after the TIFF gray scale information is converted into 8-bit gray scale, the *Gel tone curve* window acts at the original TIFF information level. This means that, in case a gel image is scanned as 16-bit TIFF file, the tone curve settings are applied to the full 16-bit (65,536) gray scale information which allows much more information to be magnified in particular areas of darkness. The advantages are:

- Weak bands are much better enhanced resulting in a smoother and more reliable picture.
- The tone curve acts at a level below the brightness and contrast settings and can be saved along with a particular gel. In all further imaging tools of the program, the tone curve for the particular gel is applied. Brightness and contrast settings are not specific to a particular gel.
- The user can fine-tune the tone curve to obtain optimal results.

The upper panel is a distribution plot of the densitometric values in the TIFF file over the available range. The right two windows are a part of the image *Before correction* and *After correction*, respectively.

You can scroll through the preview images by left-clicking and moving the mouse while keeping the mouse button pressed.

Left, there are two buttons, **<Linear>** and **<Logarithmic>**. Both functions introduce a number of distortion points on the tone curve, and reposition the tone curve so that it begins at the gray scale level where the first densitometric values are found, and ends at its maximum where the darkest densitometric values are found. This is a simple optimization function that rescales the used gray scale interval optimally within the available display range. The difference between linear and logarithmic is whether a linear or a logarithmic curve is used. In case of 8-bit gels, a linear curve is typically the best starting point.

There are six other buttons that are more or less self-explaining:

- **<Decrease zero level>** and **<Increase zero level>** are to decrease and increase the starting point of the curve, respectively.
- **<Enhance weak bands>** and **<Enhance dark bands>** are also complementary to each other, the first making the curve more logarithmic so that more contrast is revealed in the left part of the curve (bright area), and the second making the curve more exponential so that more contrast is revealed in the right part of the curve (dark area).
- **<Reduce contrast>** and **<Increase contrast>** make the curve more sigmoid so that the total contrast of the image is reduced or enhanced, respectively.

Press **<OK>** to save the tone curve settings.



It is also possible to edit the tone curve manually: nodes can be added by double-clicking on the curve in the *Gel tone curve* window, or can be deleted by selecting them and pressing the **Del**-key. The curve can be edited in each node by left-clicking on the node and moving it. There is a **<Reset>** button to restore the original linear zero-to-100% curve.

Select **File > Next step** (▶) to go to the next step: defining densitometric curves.

4.1.3.4 Defining densitometric curves

In this step, the window is divided in two panels (Figure 4.1.17): the *Image* panel on the left shows the strips extracted from the image file and the *Densitometric curve* panel on the right shows the densitometric curve of the selected pattern, extracted from the image file. You can move the separator between both panels to the left or to the right to allow more space for the strips or for the curves.

The program has automatically defined the densitometric curves using the information of the lane strips you entered in the previous step. Normally, you will not have to change the positions of the densitometric curves anymore, except when you want to avoid a distorted region within a pattern, e.g. due to an air bubble within the gel. If necessary, the position of a spline can be adjusted by grabbing the nodes using the mouse. Use the **Shift**-key to bend the spline locally in one node.

The blue lines represent the width of the area within which the curve will be averaged. The default value is 7 points. In most cases, you will have to optimize this value for a given type of gel images.

Call the *Fingerprint processing settings* dialog box with **Edit > Edit settings...** (⚙️). This time, the *Densitometric curves tab* is displayed (see Figure 4.1.18).

Ideally, the *Averaging thickness* for curve extraction should be chosen as broad as possible. However, smiling and distortion at the edges of the bands should be excluded (see Figure 4.1.19).

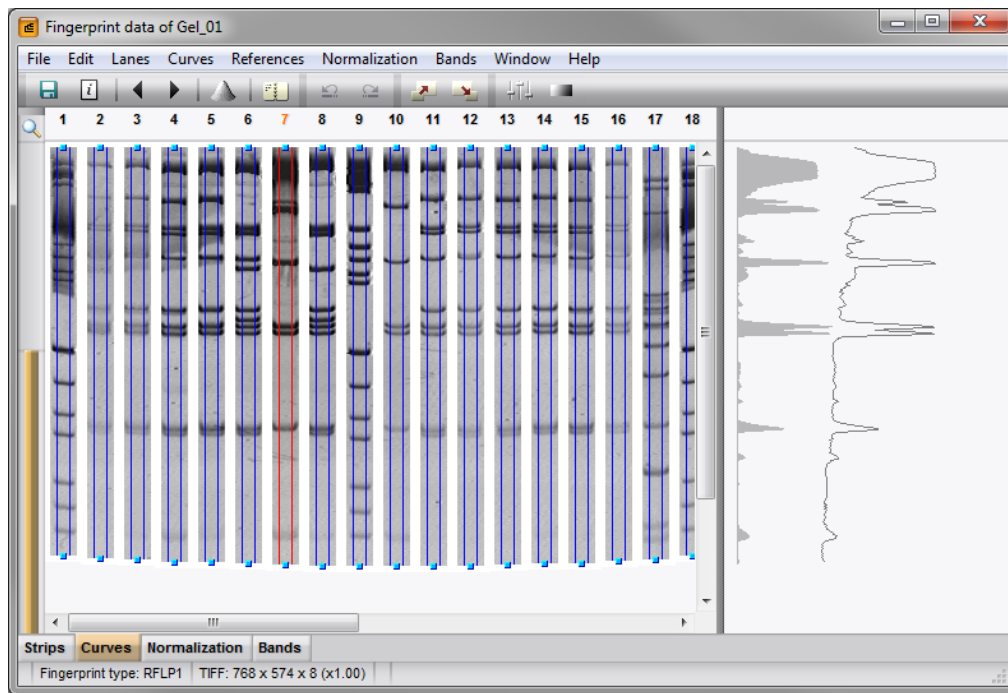


Figure 4.1.17: The *Fingerprint processing* window. Step 2: defining densitometric curves.

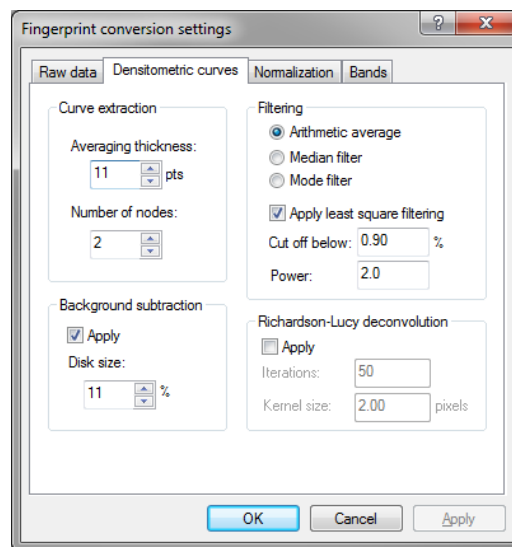


Figure 4.1.18: The *Fingerprint processing settings* dialog box, *Densitometric curves* tab.

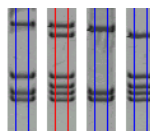


Figure 4.1.19: Optimal settings for curve averaging thickness.

The curve extraction settings include other important parameters which apply to the background removal and smoothing.

When we defined the fingerprint type, we left the **Background subtraction** disabled, because we will see how we can have the program propose the optimal settings.

Filtering is a method to make an average of the values within a specified thickness. Simple averaging is obtained with *Arithmetic average*, whereas *Median filter* and *Mode filter* are more sophisticated methods to reduce peak-like artifacts caused by spots on the patterns. Figure 4.1.20 illustrates the effect of the *Median filter* on a small spot. The latter two filters, however, reduce less noise on the curves (particularly the *Mode filter*). Only in case your gels contain hampering spots, you should use the *Mode filter*.

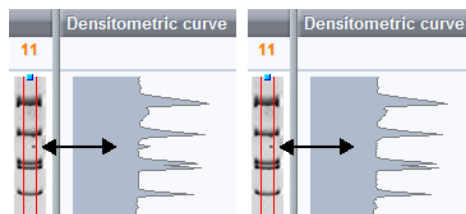


Figure 4.1.20: Result of *Arithmetic average* filtering (left) and *Median filtering* (right).

The *Least square filtering* applies to the smoothing of the profiles. This filter will remove background noise, seen as small irregular peaks, from the profile of real (broader) peaks. Like for background subtraction, the program can predict the optimal settings for least square filtering, if necessary.

Richardson-Lucy deconvolution is a method to *deblur* (sharpen) one-dimensional and two-dimensional arrays. This function sharpens and enhances the contrast of peaks in the densitometric curves. While peaks will become sharper, noise also will increase. Deconvolution actually does the opposite of least-square filtering. Since the method is iterative, the number of *Iterations* can be set (default 50). The more iterations, the stronger deconvolution will be obtained. The *Kernel size* (default 2.00) determines the resolution of the deconvolution: the smaller this value is set, the more shoulders will be split into separate peaks.

Pressing <OK> will save the settings.

The optimal settings for background subtraction and filtering settings can be determined using spectral (Fourier) analysis by selecting *Curves > Spectral analysis....* This shows the *Spectral analysis* window (see Figure 4.1.21).

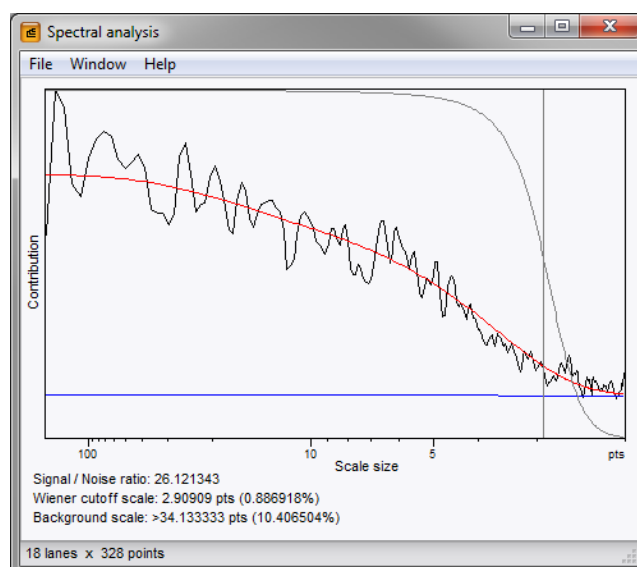


Figure 4.1.21: Spectral analysis of the patterns of a gel in the *Spectral analysis* window.

The black line is the spectral analysis of the curves in function of the frequency in number of points (logarithmic scale). Ideally, the curve should show a flat background line at the right hand side, and then slowly raise further to the left. This indicates that the scanning resolution is high enough. Another parameter which indicates the quality is the *Signal/noise ratio*, which should be above 50, if possible. The example gel in

Figure 4.1.21 is only of moderate resolution.

The **Wiener cut-off scale** determines the optimal setting for the least square filtering. Figure 4.1.21 shows an optimal setting of 0.89.

The **Background scale** is an estimation of the disk size for background subtraction. The figure shows a setting of 11%.

The optimal settings, calculated in this window, can then be entered in the *Fingerprint processing settings* dialog box.

If you want to have a better look at the curves (right panel) you can rescale them with **Edit > Rescale curves**. This will rescale the gray processed curves (background subtracted and filtering applied) to fit within the available window space. The raw curves (lines) may then fall beyond the window.

With the command **File > Print report...** or **File > Export report...**, you can generate a printed or text report of the non-normalized densitometric curves, respectively.

Select **File > Next step** (▶) to enter the next phase: normalization of the patterns.

4.1.3.5 Normalizing a gel

In the Normalization step, the *Fingerprint processing* window consists of three panels (see Figure 4.1.22): left the *Reference system* panel, which will show the reference positions and the standard pattern; the center *Image* panel shows the pattern strips; and the right *Densitometric curve* panel shows the densitometric curve of the selected pattern.

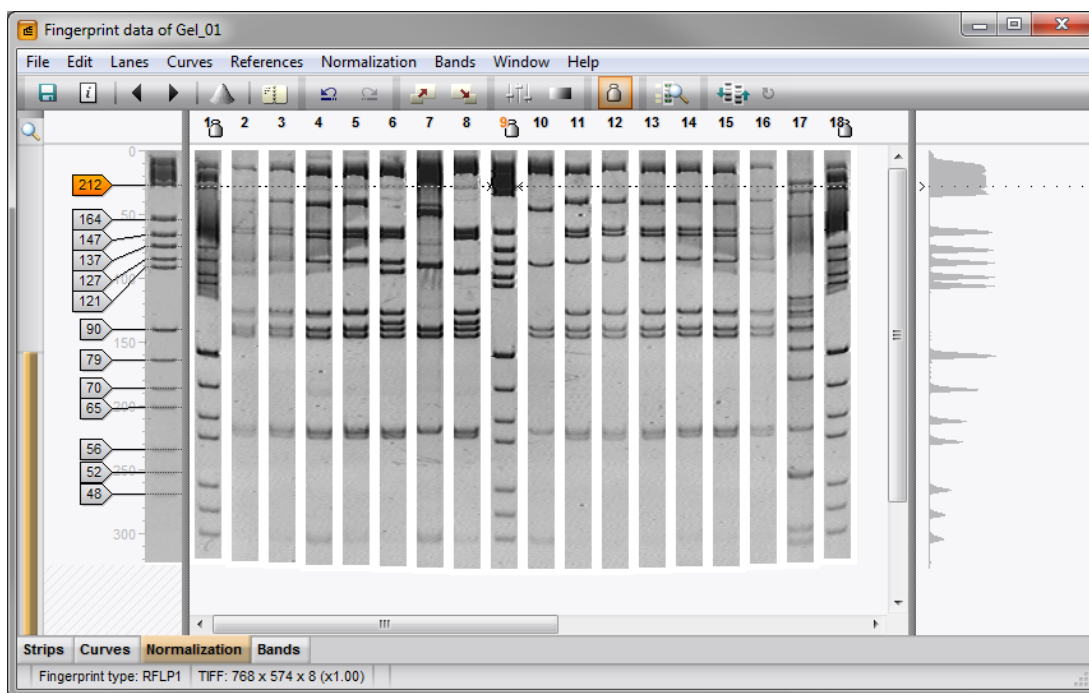




Figure 4.1.22: The *Fingerprint processing* window. Step 3: Normalization.

When setting up a new database, the normalization process of the first gel involves the following steps. Only steps 1, 4 and 5 should be performed for subsequent gels.

1. Marking the reference patterns (reference patterns are identical samples loaded at different positions on the gel for normalization purposes);

2. Identifying a suitable reference pattern on which we will define bands as *reference positions*. Reference positions are bands that will be used to align the corresponding bands on all reference patterns from the same and from other gels.
3. Defining the reference positions;
4. Assigning the bands on the reference patterns to the corresponding reference positions;
5. Updating the normalization;
6. Defining a standard (optional).

All reference patterns on the gel should be marked as such with **References > Use as reference lane** (, **Ctrl+R**). To show the gel in normalized mode, select **Normalization > Show normalized view** (, **Shift+N**).

To create a new reference system, proceed as follows:

- Choose the most suitable reference pattern on the gel to serve as standard.
- Click on a suitable band on the destined standard pattern and select **References > Add external reference position**. The *Add new reference position* dialog box will prompt you to enter a name for the band. You can enter any name, or if possible, the molecular weight of the band. In the latter case, the program will later be able to determine automatically the molecular weight regression from the sizes entered at this stage (see 4.1.5.7).
- Repeat the previous step for the next suitable band in the pattern and continue until all reference positions are defined.

Within a fingerprint type, the set of reference positions as defined, and their names, together form a *reference system*. Once a gel is normalized using the defined reference positions and saved, the reference system is saved as well. As soon as you change anything in the reference system, a position or a name, a new reference system will automatically be created in addition to the original reference system. Once a reference system has been used in one or more gels however, the program will produce a warning if you want to change anything to the reference positions.

If more than one reference system exists, one of them is the *active reference system*, i.e. the reference system to which all new gels will be normalized. Without intervention of the user, the first created reference system will always remain the default. In 4.1.5.7, it is explained how to set the active reference system and delete unused reference systems.



The message "No active reference system defined" is displayed in the left panel when processing the first gel of a fingerprint type. Once a second gel is normalized, this message will not be displayed anymore.

The normalization is done in two steps: first are the reference bands assigned to the corresponding reference positions, and then the display is updated according to the assignments made. This last step is optional, but is useful to facilitate the visual evaluation of the assignments made.

Bands can be assigned **manually** as follows:

- Click on a label of a reference position, or wherever on the gel at the height of the reference position. The reference position becomes highlighted.
- **Ctrl-click** on the reference band you want to assign to that reference position. Repeat this action for all other reference bands you want to assign to the same reference position.

- Repeat the steps above until all reference bands are assigned to their corresponding reference positions.

All assignments can be removed at once with **Normalization > Delete all assignments** (Ctrl+Shift+Del). Assignments can also be removed for the current lane with **Normalization > Delete assignments (current lane)** (Ctrl+Del) and for the currently selected reference position with **Normalization > Delete assignments (current position)** (Shift+Del).



The cursor automatically jumps to the closest peak; to avoid this, hold down the **Tab**-key while clicking on a band. This "snap to peak" behavior can be toggled on or off with **Edit > Snap to peaks**.

With **Normalization > Show normalized view** (📊, Shift+N), the gel will be shown in *normalized view*, i.e. the gel strips will be stretched or shrunk so that assigned bands on the reference patterns match with their corresponding reference positions.

To let the program assign the bands and reference positions automatically, select **Normalization > Auto assign...** (🔍).

This will open the *Auto assign reference bands* dialog box (see Figure 4.1.23).

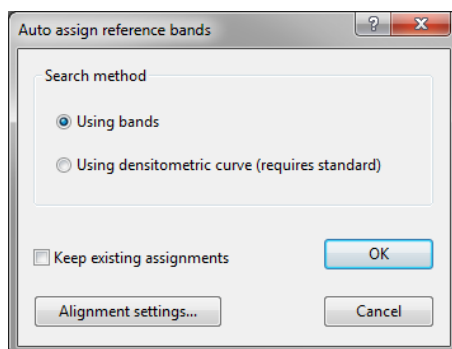


Figure 4.1.23: The *Auto assign reference bands* dialog box.

Under **Search method**, two options are available: **Using bands** and **Using densitometric curve**.

- In the **Using bands** option, the program searches for bands on the reference patterns and tries to match them optimally with the defined reference positions. This method is always applicable, even for the very first gel, when no standard is defined.
- In the **Using densitometric curve** option, a different algorithm is used, which matches the densitometric curve of the standard pattern with the curves of the reference patterns. Obviously, the option requires a standard to be defined (see 4.1.5.8 on how to define a standard). This method employs a pattern matching algorithm that works best for complex banding patterns, but is less suitable for simple patterns such as molecular weight ladders.

An option independent of the search method is **Keep existing assignments**. When this option is chosen, any assignments made previously are preserved. This option allows the user to assign a few bands manually and let the program automatically assign the remaining bands on the reference patterns. This way of working is useful to provide some initial help to the algorithm in case of very distorted or difficult gels.

Pressing <**Alignment settings**> opens the *Aligner settings* dialog box (see Figure 4.1.24).

Parameters can be adjusted for the peak detection, global alignment and local alignment algorithms:

The **Peak detection parameters** determine what is recognized by the program as a peak.

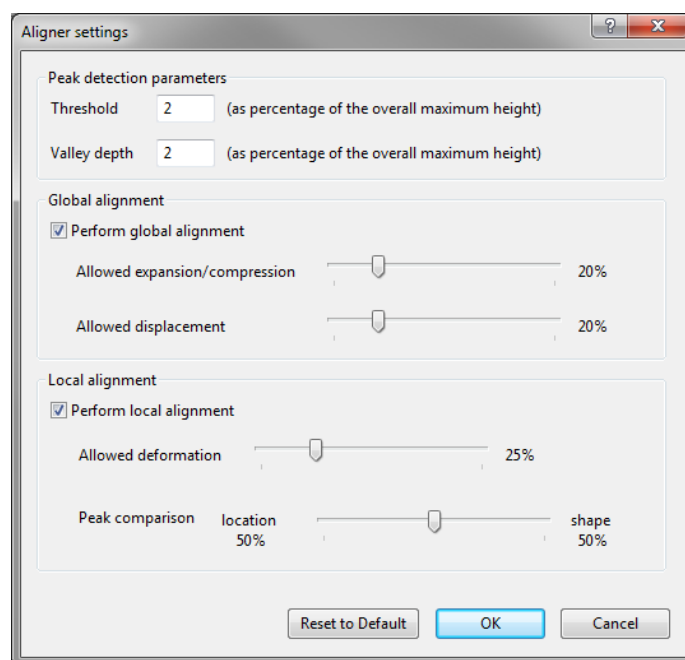


Figure 4.1.24: The *Aligner settings* dialog box.

- **Threshold** is the minimal height, expressed as a percentage of the highest peak in the profile, for which an elevation in the profile is still considered to be a peak. The default value is 2%.
- The **Valley depth** is important for peak separation: it is the minimal depth of the depression between two subsequent maxima, for which the program divides a single peak into two separate peaks. In case one maximum is higher than the other, the height between the lowest maximum and the minimum is used. Similar to the threshold, the valley depth is expressed as a percentage of the highest peak in the profile. The default value is 2%.

In a **Global alignment**, the profile **as a whole** is expanded (stretched) or compressed (shrunk) and displaced (shifted) to give the best possible fit with the reference positions. Depending on the status of the corresponding check box, a global alignment is performed or not. Not performing a global alignment can be useful in case individual reference patterns show only a minor shift, e.g. in case of fingerprints ran on an automated sequencer. Regardless of the status of the check box, when **Keep existing alignments** is checked in the *Auto assign reference bands* dialog box (Figure 4.1.23), a global alignment is not performed. Instead, the program uses the distances as obtained after the first band assignment.

- The slider for **Allowed expansion/compression** lets the user determine the maximally allowed expansion or compression of the profile, expressed as a percentage of the total profile length. The default value is 20%.
- The slider for **Allowed displacement** lets the user determine the maximally allowed displacement (shift) of the profile, expressed as a percentage of the total profile length. The default value is 20%.

In a **Local alignment**, the profile is **locally** expanded (stretched) or compressed (shrunk) to match optimally with the reference positions. Using the corresponding check box, you can either perform or not perform a local alignment.

- The **Allowed deformation** is the maximally allowed deformation, expressed as a percentage of the profile length. The default value is 25%.

- The **Peak comparison** parameter allows the user to assign more weight on the peak **location** (position) or on the peak **shape**. The shape parameter is calculated based on a curve regression and a peak size parameter. By default, both **location** and **shape** are accounted for evenly (50%). The **Peak comparison** parameter is only considered when **Using densitometric curves** was checked in the **Auto assign reference bands** dialog box (Figure 4.1.23).

Generally, the default settings perform well with most fingerprint types. Default settings can be restored by pressing **<Reset to default>**.

The settings can be saved and the *Aligner settings* dialog box closed with **<OK>**.

After an automated assignment, carefully inspect the assignments made. If some are incorrect, correct them manually as explained above.



In case most or all of the patterns on a gel contain one or more identical bands, such bands can be used for internal alignment of the gel. The software therefore creates an *internal reference position* which is saved with the gel but is not part of the reference system. An internal reference position can be created with **References > Add internal reference position**. The program then asks "Do you want to automatically search for this reference band?". If you answer **<Yes>**, it will try to find all the correct assignments, but you can change or delete assignments afterwards.

When the gel is in normalized view, a reliable way to reveal remaining mismatches is by showing the *distortion bars* with **Normalization > Show distortion bars**: these bars indicate local deviations with respect to the general shift of a reference pattern compared to the reference positions (see Figure 4.1.25). A too strong shift is seen as a zone ranging from yellow over red to black, whereas a too weak shift is indicated by a zone ranging from bright blue over dark blue to black. Slight transitions from bright yellow to bright blue are normal, as long as the color does not change abruptly. In the latter case, a wrong assignment was made.

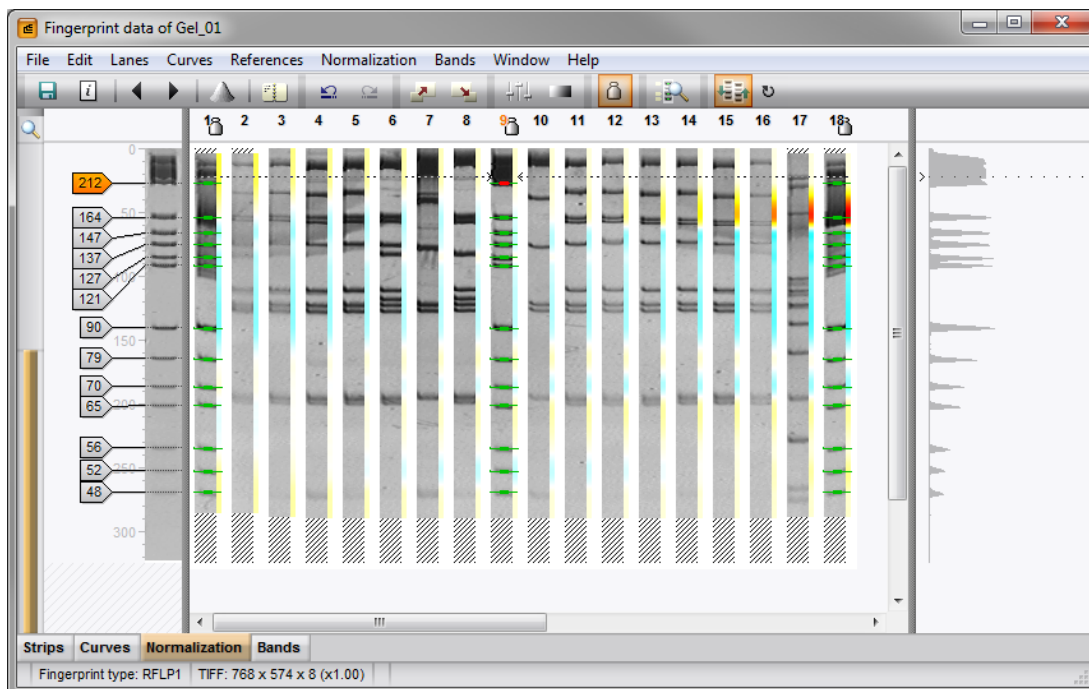




Figure 4.1.25: Distortion bars displayed on the gel image.

You can correct the misalignment by assigning the correct band manually and selecting **Normalization > Update normalization** (🔍, **Ctrl+U**). Alternatively, you can show back the original view, assign the correct band manually, and show the normalized view again. The **Show distortion bars** setting (on or off) is stored along with the fingerprint type.



If the program has difficulties in assigning the bands correctly, you can first make a few assignments manually (for example, the first and the last band of the reference patterns), then display the normalized view with **Normalization > Show normalized view** (, **Shift+N**) and then have the program find the assignments automatically with the option **Keep existing assignments** checked.

Call the *Fingerprint processing settings* dialog box with **Edit > Edit settings...** () (see Figure 4.1.26).

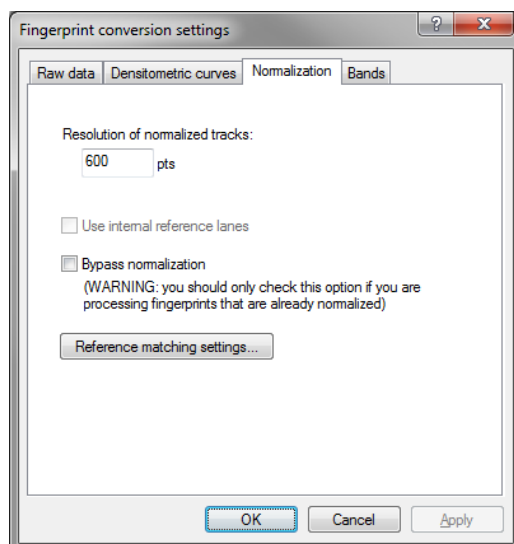



Figure 4.1.26: The *Fingerprint processing settings* dialog box, normalization tab




The third tab, the *Normalization tab* is shown, in which the **Resolution of normalized tracks** can be set. In reality, the program always stores the real length of the raw patterns. For display purposes however, the program converts the tracks to the same length at real-time, so that the gel strips are properly aligned to each other. For comparison of patterns by means of the Pearson product-moment correlation, the densitometric curves also need to be of the same length. Thus, the resolution value only influences two features: the length of the patterns shown on the screen, and the length (resolution, number of points) of the densitometric curves to be compared by the Pearson product-moment correlation coefficient. By default, the program uses 600 as resolution, but when you normalize the first gel, the program automatically uses the *average track length* for that gel as the new resolution value. Whenever you save the gel, and the value differs more than 50% from the default value, BioNumerics will ask you to copy the resolution of the current gel to the default for the fingerprint type (see 4.1.3.3).

The option **Bypass normalization** can be used to have the program process the densitometric curves of the tracks without any change. This option is only useful to import patterns in BioNumerics that are already normalized, and for which you want the values of the densitometric curves to remain exactly the same after the normalization process.

Pressing the **<Reference matching settings>** button will open the *Aligner settings* dialog box (see Figure 4.1.24).

Save the normalized gel with **File > Save** (, **Ctrl+S**).

It is possible to generate a text file or a printout of the complete alignment of the gel, by selecting **File > Export report...** or **File > Print report...**, respectively. The file lists all the reference bands defined in the reference system with their relative positions, and the corresponding bands on each reference pattern, with the absolute occurrence on the pattern in distance from the start.

Select **File > Show report** () to generate a report of all settings defined for the processed fingerprint file. The report can be saved as HTML or text with **File > Save as html** () or **File > Save as text** (), respectively.

If you are going to use band matching coefficients to compare the patterns, you should read 4.1.3.6, corresponding to the fourth step of the gel processing. If you are going to use a curve-based coefficient, you can skip that paragraph and continue with 4.1.3.8.

4.1.3.6 Defining bands and quantification

In step 3 (Normalization), select **File > Next step** (▶) to proceed to the fourth step: Defining bands and quantification. This is the last step in processing a gel, which involves defining bands and quantifying band areas and/or volumes.

Call the *Fingerprint processing settings* dialog box with **Edit > Edit settings...** (⚙) (see Figure 4.1.27).

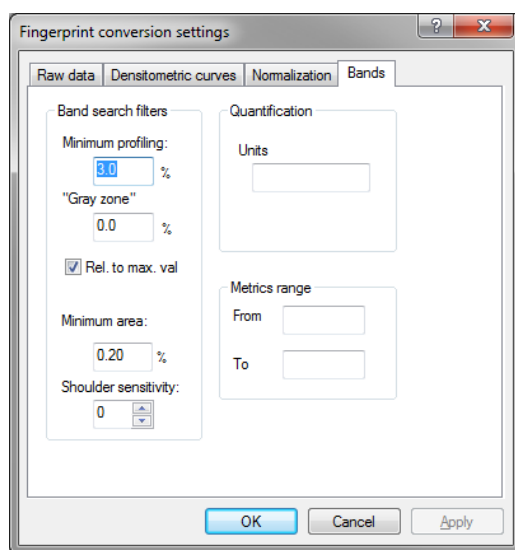


Figure 4.1.27: The *Fingerprint processing settings* dialog box, *Bands* tab.

The fourth tab, *Bands* is shown, which allows you to enter the **Band search filters** and **Metrics ranges**, as described for the *Band search* dialog box (see Figure 4.1.28).

In the **Quantification units** text box, the units in which the band concentrations are expressed, can be entered.

Select **Bands > Auto search bands** (🔍) to find bands on all the patterns. Before actually defining the bands on the patterns, the software displays the *Band search* dialog box (see Figure 4.1.28).

The **Minimum profiling** is the elevation of the band with respect to the surrounding background, expressed as a percentage. This parameter is therefore dependent on the specified **OD range**. If, for example, you increase the **OD range**, peaks will look smaller on the densitometric profiles, and a smaller minimum profiling will need to be set in order to find the same number of bands. However, when **Rel. to max. val** is checked, the minimal profiling, i.e. the minimal height of the bands will be taken relative to the highest band on that pattern. When patterns with different intensities occur on the same gel, it is recommended to enable this option. Along with the minimum profiling, it is possible to specify a **"Gray zone"**, also as a height percentage. This gray zone specifies bands that will be marked as uncertain. In comparing two patterns, the software will ignore all the positions in which one of the patterns has an uncertain band. The percentage value for the gray zone is added to the minimum profiling value. To take the example of Figure 4.1.27, all bands with a profiling of less than 5% are excluded; bands with a profiling between 5% and 10% are marked uncertain, and all bands with a profiling of more than 10% are selected (see Figure 4.1.29).

A more advanced tool based on deconvolution algorithms, **Shoulder sensitivity**, allows shoulders without a local maximum as well as doublets of bands with one maximum to be found. If you want to use the shoulder

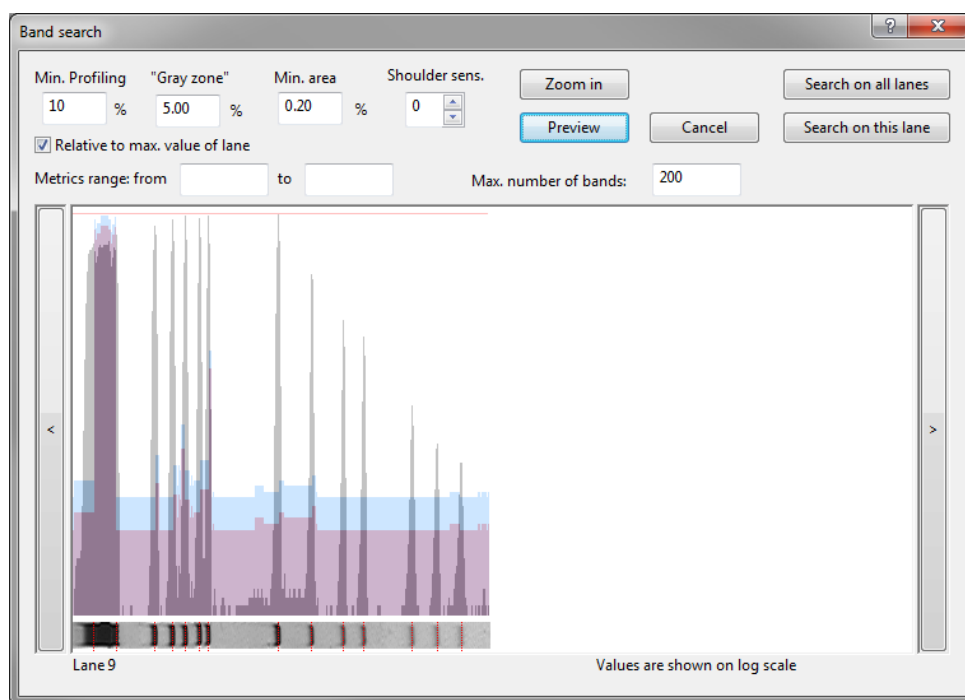


Figure 4.1.28: The *Band search* dialog box.

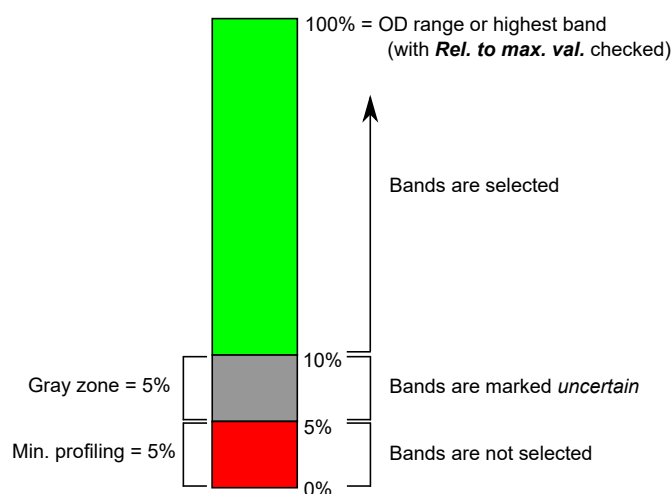


Figure 4.1.29: Understanding the meaning of the "Gray zone" of uncertain bands in relation to the minimum profiling.

sensitivity feature, we recommend to start with a sensitivity of 5, but optimal parameters may depend on the type of gels analyzed. A **Minimum area** can also be specified, as percentage of the total area of the pattern.

A fragment length range can be specified in the **Metrics range** box. If for example a range **From** 100 **To** 200 is specified, the software will only look for bands that have a fragment length between 100 and 200. When a **Metrics range** is specified and **Rel. to max. val** is checked, the minimal height of the bands will be taken relative to the highest band that is found in the fragment length range of that pattern.



The **Metrics range** is only taken into account if a calibration curve is calculated for the reference system that is used in the gel. More information on how to calculate a calibration curve can be found in 4.1.5.7.

Max. number of bands sets an upper limit to the number of bands that is retained by the algorithm. In some

cases, e.g. with low signal lanes and **Rel. to max. val** checked, an unrealistic large number of bands would be found, which obviously has an effect on performance.

The preview in the lower part of the *Band search* dialog box shows the first pattern on the gel with its curve and gel strip. Press the <Preview> button to see which bands the program finds using the current settings. A pink mask shows the threshold level based upon both the *Minimum profiling* and the *Minimum area* (if set). Only bands that exceed the threshold will be selected. If inappropriate, the settings can be changed in this preview window. The sensitivity of this search depends on the band search settings: if too many (false) peaks are found, or if real bands are undetected, you can change the search sensitivity using the band search filters as described above.

In addition, a blue mask shows the threshold level for bands that will be found as uncertain (*"Gray zone"*). All bands exceeding the pink mask but not exceeding the blue mask will become uncertain bands.

In the *Band search* dialog box, the currently selected pattern is shown and indicated in the status bar (bottom). To scroll through other patterns in the preview, press the < or > button (left and right from the curve).

You can search for bands on an individual lane by pressing <Search on this lane>, or on all lanes of the gel at once by pressing <Search on all lanes>.



If bands were already defined on the gel, the program will now ask: "There are already some bands defined on the gel. Do you want to keep existing bands?" when performing a band search. If you answer <No>, the existing bands will be deleted before the program starts a new search. By answering <Yes>, you can change the search settings and start a new search while any work done previously is preserved.

Bands that were found are marked with a green horizontal line, whereas uncertain bands are marked with a small green ellipse (see magnification in Figure 4.1.30).

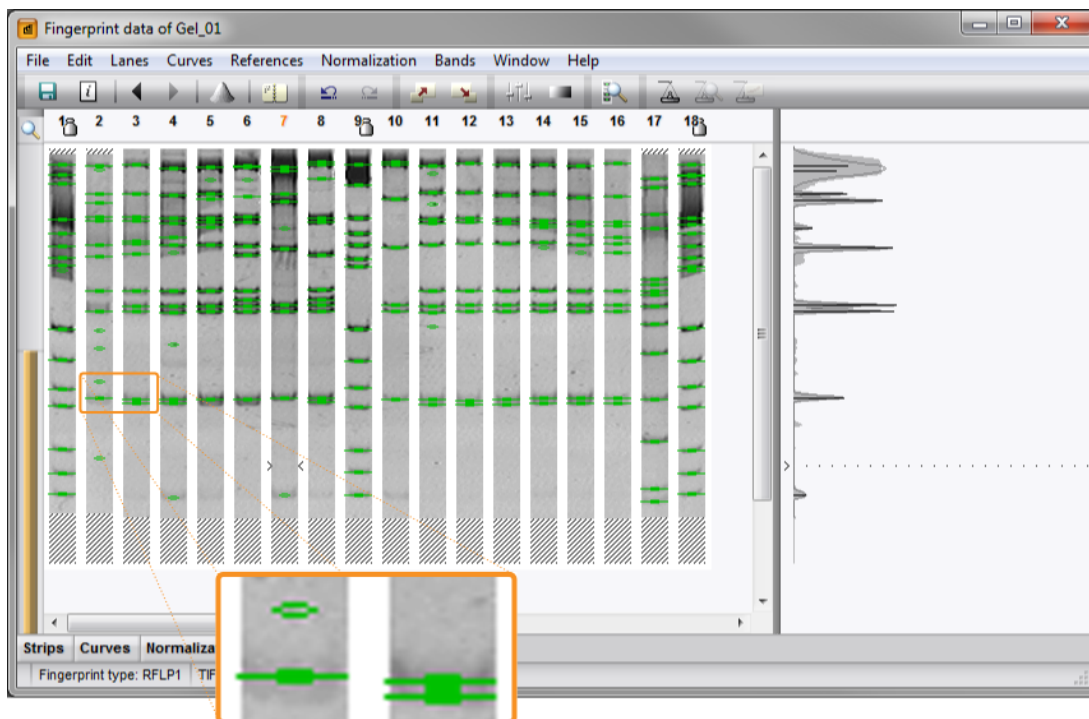


Figure 4.1.30: The *Fingerprint processing* window. Step 4. Bands.

A single band can be added with **Bands > Add new band (Enter)** or **Ctrl+click**.



The cursor automatically jumps to the closest peak; to avoid this, hold down the **Tab**-key while clicking on a band.



When there is evidence of a double band at a certain position, you can add a band over an existing one. Double bands (or multiplets) are indicated by outwards pointing arrows on the band marker: . Double uncertain bands are marked with a filled ellipse instead of an open ellipse. The clustering and identification functions using band based similarity coefficients (see 4.2 and 15) support the existence of double overlapping bands. For example, two patterns, having a single band and a double band, respectively, at the same position will be treated as having one matching and one unmatched band. Two patterns, each having a double band at the same position, will be treated as having two matching bands.

To select a group of bands, hold the **Shift**-key and drag the mouse pointer whilst holding the left mouse button. The group of bands can then be deleted with **Bands > Delete selected band(s) (Del)**, marked as uncertain with **Bands > Mark band(s) as uncertain (F5)** or marked as certain with **Bands > Mark band(s) as certain (F6)**.

Select **File > Show report** to generate a report of all settings defined for the processed fingerprint file. The report can be saved as HTML or text with **File > Save as html** and **File > Save as text** respectively.

Save the gel with **File > Save** , **Ctrl+S**.



When a gel is saved, bands that are defined outside the normalized range will automatically be removed. The software warns you for this.

4.1.3.7 Advanced band search using a size-dependent threshold

In many electrophoresis systems, the staining intensity of the bands is dependent on the size of the molecules. In DNA patterns stained with ethidium bromide for example (e.g. Pulsed-Field Gel Electrophoresis, PFGE), larger DNA molecules can capture many more ethidium bromide molecules than small DNA molecules, resulting in large size bands to appear much stronger than small size bands.

In other electrophoresis systems, the definition of the bands (sharpness) might depend on the size, which can also result in apparent different heights depending on the position on the pattern.

In such systems, a method that uses a single threshold parameter for finding bands on the patterns (i.e. the minimum profiling) might not work well: in case of PFGE for example, in the high molecular weight zone it might detect spots and irrelevant fragments whereas in the low molecular weight zone real bands might remain undetected.

In order to provide a more accurate band search for patterns with systematic dependence of intensity according to the position, BioNumerics provides a way to calculate a regression that reflects the average peak intensity for every position on the patterns in a given fingerprint type. The only requirement for this method is that a sufficient number of gels already needs to be processed, with the bands defined appropriately, before the regression can be calculated. The user can make a selection of entries from the database, and based upon that selection and the bands they contain in the fingerprint type, the regression is established.

To calculate the regression, a selection of entries should be present that contain experimental data for the fingerprint (see 3.3.8 for detailed explanation on search and select functions).

In the *Fingerprint type* window, select **Settings > Create peak intensity profile....** This pops up the *Peak intensity profile* dialog box (see Figure 4.1.31).

This dialog box shows a plot of all intensities of the selected patterns in function of the position on the pattern. Initially, the threshold factor is a flat line at 1.0. By pressing **<Calculate from peaks>**, a non-linear regression is automatically calculated from the scatter plot.

The regression line contains 5 nodes, of which the position can be changed independently by the user. To change a node's position, click and hold the left mouse button and move the node to the desired position. The regression can be reset to a flat line using the **<Reset>** button.

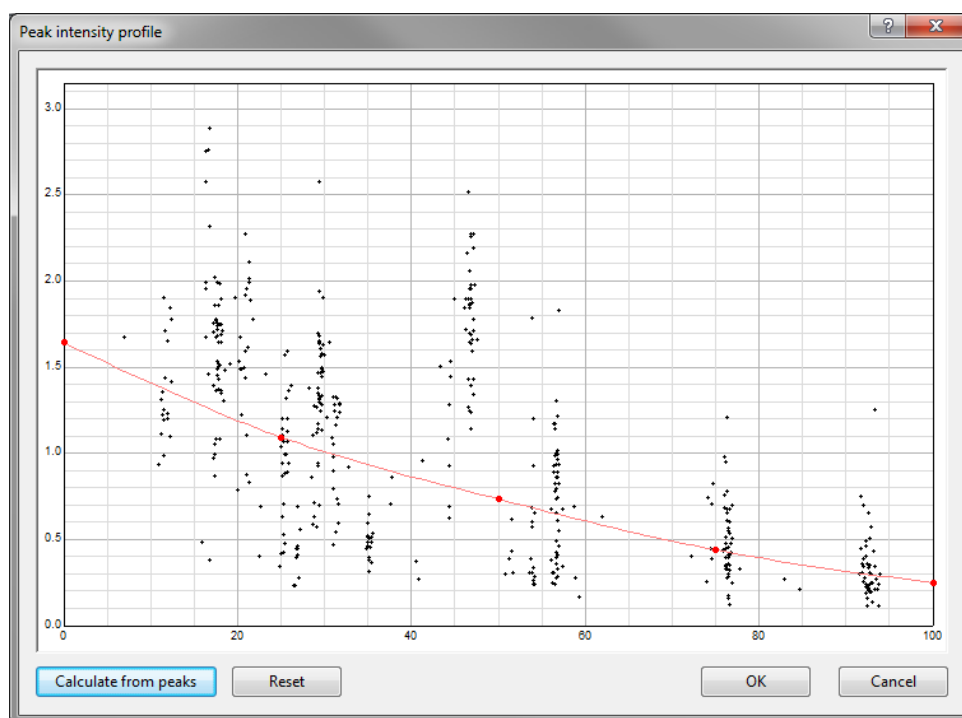



Figure 4.1.31: The *Peak intensity profile* dialog box with peak intensity regression curve.

To confirm and save the regression, press **<OK>**. The regression can be edited anytime later by opening the *Peak intensity profile* dialog box again.


As a result of creating a peak intensity regression curve, the **Minimum profiling** threshold will depend on the curve. The value entered for the **Minimum profiling** will correspond to the highest value on the intensity profile regression curve (the outermost left point in Figure 4.1.31). Therefore, after creating an intensity profile regression, you may have to increase the **Minimum profiling** setting to find the bands optimally: noise and irrelevant peaks will be filtered out in the high intensity areas whereas faint bands will still be detected in the low intensity areas.


4.1.3.8 Adding gel lanes to the database

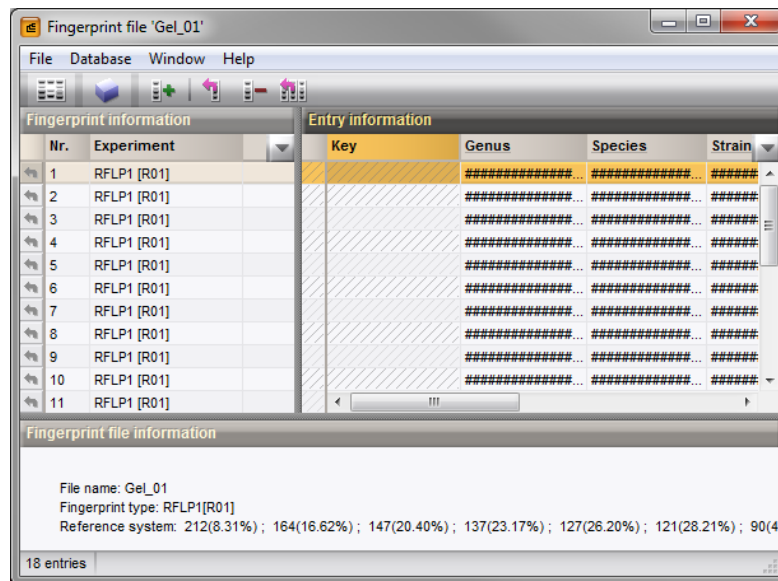
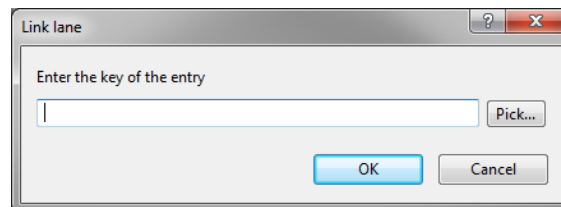
Linking gel lanes to database entries is done from the *Fingerprint* window. This window can be opened from the *Main* window by highlighting the gel file in the *Fingerprint files* panel and selecting **Edit > Open highlighted object...** (, **Enter**) (see Figure 4.1.32). After processing a gel, the *Fingerprint* window might still be open (see 4.1.3.2).

The *Fingerprint* window consists of three panels:

- The *Fingerprint information* panel lists the lanes that were defined on the gel.
- The *Entry information* panel shows the information of entries that correspond to the gel lanes.
- The *Fingerprint file information* panel shows the *Fingerprint type* of the gel, the *Reference system* according to which the gel is normalized, and the *reference positions* of this reference system.

When gel lanes are not linked to database entries yet, the *Entry information* panel will be empty and the link arrows in the *Fingerprint information* panel will be gray: .

To link a gel lane to an existing database entry, highlight the lane and select **Database > Link lane...** (). The *Link lane* dialog box pops up (see Figure 4.1.33).

Figure 4.1.32: The *Fingerprint* window.Figure 4.1.33: The *Link lane* dialog box.

The dialog box prompts you to enter the *key* of the database entry to which the experiment is to be linked. Pressing <**Pick...**> displays the *Select entry* dialog box, from which an entry can easily be picked. See 3.2.11 for more information about this dialog box.

If you try to link a lane to an entry which already has a lane of the same experiment type linked to it, the program will ask whether you want to create a **duplicate key** for this entry. This feature is useful in case you want to define experiments that are run in duplicate for one or more organisms. Rather than overwriting the first entry or disregarding duplicate entries, BioNumerics automatically considers them as duplicates and assigns an extension /#x to such duplicates. In case for a given entry a duplicate already exists (after import of another experiment), BioNumerics will automatically fill such existing duplicates that are still empty for the experiment type that is being imported. Database fields are automatically taken over from the "master" entry, i.e. the entry without extension. If the database fields from the "master" entry are changed, the /#x duplicates are automatically changed accordingly.



Levels and dependencies (3.3.10) offer a richer environment to deal with replicate experiments compared to the older concept of duplicate keys.

If you enter an entry key which does not already exist, the program asks whether you want to create an entry with that key.

Alternative to **Database > Link lane...** (🔗), a drag-and-drop procedure can be used: Drag the link arrow (🔗) from the *Fingerprint information* panel in the *Fingerprint* window over a database entry in the *Database entries* panel of the *Main* window. As soon as you pass over a database entry, the cursor shape changes into 🖱️. When the mouse button is released, the experiment is becomes linked and the link arrow will be purple: 🔗. In the *Entry information* panel, the corresponding entry information will be displayed. Entry information can also be edited in this panel: double-clicking on the entry calls the corresponding *Entry*

window and clicking twice on the same information field enables direct editing.

As soon as an experiment is linked to a database entry, the *Experiment presence* panel shows a colored dot for the experiment of this entry. You can click on the dot, which pops up the *Experiment card* window for that experiment (see 4.1.7).

In case the layout of the current gel is identical to another gel, of which the lanes are already linked to database entries, you can use **Database > Link all lanes...** (🔗).

The *Link to other file* dialog box will prompt for the name of the gel file to link to. When <OK> is pressed, the lanes of the current gel will be linked to the same database entries as the lanes of the previous gel.

If no database entries are defined for the current gel lanes, you can have the program create new entries and link the gel lanes automatically by selecting **Database > Add all lanes to database**. All unlinked lanes will be added as new entries to the database, with the gel lanes linked to these entries. If you do not wish to add all lanes to the database, highlight a lane and use **Database > Add lane to database** (➕) to add the lane individually.

A gel lane can be unlinked from a database entry using **Database > Remove link** (🔗). All entries from the gel are unlinked at once using **Database > Remove all links...**

In some cases, a gel can be composed of patterns belonging to different fingerprint types. For example, if you are running digests by three different restriction enzymes for the same set of organisms, for some remaining entries, you may want to run all three restriction enzyme digests on the same gel. In this case, you should process the gel according to one of the fingerprint types, and then, in the *Fingerprint* window, highlight a lane that belongs to another fingerprint type and use **Database > Change fingerprint type of lane....** A condition for this feature to work is that both fingerprint types are based upon the same reference system (the same set of reference markers, defined consistently using the same names). If the reference system for both fingerprint types is not the same, the software can still use the molecular weight calibration curves as a basis for conversion, if these are defined.

When a reference marker is linked to a database entry, it can be set as the standard profile with **Database > Set lane as standard....** See 4.1.5.8 for more information about this feature.

With **File > Add fingerprint information field...**, the *Add field* dialog box is called (see Figure 4.1.34).

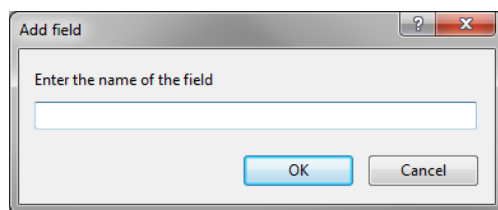


Figure 4.1.34: The *Add field* dialog box.

A fingerprint information field can be created to store information specific to individual fingerprint lanes. Recording lane-specific information could be useful e.g. to comment on PCR (RAPD, AFLP) or restriction digest (RFLP) efficiency of individual reactions. The information will appear (and can be edited) in the *Fingerprint information* panel and in the *Experiment card* window (see 4.1.7). Fingerprint information fields can also be used in the *Advanced query tool* (see 3.3.9).

4.1.3.9 Quantification of bands

The *Densitometric curve* panel in the fourth step of the *Fingerprint processing* window shows the densitometric curve of the selected pattern. For each band found, the program automatically calculates a best-fitting Gaussian curve, which makes more reliable (one-dimensional) quantification possible. Figure 4.1.35 shows a strongly zoomed band with its densitometric representation and the Gaussian fit (red). The blue points are

dragging nodes where you can change the position and the shape of the Gaussian fit for each band separately.

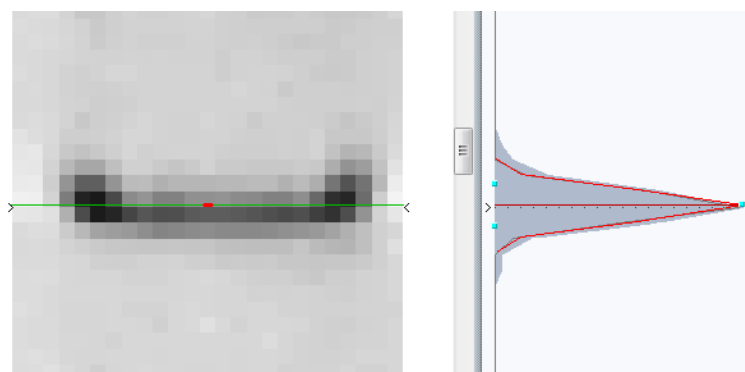


Figure 4.1.35: Zoomed band with its densitometric curve and best-fitting Gaussian approach.

It is possible to generate a text file or a printout of the complete band information of the gel, by selecting the command **File > Export report...** or **File > Print report...**, respectively. The output lists all the bands defined for each pattern with their normalized relative positions, the metrics (e.g. molecular weight), the height, and relative one-dimensional surface, as calculated by Gaussian fit.

To start calculating a *two-dimensional* quantification on the defined bands, first select **Bands > Quantification > Band quantification** (🔍). This action will bring the *Fingerprint processing* window in *quantification mode*: the quantification button now shows as 📊 and two additional band quantification buttons are displayed.

To find the surfaces (contours) of all bands, use **Bands > Quantification > Search all surfaces** (🔍). You can search the surface of the highlighted band alone with **Bands > Quantification > Search surface of band**, this can be useful e.g. if you have manually added a band later on.

Once the contours are found, the program shows for each highlighted band its *volume* in the status bar: the sum of the densitometric values within the contour.

The contours of a highlighted band can be changed manually by holding the **Ctrl**-key and dragging with the mouse to correct the upper and lower contours. This works best when zoomed in strongly on the band.

For known reference bands, you can enter a concentration value by highlighting the band and selecting **Bands > Quantification > Assign value....** Known reference bands are marked with Ⓜ.

Once multiple reference bands are assigned their concentrations, a regression to determine each unknown band concentration is calculated by selecting **Bands > Quantification > Calculate concentrations...** (📊). The *Quantification* window pops up (see Figure 4.1.36).

This window shows the real concentration in function of the band volumes, using cubic spline regression functions.

Save the gel with **File > Save** (💾, **Ctrl+S**) in order to store the quantification data.

When a text file or a printout is generated of the band information (by selecting **File > Export report...** or **File > Print report...**, respectively), the output will list all bands defined for each pattern with their normalized relative positions, the absolute volume, and if regression is done, the relative volume as determined by the calibration bands.

The *Fingerprint processing* window can be closed with **File > Exit**. The program asks: "Settings have been changed. Do you want to use the current settings as new defaults?". This question is asked when changes have been made to the fingerprint type-related settings, for example the gel strip thickness, the rolling disk size, etc.. If you answer **<Yes>**, the settings used for this gel will be saved in the fingerprint type's settings, and all new gels will be processed using the same settings.

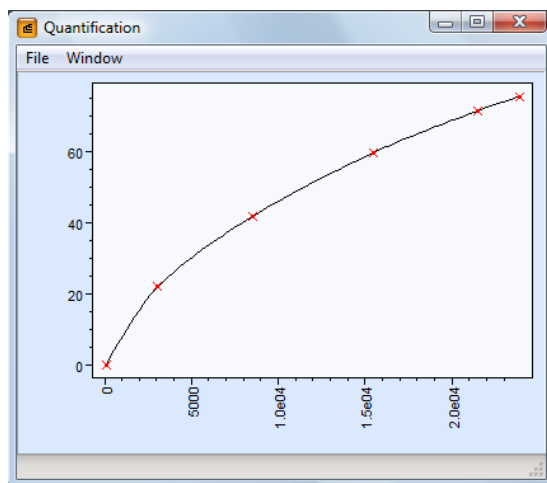


Figure 4.1.36: The *Quantification* window: concentration in function of known band volumes.



Answering **<Yes>** to the above question has the same effect as **Edit > Save as default settings...** in the *Fingerprint processing* window. Conversely, the current default settings can be copied to the current gel with **Edit > Load default settings....**

4.1.4 Importing and processing capillary sequencer curves

4.1.4.1 Import

4.1.4.1.1 Introduction

Automated sequencers using capillary electrophoresis technology were originally developed for high throughput Sanger sequencing. While they are gradually being replaced by sequencing-by-synthesis technology for this application, because of their speed, reproducibility and high resolution, such "genetic analyzers" are nowadays more often used for fragment analysis. Examples of fingerprint type experiments currently run on automated sequencers include amplified fragment length polymorphism (AFLP), terminal restriction fragment length polymorphism (T-RFLP), variable number tandem repeat (VNTR) analysis, multiplex ligation-dependent probe amplification (MLPA), etc..

Automated sequencers allow different fingerprints to be *pooled* using different color dyes and run together on the same capillary column. A *pool* (sometimes called *panel*) is a mixture of usually 4 or 5 dyes, one being a *reference sample* for fragment size calculation, the others containing each a profile for a given strain or sample. The pooling can happen through multiplex PCR or the amplification products can be mixed after PCR. The number of PCR amplicons that can be pooled together depends on (1) the number of color dyes used: if 5 color dyes are used, 4 differently labeled fragments can be pooled (one dye contains the reference sample), and (2) the possibility to combine PCR-amplicons with significantly different lengths.

Figure 4.1.37 shows a typical setup where 4 PCR products are mixed in one pool, using 5 color dyes. Since the total number of PCR targets per sample to be tested is 8, two pools are generated, each containing 4 PCR products.

When fingerprints are run on a capillary sequencer, the resulting data can have two fundamentally different formats:

- **Curve files** (also referred to as electropherograms, chromatogram files or trace files): The binary encoded raw data as produced by the capillary electrophoresis equipment. Since it is the raw data,

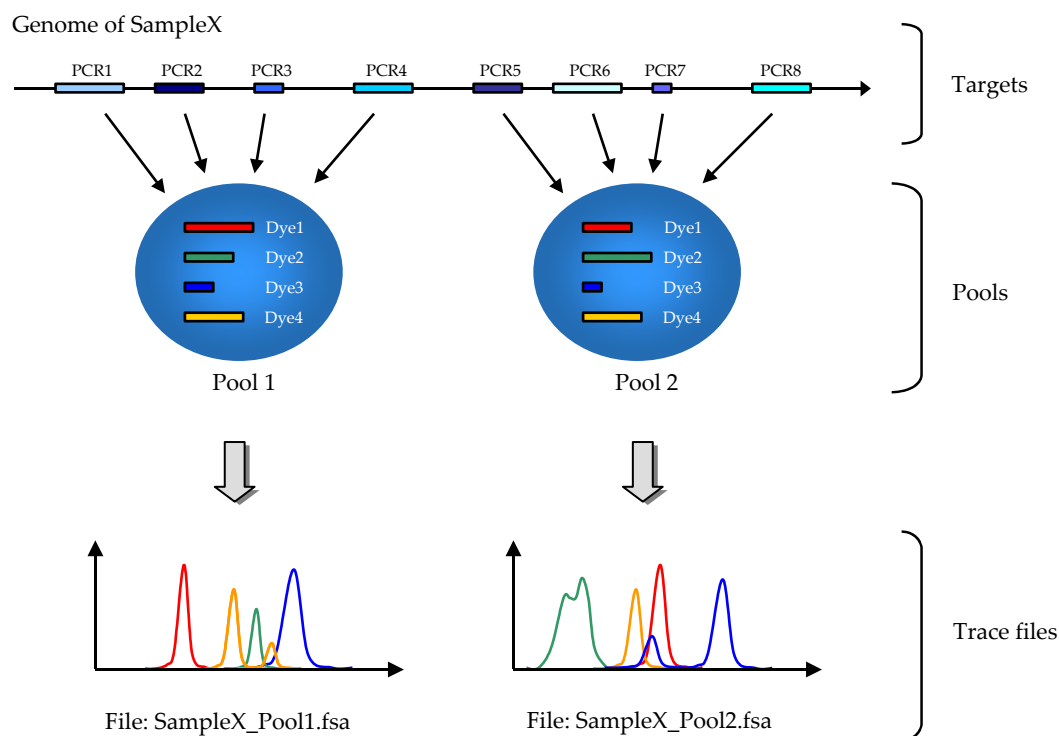


Figure 4.1.37: Schematic overview of the relation between PCR targets, pools and curve files in a typical pooled experimental setup. In this example, only one target gene is used per dye in the same pool.

it contains the complete information but some fingerprint preprocessing (e.g. normalization, band assignment, ...) in BioNumerics is required.

- **Peak tables:** Text files containing a listing of peaks from the chromatograms, with their corresponding metrics (sizes in base pairs) and peak height and/or peak area. This type of data has been processed by the software which controls the capillary electrophoresis equipment.



In contrast to the .fsa files generated by Applied Biosystems sequencers, the .scf raw curve files generated by Beckman-Coulter equipment are *not* corrected for spectral overlap of the fluorescent dyes. This is generally not a problem when a single data channel is used. However, in experimental setups where all four dyes are employed (i.e. three data channels and one reference channel), it is advised to import .crv files in BioNumerics. The latter files can be exported from the Beckman-Coulter software and contain curves which are corrected for cross talk.

4.1.4.1.2 Importing raw sequencer curve files

A batch of Applied Biosystems curve files composing one run can be downloaded from the Applied Maths website (<http://www.applied-maths.com/download/sample-data>, click on "AB sequencer trace files").

Selecting **Import curves** under **Fingerprint type data** in the **Import** dialog box and pressing <**Import**> calls the **Input** wizard page (see Figure 4.1.38).

Pressing the <**Browse**> button allows you to select the raw chromatogram files that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. Supported formats are Applied Biosystems (.fsa) and Beckman-Coulter (.scf). The number of files and total size is displayed below the list.

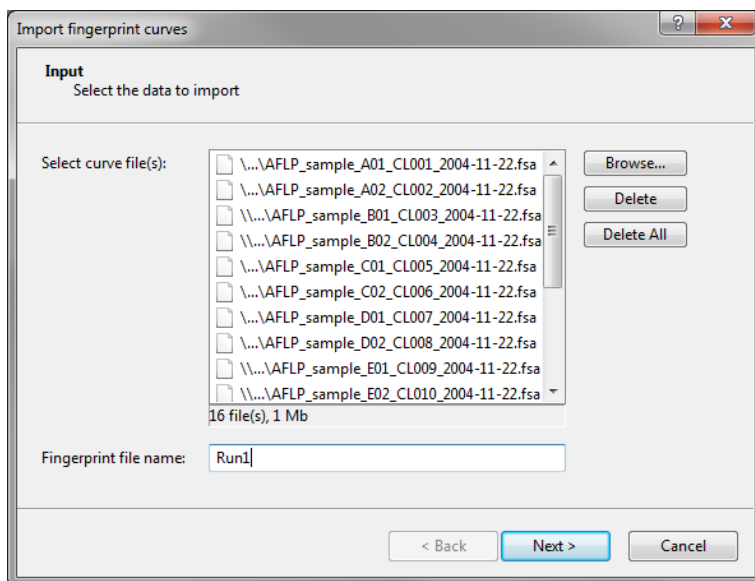


Figure 4.1.38: The *Input* wizard page.

With the **<Delete>** button all selected files are removed from the import list. All files are deleted at once from the import list when pressing **<Delete All>**.

The default suggested **Fingerprint file name** is the folder name but this can be changed to any other name. During import, BioNumerics will split the curve files into separate fingerprint files per imported dye. The fingerprint file name is composed of the **Fingerprint file name** plus a suffix referring to the dye name.

Pressing **<Next>** calls the *Import template* wizard page.

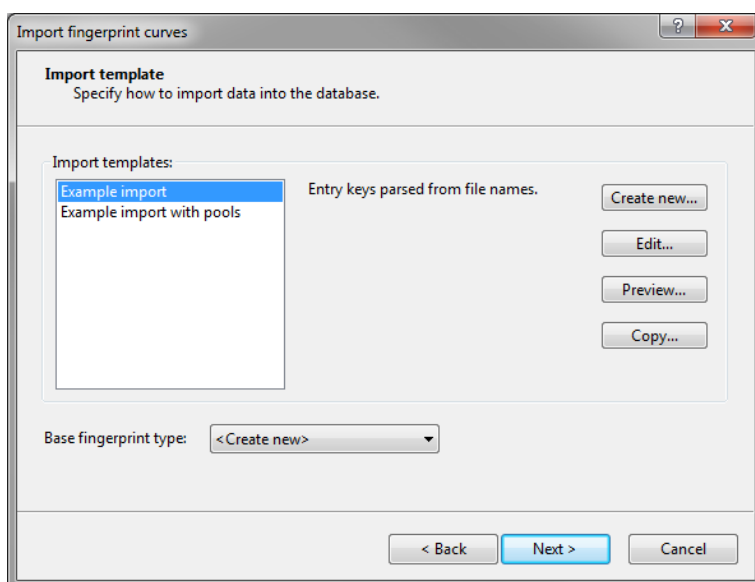


Figure 4.1.39: The *Import template* wizard page.

The way the curve information should be imported in the database can be specified with an import template. The *Import templates* panel lists all curve templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up the *Import rules* dialog box allowing you to define a new import template.

The *Import rules* dialog box lists the information present in the selected files as **Source**, their linked **Source**

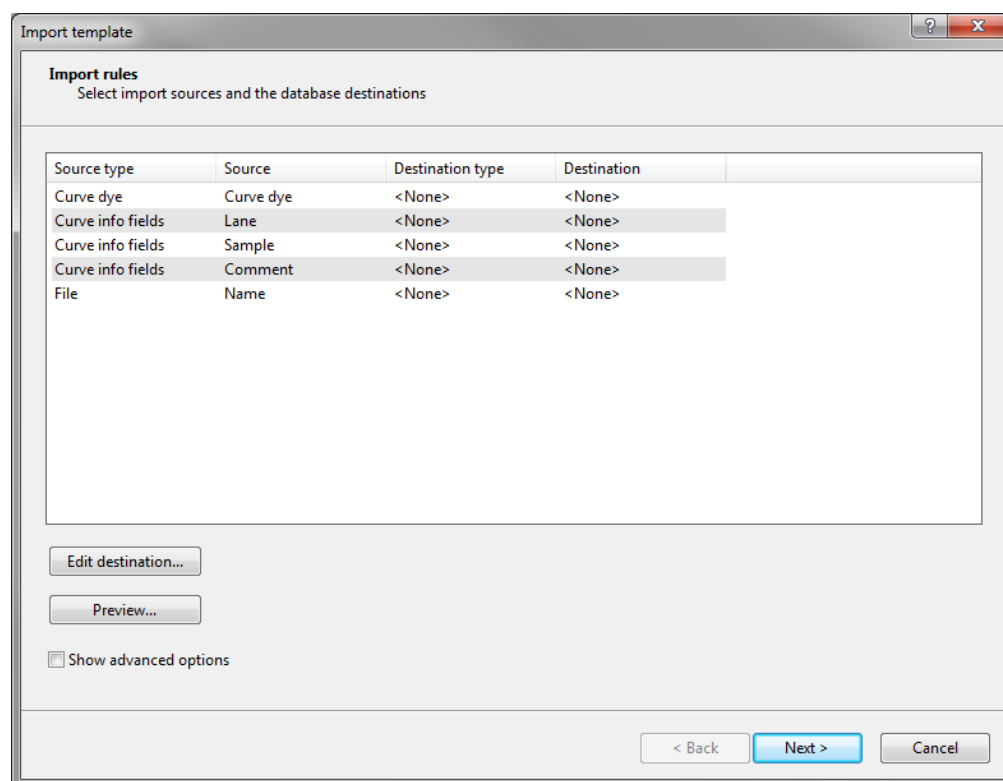


Figure 4.1.40: The *Import rules* dialog box.

type and the **Destination** component they are associated with (initially all set to <None>).

Using the last row in the grid, the (parsed) file name of the selected file can be stored in the database. The text **File** is specified in the **Source type** column and the text **Name** is displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields, lane information fields, fingerprint dyes, fingerprint pools or fingerprint type experiments.

Specifying a *destination* for one or more selected rows can be done by pressing the <**Edit destination**> button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

When only one row is selected in the grid, the information of this row can be linked to:

- The default information field **Key**.
- A **Fingerprint type** name. The (parsed) information will hold the fingerprint type name.
- A **Fingerprint pool**. The (parsed) information will hold the pool information.
- A **Fingerprint dye**. The (parsed) information will hold the dye information.
- A new or existing non-default entry information field (select the <**Create new**> option or an existing field under the topic **Entry info field**, respectively).

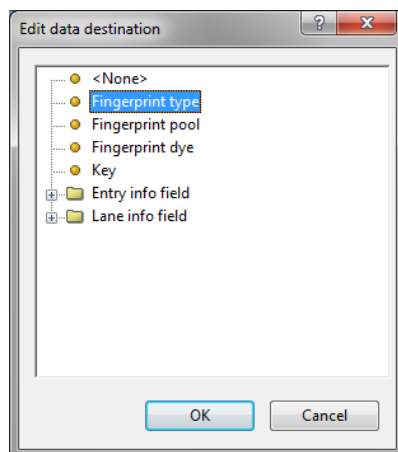


Figure 4.1.41: Edit data destination for a single selected row entry.

- A new or existing lane information field (select *<Create new>* or select an existing field under the topic *Lane info field*, respectively).

If a row is linked to a new entry information field or a new lane information field, a new dialog box pops up when pressing the *<OK>* button.

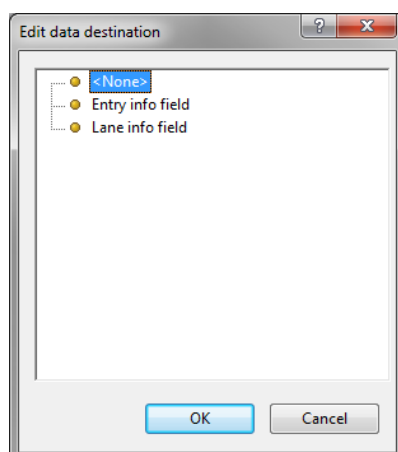


Figure 4.1.42: Edit data destination for multiple selected row entries.

When multiple rows are selected in the grid, the information of these rows can be linked to:

- Non-default entry information fields (select the *Entry info field* option).
- Lane information fields (select the *Lane info field* option).

When pressing the *<OK>* button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields or lane information fields in the database. If no entry information fields or lane information fields exist with the same name, a new dialog box pops up prompting for the names.

Pressing *<Preview>* opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the *<Close>* button.

When the *<Show advanced options>* check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in [3.3.5.5](#).

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database. Pressing the **<Next>** button calls the *Import data* dialog box.

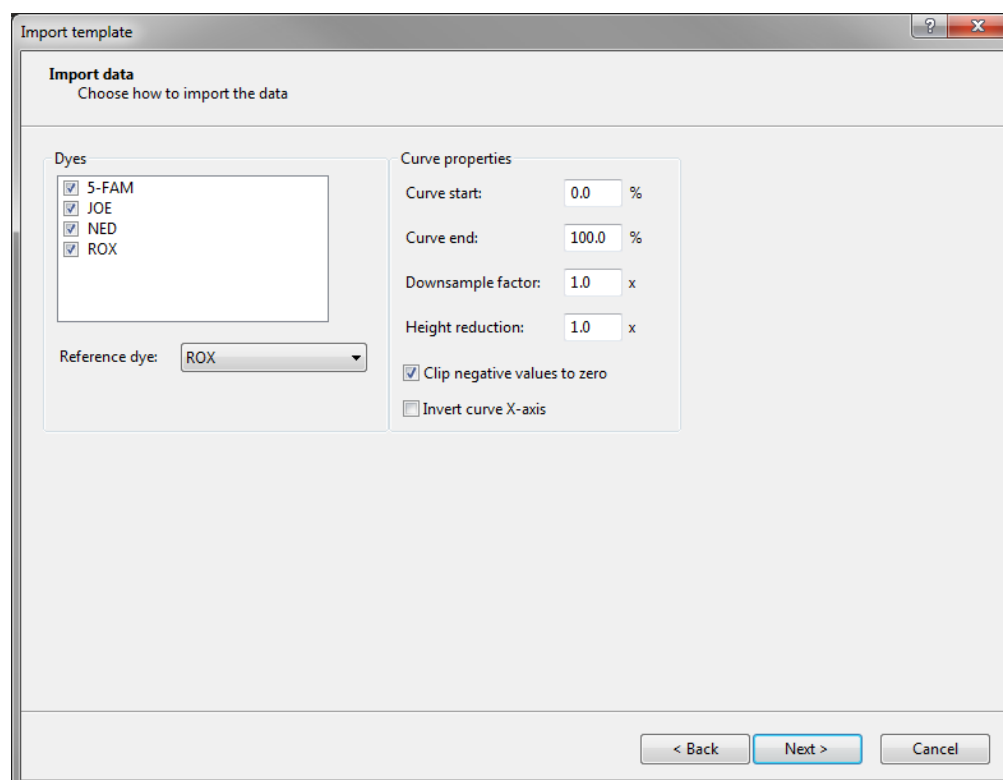


Figure 4.1.43: The *Import data* dialog box.

Dyes:

- BioNumerics reads the dye names from the curve files and displays all dyes in the grid. All channels are by default checked to be imported in the database, but can be unchecked e.g. if one or more channels do not contain any signal.
- One of the dyes can be specified as being the **Reference dye**, i.e. containing a set of molecular size markers to normalize the other channels.

Curve properties:

- The **Curve start** and **Curve stop** positions are default set to 0% and 100% respectively. Based on these settings, the complete curves are imported.
- In case of over-sampled readings (containing far more densitometric values than required to describe the curve), it is recommended to apply a **Down sample factor** so that the final resolution is less than 5000. The factor is default set to "1.0" (= no reduction of points). If the factor is set to e.g. "4", the number of points is reduced four times.
- Any peak with a height exceeding the OD range of the fingerprint experiments will appear truncated. To avoid this, a **Height reduction** can be applied. The **Height reduction** factor is by default set to "1.0" (= no height reduction). If the **Height reduction** is set to e.g. "2", the heights are reduced by a factor two.
- Points in the curves that have negative values can be scaled to zero with **Clip negative values to zero**.

- Checking the option ***Invert curve X-axis*** will invert the curves (fragments at the top of the curve will appear at the bottom and vice-versa) right before they are added to the database, i.e. **after** any resampling has been applied.

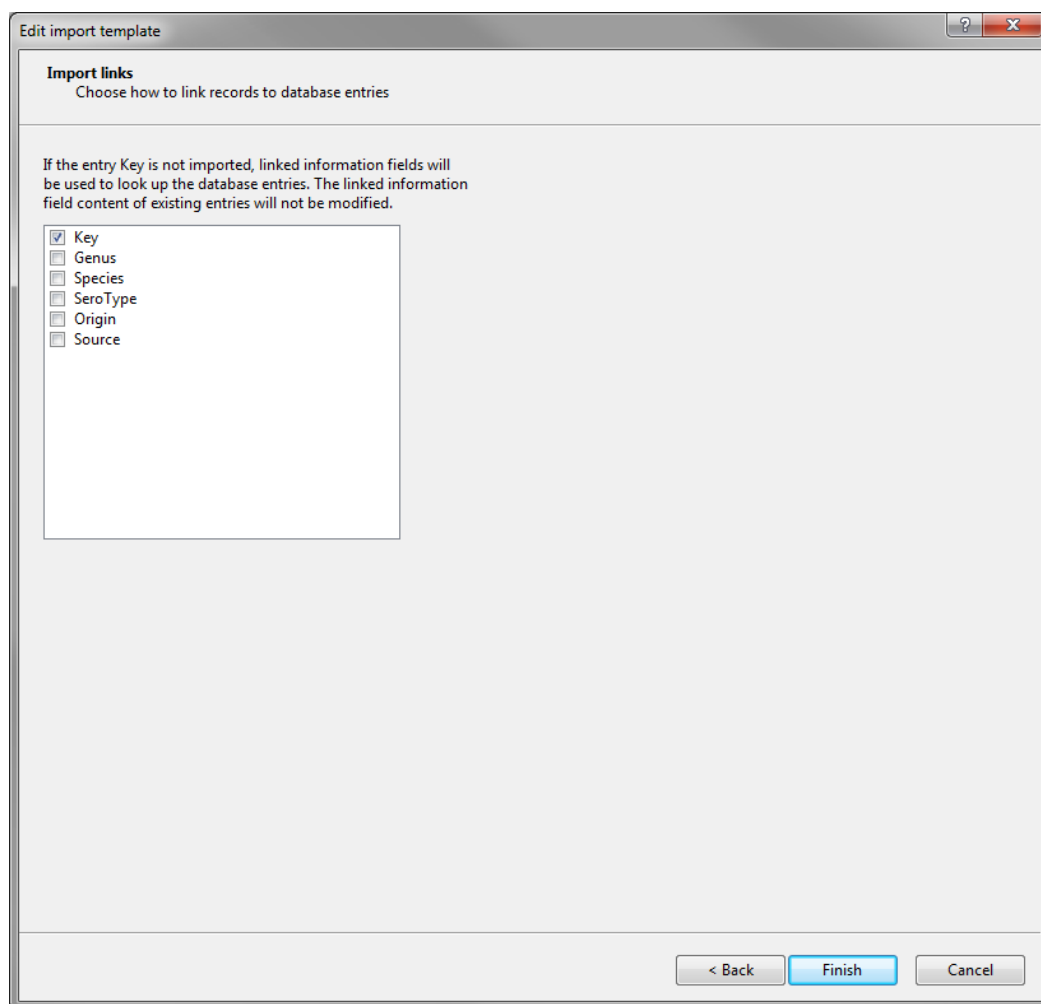


Figure 4.1.44: Specify the entry link field.

- If a row in the grid is linked to the ***Key*** field in the database, ***Key*** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the ***Key*** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the ***Key*** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option ***Create x entries*** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

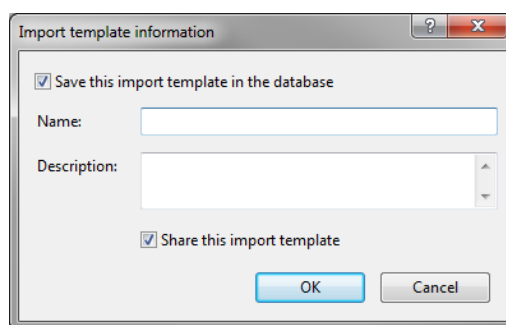


Figure 4.1.45: The *Import template information* dialog box.

Each import template has its own unique *Name*.

Optionally, a descriptive text string can be entered in the *Description* input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option *Save this import template in the database* is checked.

Check or uncheck the option *Share this import template* when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template *Name* is shown in the *Import templates panel* and is automatically selected. The template *Description* is shown in the right panel (see Figure 8.1.104).

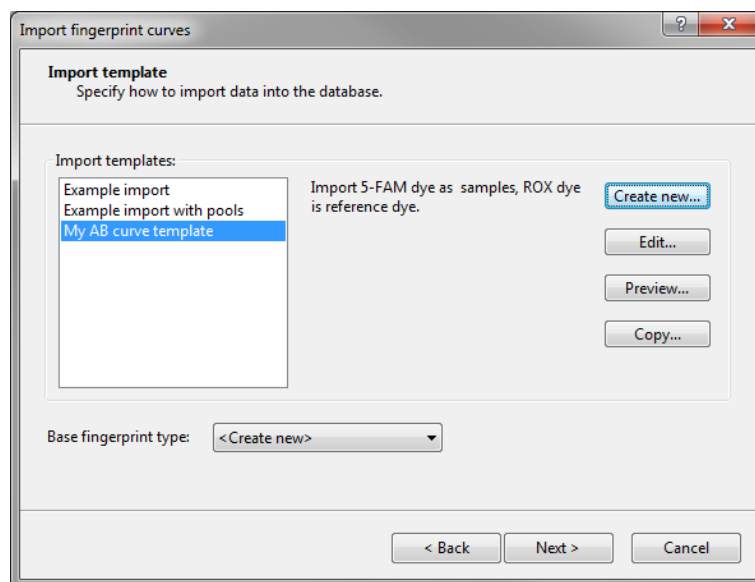


Figure 4.1.46: Import template added to the list.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the *Source* column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays

the parsed information using the template settings. The preview can be closed with the **<Close>** button.

A **Base fingerprint type** experiment needs to be specified. The settings of the base fingerprint type (including the reference system and band search settings) will be copied to all fingerprint types that are created when fingerprint data is imported in the database. An existing fingerprint type experiment can be selected from the list, or a new experiment can be created (**Create New**).

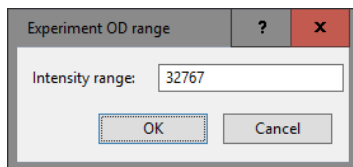


Figure 4.1.47: The *Experiment OD range* dialog box.

When new fingerprint experiment are created during import, the OD (or intensity) range should be specified in the *Experiment OD range* dialog box, corresponding to the files at hand. Default **15-bit (32768 values)** is specified ($2^{15} = 32768$ possible OD values), but this can be changed to any other bit value: e.g. **12-bit (4096 values)**, **16-bit (65536 values)**,

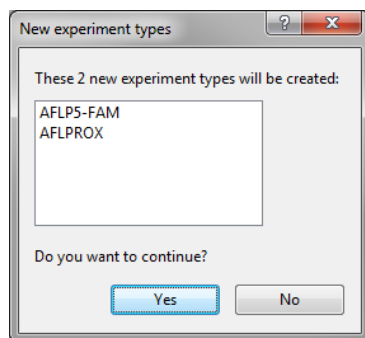


Figure 4.1.48: Missing fingerprint types.

A fingerprint type needs to be present in the database for each pool (if present) and dye combination. The names of these fingerprint types are composed of the base fingerprint type name, followed by the pool name (if present), and the name of the dye (e.g. **AFLPROX**). If one or more of these fingerprint types are not present in the database, the *New experiment types* dialog box pops up, listing all missing fingerprint types. The user needs to confirm the creation of the missing fingerprints.

The *Database links dialog* prompts for some additional settings:

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing **<Next>** calls the *Processing* wizard page.

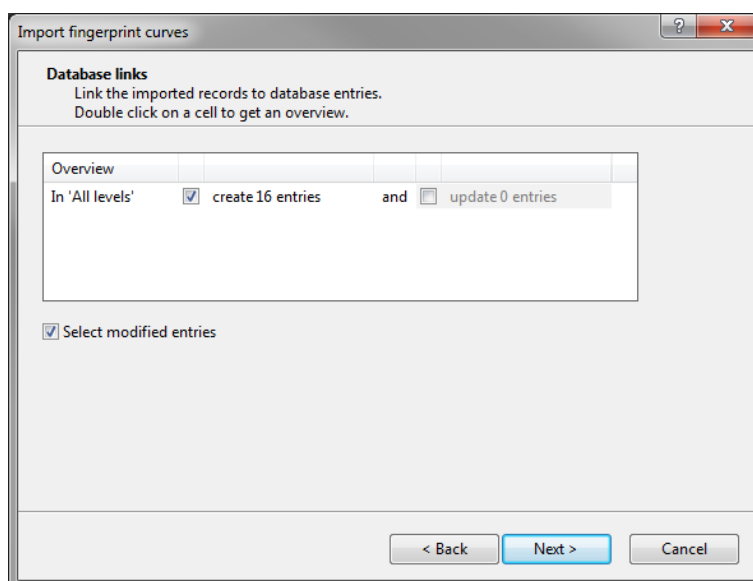


Figure 4.1.49: Link the entries to new or existing in the database.

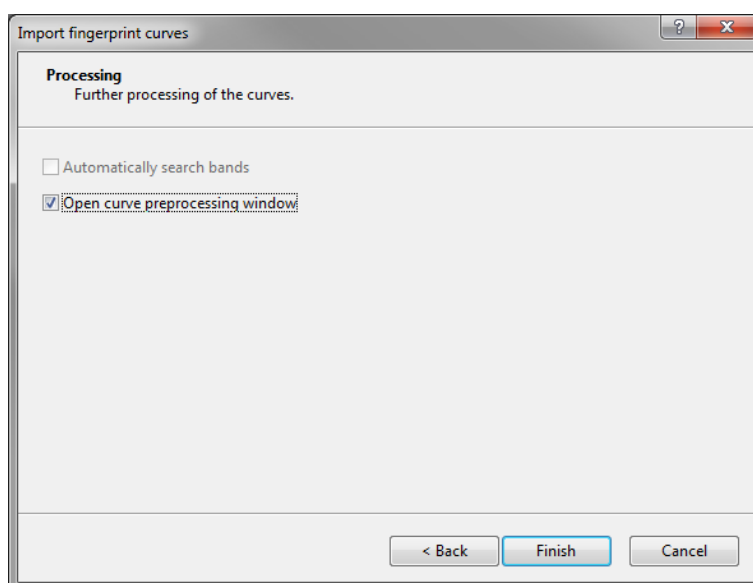


Figure 4.1.50: The *Processing* wizard page.

When the option ***Automatically search bands*** is checked, an automatic search for data bands is performed in the *Fingerprint curve processing* window using the data band settings specified for each fingerprint type experiment included in the import. This option will not be available if no reference system is available for a fingerprint type experiment.

When the option ***Open curve preprocessing window*** is checked the *Fingerprint curve processing* window will open, displaying all imported fingerprint curves.

Pressing **<Finish>** will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.



Mapped lane field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

For each dye checked in the *Dyes panel* of the *Import data* dialog box, a new fingerprint file is created, composed of the file name specified (see Figure 4.1.38) and the name of the dye. These files are displayed in the *Fingerprint files* panel. The reference file is shown in the **Link** column. Double-clicking on a fingerprint file opens the *Fingerprint* window. If lane information was imported with the individual lanes, this information is displayed in the *Fingerprint information* panel (see Figure 4.1.51).

Fingerprint information					Entry information		
Nr.	Experiment	Lane	Sample	Comment	Key	Level	Modified date
1	AFLPS-FAM [None]	1	AFLP_sample		CL001		2014-02-17 14:20:49
2	AFLPS-FAM [None]	2	AFLP_sample		CL002		2014-02-17 14:20:49
3	AFLPS-FAM [None]	3	AFLP_sample		CL003		2014-02-17 14:20:49
4	AFLPS-FAM [None]	4	AFLP_sample		CL004		2014-02-17 14:20:49
5	AFLPS-FAM [None]	5	AFLP_sample		CL005		2014-02-17 14:20:49
6	AFLPS-FAM [None]	6	AFLP_sample		CL006		2014-02-17 14:20:49
7	AFLPS-FAM [None]	7	AFLP_sample		CL007		2014-02-17 14:20:49
8	AFLPS-FAM [None]	8	AFLP_sample		CL008		2014-02-17 14:20:49
9	AFLPS-FAM [None]	9	AFLP_sample		CL009		2014-02-17 14:20:49
10	AFLPS-FAM [None]	10	AFLP_sample		CL010		2014-02-17 14:20:49
11	AFLPS-FAM [None]	11	AFLP_sample		CL011		2014-02-17 14:20:49
12	AFLPS-FAM [None]	12	AFLP_sample		CL012		2014-02-17 14:20:49
13	AFLPS-FAM [None]	13	AFLP_sample		CL013		2014-02-17 14:20:49
14	AFLPS-FAM [None]	14	AFLP_sample		CL014		2014-02-17 14:20:49
15	AFLPS-FAM [None]	15	AFLP_sample				

Fingerprint file information

File name: Run1_5-FAM
Fingerprint type: AFLP[None]
Reference system:

16 entries

Figure 4.1.51: The *Fingerprint* window.

The imported fingerprint lanes are linked to new or existing entries in the database depending on the options checked in the *Database links dialog*.

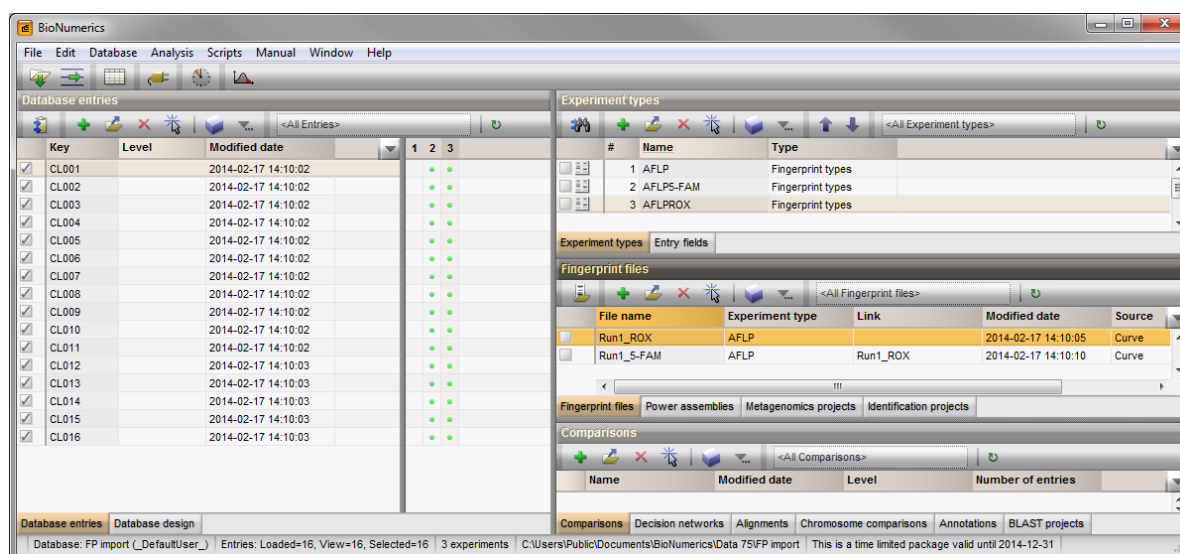
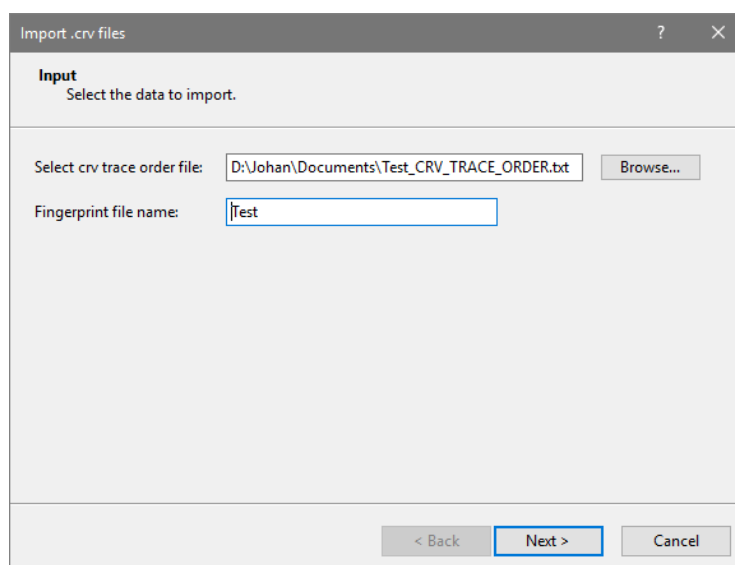
The lanes are linked to the corresponding fingerprint "dye" type. The names of these fingerprint types are composed of the base fingerprint type name, followed by the pool name (if present), and the name of the dye. The fingerprint type experiments are displayed in the *Experiment types* panel.

When the option **Open curve preprocessing window** was checked the *Processing* wizard page, the *Fingerprint curve processing* window pops up, displaying all imported fingerprint curves. See 4.1.4.2 for more information about the processing of curves.

4.1.4.1.3 Importing CRV files

CRV files are not truly raw fingerprint curve data, but rather a format which can be exported by the Beckman-Coulter software that contains chromatograms corrected for spectral overlap of the fluorescent dyes. A single .crv file contains data from several samples, for a given channel. Typically, up to four .crv files and a single .txt file are generated per export. The text file has the suffix _CRV_TRACE_ORDER.txt and contains the sample names in the same order as they occur in the .crv files.

To import a set of CRV files, select **Import .crv files** under *Fingerprint type data* in the *Import* dialog box and press <Import>. This action starts the *Import .crv files* wizard (see Figure 4.1.53).

Figure 4.1.52: The *Main* window after import of the data.Figure 4.1.53: The *Input* wizard page in the *Import .crv files* wizard.

Pressing the **<Browse>** button allows you to select the text file containing the order of the traces. BioNumerics will look in the same directory for the corresponding .crv files.

The suggested **Fingerprint file name** is the folder name, but can be changed to any other name. The actual fingerprint file names are composed of the **Fingerprint file name** shown here, plus a suffix referring to the dye name.

Pressing **<Next>** calls the *Import rules* dialog box (see also 4.1.4.1.2).

In case of CRV files import, the **Source** 'Sample' contains the sample names as obtained from the text file. If the text is too long to store in an information field, it needs to be parsed (press the **<Edit parsing...>** button, which comes available when **Show advanced options** is checked).

Pressing **<Next>** calls the *Import data* dialog box (see Figure 4.1.54).

BioNumerics reads the dye names from the curve files and displays all dyes in the grid. All channels are by default checked to be imported in the database, but can be unchecked e.g. if one or more channels do not

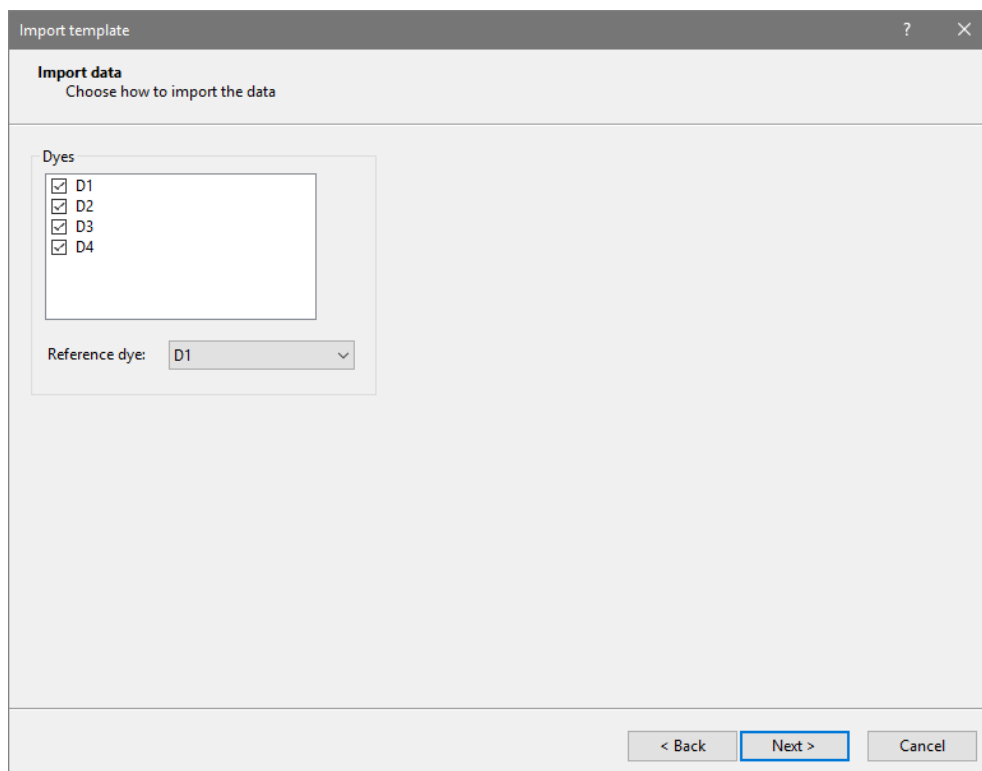


Figure 4.1.54: The *Import data* dialog box.

contain any signal.

One of the dyes can be specified as being the **Reference dye**, i.e. containing a set of molecular size markers to normalize the other channels.

Pressing **<Next>** opens the *Import links* dialog box. The remaining part of the .crv files import procedure is actually identical to the import of raw curve files (see 4.1.4.1.2).

4.1.4.1.4 Importing peak data from peak table files

BioNumerics allows the input of band size and band position tables, and reconstruct fingerprints of these, based upon the size and the amplitude (area or height) of the peaks.

An example Beckman peak file can be downloaded from the Applied Maths website (<http://www.applied-maths.com/download/sample-data>, click on "VNTR sample peak table").

An example Applied Biosystems peak file can be downloaded from the Applied Maths website (<http://www.applied-maths.com/download/sample-data>, click on "MIRU-VNTR sample data").

Selecting the **Import peak table** option from the *Import* dialog box and pressing **<Import>** calls the *Input wizard* page (see Figure 4.1.55).

Pressing the **<Browse>** button allows you to select the peak table file that you want to import, located on your computer, external drive or on a network location.

If the option **Import as fingerprint file** is checked, densitometric curves are created using the height, size and area information that is present in the peak file. For each **Dye** and **Fingerprint file name** combination, a new fingerprint file will be created. The default suggested **Fingerprint file name** is the file name but this can be changed to any other name.

When the option **Import as fingerprint file** is unchecked, the peaks will directly be imported in the database.

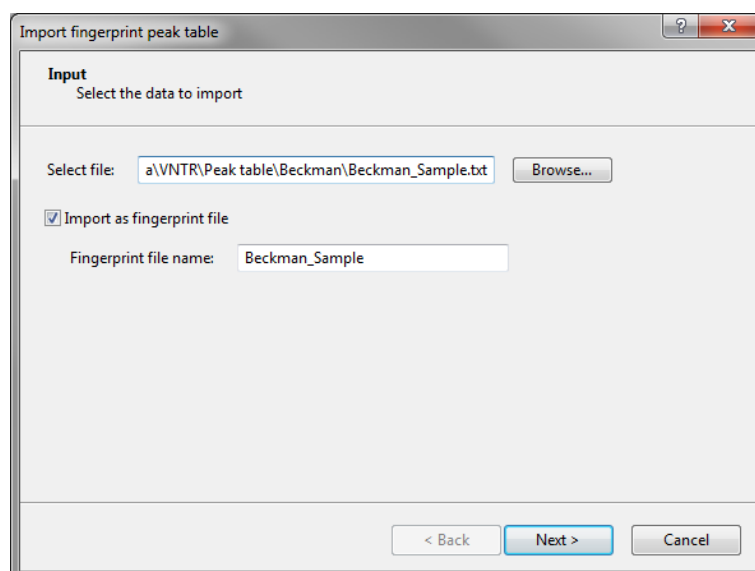


Figure 4.1.55: The *Input* wizard page.

Since no fingerprint files are created when this option is checked, no file name needs to be defined.

Pressing <*Next*> calls the *Import template* wizard page.

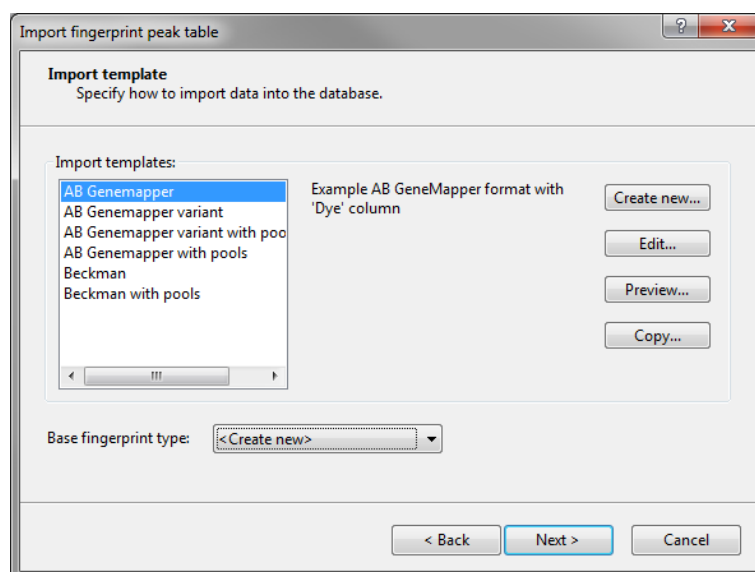


Figure 4.1.56: The *Import template* wizard page.

The way the peak information should be imported in the database can be specified with an import template. The *Import templates* panel lists all curve templates that have been created and stored in the database.

The import comes with five predefined formats to make commonly used *AB GeneMapper* and *Beckman* peak files easy to import. All predefined formats require the presence of sample information (*Sample File*), size information (*Size*), peak height information (*Height*), peak area information (*Area*), marker (*Marker*) and dye (*Dye*) information in the peak files. Each predefined format has its own set of names that are expected to be present in the peak file (Table 4.1.1). If a format name could not be matched with a column name in the peak file, this is displayed as an error message in the preview.

Pressing the <*Create new*> button brings up the *Import rules* dialog box allowing you to define a new import template. Peak files that do not contain all the information required by the predefined formats (e.g.

	SAMPLE FILE	DYE	SIZE	HEIGHT	AREA	MARKER
Beckman	RN	dye	est frag size	pk height	pk area	size std
Applied Biosystems	sample file	dye/sample peak	size	height	area	marker

Table 4.1.1: Columns names expected by the formats.

peak file without area information) can be imported using this option.

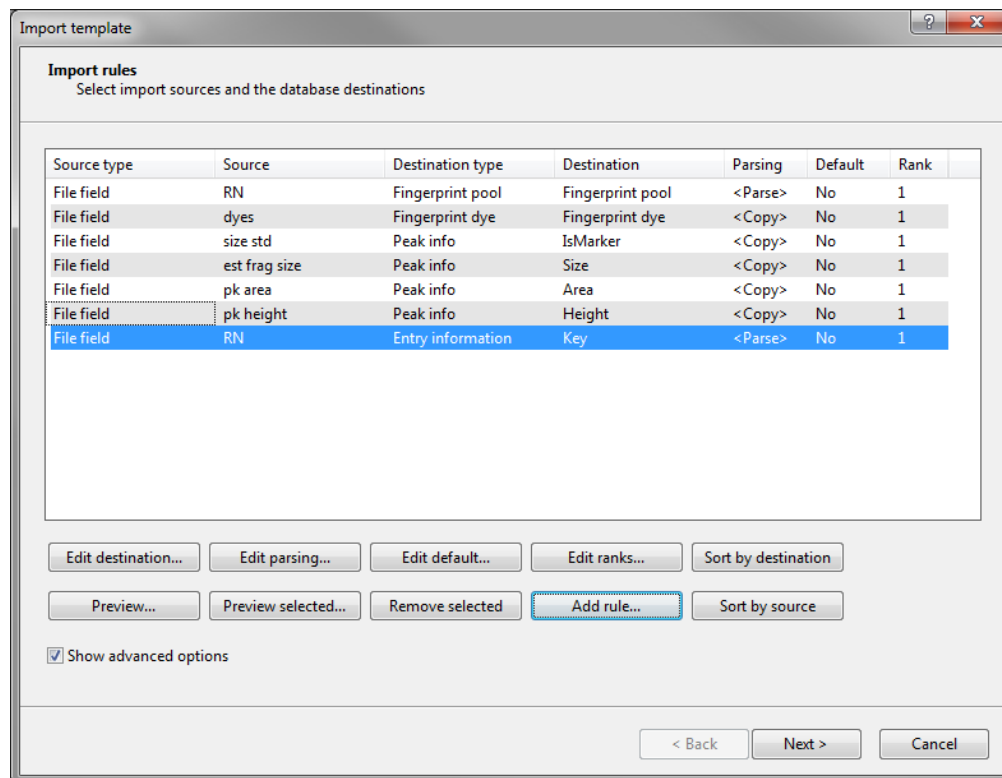


Figure 4.1.57: The *Import rules* dialog box.

The *Import rules* dialog box lists the information present in the selected files as **Source**, their linked **Source type** and the **Destination** component they are associated with (initially all set to <None>).

The rows in the grid can be associated with new or existing entry information fields, lane information fields, peak information, fingerprint dyes, fingerprint pools or fingerprint type experiments.

Specifying a *destination* for one or more selected rows can be done by pressing the <**Edit destination**> button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

When only one row is selected in the grid, the information of this row can be linked to:

- The default information field **Key**.
- A **Fingerprint type** name. The (parsed) information will hold the fingerprint type name.

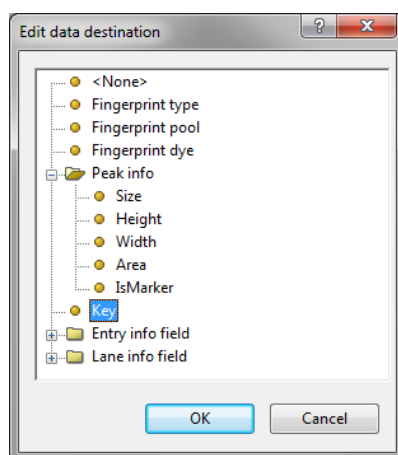


Figure 4.1.58: Edit data destination for a single selected row entry.

- A **Fingerprint pool**. The (parsed) information will hold the pool information.
- A **Fingerprint dye**. The (parsed) information will hold the dye information.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing lane information field (select **<Create new>** or select an existing field under the topic **Lane info field**, respectively).
- A **Peak info** information: **Size**, **Height**, **Width**, **Area**, or **Marker**.



When the **Marker** peak information is mapped to a column in the peak file, BioNumerics will only import lane information for samples for which no marker information is present in the marker column, and for samples for which the text "No" is supplied in the marker column.

If a row is linked to a new entry information field or a new lane information field, a new dialog box pops up when pressing the **<OK>** button.

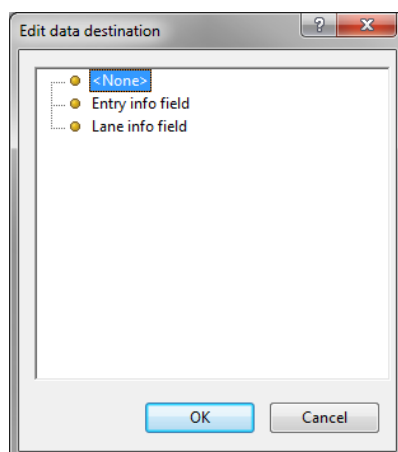


Figure 4.1.59: Edit data destination for multiple selected row entries.

When multiple rows are selected in the grid, the information of these rows can be linked to:

- Non-default entry information fields (select the **Entry info field** option).

- Lane information fields (select the ***Lane info field*** option).

When pressing the **<OK>** button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields or lane information fields in the database. If no entry information fields or lane information fields exist with the same name, a new dialog box pops up prompting for the names.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls the *Import data* dialog box.

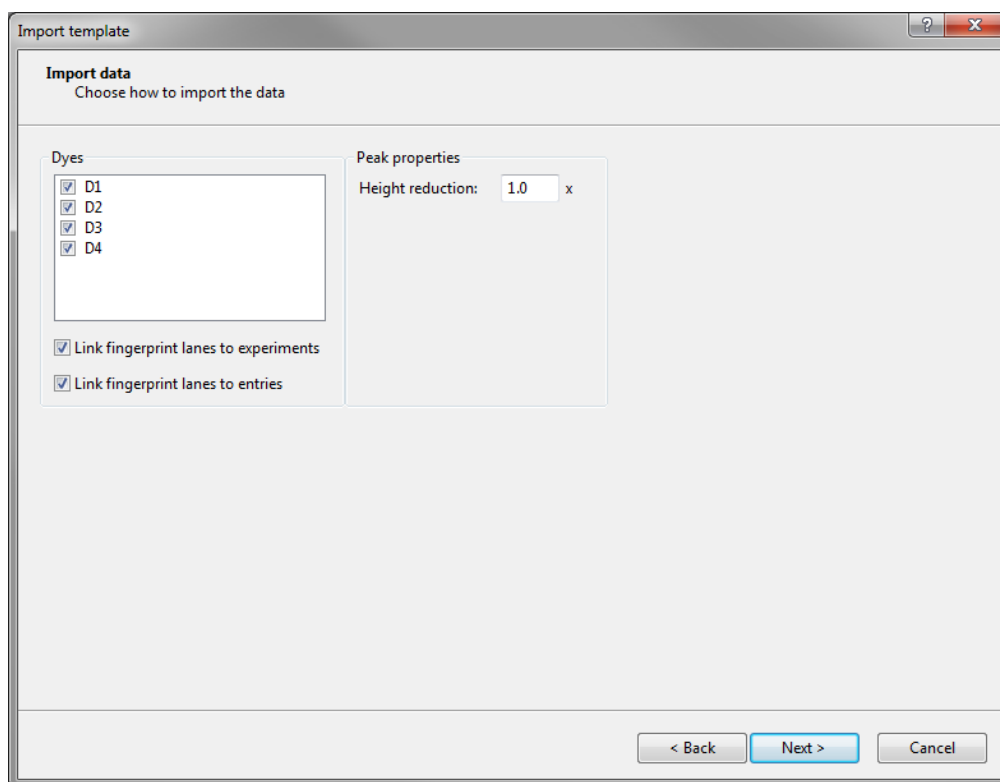


Figure 4.1.60: The *Import data* dialog box: choose how to import the data.

Dyes:

- BioNumerics reads the dye names from the mapped ***Fingerprint dye*** column and displays all dyes in the grid. All channels are by default checked to be imported in the database, but can be unchecked e.g. if one or more channels do not contain any signal.

Peak properties:

- Any peak with a height exceeding the OD range of the fingerprint experiments will appear truncated. To avoid this, a ***Height reduction*** can be applied. The ***Height reduction*** factor is by default set to "1.0" (= no height reduction). If the ***Height reduction*** is set to e.g. "2", the heights are reduced by a factor two.

When the option **Import as fingerprint file** was checked in the first step of the import wizard, two more options appear:

- When the option **Link fingerprint lanes to experiments** is checked, the lanes will be linked to the corresponding "dye" fingerprint type experiment. The names of these fingerprint types are composed of the base fingerprint type name, followed by the pool name (if present), and the name of the dye. When the option is unchecked, the dyes will be imported in the database, and the lanes will be linked to the selected base experiment. As a consequence, the lanes cannot automatically be linked to entries in the database (the option **Link fingerprint lanes to entries** is not available).
- When **Link fingerprint lanes to entries** is checked, the lanes will be linked to new and/or existing entries in the database, depending on the settings specified in the last step of the import wizard.

Pressing the **<Next>** button will display the **Import links** dialog box when the option **Link fingerprint lanes to entries** is checked. The same dialog is displayed when the option **Import as fingerprint file** was unchecked in the first step of the import wizard. Otherwise the **Import template information** dialog box pops up.

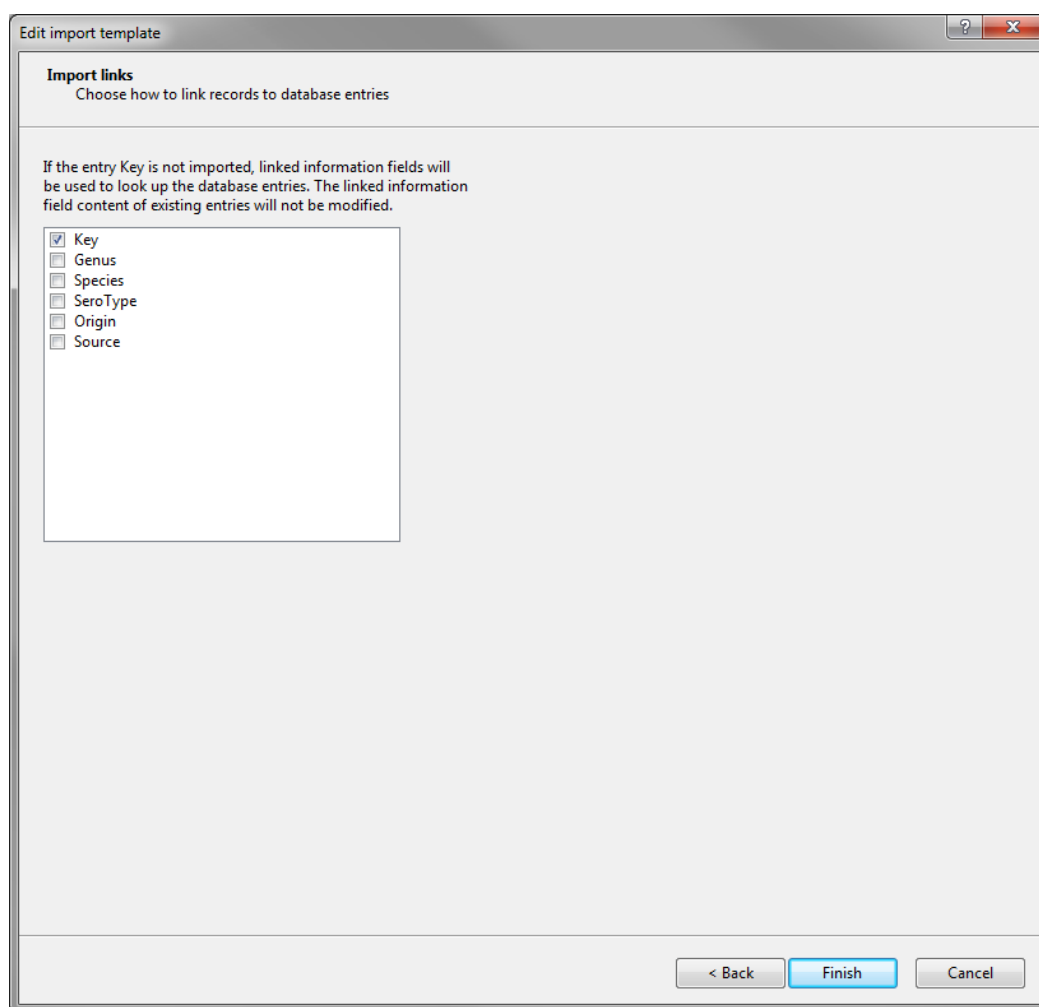


Figure 4.1.61: Specify the entry link field.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.

- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

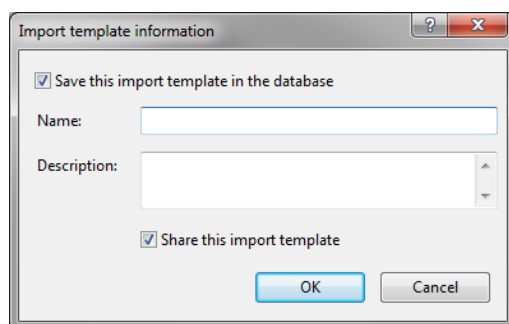


Figure 4.1.62: The *Import template information* dialog box.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel (see Figure 4.1.63).

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

A **Base fingerprint type** experiment needs to be specified. The settings of the base fingerprint type (including the reference system and band search settings) will be copied to all fingerprint types that are created when fingerprint data is imported in the database. An existing fingerprint type experiment can be selected from the list, or a new experiment can be created (**Create New**).

When a new base fingerprint type experiment is created and added to the database the *Experiment settings*

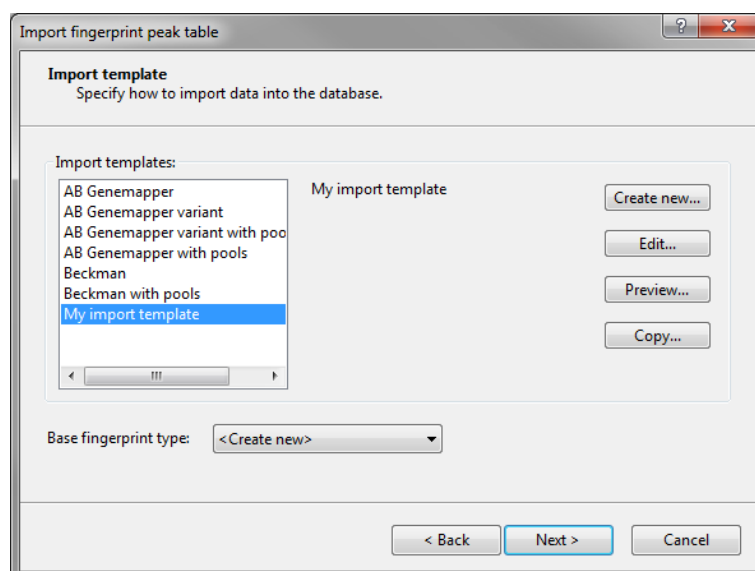
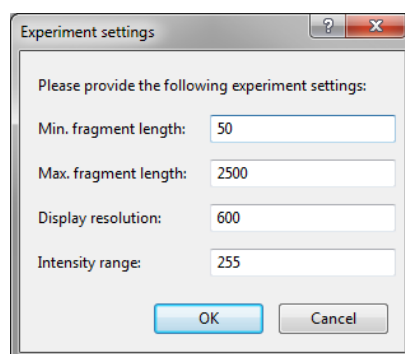


Figure 4.1.63: Import template added to the list.

Figure 4.1.64: The *Experiment settings* wizard page.

wizard page will prompt for some experiment settings. The same dialog will be displayed if a base fingerprint type experiment is selected for which no a reference system has been defined:

- A linear reference system will be created between the user-defined **Minimum** and **Maximum fragment length** positions.
- The **Display resolution** defines the resolution (track length, expressed in points) the traces will be rescaled to after normalization. A number should be entered that is equal to or less than the resolution of the imported raw traces.
- The **Intensity range** is the number of intensity levels the peak table consist of (dynamic range). This number is sometimes also indicated as a bit depth: 8-bit corresponds to 256, 12-bit to 4096, and 16-bit to 65536 intensity levels. Other values can also be entered.

If the option **Link fingerprint lanes to experiments** was checked or the option **Import as fingerprint file** was unchecked, a fingerprint type needs to be present in the database for each pool (if present) and dye combination. The names of these fingerprint types are composed of the base fingerprint type name, followed by the pool name (if present), and the name of the dye (e.g. **VNTRMM1D1**). If one or more of these fingerprint types are not present in the database, the *New experiment types* dialog box pops up, listing all missing fingerprint types. The user needs to confirm the creation of the missing fingerprints.

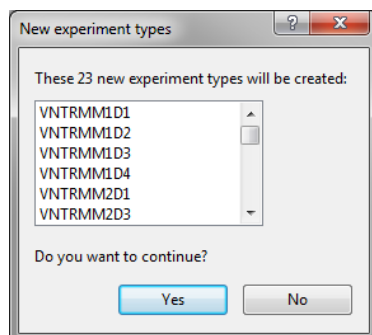


Figure 4.1.65: Missing fingerprints.

The reference system of the base fingerprint type experiment is copied to all new fingerprint experiments and a linear calibration curve is calculated. A warning message pops up when a logarithmic calibration curve is detected for an existing reference system.

The *Database links dialog* opens when the option *Link fingerprint lanes to entries* was checked. The same dialog is displayed when the option *Import as fingerprint file* was unchecked in the first step of the import wizard.

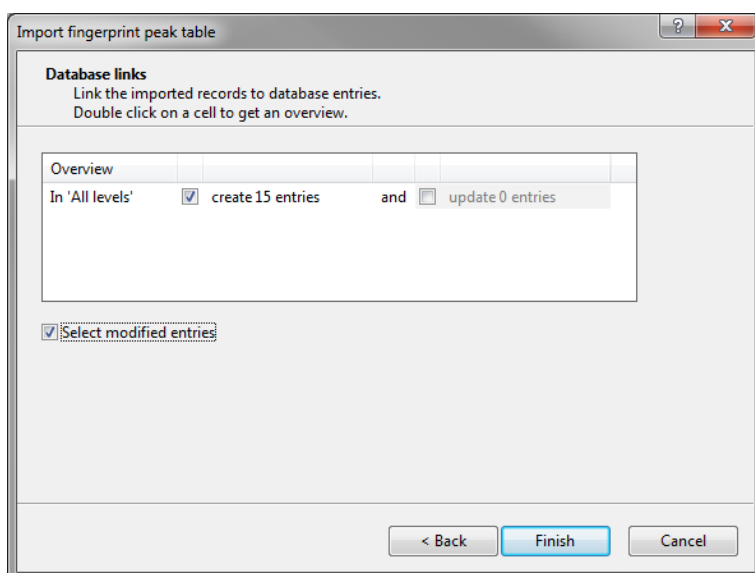


Figure 4.1.66: The *Database links dialog*.

The *Database links dialog* prompts for some additional settings:

- When *Create x entries* is checked, the import tool is allowed to create the new entries in the database.
- Check the option *Update x entries* if you want the software to be able to update the information for existing entries.
- If the option *Select modified entries* is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the *Create x entries* option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing <Next> will start with the import of the data.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

If the option **Import as fingerprint file** was checked in the first step of the import wizard, fingerprint files are created for each imported dye. The names of the files are composed of the **Fingerprint file name** and the name of the dye. If no dyes are present in the peak file, only one fingerprint file is created.

BioNumerics reads the band positions from the mapped size column, the peak heights from the mapped height column, the area information from the mapped area column and generates densitometric curves using this information.

The lane information is linked to new or existing entries depending on the options checked in the *Database links dialog*. When the option **Link fingerprint lanes to entries** was unchecked no entries are created and the lanes are not linked.

If the option **Link fingerprint lanes to experiments** was unchecked, all lanes are linked to the selected base experiment. If the option **Link fingerprint lanes to experiments** was checked, all lanes are linked to their appropriate fingerprint "dye" experiment type. The names of these fingerprint types are composed of the base fingerprint type name, followed by the pool name (if present), and the name of the dye. The fingerprint type experiments are displayed in the *Experiment types* panel.

If the option **Import as fingerprint file** was unchecked in the first step of the import wizard, the peaks are imported directly into the database and linked to their appropriate fingerprint "dye" experiment type and database entries.



When fingerprints are imported without creating synthetic profiles (= option **Import as fingerprint file** unchecked), it will not be possible to edit the bands in the *Comparison* window.




Entries for which fingerprint data was imported are selected in the database if **Select imported isolates** was checked in the *Database links dialog*.

4.1.4.2 Processing steps

Due to the nature of the data, the processing of capillary sequencer curves is somewhat different from gel images (see 4.1.3.2) and consists of following steps:

1. Curve pre-processing (optional)
2. Defining bands and quantification
3. Normalization

4.1.4.3 Data visualization

To start processing a capillary electrophoresis run, highlight one of the channels in the *Fingerprint files* panel and select **Open fingerprint data...** (). Alternatively, you can first open the *Fingerprint* window with **Edit > Open highlighted object...** (, **Enter**) and then select **File > Edit fingerprint data...** (.

Fingerprint data from automated sequencers are opened in the *Fingerprint curve processing* window (see Figure 4.1.67).



Whether the *Fingerprint processing* window or the *Fingerprint curve processing* window is opened for fingerprint processing, depends on the **Data type** set for the fingerprint type (see 4.1.5.2).

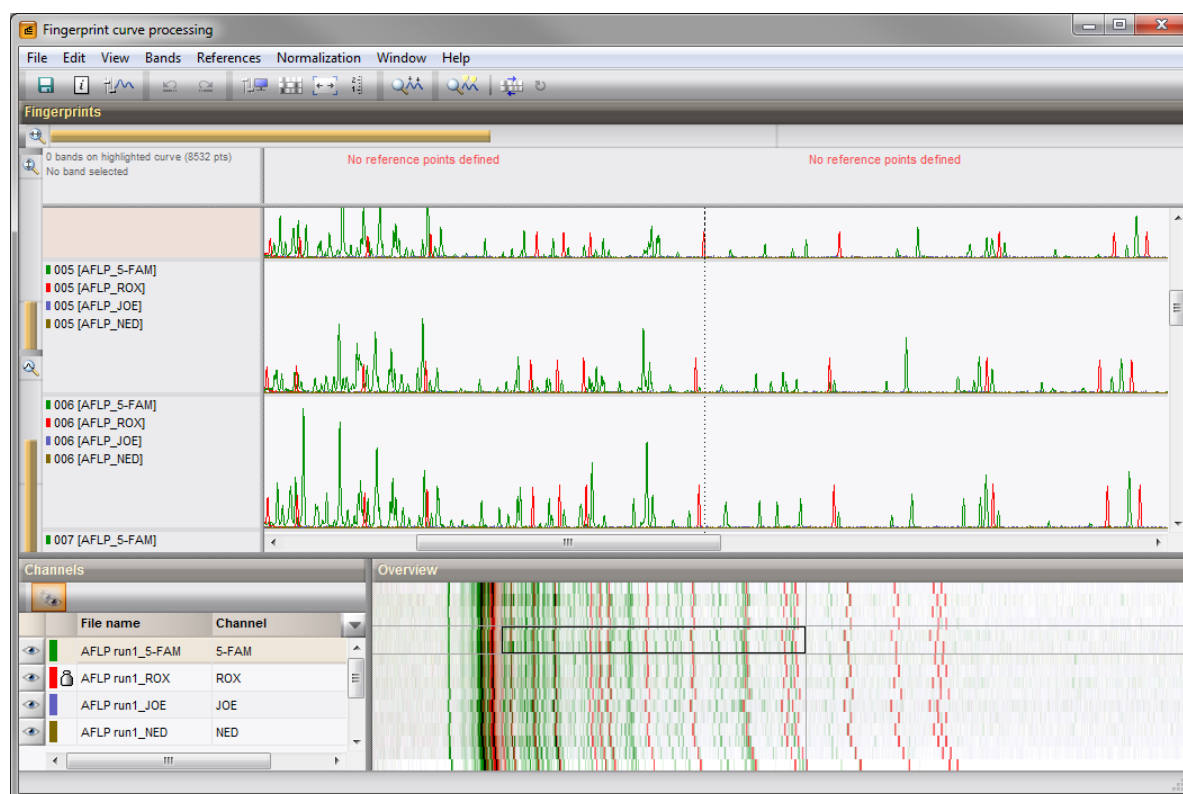






Figure 4.1.67: The *Fingerprint curve processing* window, for processing of capillary electrophoresis curves.

The *Fingerprint curve processing* window consists of three dockable panels: the *Fingerprints* panel, *Channels* panel and *Overview* panel.




All channels (corresponding to fluorescent dyes) from a run are automatically loaded in the *Fingerprint curve processing* window. Which channel or combination of channels is being displayed, can be determined in the *Channels* panel.


By default, when the *Fingerprint curve processing* window is opened, **Multi-channel view** is enabled and all channels are shown. To toggle the display of an individual channel, click on the  icon left of the channel in the *Channels* panel. When the channel is hidden from the view, the icon shows as .


To toggle between multi-channel and single-channel view, select **View > Multi-channel view** (, **Ctrl+E**). Switching off multi-channel view disables all channels except the one which is highlighted in the *Channels* panel.

The *Fingerprints* panel shows the data from the selected channels. The data can be shown as curves (chromatograms) or as reconstructed gel images via **View > Pseudogel view** (.

To optimize the display of any type of fingerprint curves, there are three zoom sliders (2.3.7) available in this panel:

- Horizontal zooming (): Increases or decreases the horizontal space taken in by the fingerprints. This zoom slider is located by default in the top part of the panel.
- Vertical zooming (): Increases or decreases the vertical space taken in by the samples. This zoom slider is located by default in the left upper part of the panel.
- Signal intensity zooming (): Increases or decreases the intensity of the signal (i.e. the height of the peaks in the fingerprints). This zoom slider is located by default in the left lower part of the panel.

Fingerprints can be fit automatically in the available horizontal space by selecting **View > Fit horizontally** .

By default, information about the sample (key / experiment combination) is displayed on the left. Instead of the sample info, a vertical scale can be shown by selecting **View > Show vertical scale** .

The default order of the profiles in the *Fingerprints* panel corresponds to the original sample order. The profiles can also be sorted according to experiment type with **View > Sort by experiment type**, which is particularly useful if a sample pooling strategy was used. In case the order was modified during assignment of bands to reference positions (see 4.1.4.6), the original order can be restored with **View > Show original ordering**.

When the *Fingerprint curve processing* window was opened by clicking on a dot in the *Experiment presence* panel or from the *MLVA plugin*, the view in the *Fingerprints* panel might be limited to a single sample (lane). To display all samples that originate from the same run again, select **View > Show all lanes**.

When metrics information is available, i.e. after normalization and calculation of a calibration curve (see 4.1.4.6), the top part of the *Fingerprints* panel shows the reference system and a metrics scale. The metrics scale is automatically adjusted to the horizontal zoom.

As soon as bands are assigned (see 4.1.4.4), band information for the selected band(s) is displayed in the top left part of the *Fingerprints* panel (see Figure 4.1.68).

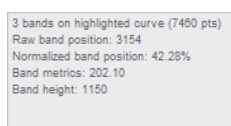


Figure 4.1.68: Information about the selected band(s) in the top left of the *Fingerprints* panel.

The *Overview* panel facilitates navigation through the fingerprint data. All data are automatically fit into this panel (X and Y axes, as well as the signal intensity) in pseudo-gel representation. A rectangle (thick lines) with guides (thin lines) indicates which part of the data is currently visualized in the *Fingerprints* panel (see Figure 4.1.69).

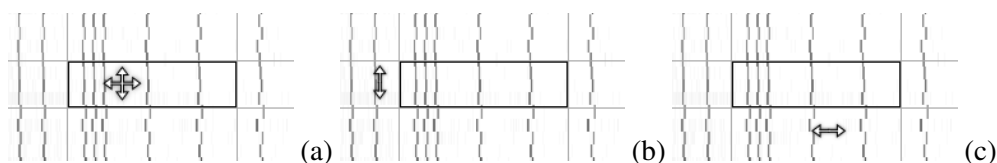


Figure 4.1.69: Navigating through curve data via the *Overview* panel: (a) movement unrestricted, (b) vertical navigation and (c) horizontal navigation.

When hovering over the rectangle, the mouse cursor changes into a four-headed arrow (Figure 4.1.69 a). This allows you to drag the rectangle to any other position in the *Overview* panel and the *Fingerprints* panel will be instantly updated.

When hovering left or right from the rectangle (between the two horizontal guides), the mouse cursor changes into a two-headed vertical arrow (Figure 4.1.69 b). Dragging the mouse moves the rectangle in the vertical direction only. This can be useful e.g. to examine a certain region on all profiles in the run.

When hovering above or below the rectangle (between the two vertical guides), the mouse cursor changes into a two-headed horizontal arrow (Figure 4.1.69 c). Dragging the mouse moves the rectangle in the horizontal direction only. This can be useful e.g. to inspect the full curve from a single sample.

If the fingerprint curves are shown in normalized view with distortion bars displayed (see 4.1.4.6), the distortion bars are also shown in the *Overview* panel. This makes it easy to find normalization errors in a

batch of curves.

Many of the colors and other display options in the *Fingerprint curve processing* window can be customized. Select **View > Display settings...** (🖨️) to show the *Display settings* dialog box (see Figure 4.1.70).

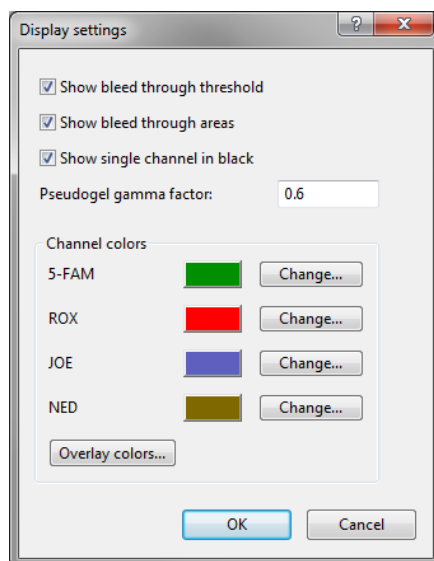


Figure 4.1.70: The *Display settings* dialog box.

With **Show bleed through threshold**, the user-defined threshold for bleed through can be displayed on the curves. Checking **Show bleed through areas** displays the bleed through areas (source and effect) on the curves and pseudo-gel views. For more information on bleed through, see 4.1.4.8.

If **Show single channel in black** is checked, curves or bands on pseudo-gels will be displayed in black (highest contrast) when the multi-channel view is disabled (see 4.1.4.3). When this setting is switched off, the curves or bands will be displayed in the colors as specified for the channel (see below). The latter colors are always used when the multi-channel view is enabled.

The **Pseudogel gamma factor** can be specified in the corresponding text box.

Under **Channel colors**, the colors in which the individual channels (dyes) are displayed can be set. Pressing <Change...> calls the *Color* dialog box, from which any desired color can be picked. Users may prefer to set this to the actual color of the fluorescent dye.

Pressing <Overlay colors...> allows the overlay colors to be set (see Figure 4.1.71).

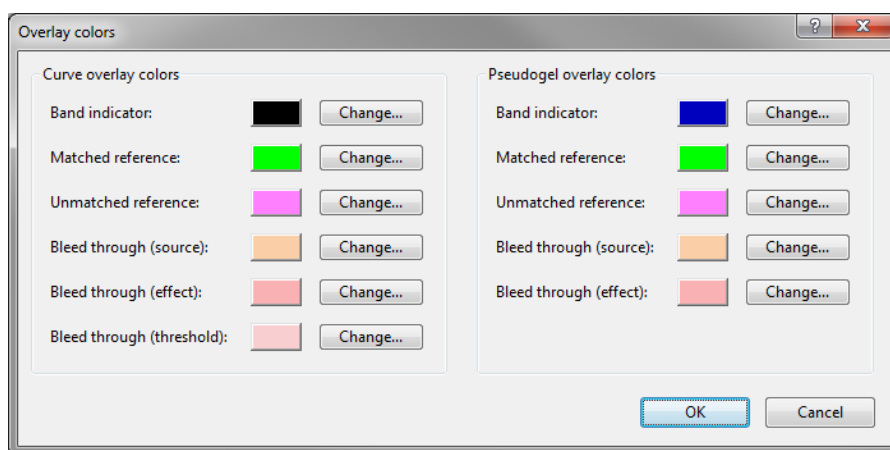


Figure 4.1.71: The *Overlay colors* dialog box.

For the curve display mode and the pseudo-gel display mode individually, following set of overlay colors can be chosen:

- **Band indicator:** Color in which band positions are indicated.
- **Matched reference:** Color of the triangle, indicating a (size marker) band that has been assigned to an external reference position.
- **Unmatched reference:** Color of the rhomb, indicating an external band that has been unlinked from an external reference position.
- **Bleed through (source):** Color of the vertical area in which a peak exceeding the bleed-through threshold (see 4.1.4.8) occurs. Since the bleed-through causing peak is indicated, the color is applied in the channel where this peak occurs.
- **Bleed through (effect):** Similar as above. However, since the results (i.e. possible pull-up peaks) are indicated, this color is applied to all channels except the channel in which the large peak occurs.
- **Bleed through (threshold):** Color of the horizontal area that flags possible bleed-through peaks. Obviously, this option is absent from the *Pseudogel overlay colors*.

Pressing <Change...> will open the *Color* dialog box.

4.1.4.4 Defining reference bands

Before any other actions can be undertaken, reference bands (or peaks) need to be defined on the reference channel. BioNumerics has an automated band search function that facilitates this task.

Since the automatic reference peak finding algorithm works on all visible channels, make sure to only display the reference channel. Next, select **Bands > Search reference bands...** (🔍, **Ctrl+F**) to open the *Search reference bands* dialog box.

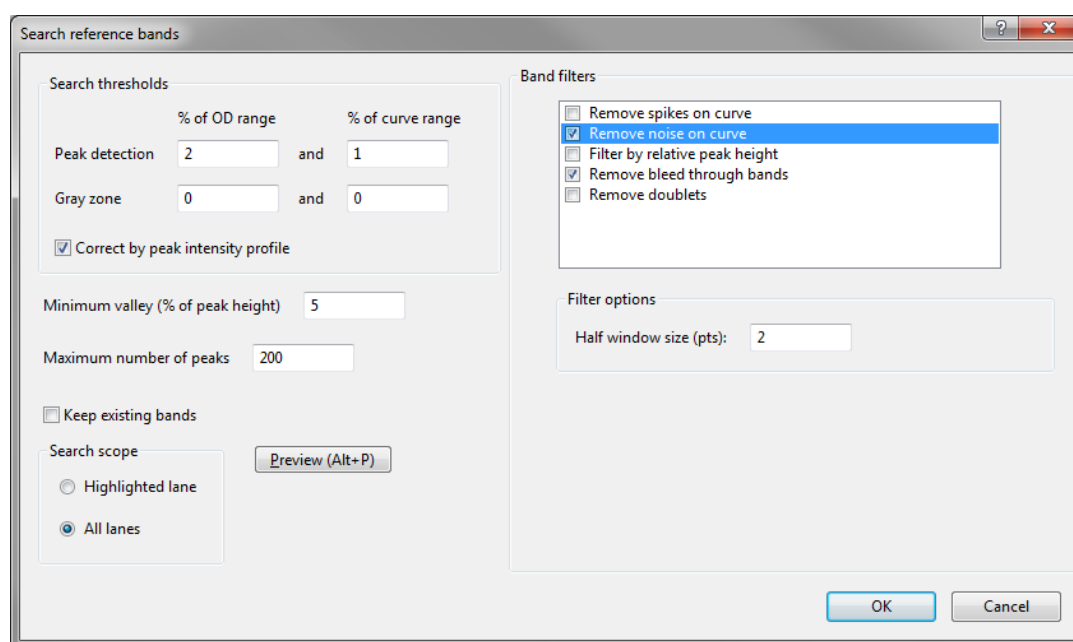


Figure 4.1.72: The *Search reference bands* dialog box.

Two types of **Search thresholds** are in use when searching bands: an absolute threshold, expressed as a percentage of the fingerprint's OD range, and a relative threshold, expressed as a percentage of the highest

value found on the individual curve (curve range). A band will only be detected if *both* conditions are fulfilled. The **Peak detection** thresholds determine when an elevation on the curve is considered a band. Optionally, the thresholds for **Gray zone** define an additional range for uncertain bands (see also 4.1.3.6). When both thresholds for **Gray zone** are zero (the default settings), no uncertain bands will be defined, i.e. bands will be either present or absent.

The option **Correct by peak intensity profile** allows to use size-dependent thresholds to correct for the typical "ski-sloping" of capillary sequencer fingerprints when a peak intensity profile is calculated (see 4.1.3.7).

The minimum valley, expressed as a percentage of the highest peak (**Minimum valley (% of peak height)**), can be used to discriminate between peaks that are in close proximity.

The upper limit for **Maximum number of peaks** is there merely for performance reasons: when very low **Search thresholds** are used, it will prevent an unrealistic high number of peaks from being saved to the database.

In combination with the above-mentioned thresholds for band detection, a number of **Band filters** can be activated that act on the set of found bands. Following **Band filters** are available:

- **Remove spikes on curve:** Removes spikes or very sharp peaks, which are typically caused by tiny air bubbles or dust particles in the capillary tubes. The **Spike width**, i.e. the maximum width for a peak to be considered a spike, can be specified.
- **Remove noise on curve:** This smoothing function is the only filter which is actually ran *before* peaks are searched. It does not alter the curve data in any way, but avoids band assignments to local peak maxima in case of noisy curve data. The **Half window size** determines the extend of the smoothing. This filter is by default enabled, with a **Half window size** of "2", i.e. a very subtle smoothing is applied.
- **Filter by relative peak height:** This filter removes bands in the vicinity of other bands if they are smaller than the **Minimum relative height (%)** and closer than the **Maximum distance (%)**.
- **Remove bleed through bands:** Deletes all bands in bleed through regions that have a height exceeding the **Remove below (% of OD range)** value. Bands with peak heights in the **Gray zone** will be labeled as uncertain. Both bleed through filtering settings are expressed as a percentage of the OD range.
- **Remove doublets:** Filters out double bands or doublets. Doublets are defined by the **Maximum distance (pts)** between the two maxima and the **Maximum valley (% of peaks)** between them. In case a doublet is found, the option **Collapsed position** determines where the position of the doublet is defined relative to the two compound peaks: **Weighted average** (average position, weighted by the height of the compound peaks), **Left** (the position of the left compound peak) or **Right** (the position of the right compound peak).

When the <**Preview**> button is pressed (keyboard shortcut **Alt+P**), the software indicates in the *Fingerprint curve processing* window which bands will be found when applying the band search current settings.

When **Keep existing bands** is checked, currently defined bands are retained and newly found bands added to this set. This option is not compatible with the preview functionality.

The **Search scope** can be limited to the currently highlighted sample in the *Fingerprints* panel (**Highlighted lane**) or can be **All lanes of visible channels**. Note that bands are only searched in lanes that are displayed in the *Fingerprints* panel (see 4.1.4.3). In addition to the automated band search tool, bands can also be defined manually: select **Bands > Add new band (Enter)** to add a band at the cursor position. When the option **Bands > Delete highlighted bands (Del)** is checked, the cursor will automatically jump to the nearest peak maximum.

Removing one or more selected bands can be done with **Bands > Delete highlighted bands (Del)**. Bands can be selected with **Ctrl+click** or a contiguous selection of bands can be made by holding the **Shift**-key on the keyboard and dragging a rectangle with the mouse.

A selection of bands can be marked as uncertain with **Bands > Mark band(s) as uncertain (F5)** or marked as certain with **Bands > Mark band(s) as certain (F6)**. Note that uncertain bands are indicated with a dashed line and certain bands with a solid line at the band's position.

4.1.4.5 Creating a reference system

Typically in capillary sequencer fingerprints, a commercially available size marker is applied in one of the channels for normalization purposes (inline reference). To have a reference system automatically created based on a lane containing commercial size marker, first highlight a suitable lane and then select **References > Define size standard....** This will display the *Size standard* dialog box, from which a size marker can be selected (see Figure 4.1.73).



A reference should only be created once, with the first sequencer run imported. When attempting to create a reference system when a reference system is already defined, a warning message will appear.



Before creating a reference system based on a size marker, make sure that bands are defined (see 4.1.4.4).

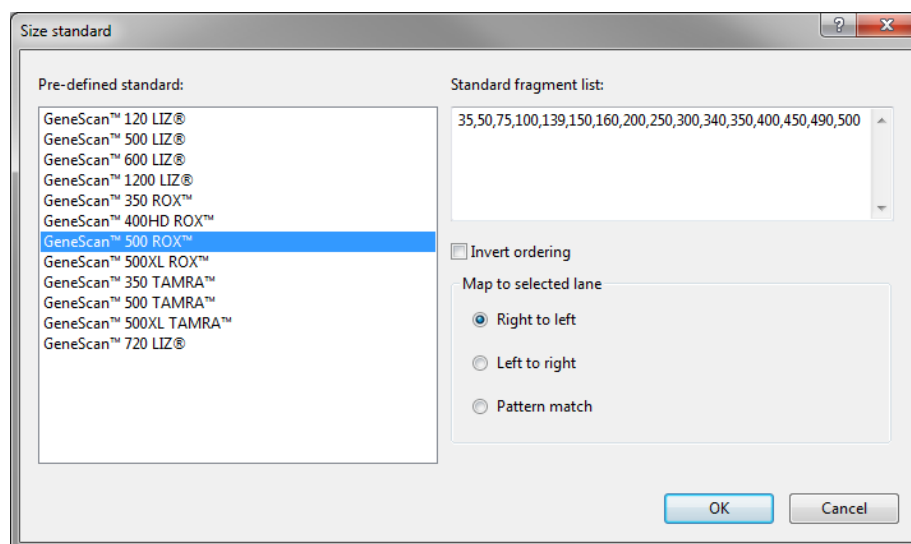


Figure 4.1.73: The *Size standard* dialog box.

The list on the left-hand side shows all pre-defined size markers. Highlighting a size marker will display a comma-separated list of fragment sizes in the **Standard fragment list** on the right-hand side. This list can be freely edited, e.g. in case of a custom size marker or when using a commercial size marker that is not available from the **Pre-defined standard** list.

The algorithm to map the fragment list to the bands found in the highlighted lane will assume that the smallest fragment will be situated on the left and the largest fragment on the right, unless the option **Invert ordering** is checked.

Three options are available for mapping the fragment list to the highlighted lane:

- **Right to left:** Starting from the right-hand side of the selected lane, the algorithm will create a reference position for each band and name it after the DNA fragment's size.
- **Left to right:** Starting from the left-hand side of the selected lane, the algorithm will create a reference position for each band and name it after the DNA fragment's size. This option will be rarely used, unless the primer-dimer peak has been manually removed from the lane.

- **Pattern match:** This uses an advanced algorithm that allows to skip peaks (e.g. peaks caused by bleed through) that do not fit in the pattern of the size marker.

Press <OK> to create the reference system.

An information message will appear (see Figure 4.1.74).

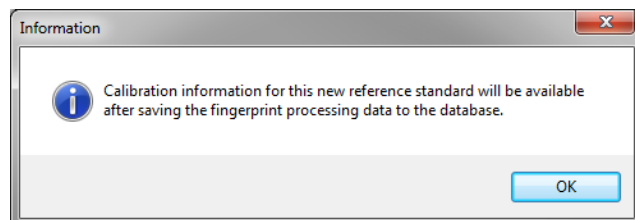


Figure 4.1.74: Information message that appears after creating a reference system.

When the fingerprint data are now saved to the database with **File > Save** (📁, **Ctrl+S**), the software will automatically create the reference system and calibration curve for each of the fingerprint types. Since this allows the calculation of metrics information, a metrics scale now becomes available in the upper part of the *Fingerprints* panel (see 4.1.4.3) and band search filters that rely on metrics information (see 4.1.4.4) can now be used.

Reference positions can also be individually removed (using **References > Delete reference position**), added (using **References > Add new reference position**) or renamed (using **References > Rename reference position...**).

4.1.4.6 Normalization

Normalization is achieved by assigning bands in the reference channel to external reference positions.



Before proceeding with the normalization, it is a good idea to check if bands in the reference channel are assigned properly. Although the normalization algorithm can cope with some additional bands in the reference channel (e.g. as a result from bleed through), it is obvious that normalization will be imperfect if a series of marker bands is missing.

To normalize a complete run at once, select **Normalization > Auto assign reference positions (all lanes)...** (📊, **Ctrl+A**). Alternatively, to normalize only the currently highlighted lane, select **Normalization > Auto assign reference positions (current)...** (**Ctrl+Shift+A**). Both actions will display the *Auto assign reference positions* dialog box (see Figure 4.1.75).

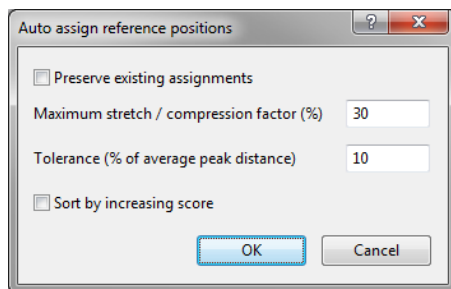



Figure 4.1.75: The *Auto assign reference positions* dialog box.

When **Preserve existing assignments** is checked, previously made assignments will be kept.



Maximum stretch / compression factor (%) Tolerance (% of average peak distance)

If **Sort by increasing score** is checked, the order of the samples in the *Fingerprints* panel will be rearranged, so that the profiles that had to be distorted the most in order to fit with the reference system, appear on top of the list. In practice, this means that only the profiles in the upper part of the *Fingerprints* panel need to be checked for correct assignments. When these profiles are OK, one can be sure that the remaining profiles are also correctly normalized. This option is unavailable when this dialog was called via **Normalization > Auto assign reference positions (current)... (Ctrl+Shift+A)**.


When the assignment of the marker bands to reference positions is made, the data can be shown in normalized mode with **Normalization > Show normalized view** (, **Shift+N**). Using the *distortion bars*, the quality of the normalization can be assessed. For capillary sequencer data, typically only very small distortions are to be expected and any bright colors of the distortion bars are indicative of a normalization issue. Distortion bars are switched on by default, but can be toggled on and off with **Normalization > Show distortion bars**.

All assignments to reference positions can be removed with **Normalization > Delete all assignments** or assignments can be deleted for selected bands (made via **Shift** and drag) with **Normalization > Delete selected assignments (Ctrl+Del)**.




The *Fingerprint curve processing* window contains a multi-level undo function: select **Edit > Undo** (, **Ctrl+Z**) to undo the last action and **Edit > Redo** (, **Ctrl+Y**) to redo the last undone action.

Assignments to reference positions can also be made for individual marker bands: click on the reference position and the band to highlight both and select **Normalization > Assign reference position (Ctrl+Enter)**.

Note that the view is not updated automatically: to reflect the changes made to marker band assignments, select **Normalization > Update normalization** (, **Ctrl+U**).

4.1.4.7 Defining data bands

Optionally, data bands (or peaks) can be defined on the data channels. BioNumerics has an automated band search function that facilitates this task.

Since the automatic data peak finding algorithm works on all visible channels, make sure to only display the data channels on which you want to search for bands. Next, select **Bands > Search data bands...** (, **Ctrl+Shift+F**) to open the *Search data bands* dialog box.

Two types of **Search thresholds** are in use when searching bands: an absolute threshold, expressed as a percentage of the fingerprint's OD range, and a relative threshold, expressed as a percentage of the highest value found on the individual curve (curve range). A band will only be detected if *both* conditions are fulfilled. The **Peak detection** thresholds determine when an elevation on the curve is considered a band. Optionally, the thresholds for **Gray zone** define an additional range for uncertain bands (see also 4.1.3.6). When both thresholds for **Gray zone** are zero (the default settings), no uncertain bands will be defined, i.e. bands will be either present or absent.

The option **Correct by peak intensity profile** allows to use size-dependent thresholds to correct for the typical "ski-sloping" of capillary sequencer fingerprints when a peak intensity profile is calculated (see 4.1.3.7).

The minimum valley, expressed as a percentage of the highest peak (**Minimum valley (% of peak height)**), can be used to discriminate between peaks that are in close proximity.

The upper limit for **Maximum number of peaks** is there merely for performance reasons: when very low **Search thresholds** are used, it will prevent an unrealistic high number of peaks from being saved to the database.

In combination with the above-mentioned thresholds for band detection, a number of **Band filters** can be activated that act on the set of found bands. Following **Band filters** are available:

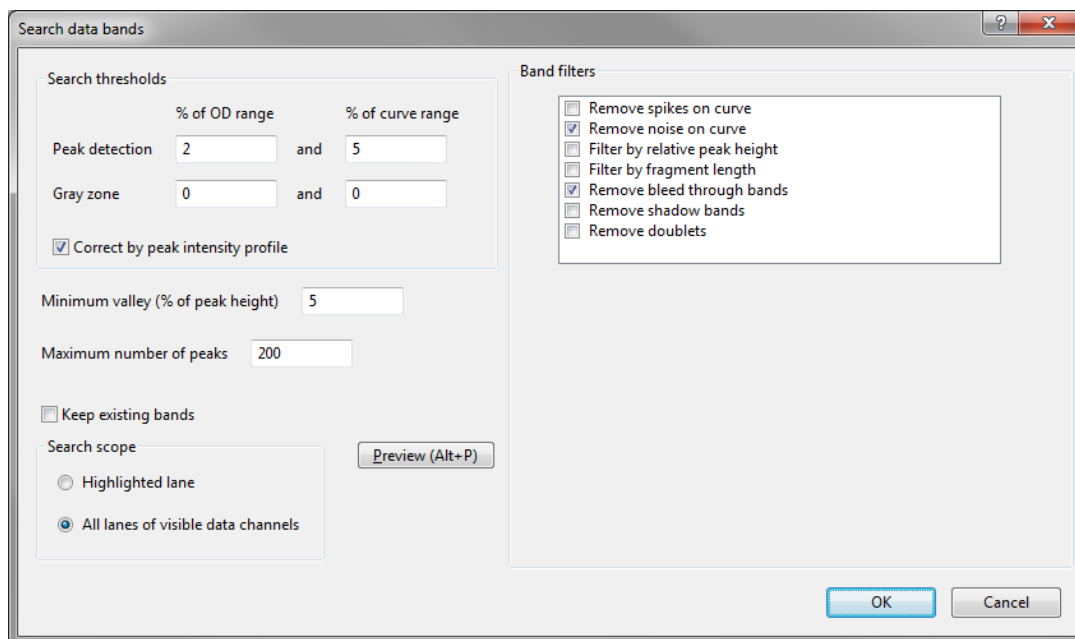


Figure 4.1.76: The *Search data bands* dialog box.

- **Remove spikes on curve:** Removes spikes or very sharp peaks, which are typically caused by tiny air bubbles or dust particles in the capillary tubes. The *Spike width*, i.e. the maximum width for a peak to be considered a spike, can be specified.
- **Remove noise on curve:** This smoothing function is the only filter which is actually ran *before* peaks are searched. It does not alter the curve data in any way, but avoids band assignments to local peak maxima in case of noisy curve data. The *Half window size* determines the extend of the smoothing. This filter is by default enabled, with a *Half window size* of “2”, i.e. a very subtle smoothing is applied.
- **Filter by relative peak height:** This filter removes bands in the vicinity of other bands if they are smaller than the *Minimum relative height (%)* and closer than the *Maximum distance (%)*.
- **Filter by fragment length:** Removes all bands with a fragment length smaller than the *Minimum fragment length* or larger than the *Maximum fragment length*. This filter is useful e.g. in MLVA analysis where the expected band size is known. The option *Non-reference channels only* allows the reference channel to be excluded from this filter. Since this filter uses band metrics, it requires a reference system and calibration curve to be defined.
- **Remove bleed through bands:** Deletes all bands in bleed through regions that have a height exceeding the *Remove below (% of OD range)* value. Bands with peak heights in the *Gray zone* will be labeled as uncertain. Both bleed through filtering settings are expressed as a percentage of the OD range.
- **Remove shadow bands:** Shadow or stutter bands are smaller peaks that occur immediately before or after a large peaks. They are caused by PCR artifacts. This filter will remove any bands with a height smaller than the value specified for *Maximum relative size (%)* and that occur within the *Maximum left distance (bp)* and *Maximum right distance (bp)* from the peak. Since this filter uses band metrics, it requires a reference system and calibration curve to be defined.
- **Remove doublets:** Filters out double bands or doublets. Doublets are defined by the *Maximum distance (pts)* between the two maxima and the *Maximum valley (% of peaks)* between them. In case a doublet is found, the option *Collapsed position* determines where the position of the doublet is defined relative to the two compound peaks: *Weighted average* (average position, weighted by the height of the compound peaks), *Left* (the position of the left compound peak) or *Right* (the position of the right compound peak).


When the **<Preview>** button is pressed (keyboard shortcut **Alt+P**), the software indicates in the *Fingerprint curve processing* window which bands will be found when applying the band search current settings.

When **Keep existing bands** is checked, currently defined bands are retained and newly found bands added to this set. This option is not compatible with the preview functionality.

The **Search scope** can be limited to the currently highlighted sample in the *Fingerprints* panel (**Highlighted lane**) or can be **All lanes of visible channels**. Note that bands are only searched in lanes that are displayed in the *Fingerprints* panel (see 4.1.4.3).

4.1.4.8 Dealing with bleed through

When peaks in a certain channel are over-saturated, they can give rise to bleed through or pull-up peaks in other channels. Bleed through peaks are therefore artifacts, which can be mistaken with signal peaks. To avoid such mistakes, BioNumerics offers functionality to label bleed through areas and automatically remove peaks from such areas.

Select **File > Curve processing settings...** () to open the *Curve processing settings* dialog box (see Figure 4.1.77).

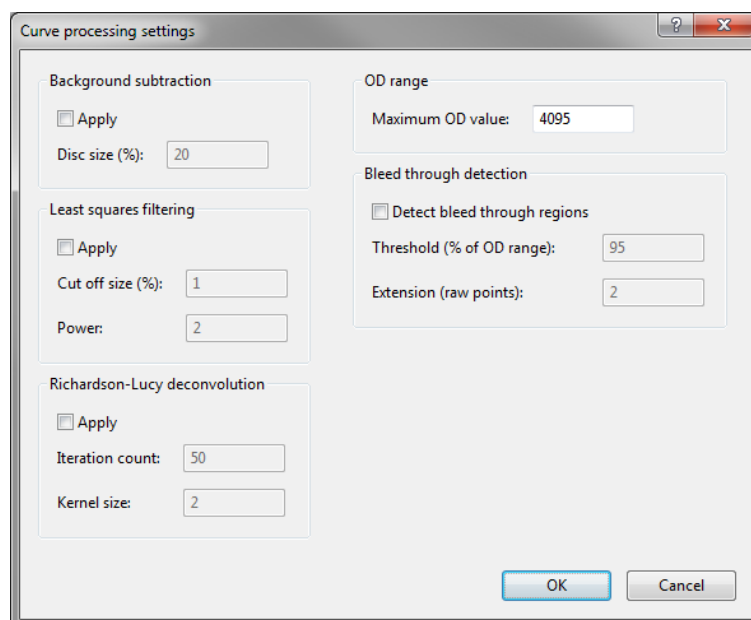


Figure 4.1.77: The *Curve processing settings* dialog box.

The options for **Background subtraction**, **Least squares filtering** and **Richardson-Lucy deconvolution** are the same as discussed for gel images (see 4.1.3.3) and they will be rarely used for capillary sequencer fingerprints.

The **OD range** is important for several reasons:

- It determines the scale of the Y-axis in the *Fingerprints* panel of the *Fingerprint curve processing* window (see 4.1.4.3).
- One of the two band search thresholds is expressed relative to the OD range (see 4.1.4.4).
- For defining bleed through (see below).

The OD range spans from zero to the **Maximum OD value**, which can be entered in the corresponding text box.

When the option *Detect bleed through regions* is checked, all curve regions higher than the *Threshold (% of OD range)*, will be considered as bleed through regions. Optionally, the bleed through regions can be extended with a number of points, entered in the *Extension (raw points)* input box. These points correspond to points on the raw electropherograms and are added in both directions.


4.1.5 Editing the fingerprint type settings

4.1.5.1 Overview of available settings

Being a relatively complex experiment type in terms of processing, a fingerprint type has a large number of settings, organized in following logical groups:

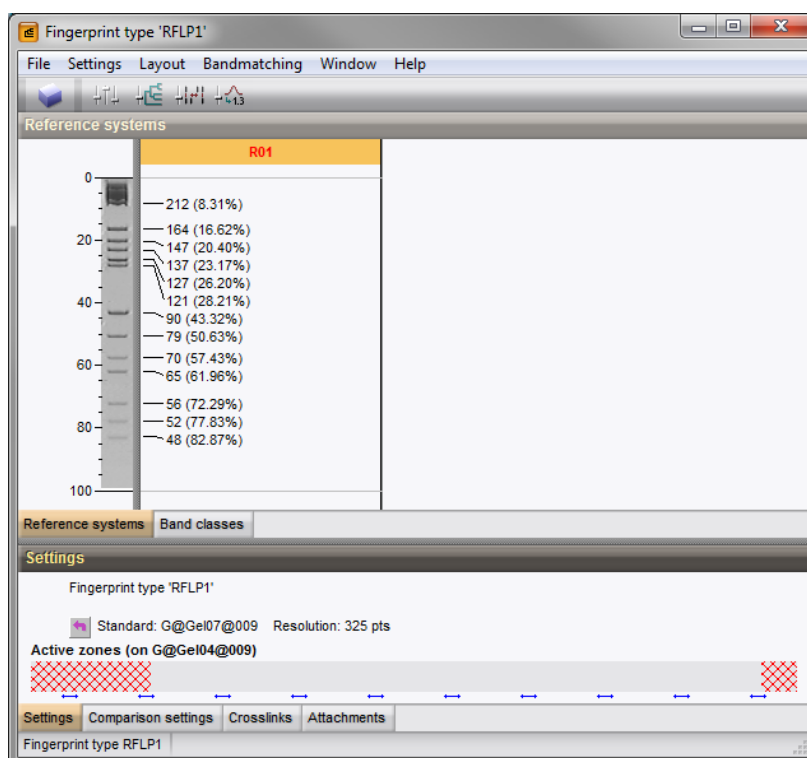
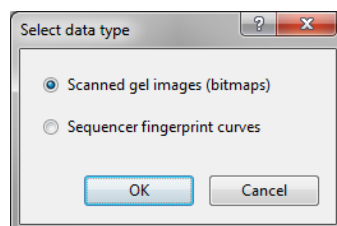
- Fingerprint data type
- Fingerprint conversion (or general) settings
- Layout settings
- Comparative quantification settings
- Comparison settings
- Reference system(s) and metrics calibration curve(s)
- Standard profile
- Active zones
- Band classes

All above settings can be accessed via the *Fingerprint type* window and are discussed in following paragraphs.

The *Fingerprint type* window (Figure 4.1.78) can be opened by clicking on the fingerprint type in the *Experiment types* panel and selecting *Edit > Open highlighted object...* (, **Enter**). Alternatively, simply double-click on the fingerprint type.

This window consists of six dockable panels:

- The *Reference systems* panel lists all reference systems (see 4.1.5.7) that are defined for the fingerprint type.
- In the *Band Classes* panel, band classes (see 4.3.6) are listed and can be edited.
- The *Settings* panel shows some of the settings used, like the standard profile (4.1.5.8) and the active zones on the fingerprint, that are used for comparison (4.1.5.9).
- The fingerprint comparison settings (see 4.2) are listed in the *Comparison settings* panel.
- Cross links (see 3.2.15) from the fingerprint type experiment to other database objects can be created from the *Crosslinks* panel.
- Attachments (see 3.2.13) can be added to the fingerprint type experiment from the *Attachments* panel.

Figure 4.1.78: The *Fingerprint type* window.Figure 4.1.79: The *Select data type* dialog box.

4.1.5.2 Fingerprint data type

Select **Settings > Data type...** to open the *Select data type* dialog box (see Figure 4.1.79).


Fingerprint types can have one of two available data types:

- **Scanned gel images (bitmaps):** The raw data are available as photographs or scans from two-dimensional gels. If this data type is selected, the *Fingerprint processing* window will open for data preprocessing.
- **Sequencer fingerprint curves:** Curves (electropherograms) originating from automated capillary sequencer equipment. If this data type is selected, the *Fingerprint curve processing* window will be used for data preprocessing.



The fingerprint data type is also displayed in the *Fingerprint processing settings* dialog box. However, it is read-only in that dialog box and it can only be set from the *Select data type* dialog box.

4.1.5.3 General settings

The *Fingerprint type* window allows you to change all settings which were defined when creating the fingerprint type and when processing the first gel with **Settings > General settings...** .

This calls the *Fingerprint processing settings* dialog box, which was discussed in [4.1.3.3](#), [4.1.3.4](#), [4.1.3.5](#) and [4.1.3.6](#).

The values that are stored with the fingerprint type are the default general settings, which can be overridden on a gel per gel basis in the *Fingerprint* window.

4.1.5.4 Layout settings


The default brightness and contrast settings can be changed with **Layout > Brightness & contrast...** This opens the *Image brightness & contrast* dialog box, which was discussed in [4.1.3.3](#). The same dialog box can also be called from the *Comparison* window (see [4.2.2](#)).

The brightness and contrast settings are applied on all profiles from the same fingerprint type. In case adjustments need to be made for individual gels, it is recommended to use the *Gel tone curve* window instead (see [4.1.3.3](#)).


Further layout settings include:

- **Layout > Show space between gelstrips:** If checked, a small white area is shown between individual profiles when displayed in the *Experiment data* panel of the *Comparison* window.
- **Layout > Show curves as images:** If checked, an artificial gel image will be created based on the densitometric curves and displayed in the *Experiment data* panel of the *Comparison* window.
- **Layout > Rescale curves:** This will rescale the densitometric curves so that they fit in the OD range specified for the fingerprint type.
- **Layout > Show normalized gelcards:** If checked, the normalized profiles are displayed in the *Experiment card* window (see [4.1.7](#)) instead of the raw profiles.
- **Layout > Hide inactive zones:** If checked, the inactive regions of a fingerprint are hidden when the profiles are displayed in the *Experiment data* panel of the *Comparison* window.

4.1.5.5 Comparative quantification settings

The quantification settings can be displayed with **Settings > Comparative quantification...** . This pops up the *Comparative quantification* dialog box, discussed in [4.3.8](#).

4.1.5.6 Comparison settings

The comparison settings are the default settings used to compare the fingerprints. They can be accessed with **Settings > Comparison settings...** , which calls the *Comparison settings* wizard. See [4.2](#) for a detailed explanation. The default comparison settings can be overruled in the *Comparison* window for individual analyses.

The comparison settings that are currently defined for the fingerprint type are shown in the *Comparison settings* panel (see Figure [4.1.78](#)).

4.1.5.7 Reference systems and calibration curves

A *reference system* consists of a set of *reference positions*. If metrics are known for each reference position, a calibration curve can be created to infer the metrics of any band.

The *molecular sizes* of the bands are not calculated within a particular gel file, but for a whole reference system. This means that, once you have created a reference system and normalized one gel, you can define the molecular size regression for all further gels that will be normalized using the same reference system.

In most cases, a reference system is created based on a size standard during the processing of the first gel (see 4.1.3.5). The *Fingerprint type* window then shows the defined reference positions in relation to the distance on the pattern (in percentage), and calls this reference system **R01**. Other reference systems (if created automatically) will be called **R02**, **R03**, etc.. In Figure 4.1.78, **R01** is shown in red because it is the active reference system.

Additionally, a new reference system also can be created in the *Fingerprint type* window directly. Two approaches are possible:

1. Entering positions on the gel (running distances) and the corresponding band sizes. Based upon the positions and the corresponding sizes, the program is able to establish a regression curve, upon which all imported bands can be mapped. This option is particularly suitable when you know the exact positions of the size markers in a gel system, and you want to reproduce the real regression exactly.
2. Allow the program to create its own regression curve between a defined maximum and minimum molecular weight, so that it can map the imported bands on this synthetic regression curve. This method is useful if you want to import band tables of which you know nothing else than the sizes.



Once a new reference system is defined, it is not possible to change it anymore! If you want to change a self-made reference system once it is saved, you will have to delete it and create it again.

For the first approach, select **Settings > New reference system (positions)...** in the *Fingerprint type* window. The *New reference system* dialog box shown in Figure 4.1.80 allows all known reference bands to be entered.

Position:	Name:
8.31	212
16.62	164
20.40	147
23.17	137
26.20	127
28.21	121
43.32	90
50.63	79
57.43	70
61.96	65
72.29	56
77.83	52
82.87	48

Figure 4.1.80: The *New reference system* dialog box, to define a new reference system based upon known band positions and sizes.

A **Name** for the new reference system can be entered in the corresponding text box.

Pressing the **<Add>** button calls the *Add reference point* dialog box (see Figure 4.1.81).

In this dialog box, a reference point, defined by a **Position** (running distance) and a **Name** (size) can be entered. Pressing **<OK>** in the *Add reference point* dialog box will add the position to the new reference system.

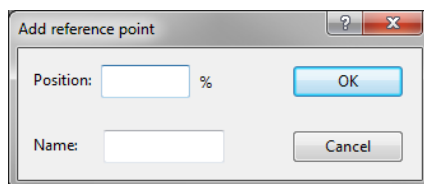


Figure 4.1.81: The *Add reference point* dialog box.

All reference points from the active reference system can be added to the list with the *<Copy from active>* button.

When finished adding reference points, press *<OK>*.

For the second option, select **Settings > New reference system (curve)...** in the *Fingerprint type* window to display the *New reference system* dialog box (see Figure 4.1.82).

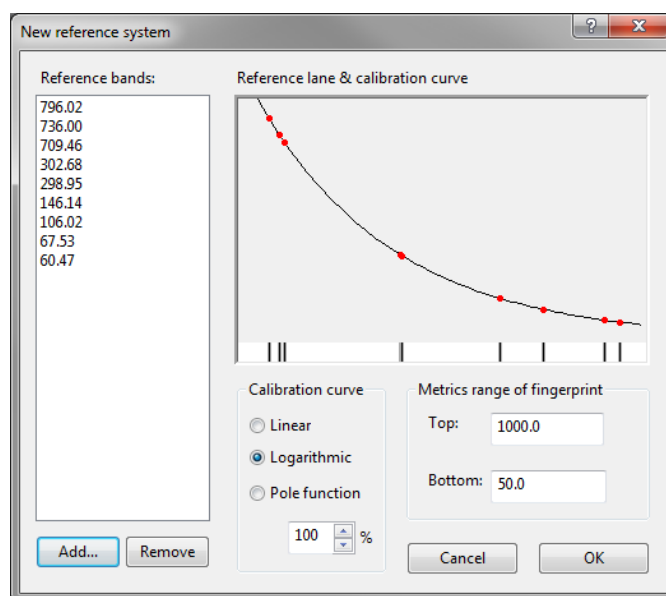


Figure 4.1.82: Defining a new reference system using a synthetic regression curve between user-defined size limits in the *New reference system* dialog box.

This dialog box allows the size range to be specified as well as the type and strength of the regression.

Under **Metrics range of fingerprint**, the highest expected value should be entered as **Top** and the lowest expected value as **Bottom**.

Press the *<Add>* button to add the metrics (sizes) for all reference bands available in the fingerprint type. The *Add new reference position* dialog box will appear.

Enter the metrics value of a reference point in the text box and press *<OK>*.

The reference bands are shown as red dots on the regression curve. This makes the adjustment of the **Calibration curve** easier.

Optimize the **Calibration curve** and the strength (in percent) to obtain the best spread of the reference bands.

When finished, press *<OK>* to save the new reference system.

To create a calibration curve for an existing reference system, highlight it in the *Fingerprint type* window and select **Settings > Edit reference system**. Alternatively, simply double-click on the reference system to display the *Fingerprint Reference system* window (see Figure 4.1.83).

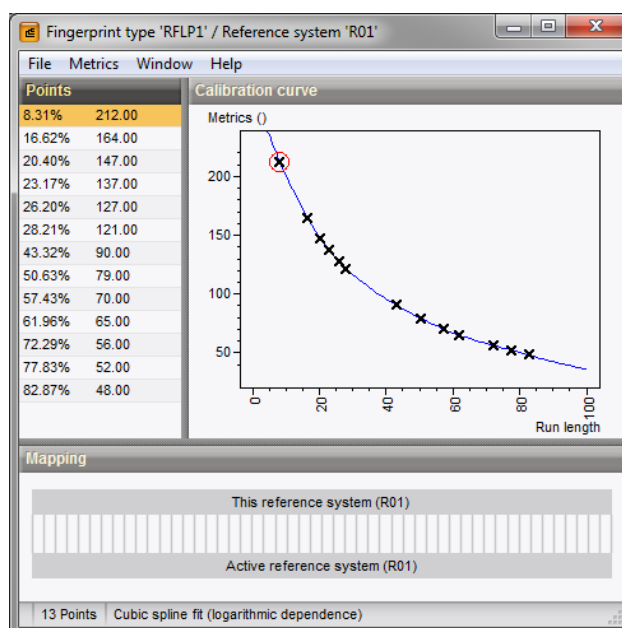


Figure 4.1.83: The *Fingerprint Reference* system window, showing molecular weight regression and remapping function to the active reference system (if different).

Initially, the calibration cannot be calculated, since the program does not know where to take the marker points from. The message "Could not calculate calibration curve. Not enough markers" is displayed.

You can add the markers manually via **Metrics > Add marker...**, but if you have entered the molecular weights as names for the reference positions (see 4.1.3.5), an easier solution is to copy these molecular weights with **Metrics > Copy markers from reference system...**

The result is a regression curve, similar as shown in Figure 4.1.83. As regression function, you use **Metrics > First degree fit**, **Metrics > Third degree fit**, **Metrics > Cubic spline fit** or **Metrics > Pole fit**, and each of these functions can be combined with **Metrics > Logarithmic dependence**.

Use **Metrics > Assign units...** to choose a unit such as "bp" (base pairs), "Da" (Dalton), etc..

The *Fingerprint Reference* system window can be closed with **File > Exit**.

Under normal circumstances, a reference system is created once initially and is never changed afterwards. In some cases however, it can be required that a second reference system is created. Some examples are:

1. The gel used originally for defining the reference positions appears to be an aberrant one, so that repositioning the reference positions is required to allow most other gels to be normalized easily.
2. One or more bands defined as reference positions are found to be unreliable or inappropriate and should be deleted or replaced with another band.
3. The user switches to a new reference pattern for the fingerprint type.
4. Gels of the same fingerprint type are imported from a second database and need to be analyzed together with gels from the first database.

Case 1, shown in Figure 4.1.84, results in two reference systems with the same reference position *names*, but having different *% distances* on the gel. Gels processed under both reference systems are perfectly compatible and there is no loss of accuracy compared to gels analyzed under the same reference system.

The same situation can arise if gels are imported from another database, which have been processed under a different reference system (case 4), but where the same marker pattern is used and the reference positions

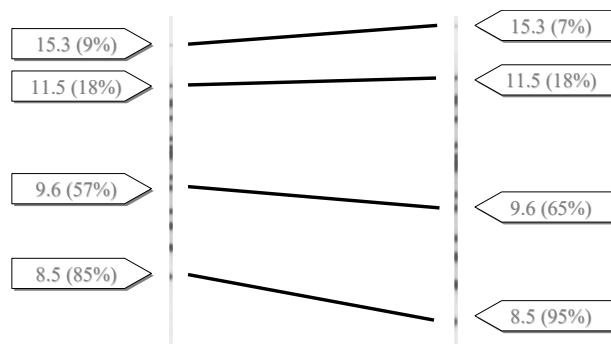


Figure 4.1.84: Example of different reference systems in the same fingerprint type for which remapping causes no loss of accuracy. See text for explanation.

have been given the same *name* (even though the % distances are different).

Case 2 may result in a new reference system with more or less bands, or with bands having a different name (Figure 4.1.85). In either case, the new reference system will not be automatically compatible with the original, and compatibility can only be obtained by creating a molecular weight regression curve for both reference systems. Both reference systems can then be remapped onto each other, which inevitably causes some loss in accuracy. The degree of compatibility depends on the number of reference positions in both systems, the amount of overlap between regression curves, the predictability of the regression curve using one of the available methods, the spread of calibration points (reference positions), the definition of the reference bands, etc..

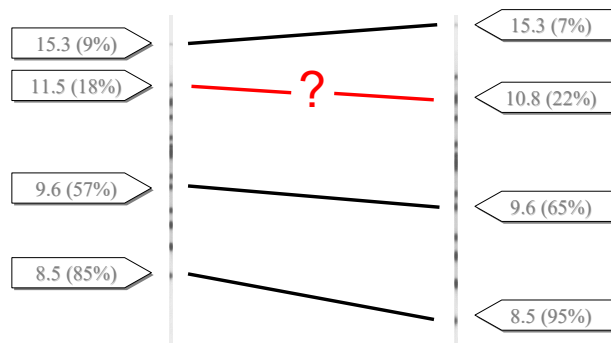


Figure 4.1.85: Example of different reference systems in the same fingerprint type for which remapping relies on molecular weight regression curves for both reference systems and as such, causes some loss of accuracy. See text for explanation.

Case 3 obviously causes a situation where reference positions have different names, since one can assume that a new marker has different bands, and results in a situation where remapping is required.

When more than one reference system is present in a fingerprint type, one of the reference systems is specified as the "active" reference system. The active reference system is the one to which all new gels will be normalized. By default, the first created reference system is the active one. The name of the active reference system is shown in red in the *Fingerprint type* window. To change the active reference system, highlight the reference system to become the active one and **Settings > Set as active reference system**.

To remove a reference system that is not used anymore, highlight the reference system in the *Fingerprint type* window and select **Settings > Remove reference system**. The program asks "Do you want to check if this reference system is in use?". For large databases, this may take a long time. If you answer <No> to this question, the selected reference system is removed, regardless of whether it is used in gels or not. By opening and saving a gel that was processed under the removed reference system however, it will be restored. By answering <Yes>, the program checks the database for gels normalized with the reference


system, and if any such gels are found, the reference system is not removed.



To avoid any possible conflict situations, it is recommended to allow the program to scan the database for the presence of gels normalized with the reference system and not to remove any reference systems that are in use.

4.1.5.8 Standard profile

In the *Fingerprint type* window displayed in Figure 4.1.78, the panel left from the reference system is blank: the fingerprint type misses a *standard* pattern. The standard pattern actually has no essential contribution to the normalization; it is only intended to show a normalized reference pattern next to the reference positions, in order to make visual assignment of bands to the reference positions easier. Another feature for which the standard is required is the automated normalization by pattern recognition (*Using densitometric curve* in the *Auto assign reference bands* dialog box). This algorithm requires a curve of a normalized reference pattern to be present in order to be able to align other reference patterns to it.

The profile that will be used as standard, should already be linked to a database entry (see 4.1.3.8). When this condition is met, a standard can be assigned by dragging the link arrow  in the *Settings* panel of the *Fingerprint type* window to a database entry in the *Main* window. Alternatively, the standard can be set in the *Fingerprint* window (see 4.1.3.8).

The standard pattern is displayed in the *Reference systems* panel next to the reference positions in the *Fingerprint type* window, and the database entry key of the standard is indicated next to the link arrow (Figure 4.1.78). From this point on, all further gels that are normalized will display the standard pattern left from the gel panel in the normalization step. This makes manual association of peaks easier and allows automated alignment using curve matching.



One can change the standard pattern at any time later on, e.g. if another reference pattern appears to be more suitable for this purpose.

4.1.5.9 Active zones on fingerprints

In the *Settings* panel in the *Fingerprint type* window the fingerprint of the selected database entry is shown (see Figure 4.1.78). For each fingerprint type, it is possible to define excluded zones. More information on how to define excluded zones on fingerprints can be found in 4.2.3.

4.1.5.10 Band classes

All band classes defined for the fingerprint type, together with their relative positions, metrics values and optional custom fields (see 3.2.6), are listed and can be edited in the *Band Classes* panel. See Figure 4.1.86 for an example.

To add a band class to the list, select **Bandmatching > Add new band class...** in the *Fingerprint type* window. However, in most cases it is more convenient to define band classes in the *Comparison* window (see 4.3) and save them to the fingerprint type as explained in 4.3.6.

Select **Bandmatching > Remove band class...** to remove a band class from the list.

In addition to the default fields Name, Position (%) and Position (metrics), extra band class information fields can be created with **Bandmatching > Add information field....** In the *Information field* dialog box that pops up, enter the name of the field and press <OK>.

The highlighted band class information field can be removed with **Bandmatching > Remove information field....** The software will ask for confirmation before actually deleting the field.

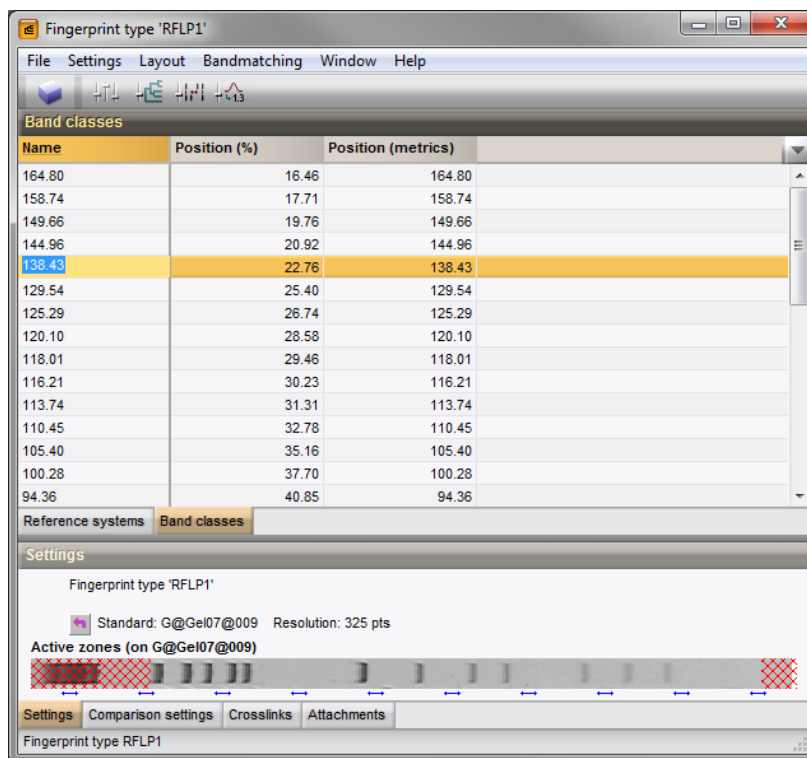


Figure 4.1.86: The *Fingerprint type* window for **RFLP1** in **DemoBase Connected** with the *Band Classes* panel displayed: band classes are saved with the experiment type.

The band class names and any information in band class information fields can be edited by clicking twice in the information fields (see 3.2.6 on editing information in object grid panels). Note that the band class positions (Position (%)) and Position (metrics)) are fixed and cannot be edited.

Similar to characters (see 6.1.2.11), *views* can be created for band classes. This allows you to maintain pre-defined sets of band classes for further analysis. Two types of band class views can be generated: subset based or query based views.

- Use a **subset based** band class view for a fixed list of band classes, corresponding to a band class selection.
- Use a **query based** band class view if the information on which to filter is contained in a band class information field and/or when the list of band classes is likely to change in the future. If the information in the band class info field is updated, the list of band classes returned by the band class view will be updated as well.

To create a subset based query, first select the band classes you like to define a subset for. Selected band classes will be indicated by a check mark in the left-most column.

Next, select **Bandmatching > Band class views > Manage user defined views...** or choose **<Manage user defined views...>** from the drop-down list in the header of the *Band Classes* panel. This action opens the *Manage band class views* dialog box (see Figure 4.1.87).

The list in the upper part of the dialog box shows all currently defined views on the band classes of the fingerprint type (if any), with their Name and Type (the latter will be either Subset or Query).

Press the **<Add>** button to create a new band class view. The *New character view* dialog box pops up (see Figure 4.1.88).

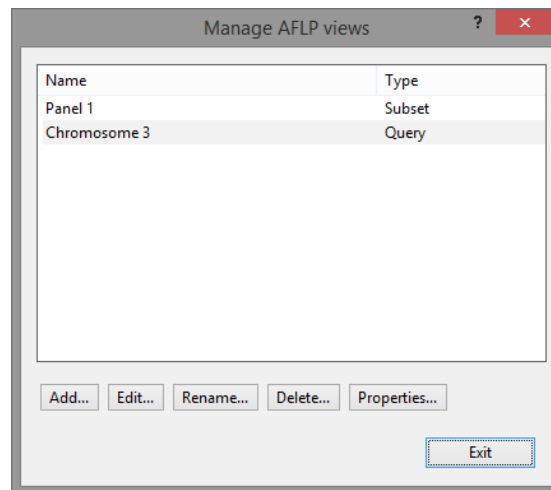


Figure 4.1.87: The *Manage band class views* dialog box, displaying a subset based and a query based band class view.

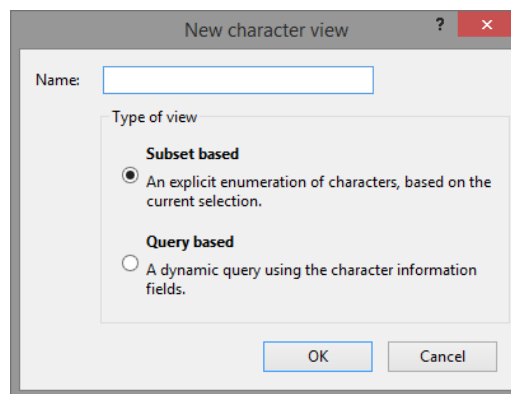


Figure 4.1.88: The *New character view* dialog box.

The dialog prompts you to enter a **Name** for the new band class view. It also allows to select the **Type of view** to create: either **Subset based** or **Query based**.

For a subset based band class view, simply enter a name (e.g. “MySubSet”) and press <**OK**>. The *New character view* dialog box will close and the *Band Classes* panel automatically switches to the newly created character view. The *Manage band class views* dialog box can then be closed.



Two minor differences exist between subset based *band class* views and the more general concept of subset based *object* views:

- Once created, subset based band class views cannot be edited anymore by adding or removing selected band classes.
- The band class order in a subset based band class view cannot be specified explicitly in the view and is instead governed by the global ordering of the band classes in the fingerprint type.

To create a query based view, select **Characters > Character Views > Manage user defined views...** or choose <**Manage user defined views...**> from the drop-down list in the header of the *Band Classes* panel to open the *Manage band class views* dialog box again (see Figure 4.1.87).

Enter a name for the band class view (e.g. “MyQuery”), check the **Query based** option and press <**OK**>.

The *Query view editor* dialog box will open. This dialog allows you to create a query on the band class names and information stored in any of the band class information fields (see 6.1.2.9) that are defined for the fingerprint type. The functionality of this dialog box is described in detail in 3.2.2.

Pressing <OK> in the *Query view editor* dialog box will close this dialog and the *Band Classes* panel will automatically switch to the newly created band class view.

In the *Manage band class views* dialog box (see Figure 4.1.87), existing band class views can be managed. Following commands all work on the highlighted band class view in the list:

- Query based views can be modified with <Edit>. This action will call the *Query view editor* dialog box again. Note that subset based views cannot be edited; they should be deleted and created again with an updated band class selection.
- Pressing <Rename> will show the *Rename character view* dialog box, in which a new name for the band class view can be entered.
- A band class view can be deleted by pressing the <Delete> button. The software will ask for confirmation before actually deleting the view.
- The object access properties for the band class view can be edited by pressing the <Properties> button. This action will open the *Object access* dialog box, as discussed in 3.2.3.

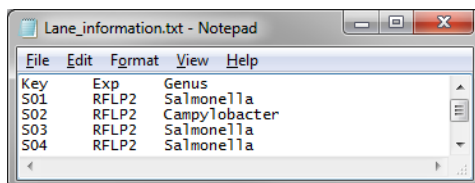
4.1.6 Importing fingerprint information from a text file

Fingerprint information, stored in a text file, can be imported in the database using the command **Database > Import information fields**, which can be launched from the *Fingerprint* window.

Double-clicking on the fingerprint file in the *Fingerprint files* panel in the *Main* window opens the *Fingerprint* window, listing the lanes defined for the gel in the *Fingerprint information* panel (empty if no lanes have been defined yet). The name of the fingerprint type experiment to which the lanes are linked and the reference system that was used, are displayed in the 'Experiment' column. When lanes are linked to entries in the database, the linked entry information is displayed in the *Entry information* panel.

With the import option **Database > Import information fields**, it is also possible to link unlinked lanes to new or existing entries in the database, re-link lanes to other entries, and re-associate the lanes with other existing fingerprint experiment types using this import option.

The text file should contain a well-defined table with rows corresponding to the lanes and columns corresponding to fingerprint or entry information fields. For each lane that is present in the *Fingerprint* window there should be a row in the text file. Empty lines are allowed. The header of the table should contain the fingerprint or entry information field names. There should be no extra rows or columns besides the table (see Figure 4.1.89 for an example).



Key	Exp	Genus
S01	RFLP2	Salmonella
S02	RFLP2	Campylobacter
S03	RFLP2	Salmonella
S04	RFLP2	Salmonella

Figure 4.1.89: Importing information from a text file.

During import, the information in the first row below the header information, will be linked to the first lane in the *Fingerprint* window; the second row will be linked to the second lane; etc.

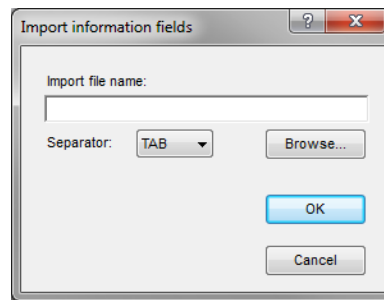


Figure 4.1.90: The *Import information fields* dialog box.

Selecting **Database > Import information fields** calls a new dialog (see Figure 4.1.90).

This dialog prompts for the text file and the separator:

- Pressing the **<Browse>** button allows you to select the file that you want to import, located on your computer, external drive or on a network location.
- Three different text file separators are currently supported and can be selected from the **Separator** drop-down list: "TAB", "Comma", "Semicolon". The separator that corresponds to the selected file should be picked from the list.

Browse for the file, select the appropriate field **Separator** from the list and press **<OK>**.

Only when the number of rows detected in the selected file (without taking into account the first row containing the header information) corresponds to the number of rows present in the *Fingerprint* window, the next step is displayed when pressing **<OK>** (see Figure 4.1.91). If the import routine is unable to open the selected file or if the number of lanes do not correspond, an error is generated.

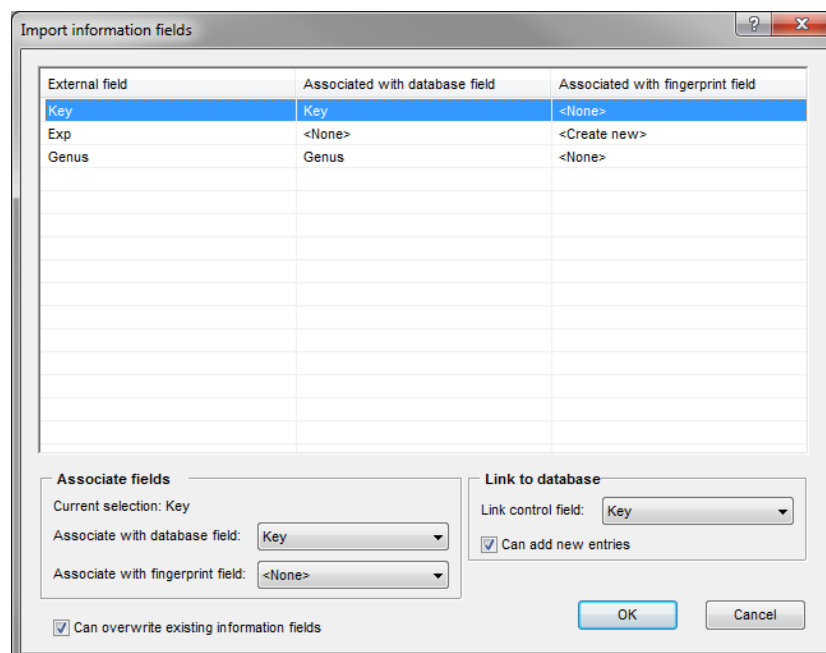


Figure 4.1.91: Linking information to entry and/or fingerprint information fields.

Each column detected in the selected text file corresponds to a row in the grid. Each column in the text file can be linked to:

- A new or existing non-default entry information field (select **<Create new>** or an existing non-default entry information field from the **Database field** list, respectively). The information will be imported and stored in the selected non-default entry information field of the entries to which the lanes in the *Fingerprint* window are attached after import (the information is updated in the *Entry information* panel).
- A new or existing non-default fingerprint information field (select **<Create new>** or an existing non-default fingerprint information field from the **Fingerprint field** list respectively). The information will be imported and stored in the selected non-default fingerprint information field for the lanes that are listed in the *Fingerprint* window (the information is updated in the *Fingerprint information* panel).
- The **Key** field (select “Key” from the **Database field** list). The lanes in the *Fingerprint* window will be linked to entries in the database based upon the imported key information. If no entries exist in the database with this key information, the entries are created by the software if the option **Can add new entries** is checked. The ‘Key’ field in the *Entry information* panel is updated.
- An existing fingerprint type experiment (select **<Experiment>** from the **Fingerprint field** list). The lanes in the *Fingerprint* window will be linked to the fingerprint type as specified in the text file column. After import, the information in the ‘Experiment’ column of the *Fingerprint information* panel is updated.

Check the option **Can overwrite existing information fields** if you want the software to be able to overwrite existing entry/lane information.

As soon as one row in the grid is linked to a non-default information field or the ‘Key’ field, the **Link to database** options are of interest:

- When the option **Can add new entries** is checked, the import routine is allowed to create new entries in the database.
- If existing entries are present in the database with the same **Link control field** information as is present in the text file, the import tool will link the imported data to these entries. If the **Link control field** is left empty, new entries are created for those lanes that are not linked already to existing entries in the database (the option **Can add new entries** needs to be checked), while existing links will be left as they are.

Pressing **<OK>** starts the import.

If columns in the text file are linked to a new entry information field or fingerprint information field (**<Create new>**), the following dialogs pop up:

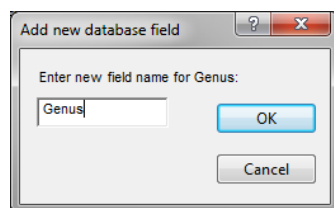


Figure 4.1.92: New database field.

Specify a new database field.

Specify a name for the new fingerprint field.

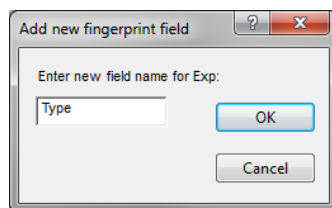


Figure 4.1.93: New fingerprint field.

4.1.7 Fingerprint experiment display

Clicking on the colored dot of a fingerprint type experiment in the *Database entries* panel pops up the *gel strip* in case the option *Scanned gel images (bitmaps)* is selected in the *Select data type* dialog box. When the *Sequencer fingerprint curves* option is selected the *curve* is loaded in the *Fingerprint curve processing* window. Alternatively, open the *Entry* window of the entry and click on the flask next the experiment name in the *Experiments* panel.

The gel strip can be displayed in two modes, a raw mode, i.e. not normalized, and a normalized mode. In the normalized mode, the band information is also shown. Band sizes are shown as molecular sizes (metrics) if the metrics regression curve is available for the reference system, or as relative distances from the top if no metrics regression curve is available.

To switch between the raw and normalized view, open the *Fingerprint type* window and select **Layout > Show normalized gelcards**. If the feature is enabled, the menu item is flagged.

The size of the card can be increased or decreased, respectively by pressing the **+**-key (plus) or the **-**-key (minus) on the keyboard.

You can right-click on the *Experiment card* window to pop up a floating menu, from which you can choose **Export normalized curve**, **Export normalized band positions**, and **Export normalized band metrics**. Selecting any of the above commands exports the corresponding information to the clipboard, from where it can be pasted as text, e.g. in Notepad.

In case multiple gel strips are shown on the screen, it is possible to line them up by right-clicking on a gel strip and choosing **Line up**. All gel strips can be closed at once using **Close all** in the floating menu.

It is possible to show or edit fingerprint lane information fields with *Information fields* (see 4.1.3.8).

4.1.8 Exporting fingerprint data

With the **Export peak table** option, listed under the topic *Fingerprint type data* in the *Export* dialog box (see Figure 4.1.94), fingerprint data and optionally entry information can be exported to a Comma Separated Values (CSV) file.

In the *Database entries* panel of the *Main* window, select the entries to export. A single entry can be selected by holding the **Ctrl**-key and left-clicking (**CTRL+click**). Check boxes for selected entries are indicated as ☒. In order to select a group of entries, hold the **Shift**-key and click on another entry. All the entries in the database can be selected using **Edit > Select all (Ctrl+A)**.

Selecting **Export peak table** under *Fingerprint type data* in the *Export* dialog box and pressing **<Export>** starts the export wizard (see Figure 4.1.95).

In the *Export peak table* dialog box, all fingerprint types defined in the database are displayed in the left panel, all entry information fields are listed in the panel on the right.

Select the *Fingerprint type(s)* and *Information fields* to export. To select multiple rows, hold the **Ctrl**-key

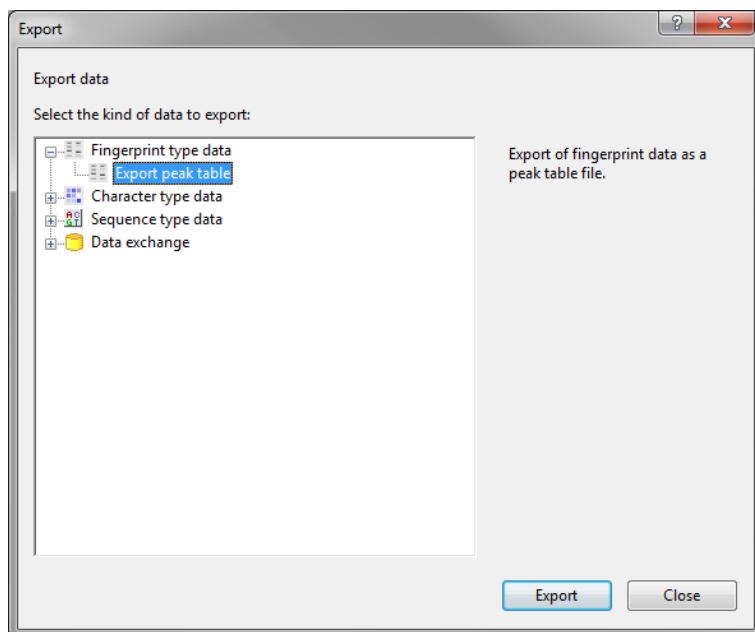


Figure 4.1.94: The *Export peak table* option in the *Export* dialog box.

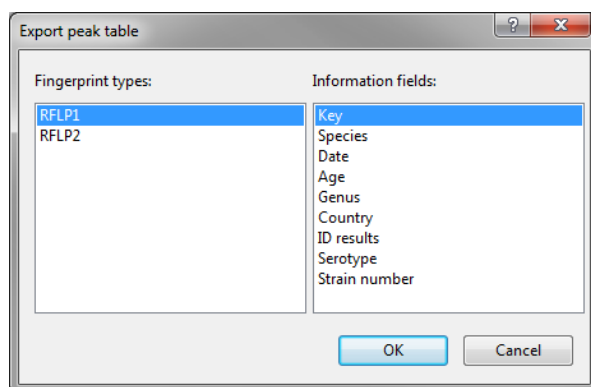


Figure 4.1.95: The *Export peak table* dialog box.

on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.

Pressing <OK> exports the entry and peak information in tabular format to a Comma Separated Values (CSV) file, that will be opened by the default CSV editor on your computer (often MS Excel).

The exported peak information includes the linked fingerprint type *Experiment*, the *Normalized* and *Metric* positions and the peak *Height* and *Width*.



When no calibration curve was calculated for the reference system included in a selected sample, the *Metric* and *Normalized* positions for this sample are the same.

Chapter 4.2


Cluster analysis of fingerprints

4.2.1 Fingerprint comparison settings

Please note that for cluster analysis of fingerprint types, the Fingerprint data module (FP) and the Tree and network inference module (TN) need to be present in your BioNumerics configuration.

Proceed as follows to calculate a cluster analysis on a fingerprint type:

In the *Comparison* window, select the fingerprint type in the *Experiments* panel on which the cluster analysis should be based. Optionally, display the normalized gel image by pressing the eye button (👁) next to the experiment name.

Select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**.... Alternatively, press the  button, in which case the following menu pops up (Figure 4.2.1).

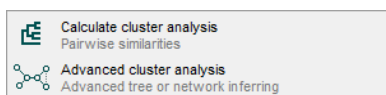


Figure 4.2.1: Cluster analysis menu popped up from the dendrogram button.

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient (see Figure 4.2.2).

The hierarchical representation on the left provides an overview of the available coefficients. Depending on the selected coefficient, the relevant settings are displayed on the right. The coefficients are subdivided in two categories: **Curve based** and **Band based**. The **Curve based** category has another subdivision, which is **Including error**. Each of the categories can be collapsed by clicking on the small '-' (minus) sign that precedes the category name.

All coefficients from the **Curve based** category provide similarities based upon densitometric curves.

The Pearson product-moment correlation (**Pearson correlation**) is calculated as:

$$r = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \sqrt{\sum_{i=1}^n y_i^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2}}$$

with x_i and y_i the densitometric values of both profiles and n the number of points in the curves.

The **Cosine coefficient** is calculated as:

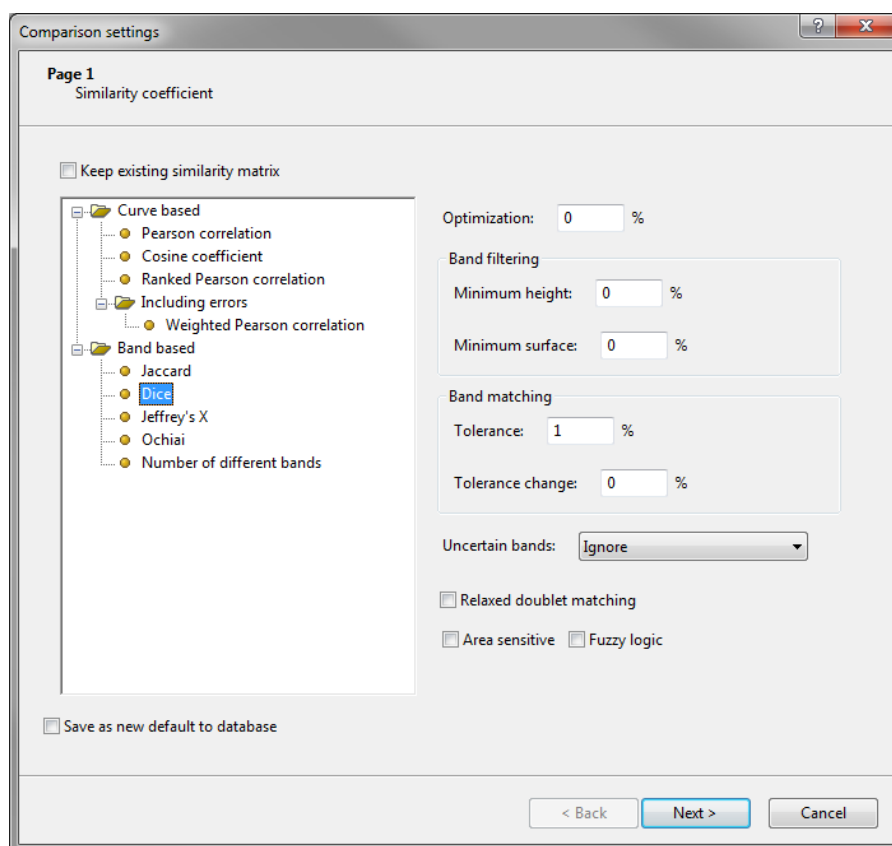


Figure 4.2.2: The *Similarity coefficient* wizard page, which deals with the choice of the similarity coefficient.

$$r = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

The **Ranked Pearson correlation** is a variant of the Pearson correlation coefficient that is less sensitive to outliers, since it is based on *ranks* instead of on the actual densitometric values. The densitometric values of each curve are first sorted and replaced by their rank. Then a Pearson correlation coefficient is calculated on the ranks.

The **Weighted Pearson correlation** should only be applied when the error (uncertainty) on the curves is known. This could be the case e.g. in MALDI-TOF analysis when several "shots" are combined in an average profile, for which the standard deviation on each value can be used to estimate the error. To calculate the **Weighted Pearson correlation**, each sum in the formula of the Pearson correlation coefficient is weighted. The weight decreases proportionally with the associated error (the sum of the errors on both curves). When there is no error associated with the curves, the weight becomes 1 and the formula reverts to the "normal" Pearson correlation coefficient.

Three parameters can be specified for **Curve based** coefficients:

- **Optimization** is a shift that you allow between any two patterns and within which the program will look for the best possible matching.
- **Curve smoothing** can be applied to remove noise from the densitometric curves. This is similar to the least square filtering, available during gel preprocessing (see 4.1.3.4). In contrast to the latter, applying curve smoothing in the comparison settings will not alter the original data.

- **Negative similarities** can be dealt with in different ways. If **Clip to zero** is selected, negative similarity values will be replaced by zero (no correlation). When **Unchanged** is set, the program will calculate with the negative values. **Absolute value** will treat negative and positive similarity values in the same way. **Negative similarities** values can only be obtained with the **Pearson correlation coefficient**, therefore the option is not available when any of the other coefficients is selected.

Band based category: Five different binary coefficients measure the similarity based upon common and different bands.

The **Jaccard** coefficient is calculated as:

$$S_J = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

with N_A and N_B the number of bands in profile A and B, respectively, and N_{AB} the number of common bands between the two profiles.

The **Dice** coefficient is calculated as:

$$S_D = \frac{2N_{AB}}{N_A + N_B}$$

The **Jeffrey's X** coefficient is calculated as:

$$S_X = \frac{1}{2} \left(\frac{N_{AB}}{N_A} + \frac{N_{AB}}{N_B} \right)$$

The **Ochiai** coefficient is calculated as:

$$S_O = \frac{N_{AB}}{\sqrt{N_A N_B}}$$

The **Different bands** coefficient is essentially a distance coefficient as it simply counts the number of different bands in two patterns. It is converted into a similarity by subtracting this distance value from 1. It is calculated as:

$$S_B = 1 - ((N_A + N_B) - 2N_{AB})$$

Following options are available for each of the five binary **Band based** coefficients:

- **Optimization** is a shift that one allows between any two patterns and within which the program will look for the best possible matching. This parameter applies for both curve-based and band matching coefficients.
- **Band matching** includes the **Position tolerance**, which is the maximal shift (expressed as a percentage of the pattern length) allowed between two bands to consider them as matching. With **Tolerance change**, a gradual increase or decrease in tolerance towards the end of the fingerprint can be specified. To understand the utility of **Position tolerance** in addition to **Optimization**, see the example in [Figure 4.2.3](#).
- **Band filtering** parameters are **Minimum height** and **Minimum surface**, which can be used to exclude weak or irrelevant bands. **Minimum height** is entered as a percentage of the OD range of the fingerprint. **Minimum surface** is expressed as a percentage of the total surface of all bands in the profile.

- **Uncertain bands** allows you to either **Ignore** or **Include** uncertain bands (see 4.1.3.6). When **Ignore** is chosen (the default option), uncertain bands are not taken into account. This means that in a pairwise comparison, an uncertain band is not penalized if there is no matching band on the other pattern. Conversely, if there is a band on the other pattern that matches an uncertain band, it will also be ignored in that comparison. When **Include** is selected from the drop-down list, uncertain bands are treated in the same way as certain bands, which means that an uncertain band which is not complemented by a band in the other pattern, is penalized.
- **Relaxed doublet matching**: this option allows a single band to match with two bands of a doublet, on condition that both bands of the doublet fall within the tolerance window from the single band.
- **Area sensitive**: this option makes the coefficient take into account differences in area between two matching bands. If for each matching band the areas on both patterns are exactly the same, the coefficient reduces to a normal binary coefficient; the more the areas differ, the lower the similarity will be.
- **Fuzzy logic** option: instead of a yes/no decision whether two bands are matching or not, the program lets the matching value gradually decrease between 1 and 0 with the distance between the bands, limited by the set band tolerance.

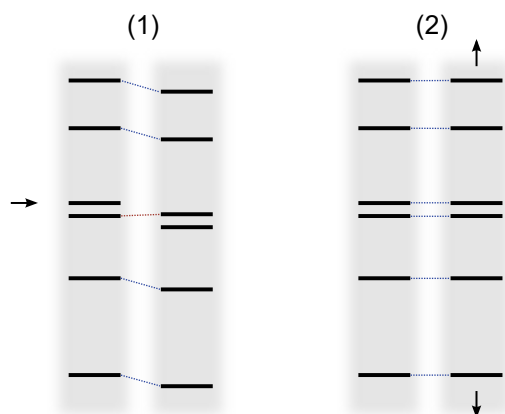


Figure 4.2.3: Effect of **Tolerance** (left) and **Optimization** (right) on the matching between patterns.

If a similarity matrix already exists for the selected experiment, an option **Keep existing similarity matrix** appears. When checked, the previously calculated similarity matrix will be used and all coefficient options (for both **Curve based** and **Band based** coefficients) will appear gray (disabled).



In 4.2.4, is discussed how to have the program automatically calculate the optimal position tolerance and optimization values for a fingerprint type.



The **Ignore** option for uncertain bands will only work when both **Fuzzy logic** and **Area sensitive** are disabled in the *Similarity coefficient* wizard page.

Check **Save as new default to database** if one wants the specified comparison settings to be saved in the database as default settings. When **Save as new default to database** is unchecked, the last comparison settings will only apply during the current *session* of BioNumerics; if the software is closed, the settings will not be saved.

The settings as defined in the *Comparison settings* wizard are stored along with the fingerprint type. A dialog box with the same settings can be called from the *Fingerprint type* window (4.1.5).

Select a similarity coefficient from the tree representation on the left and enter its parameters (see 4.2.4 how to optimize these values).

Press <**Next**> to go to the *Cluster analysis* wizard page.

Cluster analysis wizard page deals with the calculation of a dendrogram from the similarity matrix and is discussed in 13.2.6.

Select a clustering algorithm and press <**Next**> again in the *Cluster analysis* wizard page to start the cluster analysis.

When finished, a dendrogram and similarity matrix are shown for the fingerprint type.



Sometimes, when highly similar entries are displayed alongside in a cluster analysis, it can become clear that a single band or a number of bands were wrongly assigned during gel preprocessing (see 4.1.3.6). It is possible to correct such erroneous band assignments in the *Comparison* window. This procedure is described in 4.3.3.

4.2.2 Fingerprint display functions

For fingerprint types, additional information can be shown in the *Experiment data* panel. Before using these fingerprint display functions, make sure that the image of the fingerprint type is shown in the *Experiment data* panel by pressing the eye button (👁️) next to the experiment name in the *Experiments* panel.

Select **Fingerprints** > **Settings** > **Show metrics scale** (📏) to display the molecular weight scale of the selected fingerprint type. On the metric scale, active zones (see 4.2.3) will be indicated in yellow.

Select **Fingerprints** > **Show bands** (📊) to show or hide the band positions. One can also show only the band positions in the *Experiment data* panel without showing the actual image. Select **Layout** > **Show image** (🖼️) to show or hide the image.

Select **Fingerprints** > **Show densitometric curves** (📈) to show or hide small densitometric curves. One can also show the curves only, without showing the actual image.

Select **Fingerprints** > **Settings** > **Brightness & contrast...** (🖼️) to pop up the *Image brightness & contrast* dialog box for the fingerprint type (see Figure 4.1.10). The image brightness and contrast settings can be adjusted as discussed in 4.1.3.3.

In case only densitometric curves are available (e.g. in case of profiles from automated sequencers), it can be useful to display the curves as pseudo gel strips (reconstructed images). This option is selected in the *Fingerprint type* window as follows:

In the *Main* window, double-click on a fingerprint type in the *Experiments* panel to open its *Fingerprint type* window.

In the *Fingerprint type* window, select **Layout** > **Show curves as images**. If densitometric curves are now shown in a comparison, they will be displayed as pseudo gel strips.

In case densitometric curves have different intensities, the densitometric curves can be rescaled so that each curve fills the full available intensity range specified for the fingerprint type. This can be achieved as follows:

In the *Fingerprint type* window, select **Layout** > **Rescale curves**. If densitometric curves are now shown in a comparison, they will all be displayed with equal intensity.

The image of patterns can be shown with a space between the gel strips. To do so, select **Layout** > **Show space between gelstrips** in the *Fingerprint type* window.

4.2.3 Defining 'active zones' on fingerprints

When clustering fingerprints, one is not necessarily interested in comparing complete patterns. For example, when the loading well or the loading dye is comprised within the fingerprints, it may be better to exclude

such a region from the cluster analysis.

For each fingerprint type, it is possible to define regions which will be excluded from the analysis in all comparisons using that fingerprint type.

From the *Main* window, the *Fingerprint type* window of a fingerprint type can be opened by double-clicking the fingerprint type in the *Experiment types* panel. The *Fingerprint type* window is displayed as in Figure 4.2.4.

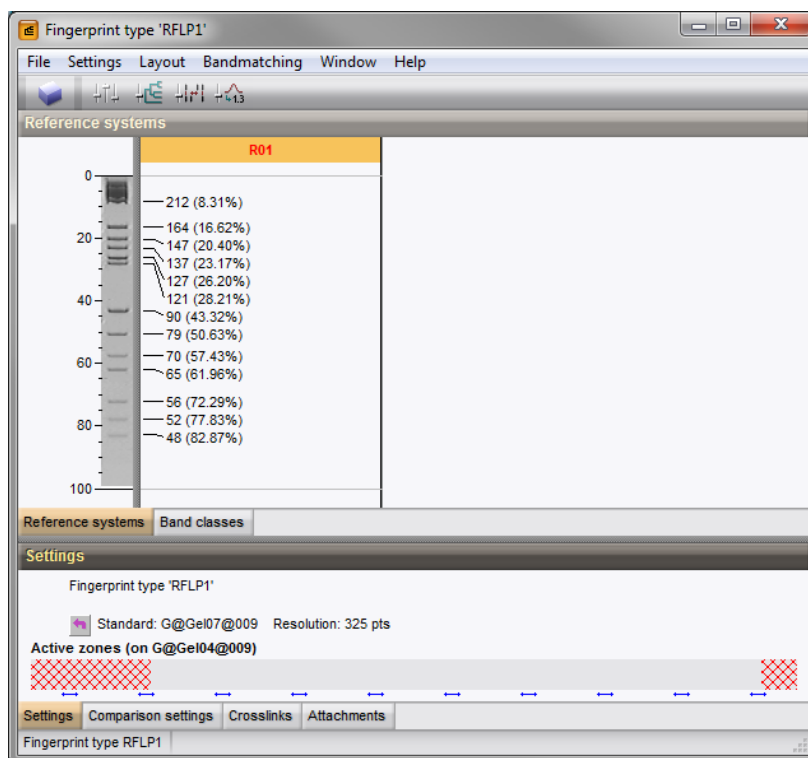


Figure 4.2.4: The *Fingerprint type* window with excluded regions defined in the *Settings* panel.

To *exclude* a region for comparison, hold the left mouse button and the **Shift**-key simultaneously while dragging the mouse pointer over the fingerprint. The excluded region becomes cross-hatched in red. In the bottom part of the window, those parts of the fingerprints that are *included* for comparison are shown as percentages (see Figure 4.2.4).

To *include* a region, hold the left mouse button (without holding the **Shift**-key), while dragging the mouse pointer over the fingerprint.



The image that is displayed at the bottom of the *Fingerprint type* window (see Figure 4.2.4), corresponds to the fingerprint of the highlighted database entry (= the "active" entry) at the moment the *Fingerprint type* window was opened. To accurately set the active zones, it might be useful to display a well-characterized profile such as a size marker.



You can exclude / include multiple regions. The defined regions apply both to comparisons based on densitometric curves and to comparisons based on band matching. Bands falling within an excluded region will not be considered for cluster analysis nor band matching analysis.

With **Layout** > **Hide inactive zones**, inactivated zones can be hidden from the view when profiles are displayed in the *Comparison* window, in the *Comparison print preview* window and on the final printout.

In the *Comparison* window, a single contiguous active zone can be set by selecting **Fingerprints** > **Settings** > **Set active zone...** (🔧). This calls the *Active fingerprint zone* dialog box (see Figure 4.2.5).

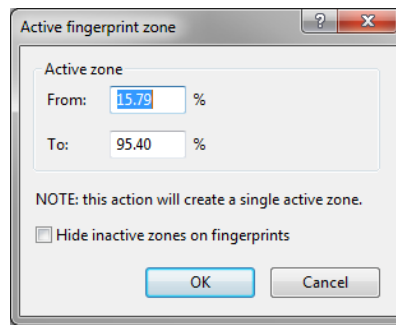


Figure 4.2.5: The *Active fingerprint zone* dialog box.

In the dialog box under **Active zone**, the start and end positions of the active zone can be entered as a percentage of the run length.

With the option **Hide inactive zones on fingerprint** checked, the inactive zones will be hidden from the view in the *Comparison* window.

Additionally, you can specify the exact start and end of the active zone(s) using a script available on Applied Maths website. The scripts can be launched from the *Main* window, using **Scripts > Browse internet...** and then selecting "Fingerprint related tools" > "Set active zones".

A cluster analysis should be recalculated to reflect any modifications to the active zones of a fingerprints.

4.2.4 Optimization of similarity coefficient parameters

BioNumerics offers a very interesting option to automatically calculate the optimal settings for similarity coefficient settings for a given fingerprint type. The principle is as follows: the user selects a number of entries which he or she wants to cluster into a comparison. The program will calculate similarity matrices with different values for a selected parameter (e.g. optimization). Within a limited range, the optimal setting for a similarity coefficient parameter yields the matrix with the highest group contrast: scores as high as possible within groups and as low as possible between groups. This translates in the highest standard deviation on the matrix of similarity values. The same process can be launched to find the best range for the next parameter (e.g. band tolerance), etc. Given the principle of the method, it is important to select entries belonging to different groups or showing a maximum of heterogeneity.

The best way to proceed is with comparison groups (see 13.3.4) already defined, e.g. based upon cluster analysis or partitioning (see 13.3.4). The program will then optimize the intergroup separation based upon these groups. If no groups are defined, the standard deviation of the whole matrix is optimized, which also works in case the comparison contains some groups of more related patterns.

Click on the fingerprint type in the *Experiments* panel and select **Clustering > Optimize similarity coefficient parameter...** The *Similarity coefficient parameter optimization* wizard appears (see Figure 4.2.6).

On the first page, the parameter to be optimized can be selected. Depending on whether a curve-based or band-based coefficient is used, different parameters are listed: **Optimization** and **Curve smoothing** for curve-based coefficients or **Optimization**, **Tolerance**, **Tolerance change**, **Minimum height** and **Minimum surface** for band-based coefficients.

The currently used coefficient is displayed.

Select the parameter to optimize, e.g. **Optimization**, and press <Next>. The second page of the *Similarity coefficient parameter optimization* wizard is displayed (Figure 4.2.7).

With **From** and **To**, a range of optimization values can be specified as the lowest and highest value that the

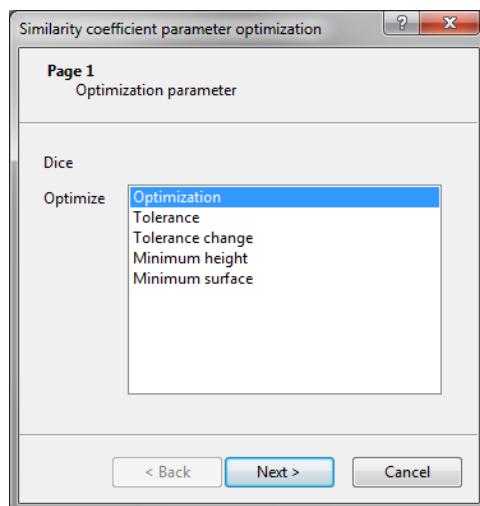


Figure 4.2.6: The first page of the *Similarity coefficient parameter optimization* wizard, where the parameter to be optimized can be selected.

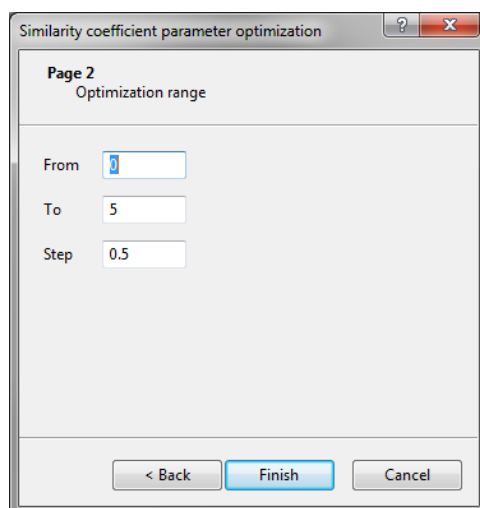


Figure 4.2.7: The second page of the *Similarity coefficient parameter optimization* wizard, where the optimization range can be specified.

program will evaluate. **Step** is the interval at which optimization values will be evaluated. It is important to keep in mind that a wide range and a small **Step** will result in a large number of optimization values to be tested and therefore in a long calculation time.

Enter a range and a step size and press **<Next>**. BioNumerics now tries to calculate the best **Optimization** value. When an optimal value was found within the range specified, this value is reported. The program asks "Copy this value to the similarity coefficient?".

If you press **<Yes>**, the comparison settings will be updated with the optimal **Optimization** value. Next, the *Parameter optimization* window appears (see Figure 4.2.8).



If the message "No optimal value was found within the given range." appears, this means that no maximum was detected in the parameter optimization curve: the curve is either flat or continuously increasing or decreasing over the examined range. A solution could be to base the parameter optimization on a more heterogeneous set of samples and/or to define comparison groups prior to launching the calculations.

The *Parameter optimization* window shows a diagram with the group separation (Y-axis) in function of the

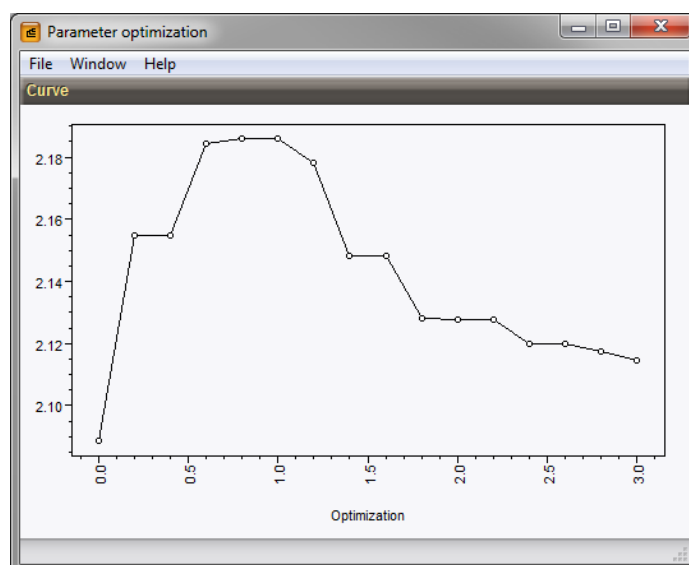


Figure 4.2.8: The *Parameter optimization* window.

allowed optimization (X-axis). Obviously, the maximum in the curve corresponds to the optimal optimization value. The window can be closed with **File** > **Exit**.

Next, to calculate the optimal tolerance, proceed as follows:

Close the *Parameter optimization* window and select **Clustering** > **Optimize similarity coefficient parameter...** again.

This time, select **Tolerance** and press <Next>.

Leave the default range and step selected and press <Next> again.

Similar as for **Optimization**, the optimal value for **Tolerance** is reported. If you answer <Yes> to the question "Copy this value to the similarity coefficient?", the comparison settings will be updated with the optimal **Tolerance** value.

The *Parameter optimization* window appears, showing a diagram with the group separation in function of the allowed **Tolerance**.

The procedure can be repeated for the similarity coefficient parameters **Tolerance change**, **Minimum height** and **Minimum surface**.

4.2.5 Exporting fingerprint information

Two options are available to export fingerprint information from the *Comparison* window in text format:

When bands are shown on the image (see 4.2.2), they can be exported with **File** > **Export** > **Export bands...**. The export file, popped up as `export.csv` in the default CSV editor (often MS Excel) or as `export.txt` in Notepad (depending on the preferences set, see 2.3.3), contains the key of the entry, and a list of band positions as relative run lengths (in percent) and molecular weight (in case a regression curve is calculated for the reference system used; see 4.1.5).

When densitometric curves are shown on the image (see 4.2.2), they can be exported with **File** > **Export** > **Export densitometric curves...**. The export file, `export.csv` or `export.txt`, contains the list entry keys separated by tabs, and a list of densitometric curves, of which the curves are listed as columns, separated by tabs.

Chapter 4.3

Band matching and polymorphism analysis

4.3.1 Introduction

Band matching is a comparison function which applies only to fingerprint types. It can be executed on any selection of entries from the database. In a first step, BioNumerics divides all the bands found among the selected patterns into *classes of common bands* (1 to 8 in Figure 4.3.1). As such, every band of a given pattern belongs to a class, and conversely, every band class is represented by a band on one or more patterns. The result is shown in Figure 4.3.1.

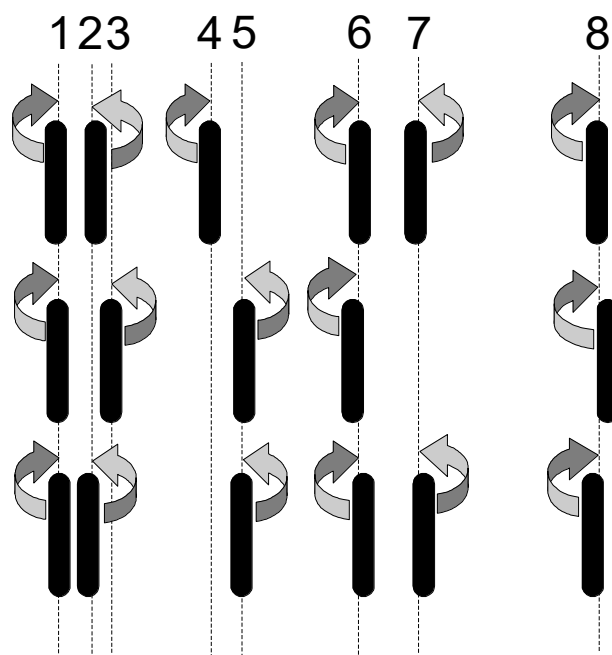


Figure 4.3.1: Comparative quantification: bands are assigned to classes.

Clearly, the number of band classes distinguished will depend on the *optimization* and the *position tolerance* that is allowed between bands considered as matching. For example, when a larger position tolerance is specified, more bands will be grouped in the same class than when a small position tolerance is chosen. In Figure 4.3.1, taking a larger position tolerance would have resulted in the merging of band classes 2 and 3, whereas a smaller position tolerance would have resulted in two separate classes for band class 8.

For each pattern, a particular band class can have two states: present or absent. This is the basis for *polymor-*

phism analysis, a tool which allows comparative binary (+/-) tables to be generated, displaying polymorphic bands between the selected patterns. These tables, created as text or tab-delineated files, are ready for export to other specialized software for statistics, genetic mapping or other further analysis. The binary table for the above example (Figure 4.3.1) is shown in Figure 4.3.2.

	1	2	3	4	5	6	7	8
Pattern 1	+	+	-	+	-	+	+	+
Pattern 2	+	-	+	-	+	+	-	+
Pattern 3	+	+	-	-	+	+	+	+

Figure 4.3.2: Binary presence/absence table of banding patterns.

Instead of using binary (+/-) data, the same tables can be generated using band intensities obtained from the curves (band heights or surfaces) or from the two-dimensional pattern contours (volumes or concentrations).

The use of band matching tables is obvious: it provides a binary or numerical character table for fingerprint type patterns, which allows a number of statistical techniques to be applied, including minimum spanning trees (16.4.4), maximum parsimony trees (16.4.3), dimensioning techniques such as principal components analysis and related techniques (17.4), and bootstrap analysis on dendrograms (13.3.7).

Band matching has a number of important applications such as screening for genetic markers (e.g. using AFLP) in plant and animal breeding and microbial community analysis using PCR-DGGE, t-RFLP or DHPLC. Please note that the Fingerprint type module (FP) needs to be present in your BioNumerics configuration before a band matching analysis can be calculated.

4.3.2 Creating a band matching

A band matching analysis is done in the *Comparison* window. Therefore, a comparison containing the entries on which you want to perform a band matching should first be created or opened.

Click on the fingerprint type in the *Experiments* panel on which you want to perform a band matching and select **Layout > Show image** (👁) or pressing the eye button (👁) next to the experiment name in the *Experiments* panel.

Select **Fingerprints > Perform band matching...** (🔍). This pops up the *Perform band matching* dialog box (see Figure 4.3.3).

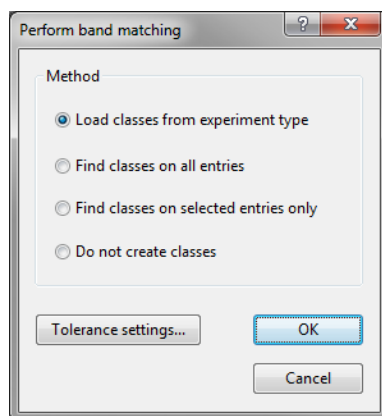


Figure 4.3.3: The *Perform band matching* dialog box.

This dialog box lists four different band matching options.

- **Load classes from experiment type:** the band classes stored with the experiment type are loaded (see 4.3.6). This way you can have perfect control on what bands to use in the analysis.
- **Find classes on all entries:** A band matching is performed on all entries within the comparison.
- **Find classes on selected entries only:** A band matching is performed on the currently selected entries only.
- **Do not create classes:** Band classes are not automatically assigned, but the fingerprint is opened in "band editing mode" for manual band assignment.

Pressing <**Tolerance settings**> opens the *Position tolerance settings* dialog box (Figure 4.3.4).

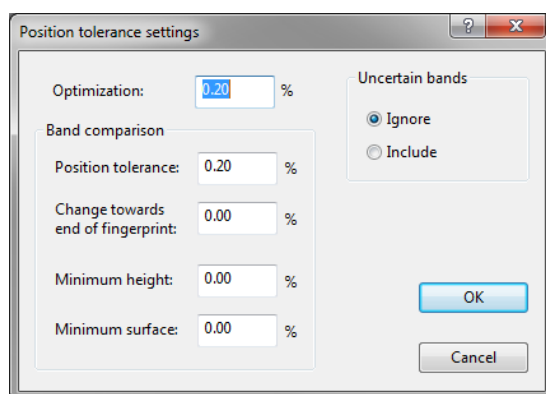


Figure 4.3.4: The *Position tolerance settings* dialog box of a fingerprint type.

The **Position tolerance** is the maximal shift allowed (in percentage of the pattern length) between two bands allowed to consider them as matching. With **Change towards end of fingerprint**, you can specify a gradual increase or decrease in tolerance.

The **Optimization** is a shift that you allow between any two patterns and within which the program will look for the best possible matching. To understand the utility of **Optimization** in addition to **Position tolerance**, see the example in Figure 4.2.3.

With **Minimum height** and **Minimum surface**, you can exclude weak or irrelevant bands.

The **Uncertain bands** option allows you to either include or ignore uncertain bands (see 4.1.3.6) in the calculation of pairwise similarity values. This setting does not affect the actual band matching.

Enter the desired tolerance settings in the *Position tolerance settings* dialog box and press <**OK**>.

In case no band classes are yet defined for the fingerprint type (see 4.3.6), select **Find classes on all entries** in the *Perform band matching* dialog box and press <**OK**>.

The program has now defined the band classes and has associated each band with a class. The band classes are shown as blue lines and the bands are linked to a class in red (Figure 4.3.5).



Band classes are only defined within active zones of the fingerprint type. Active zones can be set in the *Fingerprint type* window of the corresponding fingerprint type (see 4.2.3).

Zoom in on the image as necessary using **Layout > Zoom in** (🔍, **Ctrl+Page Up**) and **Layout > Zoom out** (🔍, **Ctrl+Page Down**) or by using the zoom sliders (see 2.3.7 for instructions on how to use the zoom sliders). The latter option allows you to zoom separately in the horizontal (🔍) and vertical (🔍) direction. Horizontal zooming can also be achieved via **Layout > Stretch (X dir)** (**Ctrl+Shift+Page Up**) and **Layout > Compress (X dir)** (**Ctrl+Shift+Page Down**). Zooming in the horizontal direction only can be an interesting

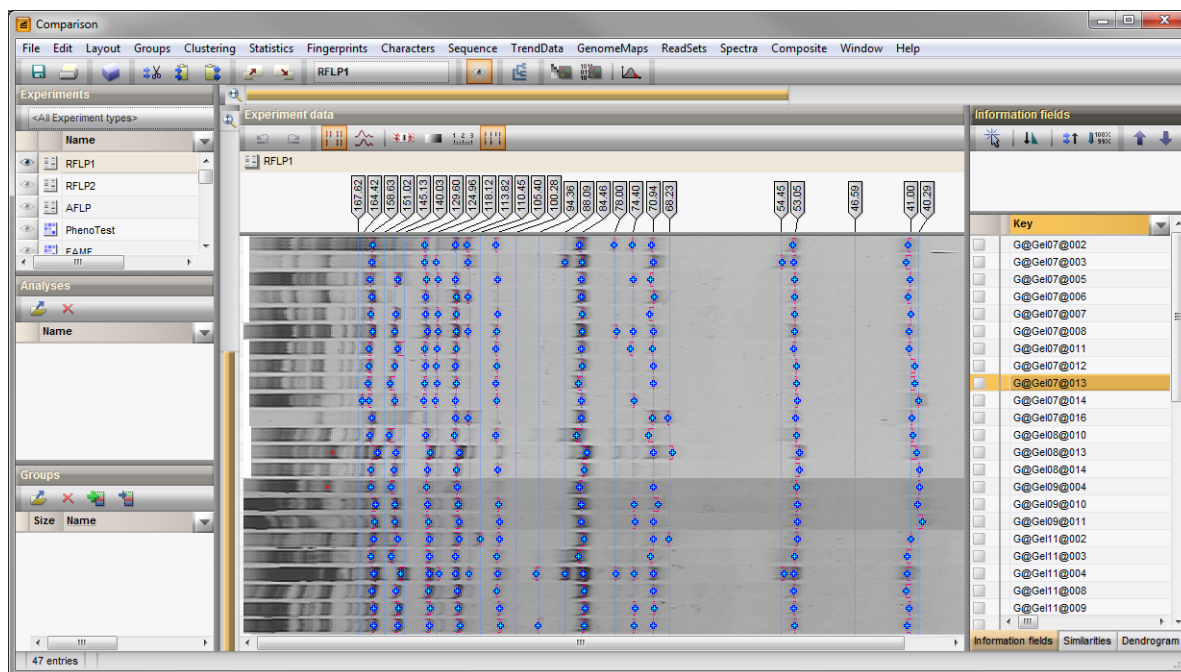


Figure 4.3.5: Band matching analysis in the *Comparison* window (to optimize display use, the *Similarities* panel and *Dendrogram* panel are docked with the *Information fields* panel as tabbed view).

option for long patterns with numerous small bands (e.g. **AFLP** in the **DemoBase Connected**). This causes the image to be enlarged in the horizontal direction only, so that sharp bands become better visible, without losing the overview of a large number of patterns.

Use **Fingerprints** > **Settings** > **Show metrics scale** () to display the molecular weight scale of the fingerprint type.

After having performed a band matching, all band classes are labeled with a band class label. The band class labels are listed on top of the image. If a band class is selected, its label is highlighted (Figure 4.3.6).

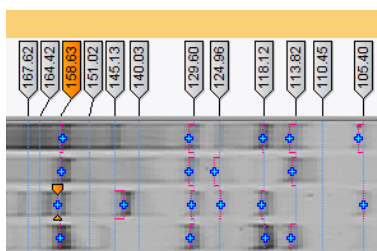


Figure 4.3.6: Band class labels.

If a regression curve is calculated for the reference system(s) of the selected fingerprint type (see 4.1.5), the metric positions of the band classes are displayed in the labels (e.g. 167.62; 164.42; ...).

Double-click on a band class label, or select **Fingerprints** > **Band class information...** (**Ctrl+I**) to open the *Band class information* dialog box (Figure 4.3.7).

This dialog box contains detailed information on the band class:

- **Name:** If a regression curve is calculated for the reference system(s), the default name of the band class label is the normalized position of the band class. This is the average position of all bands belonging to that band class, expressed as a percentage of the normalized track length. Each band

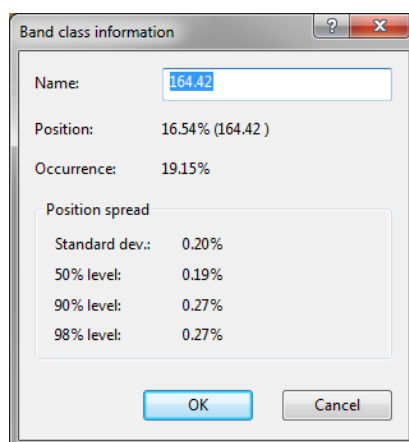


Figure 4.3.7: The *Band class information* dialog box.

class name is editable and can be changed to any name of your choice. Band class names can be edited here, but also from within the *Fingerprint type* window (see 4.3.6). When changing the band class names, the metric positions of the band classes remain present in the database (see 'Position (metrics)' column in Figure 4.1.86).

- **Position:** This reflects the relative position of the band class, derived from the regression curve.
- **Occurrence:** This corresponds to the relative occurrence of bands in the band class, expressed as a percentage of the total number of entries in the band matching analysis.
- **Position spread:** This box lists the standard deviation of the bands to the band class and the scores of the 50th, 90th and 98th percentile. The scores are the relative positions below which respectively 50, 90 and 98% of the bands are found.

Pressing <OK> will close the *Band class information* dialog box and will save the band class **Name** to the database.

4.3.3 Manual editing of bands

Band assignments that were made during gel preprocessing in the *Fingerprint processing* window (see 4.1.3.6) can be modified in the *Comparison* window and any changes made are saved on the corresponding gels. Before any editing can be done, a band matching needs to be initiated so that the fingerprint data are displayed in "band editing mode":

Select **Fingerprints > Perform band matching...** (🔍).

In the *Perform band matching* dialog box (Figure 4.3.3), select **Do not create classes** and press <OK>.

The "Perform band matching" button is now shown as 🛑 and the bands in the fingerprint type can be edited for the entries in the comparison.

To add a band at a certain position, proceed as follows:

Click on the position where the band should occur. A single selection flag indicates the selected position:



Select **Fingerprints > Add band (Enter)**. The band is now added and is displayed with double selection flags: 🚩.

An alternative procedure is to use **Ctrl+Shift+click** to add a band at a desired position. If a band class is selected, the band is automatically assigned to that class.



When *Fingerprints* > *Snap to peaks* is checked (default), the band will be created at the local maximum nearest to the cursor position. When unchecked, bands will be created at the exact cursor position.

Before a band or a number of bands can be deleted, the band(s) need to be selected first. This can be achieved in several ways:

- To select a single band, just click on it with the mouse.
- To select a number of adjacent bands, press the **Shift**-key on the keyboard while dragging a rectangle with the mouse.
- To select all bands belonging to a certain band class, double-click on any band in that band class. A band class needs to be defined first (see 4.3.2).

Selected bands are indicated with double selection flags (🚩) and can be deleted with *Fingerprints* > *Remove selected band* (Del).

When adding or deleting bands, the Undo/Redo functions are available. To undo a change, select *Fingerprints* > *Undo* (↶, Ctrl+Z). To redo an undone change, select *Fingerprints* > *Redo* (↷, Ctrl+Y).



Since removing bands is easier than adding them (the former can be done for a selection, while the latter is always done one by one), it is often more convenient to have bands searched automatically in step 4 of the gel processing with a low threshold and delete bands assigned in excess later compared to adding a large number of bands manually.

When a comparison is saved using *File* > *Save* (💾, Ctrl+S) or *File* > *Save as...*, any modified band information (added or deleted bands) is automatically saved to the corresponding gels. Modified band information can also be stored without making any modification to the comparison itself by selecting *Fingerprints* > *Save modified band information...*

4.3.4 Manual editing of a band matching

Due to shape or distribution, the program does not always assign the bands to the correct class. Therefore, you can manually correct the assignments.

For the manual band matching editing tools, a multilevel undo and redo function is available. The undo function can be accessed with *Fingerprints* > *Undo* (↶, Ctrl+Z). The redo function is accessible through *Fingerprints* > *Redo* (↷, Ctrl+Y).

In Figure 4.3.8, the band marked with the arrow is assigned to the left of two close classes, whereas it should be assigned to the right class.

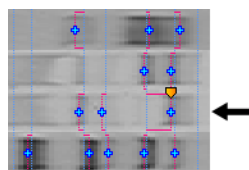


Figure 4.3.8: Detail of band class assignments.



You can easily see which bands belong to a given band class by double-clicking on the vertical blue dotted line that represents the class: all bands that belong to the class are selected with an orange flag.

Reassigning a band to another class can be done with a simple drag-and-drop procedure: Select the band that was wrongly assigned. While pressing the mouse button, drag it to the band class where it should be assigned to and release the mouse button.

If you do not wish to use an individual band in a band matching analysis, you can undo its assignment by clicking on the band that you want to unassign and dragging it outside of the gel strip.

A whole band class is deleted by clicking on a band belonging to that band class and selecting **Fingerprints > Remove band class (Shift+Del)**.

If different bands are incorrectly assigned to the same class (see Figure 4.3.9 for an example), you can create a second class by selecting a band or a number of bands which should belong to a new class and **Fingerprints > Add new band class (Shift+Enter)**.

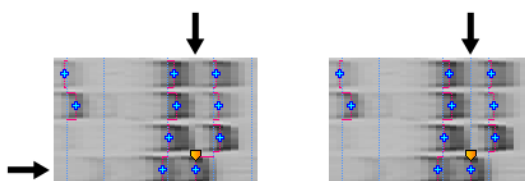


Figure 4.3.9: Splitting up a band class into two band classes.

The program asks "Do you want to auto assign bands to the new class?". If you press <No>, the new band class will contain only the selected band(s). If you select <Yes>, all bands that are closer to the new band class are automatically reassigned to that new class. In order to reassign bands to the other class, follow the drag-and-drop procedure describe above.

If bands are incorrectly assigned to different classes, you can merge the classes as follows:

Choose a band which occurs quite in the middle of the two classes and select **Fingerprints > Add new band class (Shift+Enter)**. Press <No> when the program asks to auto assign bands to the new class.

Choose a band which belongs to the left class and select **Fingerprints > Remove band class (Shift+Del)**.

Choose a band which belongs to the right class and select **Fingerprints > Remove band class (Shift+Del)**.




Select the new band class to which all the bands should belong. The band class label becomes highlighted.

Select **Fingerprints > Auto assign all bands to class**.

After reassigning bands, removing and adding bands etc., the band class position may not be the center anymore. You can correct the position of the band class with **Fingerprints > Center class position**.


If all assignments are corrected, you can save the band matching with **File > Save** (📁, Ctrl+S) or **File > Save as...**



A band matching is saved along with the comparison. When a comparison is opened and a band matching is available for the experiment type selected, the  button shows up . The graphical representation of the band matching can be displayed again by selecting **Fingerprints > Show bands** (.

4.3.5 Adding entries to a band matching


Since a band matching analysis and the associated table can be saved, it should be possible to delete entries from, or add entries to the band matching at any time.

To delete some entries, simply select the entries and **Edit > Cut selection** (, Ctrl+X).

If entries are added however, it is possible that those new entries contain bands that are not defined as a band

class yet. If you have performed some editing work to the band classes already, it would be beneficial to preserve the existing band classes, and simply associate the bands of the new entries to the existing classes, and introduce new classes in those cases where the new entries have bands that do not fit in any of the existing classes. This is achieved as follows:

In the *Main* window, select the entries you wish to add to the band matching and use **Edit > Views > Copy selection** (**Ctrl+C**).

Using **Edit > Paste selection** (, **Ctrl+V**) in the *Comparison* window, the selected entries are placed back in the band matching.

Select **Fingerprints > Search band classes...** The *Perform band matching* dialog box as in Figure 4.3.3 is shown. Check **Find classes on all entries** and press **<OK>**.

The program now asks "Remove existing band classes?". In order to preserve the existing band matching, it is important to answer **<No>** to this question.

4.3.6 Saving band classes to the fingerprint type

After having defined band classes in the *Comparison* window, you can save the band classes to the corresponding fingerprint type with **Fingerprints > Save band classes to experiment type...**

The software asks "Do you want to keep the current classes in this experiment?". If you select **<Yes>**, the existing classes in the experiment type are left untouched and if any new classes are present in the comparison, they are added to the classes in the experiment type. If **<No>** is selected, any classes in the experiment that do not exist in the current band matching are removed.

In case no band classes are present yet for the fingerprint type, you can either select **<Yes>** or **<No>** with the same result.

To illustrate that the band classes are indeed saved with the fingerprint type, open its corresponding *Fingerprint type* window and display the *Band Classes* panel. The functionality of this panel is discussed in 4.1.5.10.

If band classes are saved for a fingerprint type, the band classes can be loaded when checking **Load classes from experiment type** in the *Perform band matching* dialog box (see Figure 4.3.3).

The ability to edit, save and load band classes has several interesting implementations:

- Perfect control on what bands to use in the analysis.
- Every new set of fingerprint profiles can easily be compared based upon the predefined set of band classes.
- Band matching tables can be reduced to the relevant information, e.g. bands that carry genetic marker information.

4.3.7 Band and band class filters

When searching bands in complex patterns, especially those for which the terminal step is a PCR reaction such as AFLP patterns, it is sometimes difficult to define objective criteria as to what is a band and what is not a band. However, when the user examines a set of patterns by eye, it often becomes easier to decide whether a band is valid or not, because the user automatically compares the band with those on neighboring patterns, thus obtaining information which cannot be obtained by inspecting the pattern alone. This is more or less the way the band filters work in the band matching application of BioNumerics: in a first step, band

classes are defined over all patterns; then the relative areas of all bands of a given class are averaged, and if a band deviates more than a certain percentage from this average, it is not considered as being a matching band for this class.

Using this tool, it is possible to define more bands on the gels than one would usually do, without spending a lot of time deleting and adding bands manually. Using the band matching filters, weak bands or artifacts that do not reflect the expected intensity will be filtered out automatically, and the assignment of bands is often as reliable as after hours of band editing work.

Select **Fingerprints** > **Band class filters** > **Band class filter...** to pop up the *Band filtering settings* dialog box (Figure 4.3.10).

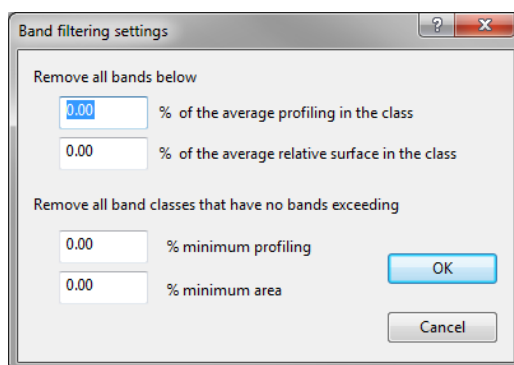


Figure 4.3.10: The *Band filtering settings* dialog box for band matching.

This dialog box exists of two parts: the upper part **Remove all bands below** is to filter individual bands within a given band class, and the lower part **Remove all band classes that have no bands exceeding** is to remove all band classes that do not contain any significant band.

Similar as for band searching, the band class filters consist of two separately working components: a *profiling* component, which is the height of the band or class, and an *area* component, which is the relative area (surface).

Within a band class, you can **Remove all bands below** a certain **% of the average profiling in the class**. If you enter 80%, this means that, if the height of a band is lower than 80% of the average profiling calculated for its class, it will not be matched with that class, and the band will be recorded negative in the band matching table. Note that the profiling of a band is an absolute measure: if a pattern as a whole is rather weak, many of its bands may be excluded from the band matching just by this fact. In such cases, we recommend to take the surface as filtering factor.

Within a band class, you also can furthermore **Remove all bands below** a certain **% of the relative surface in the class**. In this case, if you enter 80%, all bands that have a *relative* surface less than 80% of the average surface for the band class will not be matched with that class, and the bands will be recorded negative in the band matching table. Since the surface is relative to the total surface of a pattern, weak patterns in principle will not be treated differently compared to dark patterns.

In case of complex patterns such as AFLP, many band classes consist of just one weak band, spot or artifact and have no genetic or taxonomic relevance. Such band classes are just filling up the band matching table, and being treated equally important, they are disturbing the information provided by the band matching table. Therefore, BioNumerics offers the possibility to have all band classes excluded from the band matching table that do not contain at least one clear relevant band.

With **Remove all band classes that have no bands exceeding** a certain **% minimum profiling**, you can remove all irrelevant band classes based upon the minimum height of the bands included. If you enter 20%, this means that a band class for which the highest band is less high than 20% of the OD range of the fingerprint type will be considered irrelevant and will be removed.



This is again a non-relative parameter. If by incidence a band class is formed by a set of weak patterns, it may be excluded incorrectly. If this happens to be a problem, we recommend to use the more reliable feature of **% minimum area** only.

With **Remove all band classes that have no bands exceeding** a certain **% minimum area**, you can remove all irrelevant band classes based upon the minimum area of the bands included. The minimum area is defined as the area relative to the total area of a pattern. If you enter 20% here, a band class that contains no band with an area bigger than 20% of its pattern's total area will be removed from the band matching table.

4.3.8 Exporting band matching information

Band matching information can be exported as a binary (presence/absence) table or as a quantitative character table.

In the *Comparison* window, with a band matching analysis performed (see 4.3.2 for instructions), select **Fingerprints > Export band matching...** The program will ask "Export quantitative information?".

Press **<No>** to export the band matching information as a binary (presence/absence) table or **<Yes>** to export a quantitative band matching table in tab-delimited format.

The exported band intensity values are based on the *comparative quantification settings* for the used fingerprint type. This option can be defined in the *Fingerprint type* window (see 4.1.5), but it can also be changed in the *Comparison* window, by selecting **Fingerprints > Settings > Comparative quantification settings...** This action opens the *Comparative quantification* dialog box (Figure 4.3.11).

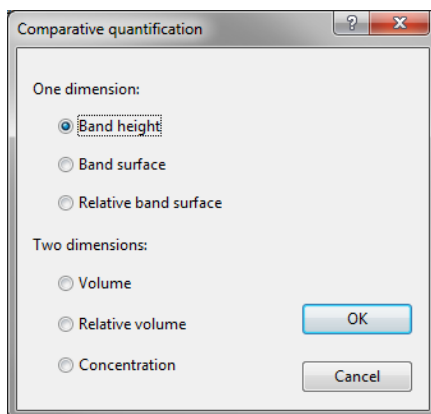


Figure 4.3.11: The *Comparative quantification* dialog box.


One dimension quantification is based on the densitometric curves extracted from the patterns (see 4.1.3.6): **Band height** is the height of the peak; **Band surface** is the area under the Gaussian curve approximating a band; **Relative band surface** is the same as band surface, but expressed as a percentage of the total band area of the pattern.

Two dimensions quantification is based on the band contours of the two-dimensional pattern images (see 4.1.3.6): **Volume** is the absolute volume within the contour; **Relative volume** is the same as a percentage of the total band volume of the pattern; **Concentration** is the physical concentration unit the user has assigned based upon regression through known calibration bands.

If no two-dimensional quantification is performed for the gels, it is obvious that one should select among the first three options.

4.3.9 Tools to display selective band classes

In a band matching analysis, it can occur that one is only interested in a specific subset of band classes, e.g. those that occur in a reference set of patterns. Following procedure allows you to confine the band matching analysis to these classes:

Select the entries in the comparison that correspond to the reference set (see 3.3.8 for instructions) and use **Fingerprints > Perform band matching...**  to create a new band matching.

In the *Perform band matching* dialog box (see Figure 4.3.3), check the option **Find classes on selected entries only**.

With this option checked, the program will only create band classes for bands found on the entries in the selection.

Use **Fingerprints > Auto assign all bands to all classes** to let the program assign the bands of the non-selected entries to the existing band classes.


BioNumerics offers an interesting tool to display only the polymorphic bands. To make this tool as flexible as possible, the polymorphic bands are only searched for within the selection list. For genetic mapping purposes, the user can select the patterns from two (or more) parent entries, and have the program display only the polymorphic band classes between these two patterns. This reduces the size of the band matching table to contain only the polymorphic bands of interest. Of course, the user can add or delete band classes afterwards, as desired (see 4.3.4).

With a band matching present, select the entries (see 3.3.8) for which you want to display the polymorphic bands.

Select **Fingerprints > Band class filters > Polymorphic classes only (for selection list)**. Only the band classes that are polymorphic between the selected two patterns are now displayed.

4.3.10 Creating a band matching table for polymorphism analysis

Before a presence/absence table as shown in Figure 4.3.2 can be displayed in BioNumerics, you will need to define a *composite data set*, containing the fingerprint type as input. A composite data set is a character table that contains all the characters of one or more experiment types (see 11.1). Such a character table is necessary to convert the band classes and represent them as presence/absence tables. See 11.1.2 on how to define a composite data set that includes the fingerprint type(s).

When a comparison is opened after the new composite data set has been defined, the composite data set is listed in the *Experiments* panel of the *Comparison* window. When a band matching is already performed on the fingerprint type (see 4.3.2), the band matching values will be automatically filled in as character values. The binary band matching table can be displayed by pressing the eye button  next to the experiment name in the *Experiments* panel (Figure 4.3.12). In order to reveal the complete information on the band classes, it may be necessary to drag the separator line between the table and its header downwards.



You can scroll between the image of gel patterns and the character table using the scroll bar at the bottom of the image panel. Once the character table is present, it is still possible to edit the band class assignments on the patterns. The character table is updated automatically.



Band classes that have been created by the user are marked with an asterisk (*).

Use **Composite > Export character table...** to export a space or tab-delineated text file of the binary band matching table.

When the program asks "Use tab-delineated fields", you should answer **<Yes>** to produce a tab-delineated

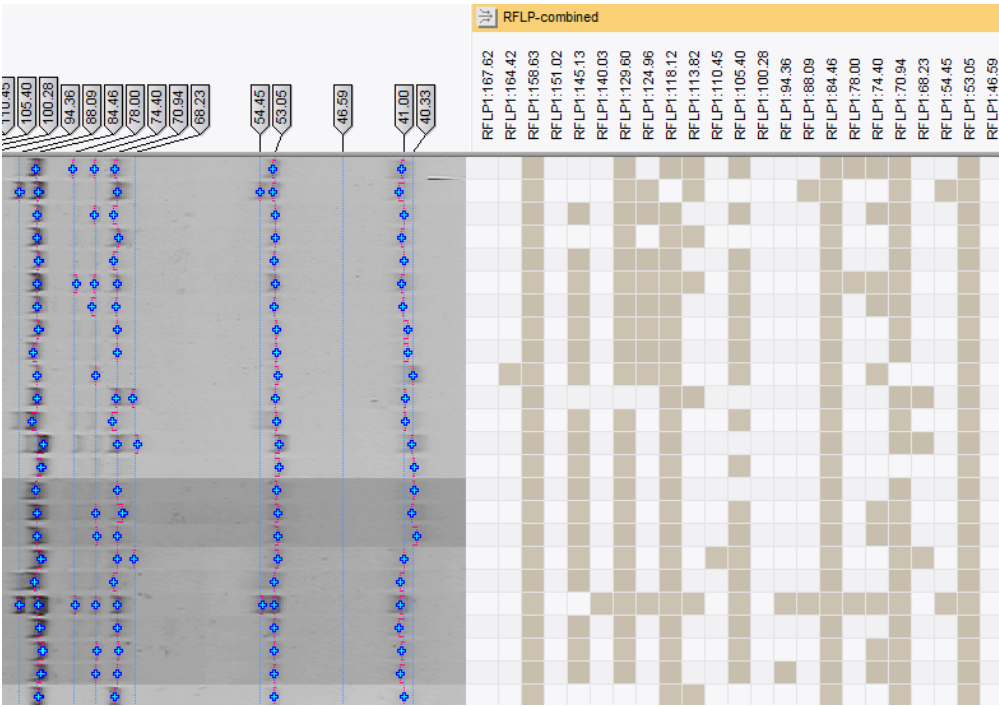


Figure 4.3.12: Binary band matching table; detail.

text file. The tab-delineated table looks as shown in Figure 4.3.13 and is in fact very similar to the one obtained via the command *Fingerprints > Export band matching....*

	RFLP1:167.62	RFLP1:164.42	RFLP1:158.63	RFLP1:151.02
G@Gel07@004	0	1	0	0
G@Gel11@005	0	1	1	0
G@Gel07@017	0	1	0	0
G@Gel11@006	0	1	1	0
G@Gel11@011	1	1	0	0
G@Gel08@016	0	1	1	0
G@Gel07@015	0	1	0	0
G@Gel07@010	0	1	1	0
G@Gel08@003	0	0	1	0
G@Gel08@006	0	0	1	0
G@Gel08@015	0	0	1	0

Figure 4.3.13: Binary band matching character table exported from BioNumerics (tab-delineated).

In the tab-delineated format, the band classes (header) and the band presence/absence table are given in columns separated by tabs. This format is the easiest to import in spreadsheet or database software packages.

To show the intensity of the bands as colors, select *Composite > Show quantification (colors)* (🎨).

In the default color map (see Figure 4.3.14 for an example), the color ranges from blue (weakest bands) over cyan, green, yellow, orange to red (darkest bands). However, other composite data set quantification colors can be defined in the preferences (see 2.3.3). The intensity is based upon the *comparative quantification settings* for the used fingerprint type. This option can be defined in the *Fingerprint type* window (see 4.1.5) or in the *Comparison* window (see 4.3.8).

To show the numerical intensities of the bands, select *Composite > Show quantification (values)* (📊). Now, with *Composite > Export character table....*, a numerical band matching table is created in text format, separated by tabs or spaces (Figure 4.3.15).

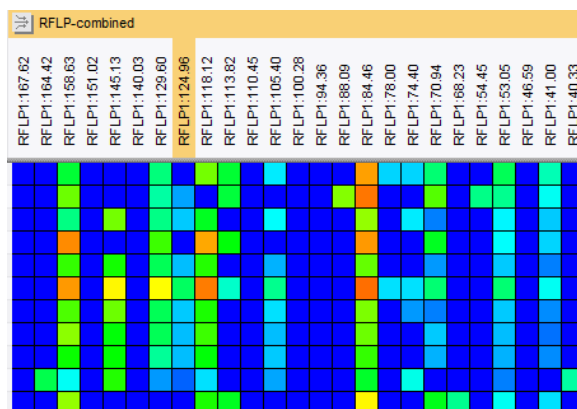


Figure 4.3.14: Intensity of bands shown in color.

RFLP1:94.36
RFLP1:88.09
RFLP1:84.46
RFLP1:78.00
RFLP1:74.40
RFLP1:70.94
RFLP1:68.23
RFLP1:54.45
RFLP1:53.05
RFLP1:46.59
RFLP1:41.00
RFLP1:40.33

HEADER:
Band classes

0.000.0025.273.482.998.950.000.008.310.0013.310.00
0.0010.390.000.008.8510.460.000.0011.780.0010.230.00
0.0013.4925.320.000.0015.510.009.287.110.005.050.00
0.000.0019.790.000.0013.710.000.005.130.005.550.00
0.000.0021.672.362.426.390.000.006.290.006.610.00
0.000.0025.060.000.003.920.000.006.330.005.270.00
0.0011.890.000.000.0010.340.005.477.440.003.690.00
0.000.0026.110.000.004.970.000.005.850.003.650.00
0.000.0026.200.003.762.590.000.005.950.004.560.00

TABLE:
Rows=entries

TABLE:
Rows=entries

Figure 4.3.15: Numerical band matching character table exported from BioNumerics (space-delineated).

4.3.11 Finding discriminative bands between entries

The use of a composite data set allows discriminative bands to be searched for in a band matching table. This feature works on the selected entries in the comparison and rearranges the characters in the composite data set according to their ability to discriminate the selected entries from the unselected ones.

Select the entries that you want to discriminate. These entries can correspond e.g. to a cluster in a dendrogram or a type designation in an information field.

Select **Composite > Discriminative characters**.

The characters (band classes) are reorganized in such a way that those characters negative (low values) for the selected entries and positive (high values) for the other entries occur left; and those characters that are positive (high values) for the selected entries and negative (low values) for the other entries occur right (see Figure 4.3.16).

In a composite data set, it is possible to list the entries according to the value of a selected character. In case of banding patterns, the entries will be ordered by the intensity of a selected band. This feature allows for a particular band the entries to be found in which the band is present or not.

Show the band table as intensity table with **Composite > Show quantification (colors)** (🎨).

Click on a band class in the band classes header and select **Composite > Sort by character** (📊).

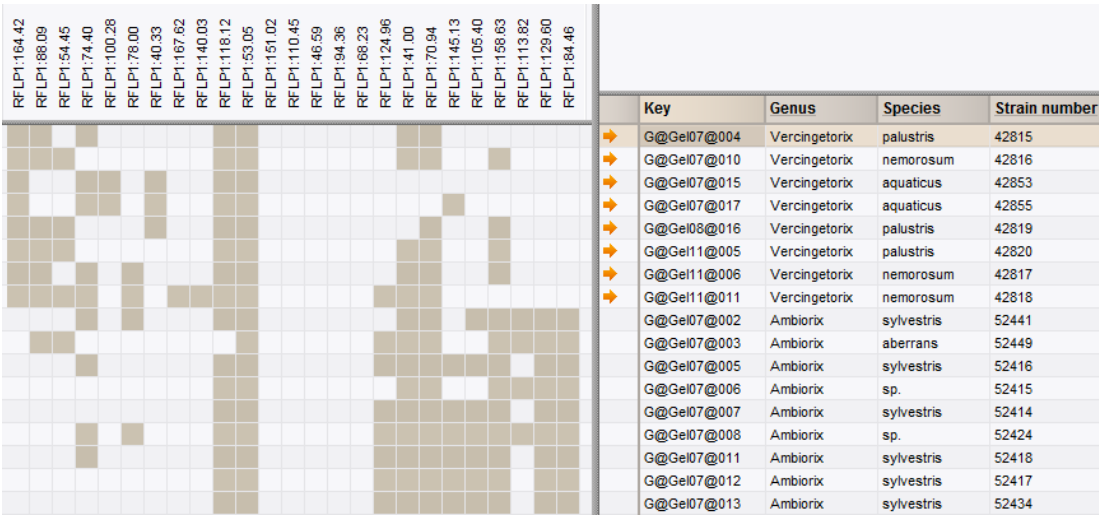


Figure 4.3.16: Discriminative bands for selected entries.

The entries are now sorted by increasing intensity of the selected band class.

Furthermore, it is possible to perform a transversal (or two-way) clustering of a band matching table. See [11.2.5](#) for a detailed description of the transversal clustering of composite data sets.

Part 5

Spectrum types

Chapter 5.1

Setting up spectrum type experiments

5.1.1 Creating a new spectrum type

To create a new spectrum type, highlight the *Experiment types* panel in the *Main* window and select **Edit** > **Create new object...** (🟢). The *Create a new experiment type* dialog box pops up (see Figure 5.1.1).

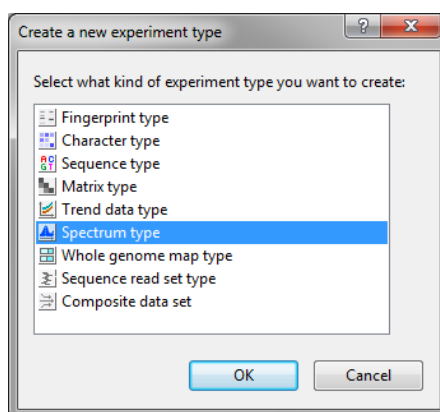


Figure 5.1.1: The *Create a new experiment type* dialog box.

This dialog box lists all experiment types, available in BioNumerics (see 2.1.2 for more information).

Click on **Spectrum type** and press <OK>. The first step of the *New spectrum type* wizard (see Figure 5.1.2) will be displayed.

Enter an experiment name and optionally change the units of the X and Y-axis. The default units are mass over charge (“m/z”) and “Intensity”, respectively.

In the second step of the *New spectrum type* wizard (Figure 5.1.3), a choice can be made which preprocessing template will be used as default preprocessing during import. The template can be changed during import, but the template selected here will be selected by default. Three predefined templates are included, but the user can define his own or edit the existing templates (see 5.2). A short description of each template can be found in the left panel. Select a template and press <Finish>.

5.1.2 Settings of a spectrum type experiment

All settings defined during the creation of the experiment type can be modified later by opening its *Spectrum type* window (see Figure 5.1.4). To open this, click on the spectrum type experiment in the *Experiment types*

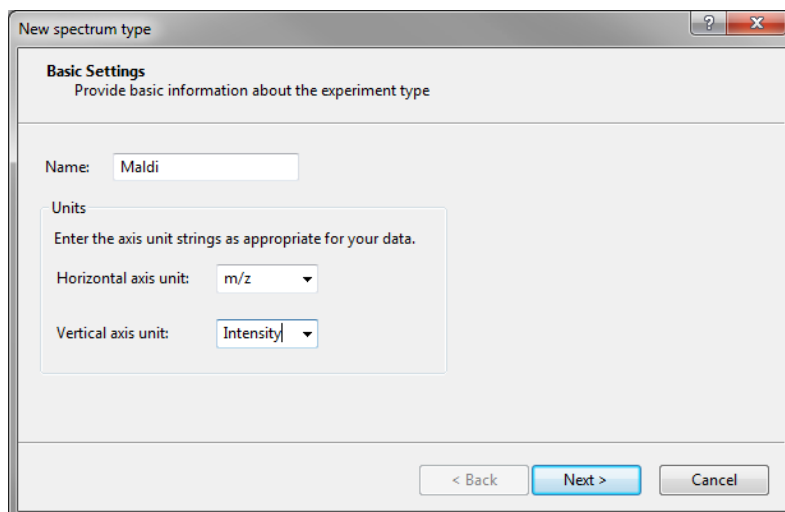


Figure 5.1.2: The *Basic Settings* wizard page.

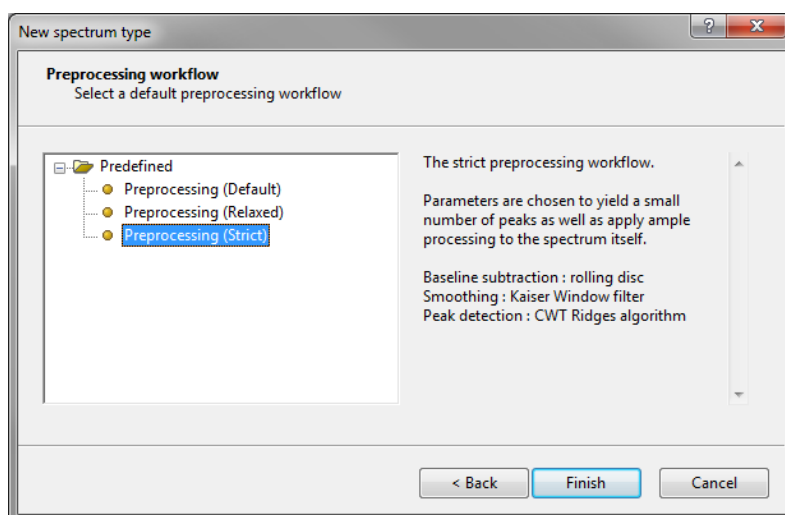


Figure 5.1.3: The *Preprocessing workflow* wizard page.

panel from the *Main* window and select **Edit > Open highlighted object...** (📁, **Enter**). Alternatively, simply double-click the spectrum type experiment.

The *Spectrum type* window consists of three dockable panels:

- The *Comparison settings* panel summarizes the comparison settings used.
- The *Crosslinks* panel lists all cross-links of the spectrum type to other database objects (see 3.2.15)
- The *Attachments* panel provides a list of all spectrum type attachments (see 3.2.13).

Selecting **Settings > General settings...** (⚙️) will open the *Spectrum experiment type settings* dialog box (see Figure 5.1.5).

In addition to the settings defined at creation (see 5.1.1), some additional parameters can be adjusted in the *General tab* of the *Spectrum experiment type settings* dialog box. The **Peak storage mode** determines how the peaks are stored in the database. By default, they are stored as peak vectors, meaning as groups of peaks. This allows for a more efficient use of storage space, but does not allow querying of individual peaks.

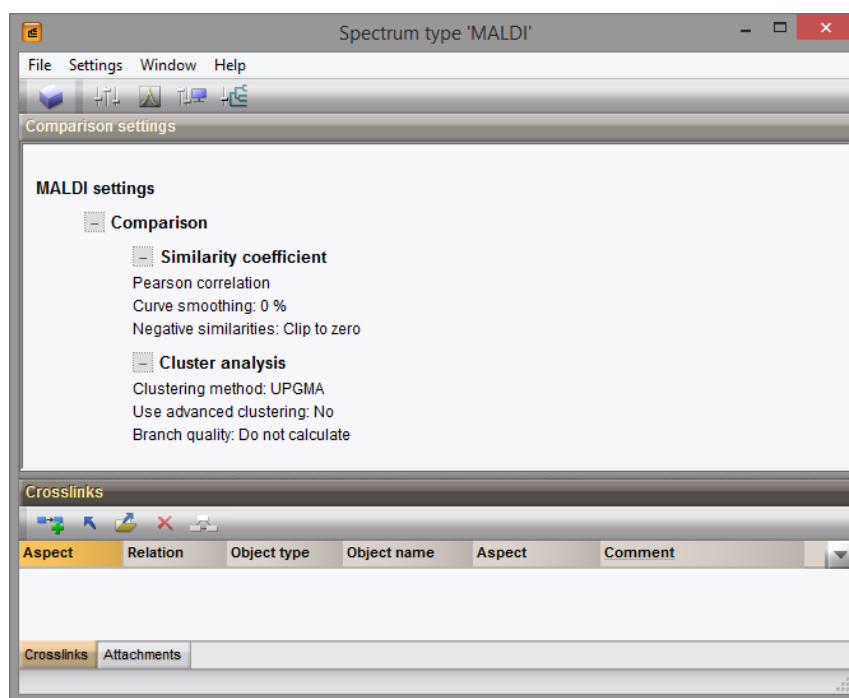


Figure 5.1.4: The *Spectrum* type window, from which all relevant settings for the spectrum type can be accessed.

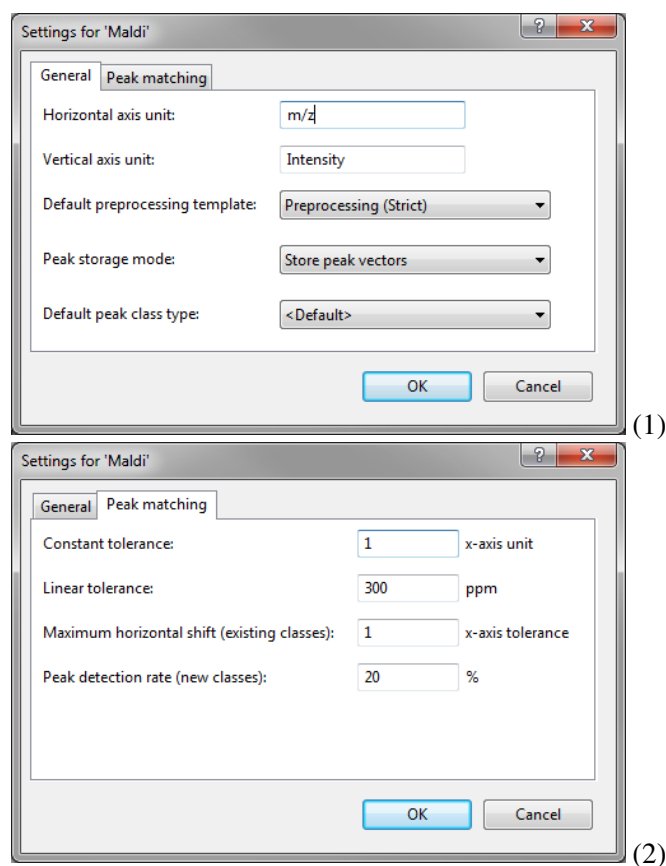


Figure 5.1.5: The *Spectrum* experiment type settings dialog box, *General* tab (1) and *Peak matching* tab (2) of a spectrum type.

Storing the peaks as individual peaks, does allow querying but requires more storage space. The **Default peak class type** can also be adjusted here.

In the *Peak matching tab*, the default parameters of the peak matching can be specified (see 5.5).

Peak class types can be created in the *Comparison* window and can be managed both from the *Comparison* window and the *Spectrum experiment type settings* dialog box. For more details, see 5.5.3.

Selecting **Settings > Peak Class Type Settings...** (🖥️) will open the *Display settings* dialog box where the user can change the default display options for the peak classes in comparisons. More details on this dialog can be found in (see 5.5).

Settings > Comparison settings... (⚙️) allows the user to define the default comparison settings of the experiment. More information on the options available in the *Comparison settings* wizard can be consulted in 5.4.

5.1.3 Importing and preprocessing of spectra

5.1.3.1 Introduction

Spectra can be imported from various formats, mass-intensity lists, peak lists, Main Spectra data (Bruker), mzML and mzXML. The software included with most mass spectrometers allows the export to one of these formats. Also, numerous conversion tools can be found online, especially for the mzML and mzXML formats. We refer you to the documentation of the manufacturer of your machine for instructions regarding the export in a correct format.

5.1.3.2 Importing spectrum experiments

With the **Import Spectrum data** option, listed under the topic **Spectrum type data** in the *Import* dialog box (see Figure 5.1.6), raw spectra or peak lists and optionally entry information can be imported from text files and linked to new or existing entries in a BioNumerics database.

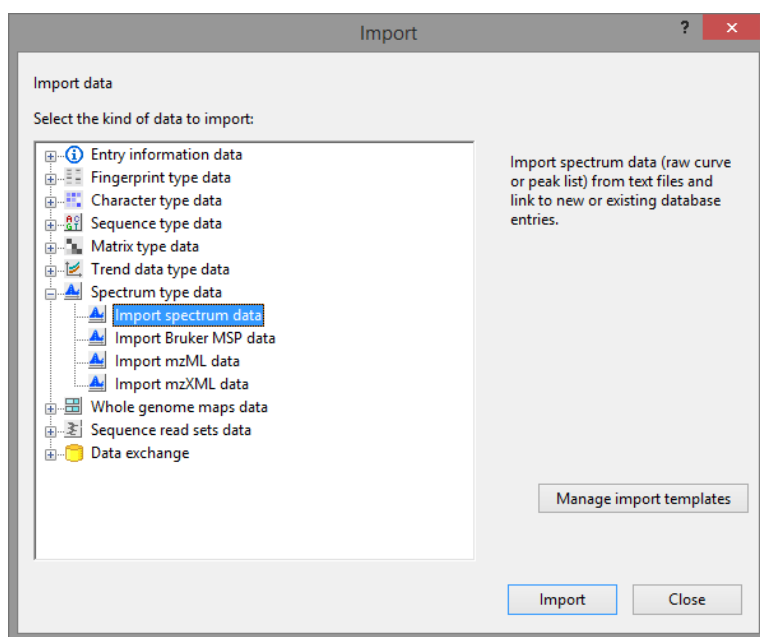


Figure 5.1.6: The **Import spectrum data** option in the *Import* dialog box.

Each file should contain two columns, where the first column corresponds to the m/z values and the second to the *intensity* values.

With the **Import Bruker MSP data** option, listed under the topic **Spectrum type data** in the **Import** dialog box (see Figure 5.1.7), spectra in Bruker MSP format can be imported from text files in the database and linked to new or existing database entries.

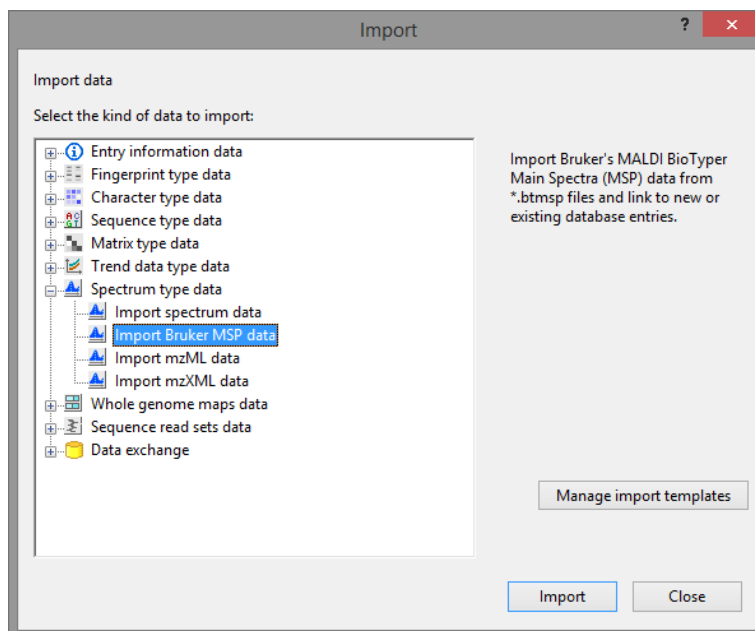


Figure 5.1.7: The **Import Bruker MSP data** option in the **Import** dialog box.

The extension of Bruker MSP files is `.btmsp`, it is a proprietary format of Bruker and can only be obtained with their software.

With the **Import mzML data** option, listed under the topic **Spectrum type data** in the **Import** dialog box (see Figure 5.1.8), processed spectra and optionally entry information can be imported from mzML files in the database and linked to new or existing database entries.

mzML is a standardized format for exchange of spectrum type data. The specifications can be found here: <http://www.psicodev.info/mzml>. For the majority of mass spectrometers, free conversion tools to mzML format can be found online.

Selecting **Import spectrum data**, **Import Bruker MSP data**, **Import mzML data** or **Import mzXML data** under **Spectrum type data** in the **Import** dialog box and pressing **<Import>** starts the import wizard.



Before spectrum data can be imported in the database using these import options, a spectrum type experiment must be defined in the database (see 5.1.1).

If the **Import spectrum data** option was selected in the **Import** dialog box, the first step of the wizard prompts for the spectrum data files (see Figure 5.1.10). Pressing the **<Browse>** button allows you to select the files that you want to import, located on your computer, external drive or on a network location. The files should be in txt format. More files can be added by clicking **<Browse>** again. After all files have been selected, press **<Next>**.



If you wish to parse information from the filename, only select spectra files where the filename is in the same format so all files can be parsed with the same rules.

If the **Import Bruker MSP data** option was selected in the **Import** dialog box, the first step of the wizard prompts for the `.btmsp` files. Pressing the **<Browse>** button allows you to select the files that you want to import, located on your computer, external drive or on a network location. More files can be added by

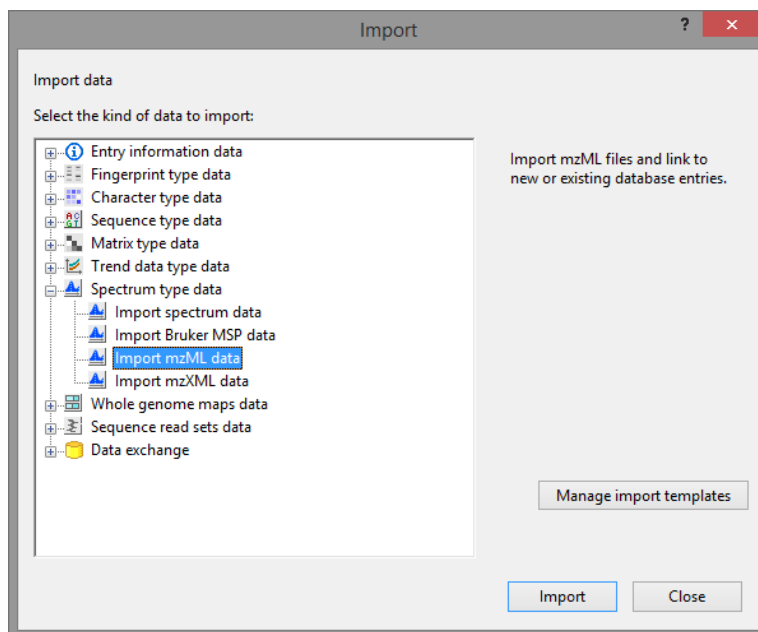


Figure 5.1.8: The *Import mzML data* option in the *Import* dialog box.

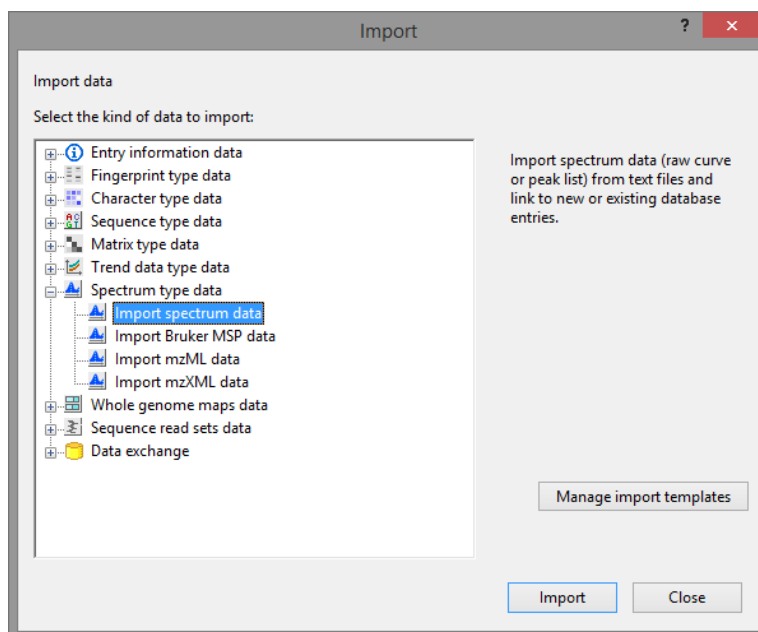
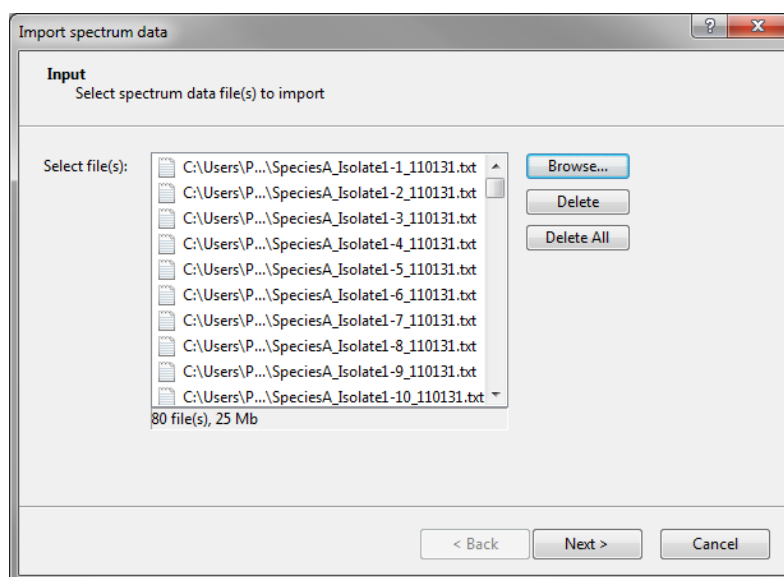
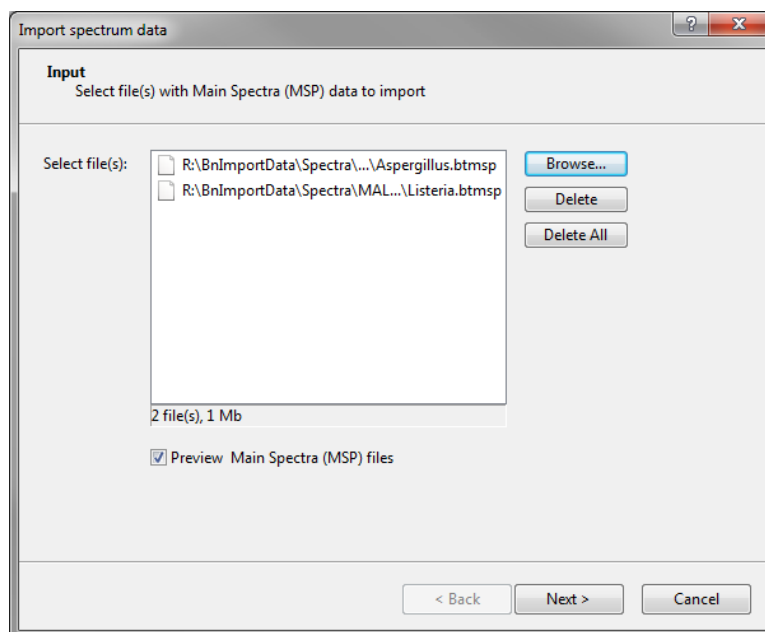


Figure 5.1.9: Importing spectra

clicking **<Browse>** again. After all files have been selected, press **<Next>**.

If **Preview Main Spectra (MSP) files** was selected in *Input* wizard page, the name of the spectra present in the MSP file is shown together with the file it belongs to. By default all spectra in the file are selected for import, but the user can use the check boxes to the left in the *Preview* wizard page to exclude spectra from the import.

If the **Import mzML data** or **Import mzXML data** option was selected in the *Import* dialog box, the first step of the wizard prompts for the mzML or mzXML files (see Figure 5.1.13). Pressing the **<Browse>** button allows you to select the files that you want to import, located on your computer, external drive or on a network location. More files can be added by clicking **<Browse>** again. After all files have been selected,

Figure 5.1.10: The *Input* wizard page.Figure 5.1.11: The *Input* wizard page.

press **<Next>**.

After all files have been selected, press **<Next>** to go to the next step. If the import routine is unable to open the selected file, an error is generated.

The way the spectrum data should be imported in the database can be specified with an import template. The *Import templates* panel lists all import templates that have been created and stored in the database.

The **Default** template will import the sequences in the database and link the sequences to new entries in the database (if the option **Create *x* entries** is checked in the final step). The keys are automatically created by the import routine.

Pressing the **<Create new>** button brings up a new dialog box, allowing you to define a new import template (see Figure 6.1.37).

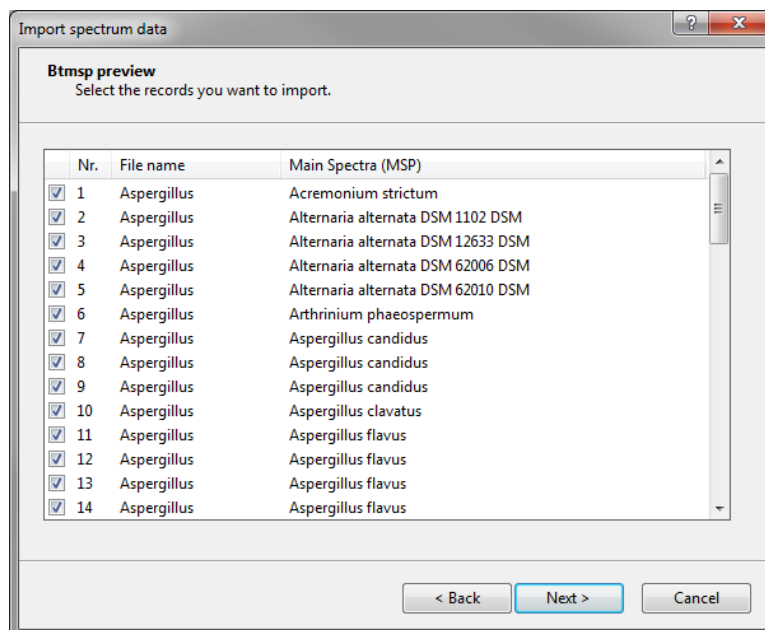


Figure 5.1.12: Preview of spectra included in Bruker MSP file

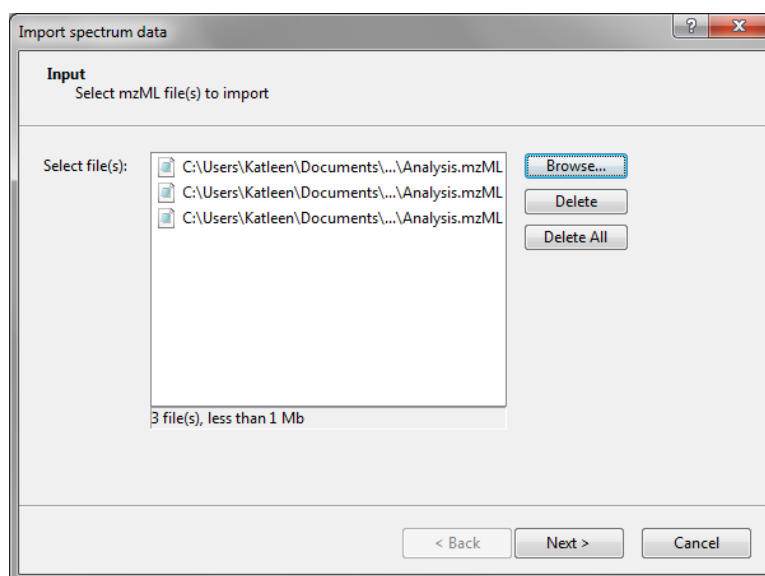


Figure 5.1.13: The *Input* wizard page.

When importing a mass-intensity list or a peak table, the only column that will be available is the filename. For MSP files and for mzML, the content of several information fields present in the file is available. Each information field in the selected MSP or mzML files corresponds to a row in the grid. The text **File field** is specified in the **Source type** column and the information field names are displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields. Initially the rows are not linked to any information in the database (the **Destination type** and **Destination** for all rows is set to **<None>**).

Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).

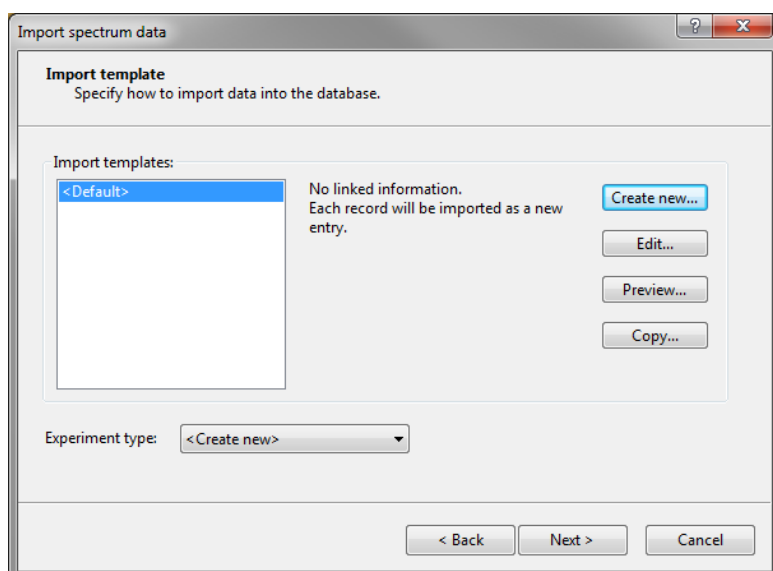


Figure 5.1.14: The *Import template* wizard page.

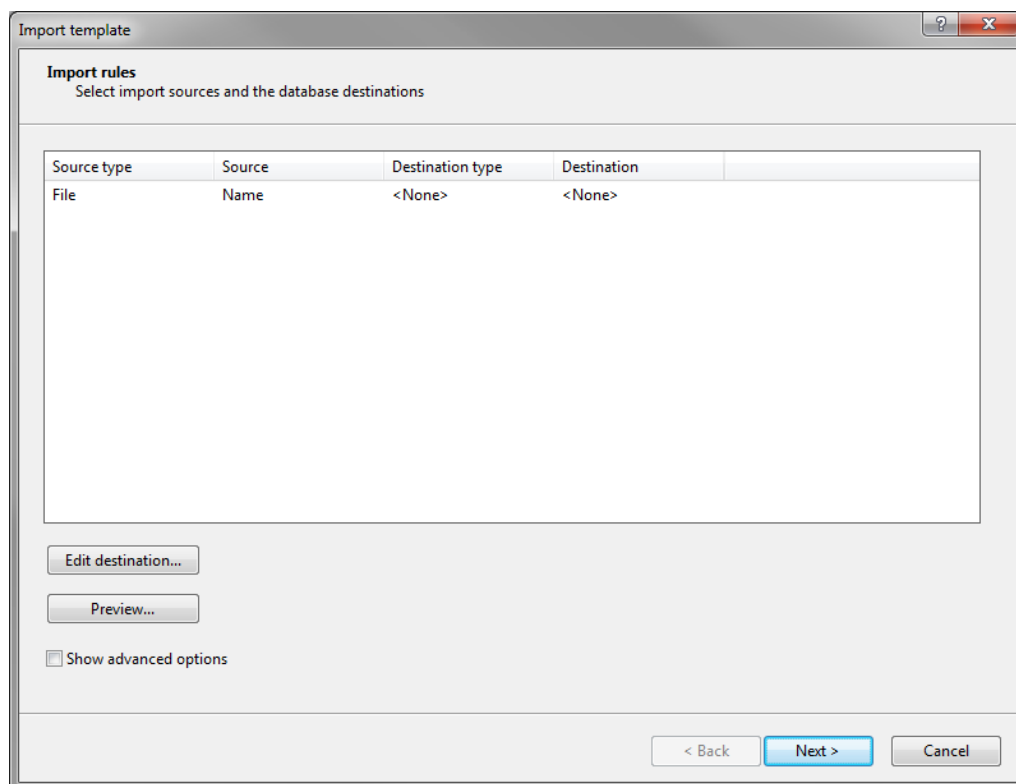


Figure 5.1.15: The *Import rules* dialog box.



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

When only one row is selected in the grid, the information of this row can be linked to (see Figure 5.1.16):

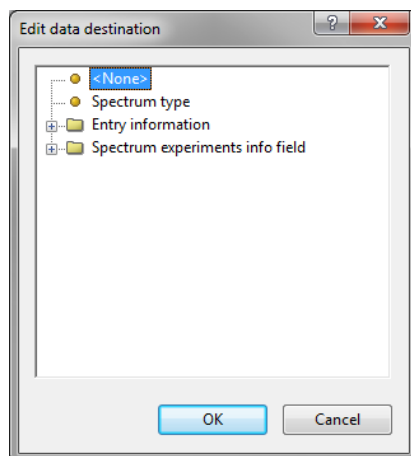


Figure 5.1.16: Edit data destination for the filename of a spectrum data file.

- The default information field **Key**.
- A new or existing spectrum type experiment **Spectrum data**.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing spectrum type information field (select **<Create new>** or select an existing field under the topic **Spectrum experiments info field**, respectively).

If a row is linked to a new entry information field, a new spectrum type experiment or a new spectrum type information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to spectrum type information fields, the text **Spectrum set info field**, followed by the name of the spectrum type experiment, is displayed in the **Destination type** column; the name of the spectrum information field is listed in the **Destination** column.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5, they will be needed in most cases when importing spectrum data from mass intensity lists or peak lists, as the file name will contain several pieces of useful information. A new rule needs to be added for each piece of information parsed from the name (**<Add rule>**). In each rule, different parsing strings can be defined to get different information out of the filename. When the **Parse component** option is checked, the mapped information is parsed using the parsing strategy as specified by the **Data parsing string**. This string should at least contain the **[DATA]** component, in which case the complete information is retained. The asterisk (*) can be used as a wildcard to omit characters from the information. In the parsing string ***-[DATA]-*** for example, any characters before the underscore and after the hyphen will be ignored.

When working in a database with levels, the parsed information can be sent to different levels and in such can be used to define the links between levels. For example, when working with technical replicates, the part of the filename that is unique for each individual spectra, can be linked to the key at the lowest level. The part of the filename that identifies the batch to which the replicate belongs, can be linked to the key

at a higher level, the batch level. All spectra with the same batch key will then be linked to this key, thus creating a link between both levels. A typical example of how an import template for spectra in a database with levels looks like, is shown in Figure 5.1.17.

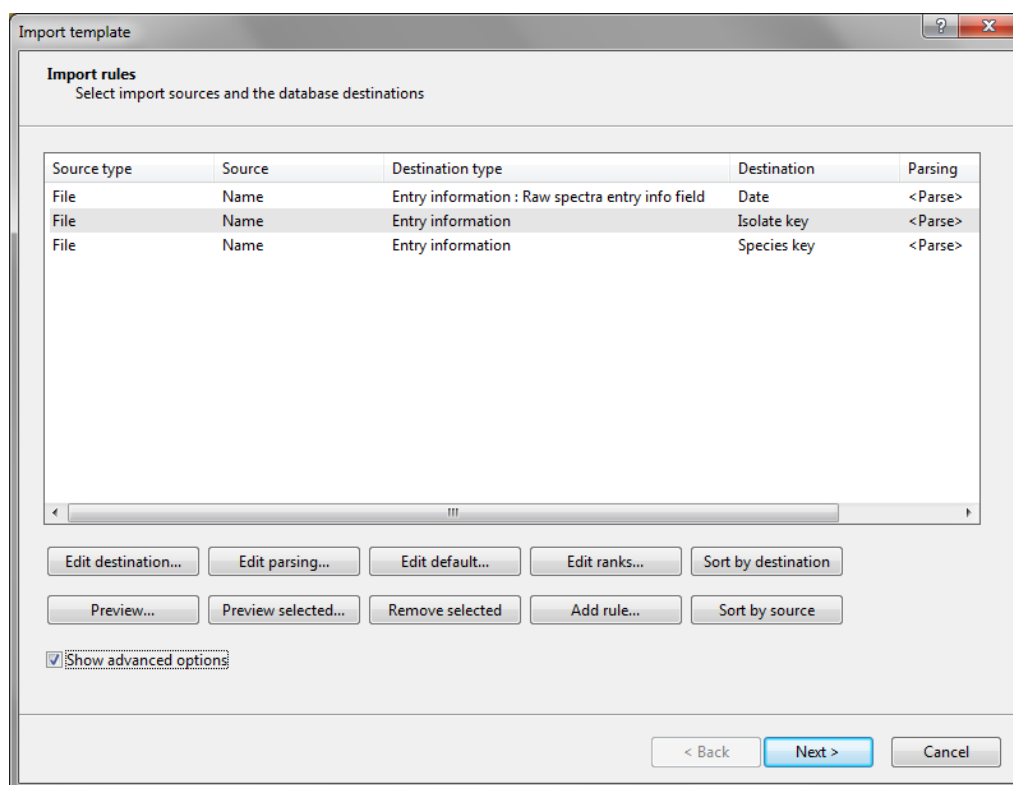


Figure 5.1.17: Example of typical import template for spectrum type data

It is strongly advised that you use the preview during each import to ensure the right information ends up in the right destination.

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

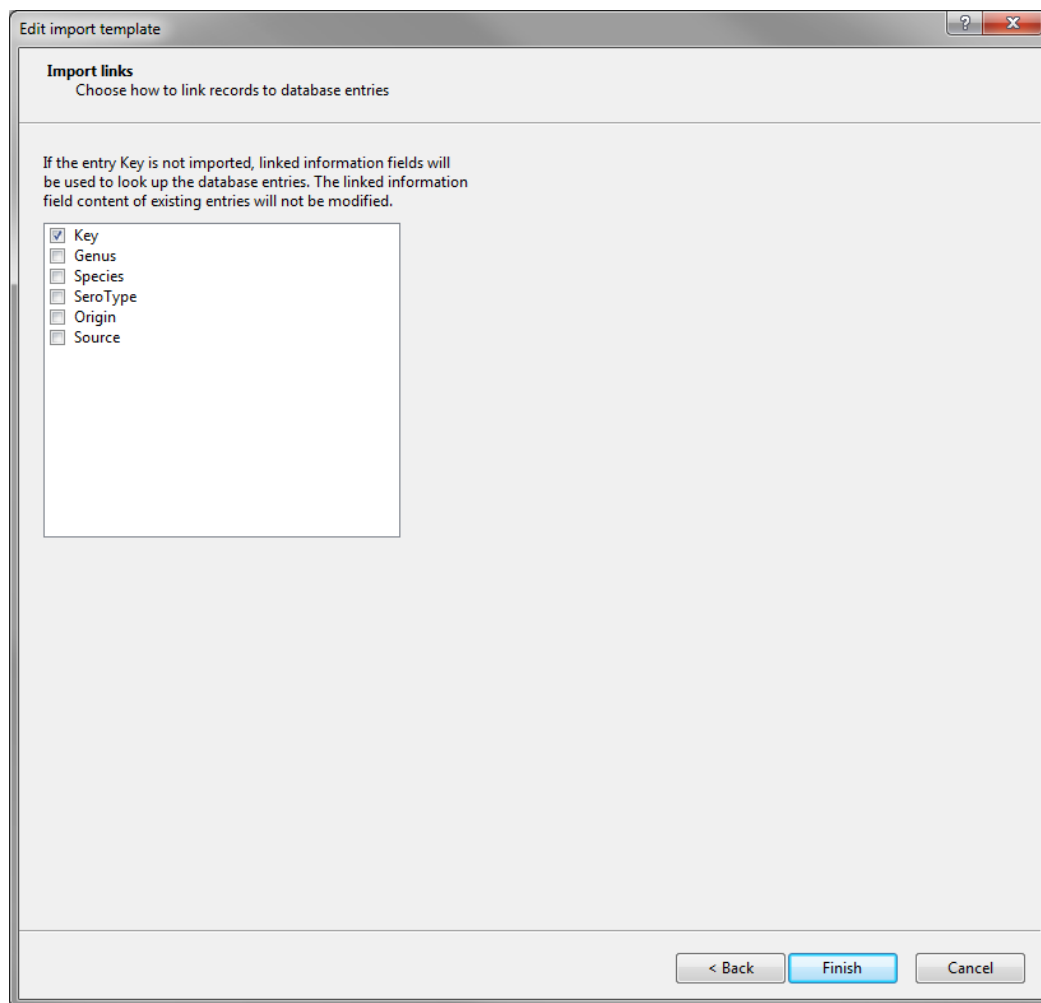


Figure 5.1.18: Specify the entry link field.

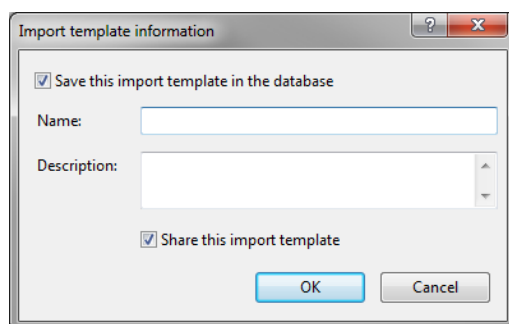


Figure 5.1.19: The *Import template information* dialog box.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database. With **<Preview>**, the user can review the results of the existing template on the currently imported spectra.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

Pressing **<Next>** opens the last step of the wizard, prompting for some final settings.

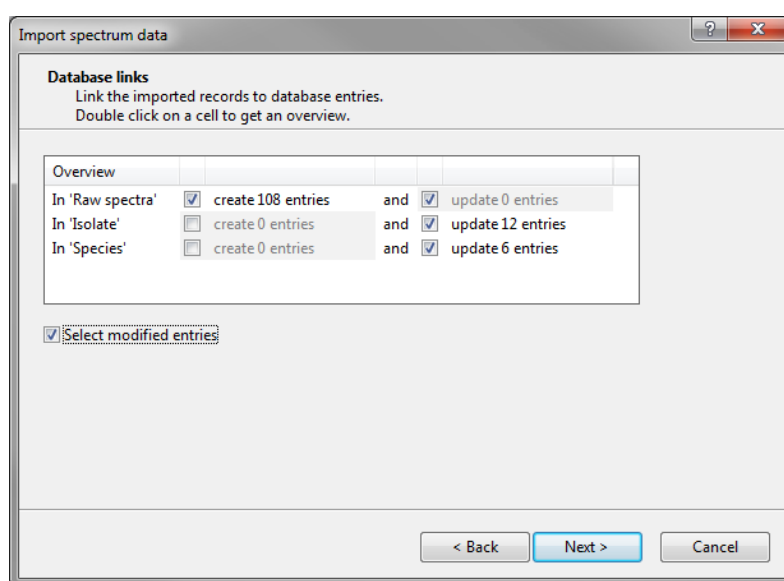


Figure 5.1.20: The *Database links* wizard page.

For each level, the *Database links* wizard page shows how many entries will be created and how many entries will be updated.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database. This can be defined for each level individually.
- Check the option **Update *x* entries** if you want the software to be able to update the entry and character information for existing entries. This can be defined for each level individually.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Clicking **<Next>** will open the final step of the wizard, *Preprocessing* wizard page (see Figure 5.1.21).

When importing raw spectrum data, options concerning the preprocessing can be defined in the *Preprocessing* wizard page. The template defined in the chosen spectrum experiment, is highlighted by default, but

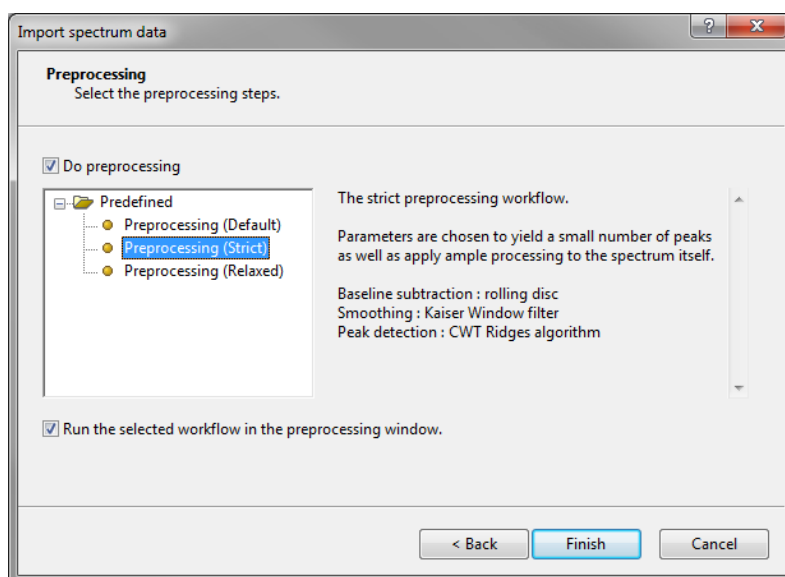


Figure 5.1.21: Select the preprocessing template

may be changed. Three predefined templates are present upon installation, but the user may define his own custom templates (see 5.2).

The preprocessing workflow may be performed in a dedicated window (the *Spectrum Preprocessing* window, see 5.1.3.3) or in the background. When importing a large number of spectra (more than 500), it is advised to run the preprocessing in the background. Clicking <Finish> will start the import, the preprocessing starts either in the *Spectrum Preprocessing* window or in the background. When a large number of spectra is imported, it may take several minutes for the import to complete.

With the **Import mzXML data** option, listed under the topic *Spectrum type data* in the *Import* dialog box (see Figure 5.1.22), processed spectra and optionally entry information can be imported from mzXML files in the database and linked to new or existing database entries.

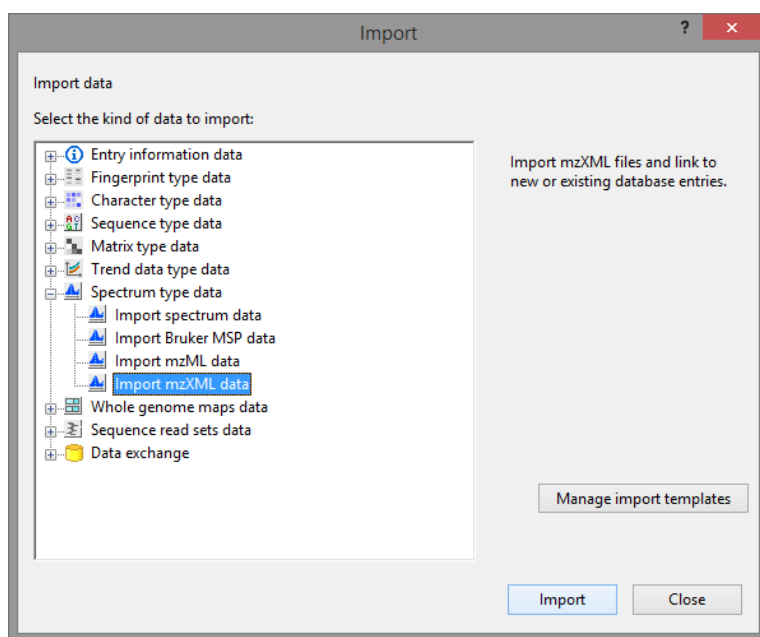


Figure 5.1.22: The **Import mzXML data** option in the *Import* dialog box.

mzXML is a standardized format for exchange of spectrum type data. The specifications can be found here: <http://tools.proteomecenter.org/wiki/index.php?title=Formats:mzXML>. For the majority of mass spectrometers, free conversion tools to mzXML format can be found online.

5.1.3.3 The preprocessing window

The preprocessing of raw spectra is necessary to remove baseline, noise,... As mentioned in 5.1.3.2, the preprocessing can be performed upon import either in the background or in a dedicated window (the *Spectrum Preprocessing* window). Afterwards, the preprocessing may be reviewed and/or altered at any moment either by opening the selected spectra in the *Spectrum Preprocessing* window (**Analysis** > **Spectrum types** > **Open preprocessing window...**) or running the default preprocessing template in the background (**Analysis** > **Spectrum types** > **Preprocess...**).

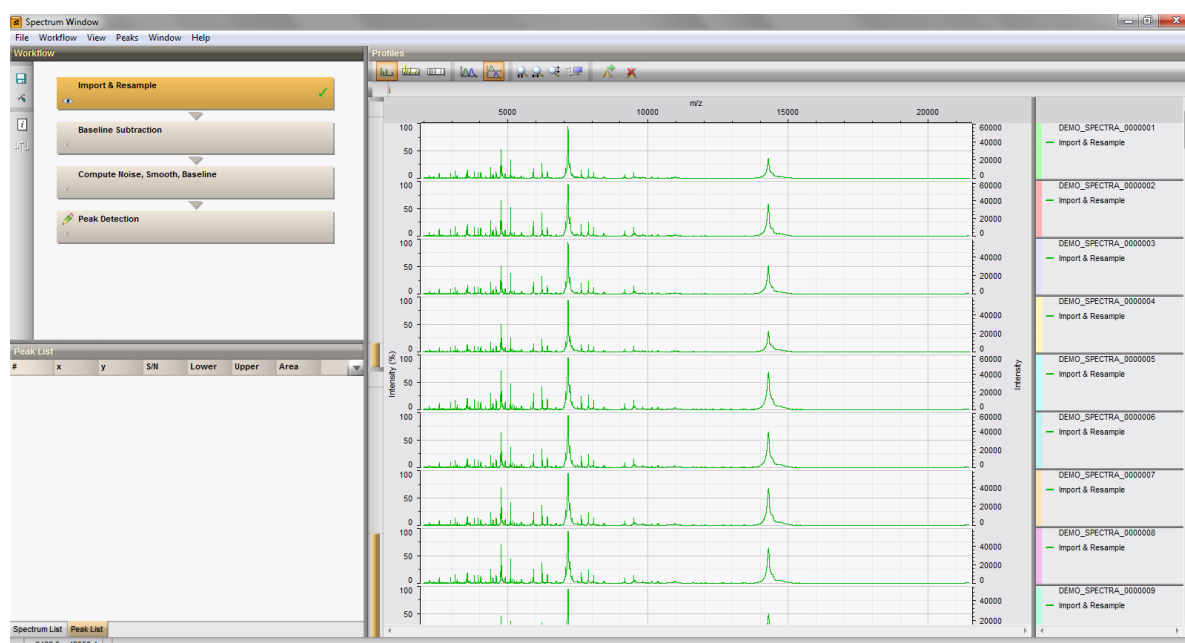


Figure 5.1.23: The *Spectrum Preprocessing* window containing raw spectra.

The *Spectrum Preprocessing* window (see Figure 5.1.23) consists of several panels, the *Workflow* panel, the *Spectrum List* panel, the *Peak List* panel and the *Profiles* panel. Each panel will be discussed in detail below.

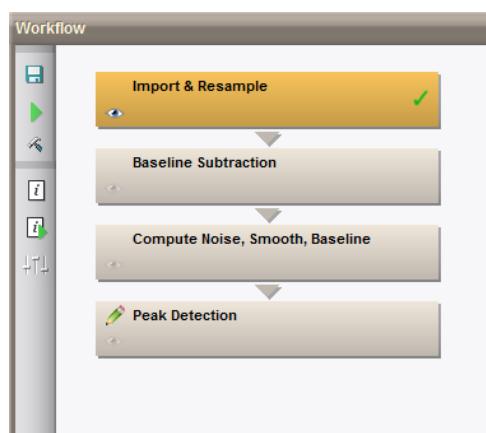





Figure 5.1.24: The *Workflow* panel of the *Spectrum Preprocessing* window with the default preprocessing template.

The *Workflow* panel (see Figure 5.1.24) contains the pipeline of the loaded preprocessing template, another template can be loaded through **File > Load workflow from template...** or **File > Import workflow from file...**. The loaded template can be altered and saved as a new template (**File > Save workflow as template...**) or exported to file (**File > Export workflow to file...**). Detailed instructions on how to make your own preprocessing templates are provided in 5.2. The default template contains four actions: Import & Resample, Baseline Subtraction, Compute Noise, Smooth, Baseline and Peak Detection. Clicking on an action will run the action and all previous actions that have not been run. Actions that have been run are marked with  on the right side of the action bar. By default the results of the latest action are visualized in the *Profiles* panel. Other actions can be visualized by clicking  in the bottom left corner of the action bar. A pencil sign () in the upper left corner means the results of this action can be manually edited in the *Profiles* panel. The other functions of the *Workflow* panel are explained in detail in 5.2.

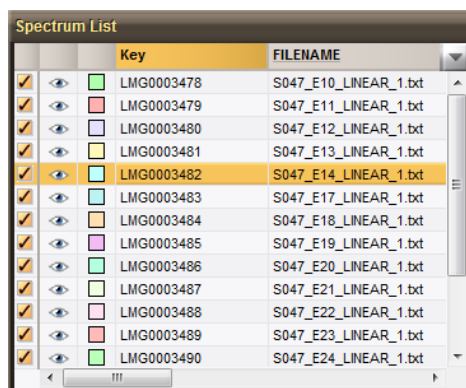





Figure 5.1.25: The *Spectrum List* panel of the *Spectrum Preprocessing* window.

All spectra loaded in the *Spectrum Preprocessing* window are listed in the *Spectrum List* panel (see Figure 5.1.25). By default all spectra are visualized, but clicking on the  will hide a visible spectrum or visualize a hidden spectrum. This panel behaves the same way as *Database entries* panel in the *Main* window, all data in the information fields can be edited.

#	x	y	Lower	Upper	S/N
0	2083.16	47.32	2079.33	2086.03	13.09
1	2088.43	53.75	2086.03	2091.30	15.94
2	2117.27	11.18	2113.89	2119.20	8.21
3	2185.34	35.79	2177.02	2187.30	10.28
4	2228.68	38.33	2227.20	2232.64	15.98
5	2257.98	181.00	2246.54	2263.46	58.02
6	2294.00	24.40	2290.98	2298.01	8.28
7	2320.69	33.69	2317.15	2324.23	11.11
8	2525.05	71.99	2522.42	2531.91	23.65
9	2563.67	13.60	2561.02	2566.86	7.79
10	2620.81	8.87	2617.59	2624.03	5.08
11	2714.53	15.73	2710.71	2718.36	7.23
12	3016.56	117.67	3006.77	3024.63	48.65
13	3064.00	266.32	3061.09	3070.39	73.98
14	3093.11	139.32	3089.02	3101.28	40.31

Figure 5.1.26: The *Peak List* panel of the *Spectrum Preprocessing* window.

In the *Peak List* panel (see Figure 5.1.26), a list of the peaks identified in the selected spectra is given. For each peak, the x and y values, the upper and lower bound on the x-axis and the single to noise ratio are provided. Selecting a peak in the *Peak List* panel will cause this peak to be selected in the *Profiles* panel and selecting a peak in the *Profiles* panel, will select this peak in the *Peak List* panel.

The *Profiles* panel (see Figure 5.1.27) is the most important panel of the *Spectrum Preprocessing* window. Here, all spectra can be viewed, either as spectra (**View > Spectra** ) , as bands (**View > Band representation** ) or an overlay of the spectra on the band representation (**View > Band representation and**

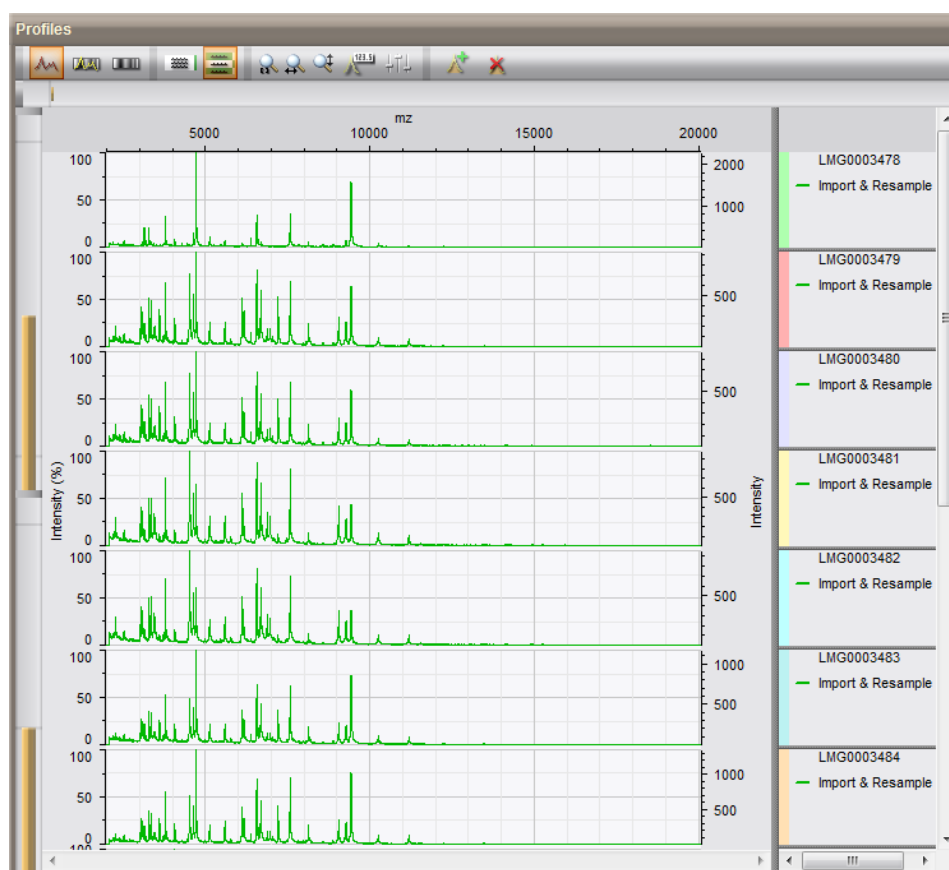









Figure 5.1.27: The *Profiles* panel of the *Spectrum Preprocessing* window.

spectra ). By default, the spectra are visualized per entry (**View > Group by entry** ), but they can also be viewed by action (**View > Group by action** ). This last view is very convenient when validating the preprocessing parameters on a small number of spectra. When several spectra are visualized in one profile, they are listed to the right of the profile. Selecting a spectrum from this list will visualize this spectra as a thicker line to the foreground of the other spectra in this profile. It is also possible to zoom in and out using the zoom sliders. For horizontal zooming, use the zoom slider above the spectra or specify a specific x-range with **View > Zoom to specified x-range** ). For vertical zooming, there are two zoom sliders, the top one will increase the intensity of the spectra within their windows, the bottom one will increase the vertical window size. To view the entire spectra in the window, select **View > Zoom to fit horizontal**  and **View > Zoom to fit vertical** .

More options for visualization regarding labels, grid lines and axis can be adjusted by selecting **View > Settings** ). This will open the *Settings* dialog box (see Figure 5.1.28).

In the *Plot settings* tab, following settings are available:

- **Line width:** the thickness of the line (in screen pixels) used to plot the spectra.
- **Plot mode:** the type of line used (either Lines, Points, Lines & Points or Hidden) to plot the spectra.
- **Separators:** the separator lines between plots of individual spectra (either Full range, Plot only or Hidden).
- **Vertical axes:** if and where a Y-axis is drawn (on both sides, on the left-hand side, on the right-hand side or not shown).
- **Grid lines:** whether major and minor grid lines are shown (either Major and Minor, only Major or Hidden).

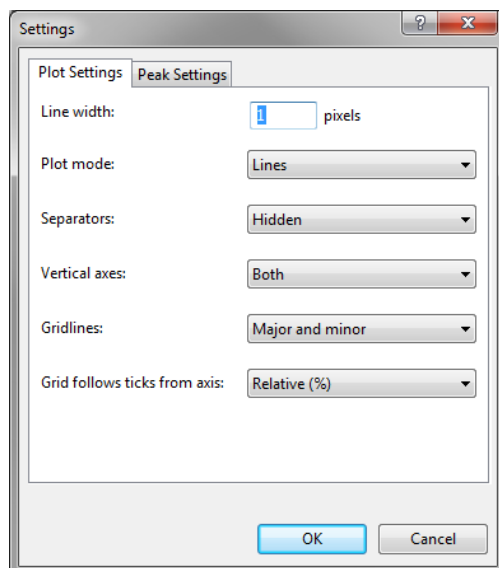


Figure 5.1.28: The *Settings* dialog box.

- **Grid follows ticks from axis:** select *Relative %* to follow the ticks on the *Y*-axis or *Absolute (Intensity)* to use absolute units.

The *Peak settings tab* groups the display settings of individual peaks:

- **Peak marker mode:** determines how the peaks are indicated (either *Auto*, *Circles*, *Crosses*, *Lines* or *None*) on the spectra.
- **Peak marker size:** a relative measure for the size of the peak marker symbol used.
- **Show selected peak boundaries:** whether or not the boundaries of the selected peak are displayed.
- **Enable peak labels:** whether or not to display peak labels.
- **Show peak labels:** if peak labels are enabled, when to display these. The peak labels can be displayed for all peaks (*Always*) or only for the *Selected Spectra* or the *Selected Peaks*.
- **Peak labels orientation:** text orientation of the peak label, either *Vertical* or *Horizontal*.
- **Peak label font size:** the font size used for the peaks labels; this allows you to determine the relative size of the peak labels.

With **File > Export...** the visualized profiles in the *Profiles* panel can be exported.

This dialog box allows you to export graphical information to a file or to the Windows clipboard in one of several available formats. In case a file is exported, a file **Name** should always be entered or browsed for via the **<Browse>** button. Exported files will open in their default editor. Information on the Windows clipboard can be pasted into other applications. Following export options are available:

- **PNG image (*.png):** exports to a Portable Network Graphics (PNG) file. PNG is a raster graphics file format that supports lossless data compression. A **Name** and **Width** (in pixels) should be specified; the height will be determined automatically.
- **JPG image (*.jpg):** exports to a Joint Photographic Experts Group (JPEG) file. JPEG or JPG is a raster graphics file format that uses a lossy data compression. A **Name** and **Width** (in pixels) should be specified, as well as a **Quality** parameter. With the latter, a tradeoff can be obtained between storage size and image quality.

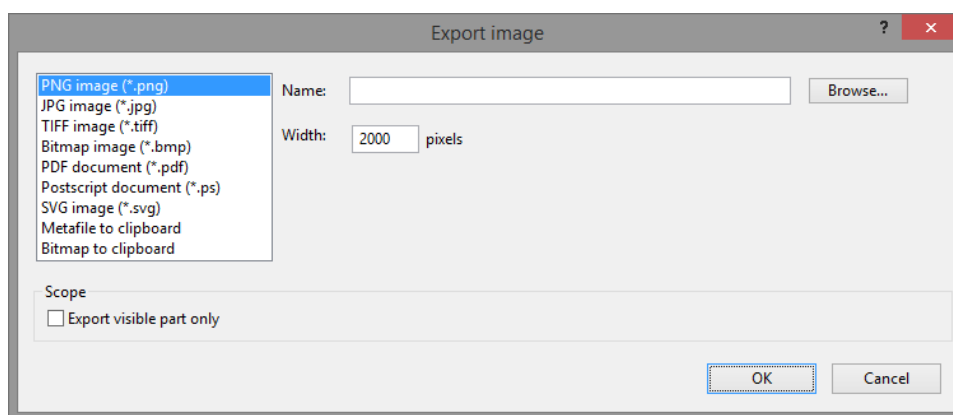


Figure 5.1.29: The *Export image* dialog box.

- **TIFF image (*.tiff)**: exports to a Tagged Image File Format (TIFF) file. TIFF is a raster graphics file format with optional lossless data compression. A **Name** and **Width** (in pixels) should be specified.
- **Bitmap image (*.bmp)**: exports to a BMP bitmap image or device independent bitmap (DIB) file. BMP is a raster graphics image file format used to store bitmap digital images, independently of the display device. A **Name** and **Width** (in pixels) should be specified.
- **PDF document (*.pdf)**: exports to a Portable Document Format (PDF) file. PDF is a file format used to present documents in a manner independent of application software, hardware, and operating systems. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **Postscript document (*.ps)**: exports to a PostScript (PS) file. PostScript is a computer language for creating vector graphics. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **SVG image (*.svg)**: exports to a Scalable Vector Graphics (SVG) file. SVG is an XML-based vector image format for two-dimensional graphics. A **Name** should be specified.
- **Metafile to clipboard**: copies the graphics as Windows enhanced metafile to the clipboard. Enhanced metafile is the standard clipboard exchange format between native Windows applications.
- **Bitmap to clipboard**: copies the graphics as a bitmap to the Windows clipboard. The **Width** (in pixels) should be specified.

For actions marked with a pencil (✎) in the workflow panel, it is possible to do some manual editing in the profiles panel. The user can add peaks at the cursor position with **Peaks > Create peak at cursor** (✎) and delete selected peaks with **Peaks > Remove selected peaks** (✖). To select multiple peaks, hold down the **Shift**-key while dragging the cursor over the peaks you wish to select. Peaks can only be selected on one profile at a time. The user must use the manual editing options with caution, rerunning the preprocessing template will cause any manual editing to be removed. A better approach would be to adjust the preprocessing template to match any manual editing the user considers necessary (see 5.2) and to only use manual editing in exceptional cases.

Chapter 5.2


Making spectrum preprocessing templates

5.2.1 Introduction


When the software is installed, three default preprocessing templates are included. We strongly advise to adapt these templates to get optimal processing results for your specific type of data. It is best to do this on a limited amount of well defined samples, for instance on a dilution series of filter-cloned isolates. Make sure to add both technical as biological replicates to ensure reproducibility with the chosen parameters. Once the preprocessing parameters are optimized, this can be saved as a new template. It is also advised to check the performance of the preprocessing template if changes are made to the experimental setup (for instance, a different matrix), or if components of the mass spectrometer are replaced.

Custom preprocessing template can be created in the *Spectrum Preprocessing* window by modifying an existing template and saving this under a different name. First, the general workflow of the template is defined (5.2.2), then the parameters in each action of the pipeline can be adjusted (5.2.3).

5.2.2 Managing the workflow

The workflow of the preprocessing analysis is visualized in the *Workflow* panel. The calculations performed during preprocessing can be divided in several steps, called actions. Each action is visualized by one rectangle in the pipeline. The results of each action on the spectra is shown in the *Profiles* panel. By default, only the last action is visualized, but clicking on the eye button () in the lower left corner of an action, will toggle the visibility of this action.

In the predefined templates there are four actions: 'Import & Resample', 'Baseline Subtraction', 'Compute Noise, Smooth, Baseline' and 'Peak Detection'. The user can add his or her own actions by selecting **Workflow > Add action...**, this will open the *Properties* dialog box.

The *Properties* dialog box allows the user to define his own actions in a pipeline. A name and description can be provided, only the name will be visible in the pipeline so make sure to provide a meaningful name. Spectra stored in the new action can be made editable by checking 'Action data is editable'. A pencil sign () will appear in the resulting action to mark it as editable.

The properties of an existing action can also be reviewed and edited in the *Properties* dialog box by selecting **Workflow > Properties...**. An action can be removed by selecting **Workflow > Remove** when the action is selected. An action can also be moved up or down in the pipeline with **Workflow > Move up** and **Workflow > Move down** respectively. With **Workflow > Properties...**, the name, description and editability of an action can be changed. To define or change the operators and parameters performed by an action, select

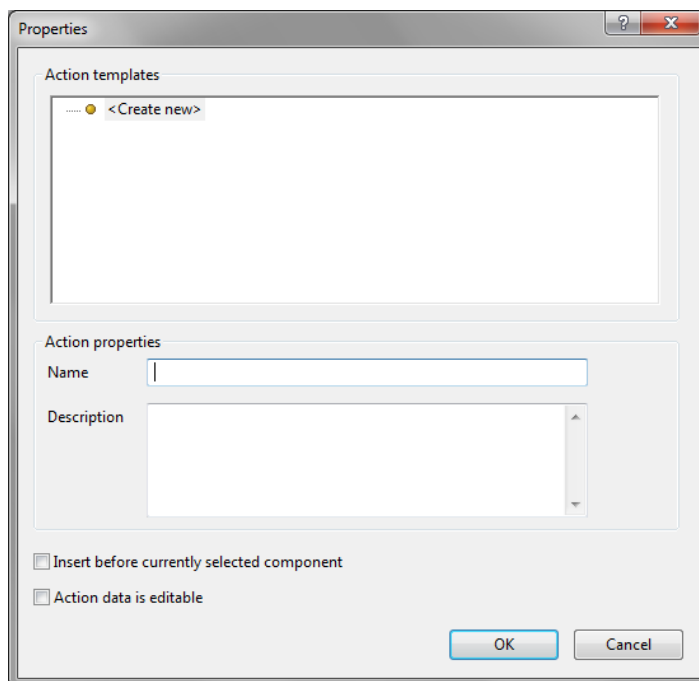


Figure 5.2.1: The *Properties* dialog box, for adding a new action to the pipeline.

Workflow > Show flow chart... The *Workflow* window will be discussed in more detail in 5.2.3.

The behavior of the pipeline can be modified with the *Settings* dialog box. Disabling 'Run actions on click' is very useful during the process of creating or modifying a template.

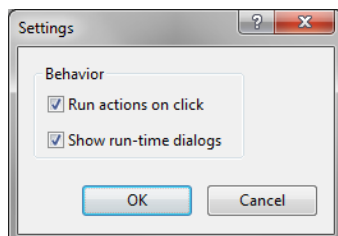


Figure 5.2.2: The *Settings* dialog box, containing the settings of the pipeline.

An overview of the operators of an executed action and their parameters can be shown by highlighting the action and clicking **Workflow > Show report...** (🔍). The report can be copied to clipboard or saved as text file (**File > Copy report (text)** (📄), **File > Copy report (html)** (📄), **File > Export report (text)**, and **File > Export report (html)**).

The user can save an adapted workflow with **File > Save workflow as template...**

In the *Save analysis template* dialog box, a **Name** and **Description** can be entered for the template. The template will be saved as in the *User-defined* category.

The template of an existing workflow can be loaded with **File > Load workflow from template...**

The *Load analysis template* dialog box contains an overview of all predefined and user-defined templates the user can choose from.

Deleting obsolete templates of workflows can be done with **File > Remove workflow templates...**

The *Remove analysis templates* dialog box lists all available templates in the database. Highlight one or more templates and press <OK> to remove them. Please note that this action cannot be undone.

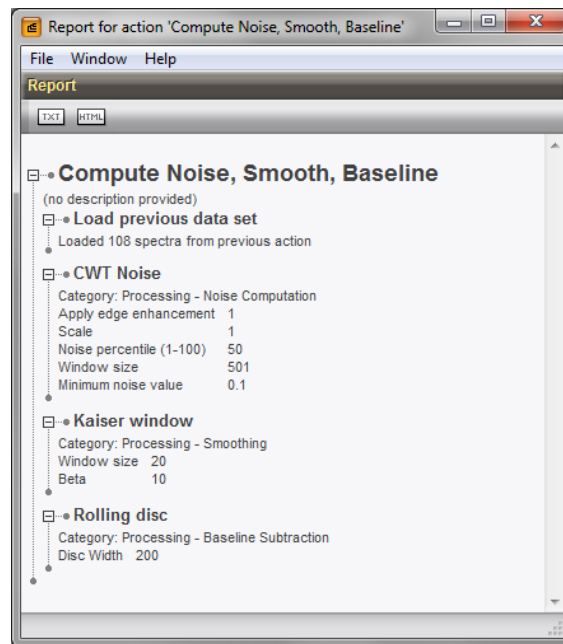


Figure 5.2.3: Example of a report for an action.

The current workflow can be exported to file in order to exchange the workflow between databases, computers, users, etc. with **File > Export workflow to file...**

In the *Export template* dialog box, the user can name the file and select its location by pressing **<Browse>**. Templates are saved as XML files.

To import a previously exported workflow, select **File > Import workflow from file...**

In the *Import template* dialog box, browse for the location of the template file and press **<OK>**.

The workflow from the file will now be loaded in the *Workflow* panel. For future use, this workflow should be saved as a template.

5.2.3 The preprocessing operators and their parameters

In the *Workflow* window (see Figure 5.2.4) of an action, the calculations and steps performed by the action can be viewed and altered. This window can be called through **Workflow > Show flow chart...** or by double-clicking on the action in the pipeline if this action has already been performed or if 'Run action on click' has been disabled. The right panel contains a tree of all available operators listed according to function and the left panel contains the current flow chart of the action. The flow chart can be edited, **File > Add operator...** (📁➕) adds the selected operator from the list at the end of the flow chart, **File > Insert operator...** (📁➡) inserts the selected operator from the list before the selected operator in the flow chart, **File > Replace operator...** (📁🔄) replaces the selected item in the flow chart with the selected operator in the list and **File > Remove operator...** (📁➖) removes the selected operator from the flow chart.

Editing the settings and parameters of an operator in the flowchart is done with **File > Operator settings...** (🔧). For each parameter available for the selected operator, a short description can be shown by clicking on the information button next to the parameter (see Figure 5.2.5).

For each action, it is possible to create a runtime dialog (**File > Build runtime dialog...** (🗨️)). This runtime dialog will be shown when running the action. The parameters of all operators in the action can be defined through this dialog. The creation of the runtime dialog is similar to the creation of power assembler runtime dialogs (see 18.3.9).

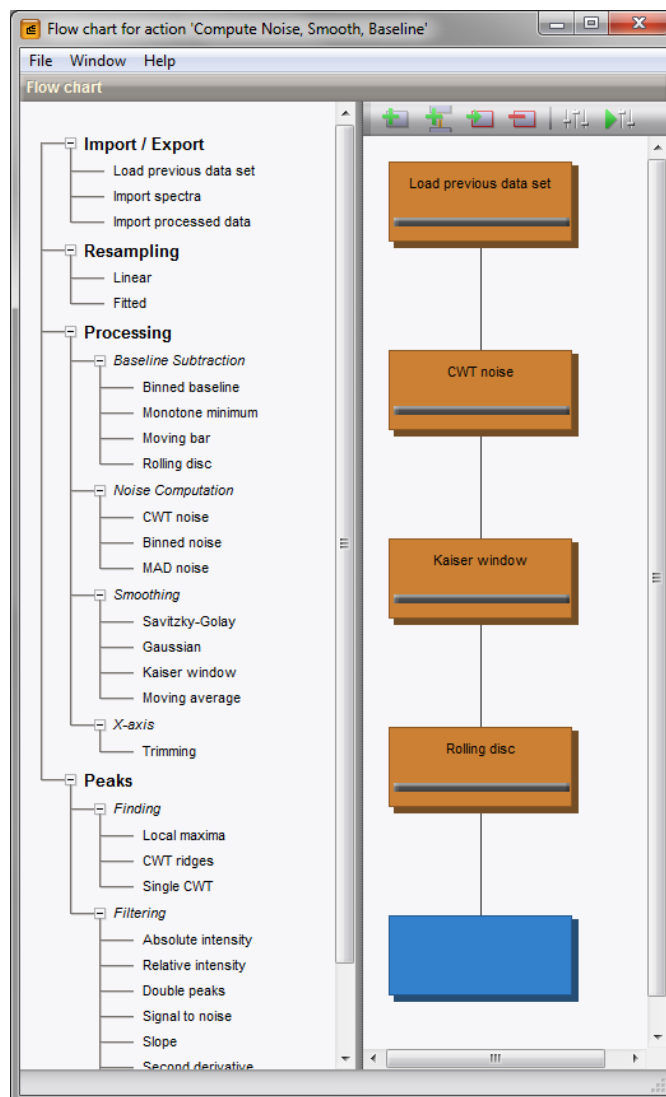


Figure 5.2.4: The *Workflow* window of an action in the preprocessing workflow

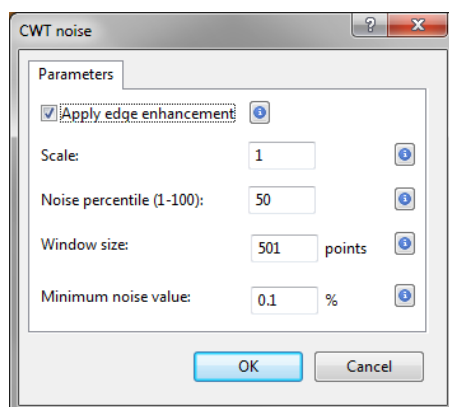


Figure 5.2.5: Changing default parameters of an operator

5.2.3.1 Import/Export

Load previous data set: Used for loading the results of a previous action. Do not use this in the first action of a pipeline.

Import Spectra: Used in the first action of the pipeline to load the spectra from the database into the preprocessing window.

Import Processed Data: Used to import processed data such as peak lists in the processing workflow.

5.2.3.2 Resampling

Linear: Resampling is done on a linear range with the distance between the points defined by the width parameter. A low width results in a high resolution spectrum, but also requires more storage space. A high width reduces both the resolution and the storage requirements. This is a very straightforward and simple approach to resampling. However, for spectra from mass spectrometers of whole bacterial extracts, this usually results in an artificial broadening of the peaks with high m/z values as the resolution of mass spectrometers decreases with increasing m/z . Therefore, this is not a suitable operator for resampling spectra for bacterial typing. For spectra in a lower and smaller m/z range, this operator is suitable however.

Fitted: Resampling is done using a fitted function, this can be either a linear or second degree polynomial function. Fitting with a linear function is identical to using linear resampling with the width set to the spectrum averaged distance between spectrum points. Fitting with a second degree polynomial function takes the decreased resolution with higher m/z into account and decreases the distance between higher m/z points to correct for this. This is the most suitable approach for spectra for bacterial typing and is included in the default preprocessing template.

5.2.3.3 Processing

5.2.3.3.1 Baseline Subtraction

Several strategies are available for the calculation of the baseline. The computed baseline is subtracted from the intensity value of the spectra.

Binned Baseline: Divide the x-range in a number of bins. Bins that do not meet certain criteria (concerning mean y-value, kurtosis and skewness of points inside the bin) are discarded. Each bin gives one baseline point, at its center, with value the minimum y-value in that bin. The baseline is the connection of those bin-points. For most spectra, the bins that are in a peak meet the criteria to be discarded, the baseline is then the connection of the baselines next to the peak. The parameters that can be defined for this operator are the size of the bins and the criteria for discarding the bin:

Bin size: Defines the size of the bins, small bins usually result in a higher baseline value, large bins result in a small baseline value as a larger bin has a higher chance of meeting the criteria for discarding.

Bin versus spectrum average multiplier: The mean intensity of the spectrum is multiplied by this parameter and compared to the mean intensity of the bin. If the intensity of the bin is smaller, the bin is discarded. A multiplier larger than one results in a higher baseline value, smaller than one in a lower baseline value.

Kurtosis upper limit: The kurtosis of each bin is computed. If the kurtosis value exceeds the value specified in the kurtosis threshold, the bin is discarded.

Skewness upper limit: The skewness of each bin is computed. If the skewness value exceeds the value specified in the skewness threshold, the bin is discarded.

Monotone Minimum: The baseline is the monotone minimum of the input spectrum: take the leftmost (i.e. first) spectrum point to be the first point of the baseline. If the y-value at the next point is smaller than the baseline, this y-value is the value of the baseline, if this y-value is larger than the baseline, the previous baseline is used. The value of the baseline at the m/z value of the second spectrum point is then either the y-value at that point, if that y-value is smaller than the baseline at the previous y-value, or the previous y-value otherwise. As a small addition it is possible to ignore leading 0 values (which would give a baseline of value max 0) and/or to start from the leftmost maximum.

Moving Bar: The moving bar method is actually a moving local minimum method, which works exactly like moving a horizontal bar on the lower side of the spectrum, pushing the bar as high as possible. This will not only subtract the baseline, but also filter out small noise. As a parameter the bar width can be adjusted. A small bar results in a high baseline value, but also to the reduction or even removal of relevant peaks. With a large bar, the removal of relevant peaks is less likely, but the baseline value will be low.

Rolling Disc: This algorithm works conceptually like rolling a disk on the lower side of the spectrum, taking the highest point of the disk at a certain x-value as the baseline y-value at that point. Similar to the moving bar, this also results in the removal of small noise peaks. The disk size can be adjusted, a small disk will result in a high baseline value, but also the reduction or even removal of relevant peaks. With a large disk, the removal of relevant peaks is less likely, but the baseline value will be low.

5.2.3.3.2 Noise Computation

There are several algorithms available for computing the noise of a spectrum. These algorithms do not make any changes to the spectra themselves, but the value for the noise can be used later in the pipeline by the algorithms for peak filtering.

CWT noise: CWT (Continuous Wavelet Transform) noise is computed by taking the CWT at a fixed scale (typically 1), giving a CWT coefficient value for each x-value. A moving window is then applied to these coefficients; the noise value at each bin position is the coefficient value at a chosen percentile in that bin.

Apply Edge Enhancement: Activate this enhancement to remove spurious peak detection at the lower and upper boundaries of the input spectrum. Because of the inherent nature of the CWT formalism the value of the wavelet-transformed spectrum is artificially high at the edges of the input spectrum. As this wavelet-transformed spectrum determines the peak signal values this will lead to spurious peak detection at the edges when applying the signal to noise filter. When the edge fix enhancement is activated the spectrum is extended beyond its input boundaries before doing the wavelet transform, effectively removing the artificial increase of the wavelet-transformed values in those regions and thus leading to correct peak detection there.

Scale: Minimum value is 1. Increasing the scale lead to wider and wider peaks being considered as noise.

Window size: Noise is computed using a moving window with size equal to 'Window size'.

Noise percentile: This parameter determines which percentile of the coefficient values in the bin is chosen as noise value. At a percentile value of 50, the median coefficient value is used. Higher percentiles result in higher noise values and downstream in lower signal to noise ratio and the detection of less peaks. A lower percentile leads to lower noise, higher signal to noise ratio and the detection of more peaks.

Minimum noise value: A minimum value can be defined for the noise to prevent superfluous detection of peaks in very flat regions. In these regions, the noise is very low and even very small signals would result in a significant signal to noise ratio and thus the detection of peaks. The value is specified as percentage of the maximum CWT value (taking into account the scale specified for this operator).

Binned noise: Binned noise uses the same algorithm as 'Binned Baseline' above for the computation of the bins (see 5.2.3.3.1). The noise value inside a bin is the standard deviation of y-values inside that bin. The resulting noise estimates are then connected, giving noise estimates at all positions. The parameters are the same as those of 'Binned Baseline', with the exception of a minimum noise value used to prevent superfluous detection of peaks in very flat regions.

MAD noise: MAD (Median of the Absolute Deviation) employs a moving window (similar to the algorithms computing the 'Moving Bar' baseline (see 5.2.3.3.1)). The noise estimate inside the window is the MAD of the y-values inside that window.

Window size: The size of the window used to compute the MAD.

Minimum noise value: A minimum value can be defined for the noise to prevent superfluous detection of peaks in very flat regions.

5.2.3.3.3 Smoothing

Savitzky-Golay: A Savitzky-Golay smoothing performs a local polynomial regression (i.e. fit a polynomial) to a set of consecutive points to determine the smoothed values of each point.

Polynomial order: The order of the polynomial curve used to fit.

Window size: The number of points used for each local fit.

Gaussian: Each point is smoothed over its neighbors using a Gaussian curve. The standard deviation used in the Gaussian function can be adjusted, a larger standard deviation results in more aggressive smoothing.

Kaiser window: A Kaiser Window is a filter typically used in signal processing. Two parameters of the function can be adjusted.

Window size: A larger window size results in a more aggressive smoothing, a smaller is a more conservative smoothing.

Beta: This corresponds to the parameter α/π in the Kaiser function. A smaller value for beta results in a more aggressive smoothing and vice versa.

Moving average: Each y-value is replaced by a local average value. The size of the window to be used for determining the average value can be adjusted. A larger window size results in more aggressive smoothing and vice versa.

5.2.3.3.4 X-axis

Trimming: A lower and upper limit on the x-axis can be set. Values of the spectrum outside these limits will be removed during preprocessing.

5.2.3.4 Peaks

The final step of the preprocessing of spectra comprises the peak detection. This consists of two steps: peak finding and peak filtering. CWT (Continuous wavelet transform) algorithms perform both these steps, but additional filtering may be applied.

Peaks finding

Local Maxima: Find all local maxima and mark them as peaks. When this algorithm is used, additional filtering is essential.

CWT ridges: This is a complex algorithm that is used in combination with the CWT noise detection. For most spectra it performs very well for detecting the relevant peaks. Brief explanation: the input spectrum is convolved with a window function for a number of different window sizes. This means that we check for peaks of various widths, from small to large. Real peaks will typically fit both small and large peak templates, We only retain peaks that are present for a number of widths, thus removing noise peaks. We then also impose a signal to noise limit, using previously computed CWT Noise. For a more in depth understanding of this algorithm, we refer to the specialized literature ([13]).

Single CWT: Computes a CWT at a single scale (window size), then performs CWT signal to noise filtering. This algorithm is a greatly simplified version of the CWT Ridges algorithm. Under normal circumstances the latter algorithm should be used, the Single CWT algorithm is suitable for testing purposes e.g. when one wants to get a good grasp of the effect of several of the parameters of the CWT Ridges algorithm.

Peaks filtering

Peak filtering is essential when the local maxima operator is used for peak finding and optional for the CWT algorithms. Several criteria for filtering are available and can be combined freely.

Absolute intensity: Impose a minimum absolute intensity cut-off on the peaks.

Relative intensity: Impose a minimum intensity cut-off on the peaks relative to the maximum signal intensity.

Double peaks: For all peaks, the corresponding local maximum is found. If multiple peaks correspond with the same local maximum, only the closest of those peaks is retained. This operator can/should be applied after the CWT Ridges algorithm, which can detect small noise bumps on very large peaks as extra peaks.

Signal to noise: Probably the most important filter: use the previously computed noise estimates (Binned Noise or MAD Noise, not CWT Noise) to filter peaks. Note that this filtering is already present in the CWT ridges algorithm.

Slope: All slope values between consecutive points are computed. Based on an input percentile value, a minimum slope value is chosen. We then attribute a fixed width to each peak and check the corresponding slope, if it is smaller than the found cutoff value the peak is rejected.

Second derivative: Only retain peaks that have a positive second derivative e.g. peaks that are local maxima.

Width: Checks the width of each peak with a chosen value, if the width is smaller, the peak is rejected. This will filter out spikes that are usually artifacts.

Trimming: Will reject all peaks outside a chosen X-range.

Chapter 5.3

Summary spectra

5.3.1 Creating summary spectra

Individual spectra can be combined into summary spectra. This is very useful when working with technical and/or biological replicates of the same strain or for creating representative spectra for subgroups or species. Using the summary spectra for further analysis will filter out variations that are caused by the technique or the handling of the strains. In a summary spectrum, only peaks that are consistently present in the individual spectra will also be present in the summary spectrum, other peaks will be averaged out.

In BioNumerics there are two majors paths to create summary spectra. One way is to use an information field or a combination of information field to distinguish which individual spectra should be included into one summary. All spectra with the same value for the defined field(s) will be included into the same summary. A second path uses the levels of BioNumerics, for a more detailed description on how to use levels, we refer to [3.3.10](#). All entries that are linked to the same key on a higher level will be included into the same summary spectrum. In general, the first path is easy in setup, but with large datasets, the data will be more difficult to organize and the user will easily lose the overview of the data. The second path requires a bit more time and attention during the setup phase, but, once correctly set up, allows for a smooth daily routine where a structured overview on the data is obtained with little effort.

To create summary spectra, first select all entries you wish to create summaries for, then select **Analysis > Spectrum types > Summarize...** This will start the *Create summary spectra* wizard wizard.

In the first page of the *Create summary spectra* wizard wizard, the user can choose between creating the summary spectra in the same level based on an information field or using the levels and the dependencies between the levels to create the summary in a higher level. The second option is only possible when the database actually contains levels.

When creating summary spectra in the same level using (an) information field(s), the second page of the *Create summary spectra* wizard is shown in Figure [5.3.2](#). The information fields in the database are listed and can be selected with the check box to the left of the information field. Several fields can be selected together, only spectra for which the values of all selected fields are identical will be summarized together. The number of summary spectra that will be created is shown at the bottom of the window.

When summarizing in the same level using information fields, a detailed overview of the summary spectra can be displayed by clicking **<Details>**. This is particularly useful if the overview is not what the user expected, for instance in case of typo's. In the detailed overview, for each summary spectrum that will be created, the number of individual spectra included as members is reported together with the common fields of these member spectra. The user can review this information and if corrections in the database are necessary, cancel the *Create summary spectra* wizard.

If you are working in a database with levels, it is possible to summarize your individual spectra to a higher level. The only choice that needs to be made is to which level will be summarized (see Figure [5.3.4](#)). All

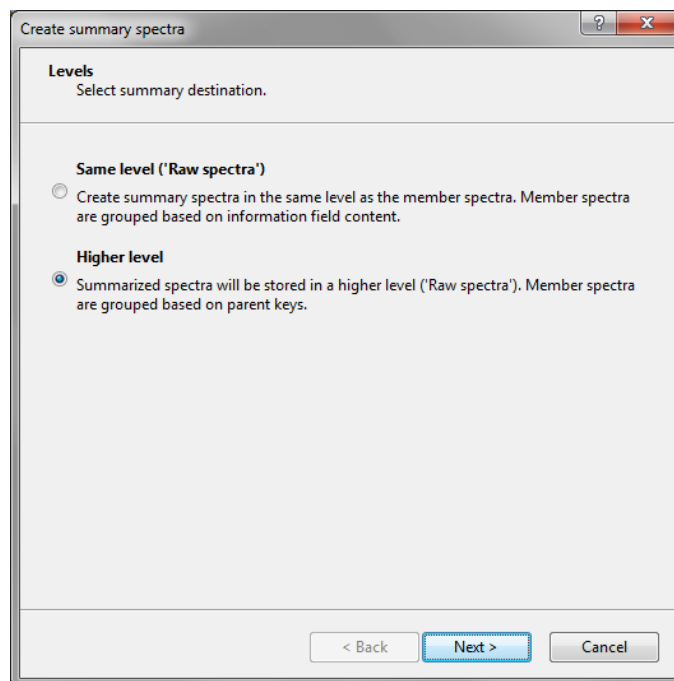


Figure 5.3.1: First page of the *Create summary spectra* wizard wizard.

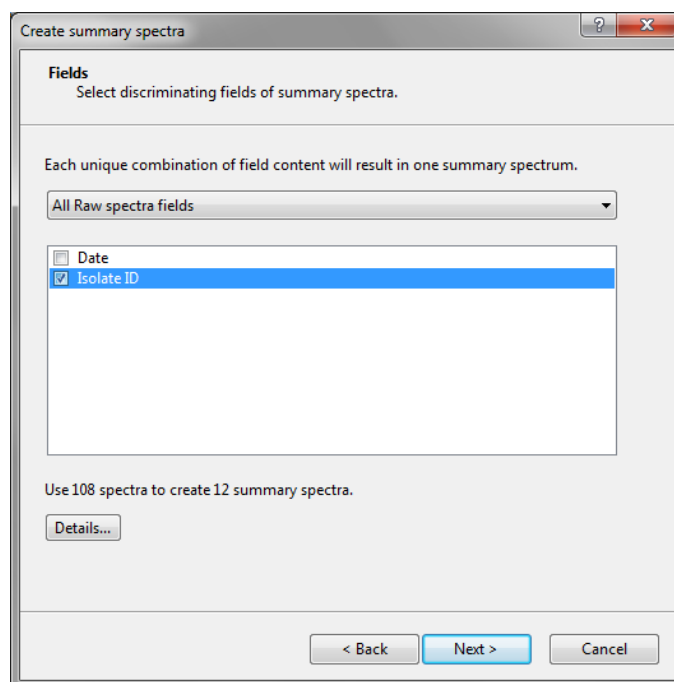


Figure 5.3.2: Second page of the *Create summary spectra* wizard wizard: Summary in same level

entries with the same parent key in the selected level will be included in the same summary spectrum. More information on how to create the dependencies between parent and child entries can be found in 3.3.10.5. If entries without parent keys to the selected level are included in the selection, a warning message is received (Figure 5.3.5). Clicking *<Yes>* unselects the entries without parent key and the creation of the summary spectra can be continued. Clicking *<No>* allows the user to either change the previously selected settings of the wizard or to cancel the wizard and make the necessary corrections in the database before proceeding.

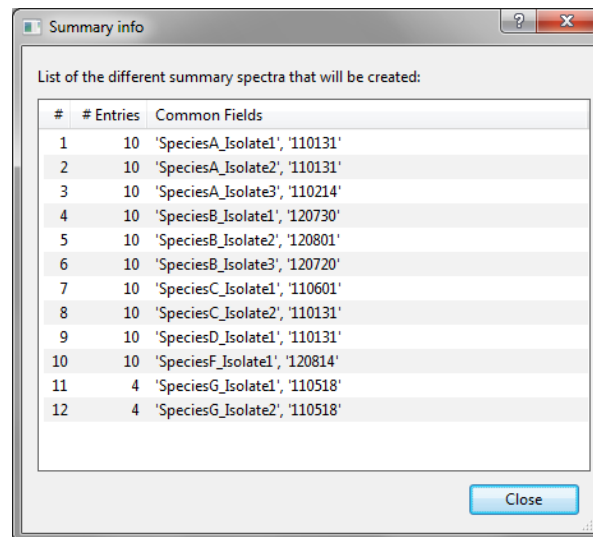


Figure 5.3.3: Example of summary info.

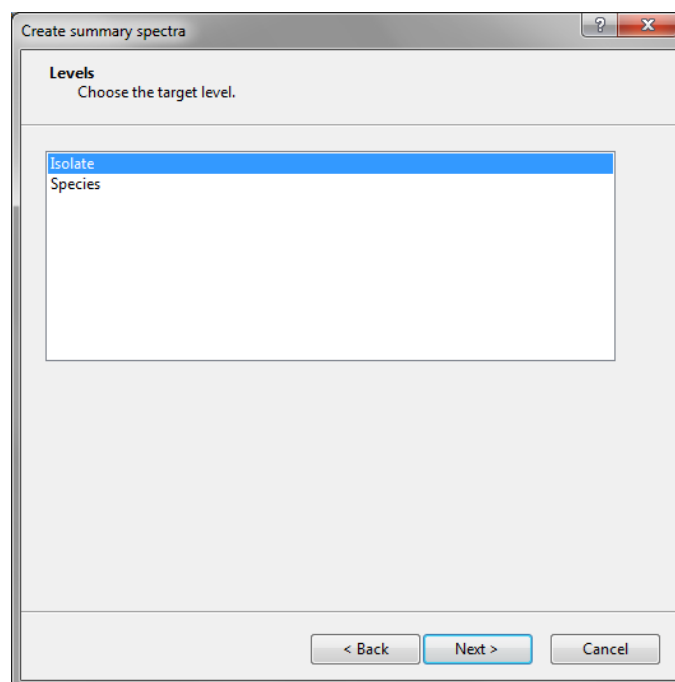


Figure 5.3.4: Second page of the *Create summary spectra* wizard wizard: Summary to higher level.

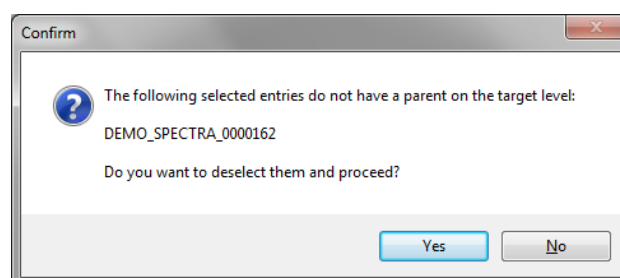


Figure 5.3.5: Error upon summarizing of entries without parent key.

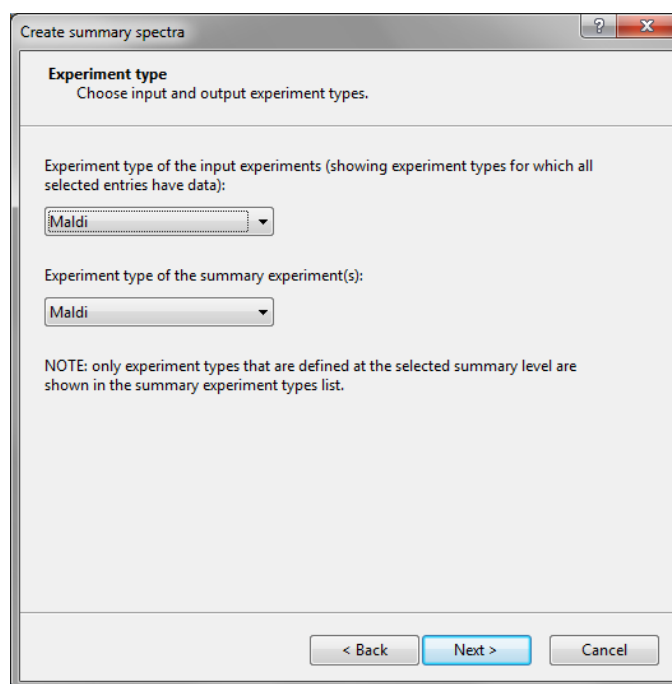


Figure 5.3.6: Third page of *Create summary spectra* wizard

If several spectrum type experiments are present in the database, the input and output experiment type can be selected independently. For the input experiment type only experiment types that are present for all selected entries are shown. If this box is empty, the selection of the entries should be reviewed. For the output experiment, only experiment types that are defined at the summary level are shown.

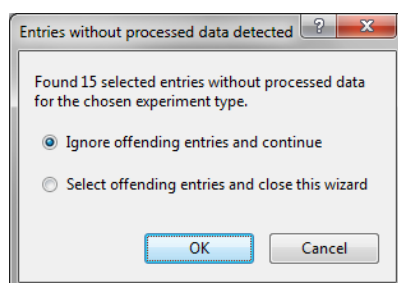


Figure 5.3.7: Error generated upon summarizing entries without preprocessed spectra.

If entries have been selected that do not contain any processed spectra for the selected experiment types, clicking *<Next>* in the third page of the *Create summary spectra* wizard, will generate an error (see Figure 5.3.7). The user can choose to continue without these entries, or to select the entries without preprocessing data and cancel the *Create summary spectra* wizard.

In the fourth step of the wizard (Figure 5.3.8), the user can select an appropriate workflow for the processing of the spectra. Three predefined templates are present: one for technical replicates, one for biological replicates and one general template for processing for instance several strains of the same subgroup or the same species. Detailed information on the content of these templates and on creating your own templates can be found in section 5.3.3.

In the final step, the user can choose how to execute the workflow. By default, the *Summary Spectrum* window will be run after finalizing the creation of the summary spectra, allowing the user to visually inspect the results of the summary and to adapt the processing workflow if necessary. If many entries are summarized (rule of thumb: more than 500, although this depends on the performance of the computer), it

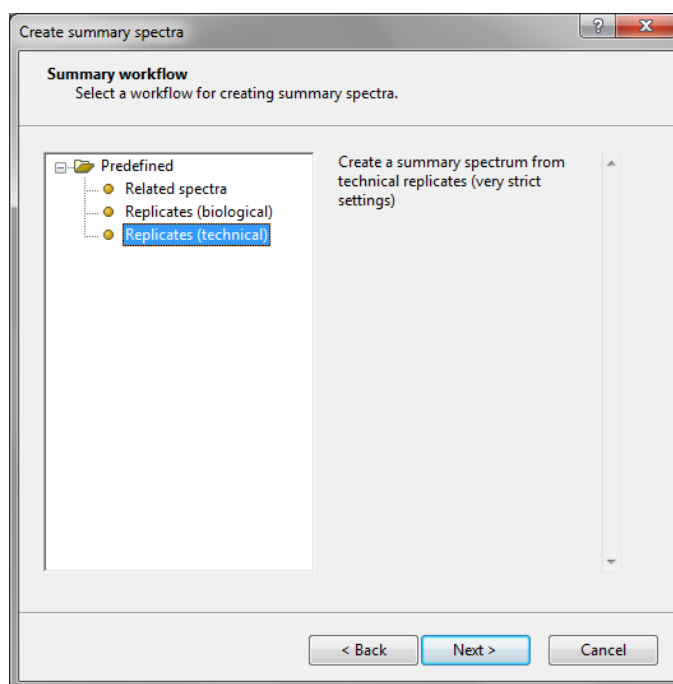


Figure 5.3.8: Fourth page of the *Create summary spectra* wizard

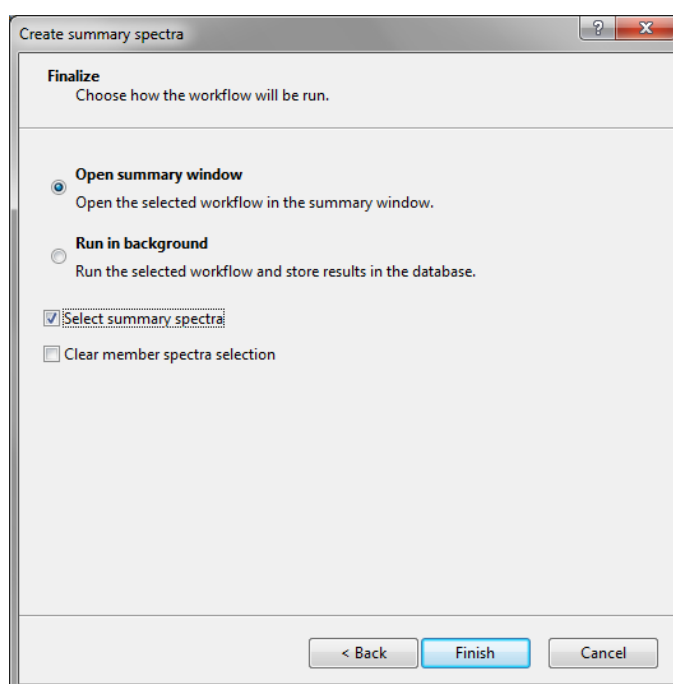


Figure 5.3.9: Final page of the *Create summary spectra* wizard

is advised to run the calculations in the background, without visualization.

5.3.2 Processing summary spectra

The *Summary Spectrum* window can be opened either by *Analysis > Spectrum types > Open summary window...* or after finalizing the *Create summary spectra* wizard. In the first case, the currently selected

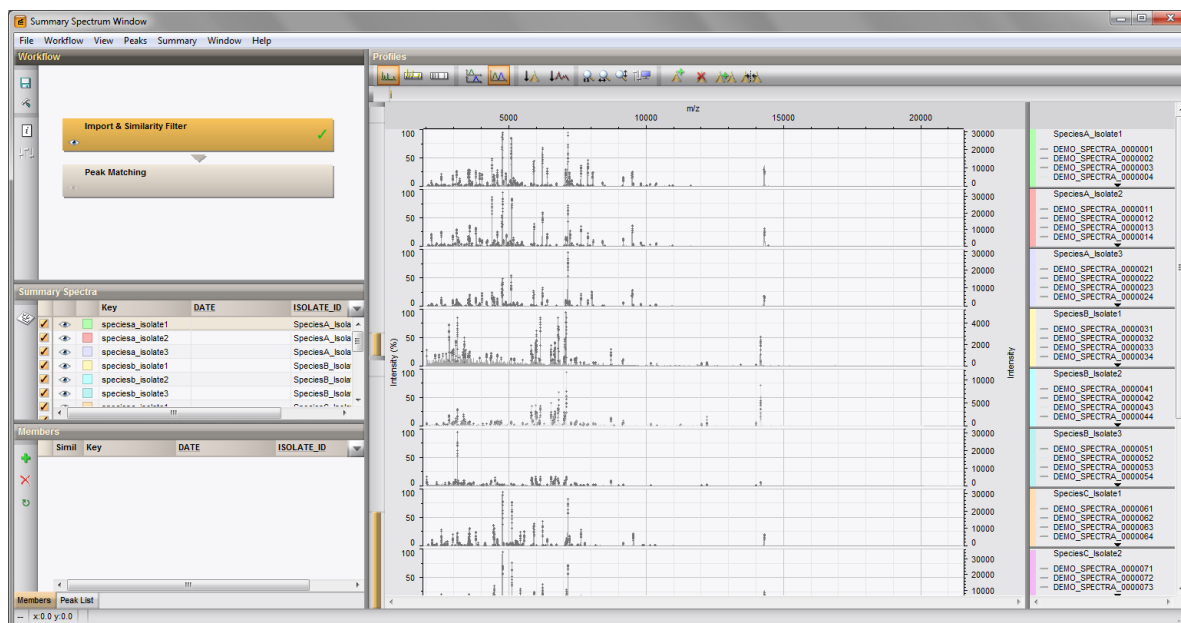


Figure 5.3.10: The *Summary Spectrum* window before running a workflow.

summary spectra will be loading into the *Summary Spectrum* window, in the second case, the newly created summary spectra are shown.

The structure of the *Summary Spectrum* window (Figure 5.3.10) is almost identical to the *Spectrum Preprocessing* window (see 5.1.3.3), only the specific difference between both windows will be discussed here.

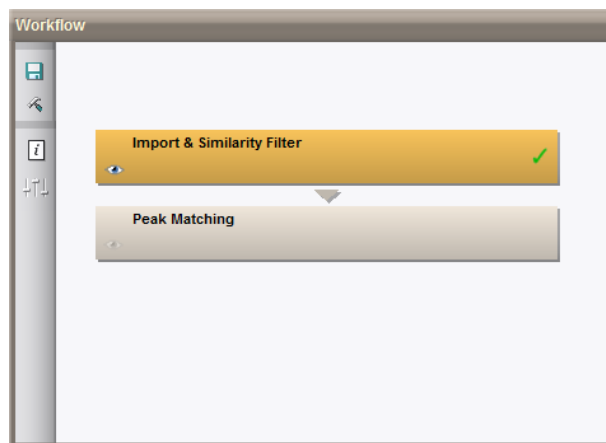
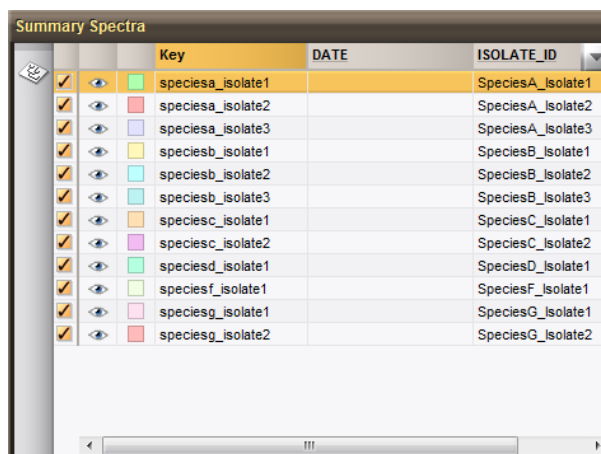


Figure 5.3.11: The *Workflow* panel of the *Summary Spectrum* window.

The *Workflow* panel is identical to the *Workflow* panel of the *Spectrum Preprocessing* window, only the workflow loaded is different. The default workflow contains two actions: Import & Similarity filter and Peak matching. The content of these actions will be explained in more detail further on (see 5.3.3).

The *Summary Spectra* panel list (Figure 5.3.12) corresponds to the *Spectrum List* panel of the *Spectrum Preprocessing* window. One additional command is available: **Summary > Create comparisons** (🔍).

The *Create summary comparisons* dialog box allows the user to choose which comparisons will be created starting from the *Summary Spectrum* window. A comparison containing only the summary spectra and one containing only the members can be created. The entries in the member comparison will be grouped according to summary spectra. The user can choose whether inactivated spectra are included in the comparison or not.



	Key	DATE	ISOLATE_ID
✓	speciesa_isolate1		SpeciesA_Isolate1
✓	speciesa_isolate2		SpeciesA_Isolate2
✓	speciesa_isolate3		SpeciesA_Isolate3
✓	speciesb_isolate1		SpeciesB_Isolate1
✓	speciesb_isolate2		SpeciesB_Isolate2
✓	speciesb_isolate3		SpeciesB_Isolate3
✓	speciesc_isolate1		SpeciesC_Isolate1
✓	speciesc_isolate2		SpeciesC_Isolate2
✓	speciesd_isolate1		SpeciesD_Isolate1
✓	speciesf_isolate1		SpeciesF_Isolate1
✓	speciesg_isolate1		SpeciesG_Isolate1
✓	speciesg_isolate2		SpeciesG_Isolate2

Figure 5.3.12: The *Summary Spectra* panel provides a list of summary spectra.

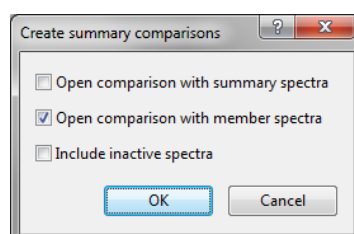
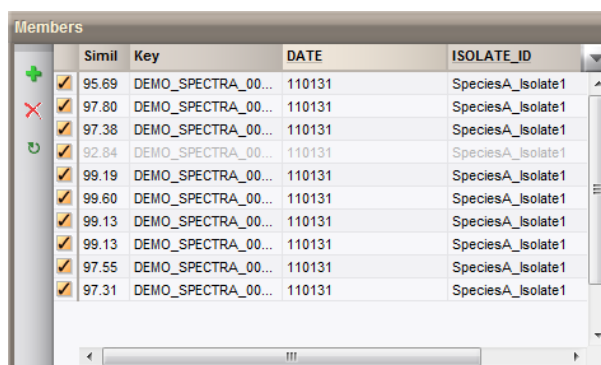


Figure 5.3.13: The *Create summary comparisons* dialog box with options for creating summary comparisons.



	Simil	Key	DATE	ISOLATE_ID
✓	95.69	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	97.80	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	97.38	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	92.84	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	99.19	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	99.60	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	99.13	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	99.13	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	97.55	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1
✓	97.31	DEMO_SPECTRA_00...	110131	SpeciesA_Isolate1

Figure 5.3.14: The *Summary Spectra* panel shows a list of member spectra.

When a summary spectrum is highlighted in the *Summary Spectra* panel, a list of the member spectra is shown in the *Members* panel (Figure 5.3.14). For each member spectrum, the similarity to the summary spectrum for the currently selected workflow action is given. Spectra that are grayed out have been inactivated by a workflow operator. Note that the *Members* panel is empty if no summary spectrum is selected in the *Summary Spectra* panel. If no workflow action is selected then all spectra will appear grayed out with similarities identical to zero.

Selecting **Summary > Add selected entries** (+) will add any selected spectra to the highlighted summary spectra. The user will be prompted to confirm this action and choose whether to rerun the workflow or not (Figure 5.3.15).

Selecting **Summary > Remove selected entries** (X) will remove any selected member spectra from the highlighted summary spectra. The user will be prompted to confirm this action and choose whether to rerun

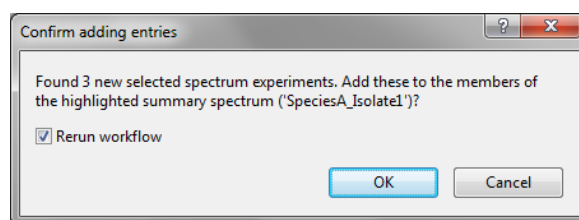


Figure 5.3.15: Confirmation for adding member spectra.

the workflow or not.

The command **Summary > Update member spectra list** (🔄) will update the list of member spectra. The criteria set for the creation of the summary spectrum will be reevaluated and members will be added if they fit these criteria. These criteria are either child-parent relationships (summary spectrum created using levels) or common information field values.

For the peaks of a summary spectrum, no S/N, lower and upper limit and area are available as these are only calculated for raw spectra. For each peak, a peak detection rate (PDR) is calculated, this is the percentage of member spectra for which this peak is detected. An example of a peak list of a summary spectrum is shown in Figure 5.3.16.

#	x	y	PDR
1	2836.57	3060.50	100.00
2	2965.02	1211.73	90.00
3	3073.83	2055.99	100.00
4	3128.31	3423.67	100.00
5	3393.53	2120.52	100.00
6	3425.47	1137.73	80.00
7	4198.83	986.72	80.00
8	4359.91	995.99	80.00
9	4576.83	965.11	80.00
10	5827.18	1297.78	90.00
11	5969.44	1714.92	100.00
12	6001.42	2496.47	90.00
13	6014.11	1680.13	80.00
14	6151.20	3661.26	100.00
15	6530.57	1332.81	90.00
16	6646.28	1138.45	90.00
17	6692.26	1779.78	100.00
18	6789.06	3339.37	100.00
19	6820.75	2980.09	100.00
20	7003.16	688.33	80.00
21	7038.70	2004.72	100.00
22	7086.48	3733.62	100.00
23	7266.85	574.16	80.00

Figure 5.3.16: Example of a peak list of a summary spectrum.

The *Profiles* panel of the *Summary Spectrum* window is very similar to the *Profiles* panel of the *Spectrum Preprocessing* window. To the right of the profiles, the key of the summary spectra is shown with a list of the member spectra. Clicking on a name of a member spectra in this list will select this spectrum.

Before the summary workflow has been executed, the profiles of all member spectra are shown in grey, the inactivated profiles in light grey. Clicking on a profile (or on the name of the member spectra either in the members list or the list to the right of the profiles) will select this profile, a selected profile is visualized in bold.

After the execution of the summary workflow, the summary spectrum will be calculated and visualized in the profiles panel in red. The peaks identified on the summary spectrum are represented as lines, adjacent lines are colored differently to allow easier visual inspection. Each peak of the member spectra that contributes to this summary peak is given the same color as the summary peak (see Figure 5.3.18).

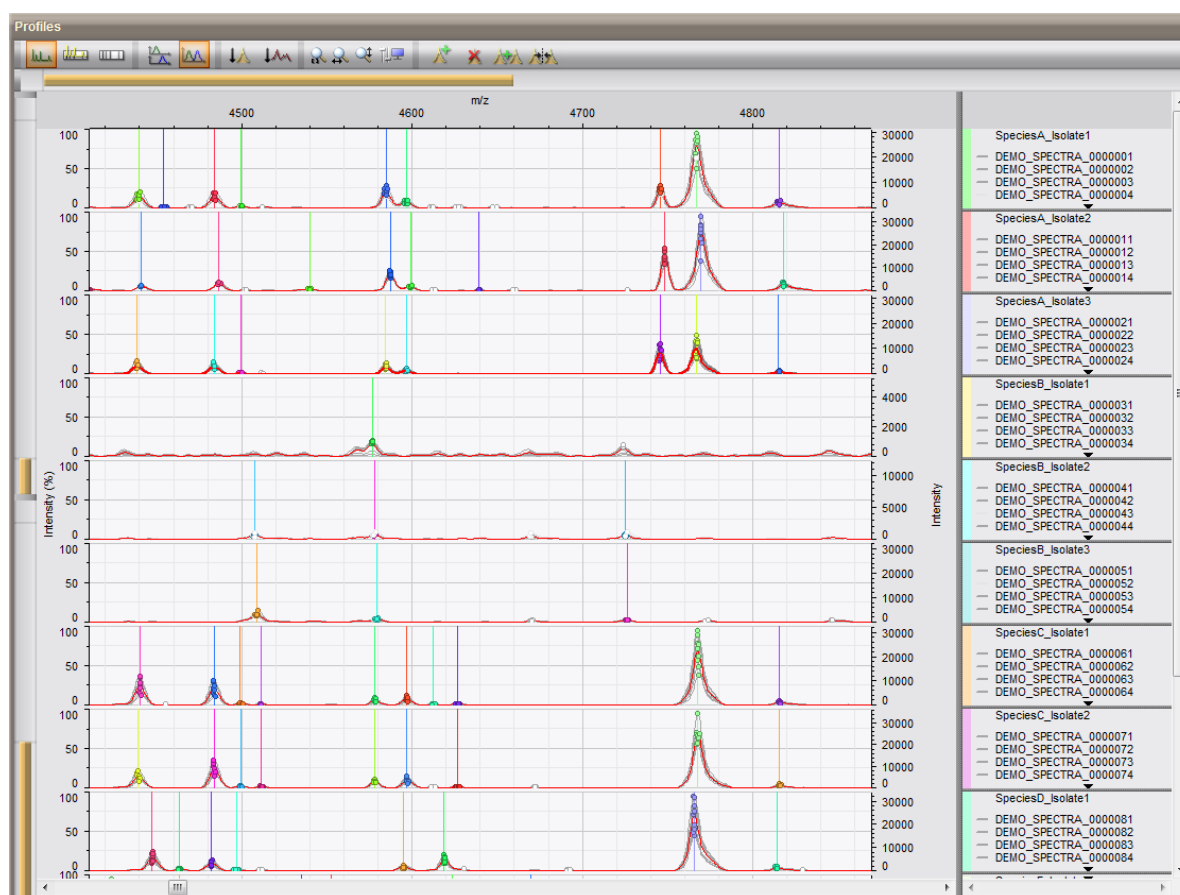


Figure 5.3.17: The *Profiles* panel, after execution of the workflow.

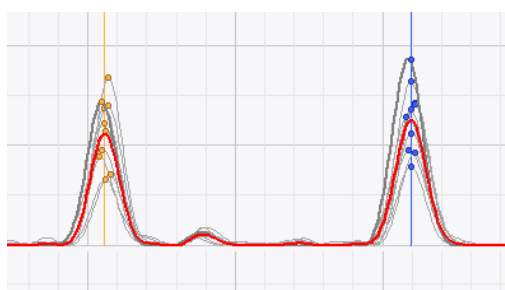





Figure 5.3.18: Close up of two summary peaks.

With the command **View > Show single summary spectrum and its members** (📊), the user can change the view of the profiles. Each member profile will be visualized with the summary spectra. This mode only shows one summary spectrum at a time, the currently shown summary spectrum can be changed by clicking on 📊 of another summary spectrum in the list of summary spectra.

By default the summary spectra are ordered alphabetically by their key. When a single summary spectrum is visualized with its individual members, two additional sort options are available. The user can sort the spectra by size of a selected peak (**View > Sort channels by selected peak** (📊)) or by similarity to the summary spectrum (**View > Sort channels by similarity with summary spectrum** (📊)).

In the profiles panel, there are a few tools available for the manual editing of the summary spectra. These tools should be used with care as manual editing may be subjective. Furthermore, if the workflow is rerun, any manual edits are overwritten. With **Peaks > Create summary peak at cursor** (📊), a summary peak is created at the position of the cursor. Individual peaks can be added to an existing summary peak, either

by selecting both the individual peaks and the summary peaks and selecting **Peaks > Assign selected peaks to selected summary peak** () or by clicking on the peak and dragging it to the desired summary peak. Deleting peak classes is possible by clicking **Peaks > Remove selected summary peaks** (). Two peak classes can be merged by selecting them (hold down shift while clicking and dragging in the profile) and clicking on **Peaks > Merge selected summary peaks** ()

5.3.3 The summary processing operators and their parameters

5.3.3.1 Background

Managing templates and workflows is identical for the creation of summary spectra as for the preprocessing of spectra. For more details on how to manage your summary pipeline, see [5.2.2](#). The specific operators for the creation of summary spectra and their parameters will be described in more detail here. There are four different classes of operators: Import/Export, Disable spectra, Peak matching and Summary.

5.3.3.2 Import/Export

The operators here are identical to the operators in the same class of the preprocessing operators, see [5.2.3.1](#).

5.3.3.3 Disable spectra

The operators in this class function as a quality control, disabling spectra that do not meet certain criteria.

Minimum intensity filter: When a value is defined as minimum intensity, a spectrum will be disabled if the maximum intensity value of this spectra is beneath the specified minimum value. The value is defined as absolute intensity, so the user should choose this value according to his or her dataset. It is also possible to define an m/z range. If a range has been defined, this operator will only be evaluated withing the given range.

Number of peaks filter: Defining a value for the minimum number of peaks, will disable spectra that have less peaks than the required minimum. This criteria can be limited to a certain m/z range. In practice, the number of peaks filter will often filter out the same spectra as the minimum intensity filter.

Similarity filter: The similarity filter will disable spectra that have a similarity to the summary spectra lower than the chosen value. The similarity to the summary is calculated with a Pearson correlation coefficient. This is very useful when summarizing technical and biological replicates as these are expected to have a high similarity. A similarity lower than usual is often the result of technical or human errors. This filter is included in the predefined templates, for technical replicates it is placed at 95%, for biological replicates at 90% and for similar spectra at 40%.

5.3.3.4 Peak matching

There is only one operator, Peak matching, in this class, containing three parameters.

Minimum peak detection rate: The minimum percentage of member spectra that should contain the peak. If a peak is found in less members, it is not considered as a summary peak. In the predefined template for technical replicates, this cutoff is placed at 75%, for biological replicates at 60% and for similar spectra at 50%.

Constant tolerance: The tolerance is calculated as a linear function of m/z:

$$\text{Position tolerance} = \text{linear tolerance} \cdot m/z + \text{constant tolerance}$$

. The tolerance increases with increasing m/z values. This parameter is the constant value in this function. At low m/z values it is the most important factor in the tolerance, at high m/z values it is negligible. In the predefined templates, the default value for this parameter is 1 x-axis unit.

Linearly varying tolerance value: The tolerance is calculated as a linear function of m/z (see Constant tolerance), so the tolerance increases with increasing m/z values. This parameter is the linear value in this function. At low m/z values, it is negligible, but at high m/z values it is the most important factor in determining the tolerance. In the predefined templates, the default value for this parameter is 300 ppm.

5.3.3.5 Summary

There are three different methods to calculate the summary spectrum:

Average: For each m/z value, the intensity value of the summary spectrum is the average of the intensity values of the member spectra at that point.

Maximum: For each m/z value, the intensity value of the summary spectrum is the maximum of the intensity values of the member spectra at that point.

Minimum: For each m/z value, the intensity value of the summary spectrum is the minimum of the intensity values of the member spectra at that point.


Chapter 5.4


Cluster analysis of spectra

5.4.1 Spectrum comparison settings

Please note that for cluster analysis of spectrum types, the Fingerprint data module (**FP**) and the Tree and network inference module (**TN**) need to be present in your BioNumerics configuration.

Proceed as follows to calculate a cluster analysis on a fingerprint type:

In the *Comparison* window, select the spectrum type in the *Experiments* panel on which the cluster analysis should be based. Optionally, display the spectra by pressing the eye button () next to the experiment name.

Select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**.... Alternatively, press the  button, in which case the following menu pops up (Figure 5.4.1).

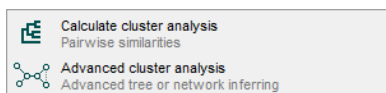


Figure 5.4.1: Cluster analysis menu popped up from the dendrogram button.

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient (see Figure 4.2.2).

The hierarchical representation on the left provides an overview of the available coefficients. Depending on the selected coefficient, the relevant settings are displayed on the right. The coefficients are subdivided in two categories: **Curve based** and **Peak based**. Each of the categories can be collapsed by clicking on the small "-" (minus) sign that precedes the category name.

All coefficients from the **Curve based** category provide similarities based upon densitometric curves.

The Pearson product-moment correlation is calculated as:

$$r = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \sqrt{\sum_{i=1}^n y_i^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2}}$$

with x_i and y_i the densitometric values of both profiles and n the number of points in the curves.

The **Cosine coefficient** is calculated as:

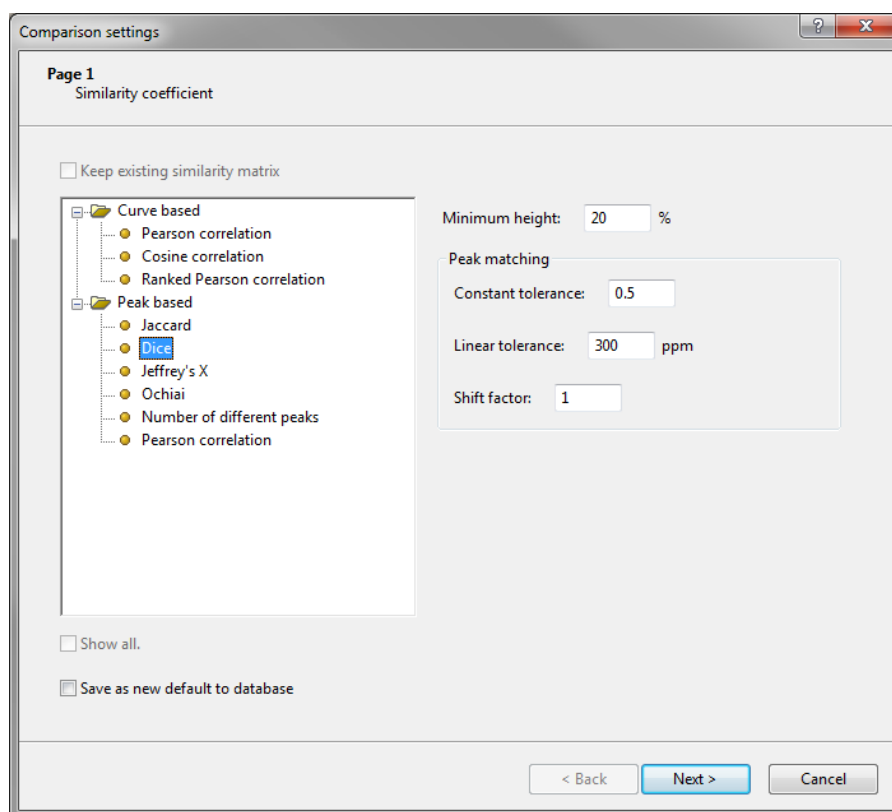


Figure 5.4.2: The *Similarity coefficient* wizard page, which deals with the choice of the similarity coefficient.

$$r = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

The **Ranked Pearson correlation** is a variant of the Pearson correlation coefficient that is less sensitive to outliers, since it is based on *ranks* instead of on the actual densitometric values. The densitometric values of each curve are first sorted and replaced by their rank. Then a Pearson correlation coefficient is calculated on the ranks.

Two parameters can be specified for **Curve based** coefficients:

- **Curve smoothing** can be applied to remove noise from the densitometric curves. If the curves have been properly preprocessed, this parameter should be left at 0%.
- **Negative similarities** can be dealt with in different ways. If **Clip to zero** is selected, negative similarity values will be replaced by zero (no correlation). When **Unchanged** is set, the program will calculate with the negative values. **Absolute value** will treat negative and positive similarity values in the same way. **Negative similarities** values can only be obtained with the **Pearson correlation coefficient**, therefore the option is not available when any of the other coefficients is selected.

Peak based category: Six different binary coefficients measure the similarity based upon common and different peaks.

The **Jaccard** coefficient is calculated as:

$$S_J = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

with N_A and N_B the number of peaks in profile A and B, respectively, and N_{AB} the number of common peaks between the two profiles.

The **Dice** coefficient is calculated as:

$$S_D = \frac{2N_{AB}}{N_A + N_B}$$

The **Jeffrey's X** coefficient is calculated as:

$$S_X = \frac{1}{2} \left(\frac{N_{AB}}{N_A} + \frac{N_{AB}}{N_B} \right)$$

The **Ochiai** coefficient is calculated as:

$$S_O = \frac{N_{AB}}{\sqrt{N_A N_B}}$$

The **Different peaks** coefficient is essentially a distance coefficient as it simply counts the number of different peaks in two patterns. It is converted into a similarity by subtracting this distance value from 1. It is calculated as:

$$S_B = 1 - ((N_A + N_B) - 2N_{AB})$$

Following options are available for each of these five binary **Peak based** coefficients:

Minimum height can be used to exclude weak or irrelevant peaks. **Minimum height** is entered as a percentage of the OD range of spectrum.

Peak matching includes the **Constant tolerance** and **Linear tolerance**, which are used to calculate the position tolerance (the maximal shift allowed between two peaks to consider them as matching). The function used to calculate the position tolerance is:

$$\text{Position tolerance} = \text{linear tolerance} \cdot m/z + \text{constant tolerance}$$

A third **Peak matching** parameter that can be adjusted is **Shift factor**. This is comparable to the optimization used in fingerprints (see 4.2.1). The entire curve will be shifted according to

$$\text{shift factor} \cdot \text{position tolerance}$$

and the peak matching is performed within this range.

The sixth peak based coefficient (**Pearson correlation**) is not binary, but it is in fact a hybrid between peak and curve based. An artificial curve is made based on the peaks and a normal Pearson product moment correlation coefficient is calculated based on these artificial curves. This has the advantage that only the peaks are taken into account, instead of the entire curve, reducing the effect of noise and that intensity differences between peaks are taken into account, unlike the binary coefficients.

If a similarity matrix already exists for the selected experiment, an option **Keep existing similarity matrix** appears. When checked, the previously calculated similarity matrix will be used and all coefficient options (for both **Curve based** and **Peak based** coefficients) will appear gray (disabled).



In 4.2.4, is discussed how to have the program automatically calculate the optimal parameters for a fingerprint type, which also applies to spectrum types.

Check

Save as new default to database if one wants the specified comparison settings to be saved in the database as

default settings. When *Save as new default to database* is unchecked, the last comparison settings will only apply during the current *session* of BioNumerics; if the software is closed, the settings will not be saved.

The settings as defined in the *Comparison settings* wizard are stored along with the spectrum type. A dialog box with the same settings can be called from the *Spectrum type* window (5.1.2).

5.4.2 Spectrum display functions

For spectrum types, additional information can be shown in the *Experiment data* panel. Before using these spectrum display functions, make sure that the image of the spectrum type is shown in the *Experiment data* panel by pressing the eye button (👁) next to the experiment name in the *Experiments* panel.

Select *Spectra* > *Spectra* (📊) to show the spectra as spectra (default), select *Spectra* > *Band representation and spectra* (📊📊) to view both a band representation and the spectra and finally, select *Spectra* > *Band representation* (📊) to view the spectra as bands.

When a peak matching is present, the display settings of the peak classes can be altered by selecting *Spectra* > *Display settings* (⚙). This will open the *Display settings* dialog box (see Figure 5.4.3).

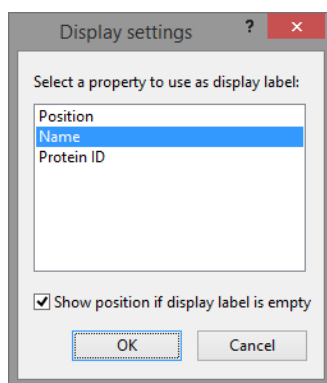


Figure 5.4.3: The *Display settings* dialog box for peak classes.

The *Display settings* dialog box can be used to adjust which field is displayed for the peak classes. By default, the Position of the peak class is displayed. One other field is available (Name) which can be used to name a peak class. Custom fields can be created by the user in the *Manage peak class types* dialog box. When the display field is different from Position, the user has the option to display the position when the chosen display field is empty.

The *Experiment data* panel offers an overview of the spectra, as it is optimized for visualizing large amount of spectra with minimal loading time, it is not suitable to review the spectra in detail. To look into certain spectra in more detail, select the spectra of interest and click *Spectra* > *Open spectrum window* (🔍). This will open the *Spectrum Preprocessing* window with no workflow loaded, a suitable environment for visualizing spectra (see 5.1.3.3).

Chapter 5.5

Peak matching

5.5.1 Introduction

Peak matching for spectrum types is very similar to band matching for fingerprint types. For a general introduction, we refer you to 4.3. Please note that to perform peak matching on spectrum types, the Fingerprint data module (FP) and the Tree and network inference module (TN) need to be present in your BioNumerics configuration.

There are three important terms for the peak matching: peak, peak class and peak class view. A *peak* is defined on the spectrum during preprocessing, performing a peak matching does not make any changes to the defined peaks. A *peak class* is defined on a group of spectra and is similar to the band classes for fingerprint types. During peak matching, peaks will be assigned to a peak class. A *peak class view* is a collection of peak classes, similar to band class views for fingerprint types. Several peaks class views can be defined containing different peak classes.

5.5.2 Creating a peak matching

A peak matching analysis is done in the *Comparison* window. Therefore, a comparison containing the entries on which you want to perform a peak matching should first be created or opened.

Click on the spectrum type in the *Experiments* panel on which you want to perform a peak matching and select **Layout > Show image** (🖼️) or press the eye button (👁️) next to the experiment name in the *Experiments* panel.

Select **Spectra > Do peak matching** (🔍). This pops up the *Peak class matching* wizard (see Figure 5.5.1).

In this dialog, the user can select the mode of the peak matching. The first time a peak matching is performed for a spectrum type, the only option available is **Recreate peak classes**. This will create new peak classes and add these to the active peak class type.

When there is already a peak matching present, the user has three options. **Existing peak classes only** will only search the spectra for peaks within the existing classes. Peaks outside the tolerance level for the matching to an existing class will be ignored. The second option **Extend Peak classes** will first perform a peak matching against the existing classes and then create new classes using the remaining, unmatched peaks. If the third option **Recreate peak classes** is used with an existing peak class type, all peak classes in this type will be replaced by the newly found classes.

In the *Peak match parameters* wizard page, the parameters of the peak matching can be adjusted. The values for **Constant tolerance** and **Linear tolerance** will determine the position tolerance. These parameters are described in more detail in the description of the peak based coefficients (5.4.1).

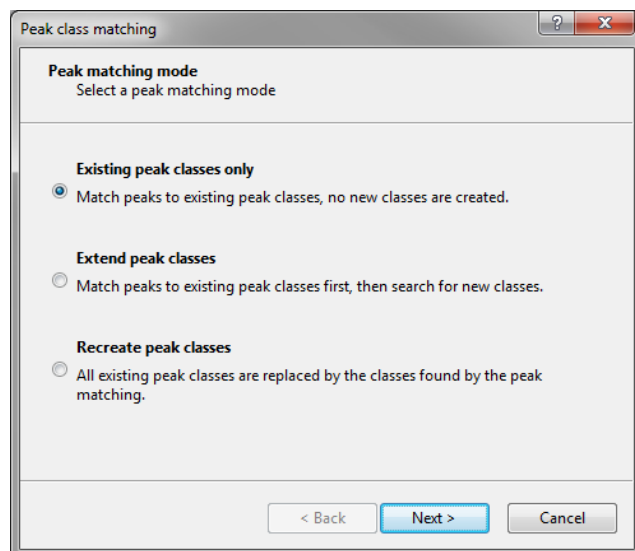


Figure 5.5.1: The *Peak matching mode* wizard page.

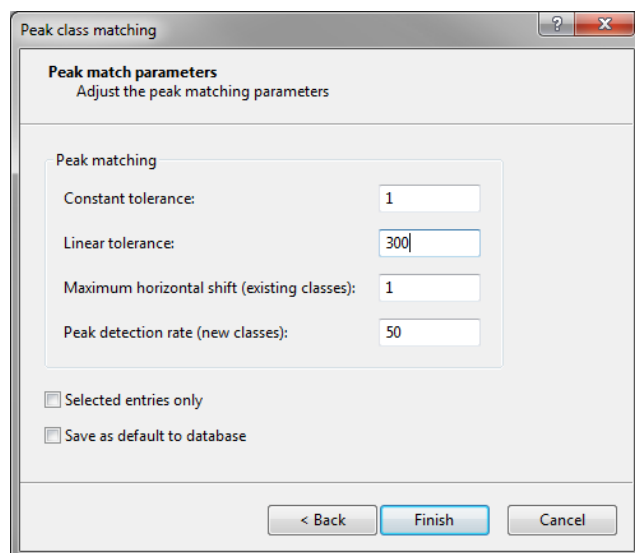


Figure 5.5.2: The *Peak match parameters* wizard page.

The *Maximal horizontal shift* is only available when matching against existing peak classes and is grayed out when the option *Recreate peak classes* was chosen in the previous page. It is identical to the shift factor described in (5.4.1).

The *Peak detection rate* will only be available when creating new classes and will be grayed out when the first option *Existing peak classes only* was chosen in the previous page. The parameter will limit the detection of peak classes to classes containing a minimum number of peaks. This is useful for example when a peak matching is performed on different isolates of one species to identify peaks that are present in most isolates of this species.

The peak matching can be limited to only the selected entries in the comparison (*Selected entries only*) and the results can be automatically saved to the database (*Save as default to database*).

Any modifications to the peak matching and to the peak classes can be saved with *Spectra > Save modified peak classes* (📁).

5.5.3 Managing peak class types

Please note that the older concept of peak class types is replaced with peak class views (see 5.5.5). As such, it will not be possible anymore to create new peak class types and the option is offered to convert existing peak class types into peak class views.

Selecting *Spectra* > *Manage peak class types* (📊) in the *Comparison* window will open up the *Manage peak class types* dialog box (see Figure 5.5.3). The same settings can be accessed from the *Spectrum type* window with *Settings* > *Manage Peak Class Types...* (📊).

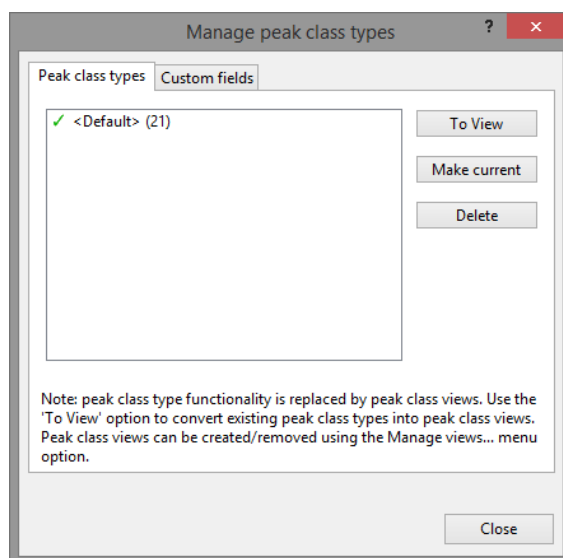


Figure 5.5.3: The *Manage peak class types* dialog box, *Peak class types* tab.

In the *Peak class types* tab of this dialog, the user can manage existing peak class types.


<**Make current**> will make the highlighted peak class type, the active type in the comparison. Double-clicking on the name of peak class type has the same effect. A green mark ✓ is placed at the peak class type that is currently active.

An obsolete peak class type can be deleted with <**Delete**>. This action cannot be undone and the user will receive a warning before the action is executed.

Clicking <**To view**> will automatically convert the peak class type into a peak class view. For more information on peak class views, see 5.5.5.

In the *Custom fields* tab, the user can create custom fields for the peak classes. Any information that needs to be saved concerning a peak class can be saved in custom fields. These custom fields can also be selected in the *Display settings* dialog box as custom field (see 5.4.2).

5.5.4 Managing peak classes

There are several functions available to manage the peak classes. If the Dimensioning and Matrix Mining  is present, these functions form a powerful tool in combination with the *Matrix Mining* window to perform an intensive data mining on the spectra. The *Matrix Mining* window can be opened with *Statistics* > *Matrix mining...*. It will contain the results of the current peak matching and if the *Comparison* window contains raw spectra, the S/N will also be present as a separate layer in the *Matrix Mining* window. The selection of peaks is synchronized between the *Matrix Mining* window and the *Comparison* window, making it possible to select peaks based on a specific analysis in the *Matrix Mining* window and continue with these peaks in

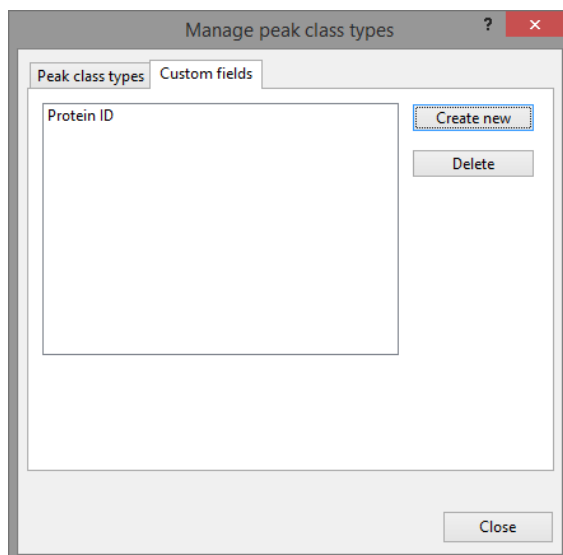


Figure 5.5.4: The *Manage peak class types* dialog box, *Custom fields* tab.

the *Comparison* window. More information on the possibilities of the *Matrix Mining* window can be found in 20.

To adjust the content of the display field and the color of the selected peak classes, make a peak class selection first and use ***Spectra* > *Edit peak class properties*** (🔧). This action will open the *Change Properties* dialog box (see Figure 5.5.5).

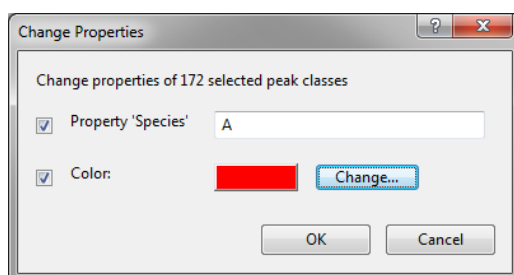


Figure 5.5.5: The *Change Properties* dialog box, to change the properties of the selected peak classes.

The user can adjust the information in the display field for the selected peak classes. To change a different field, this field needs to be set as display field first using ***Spectra* > *Display settings*** (🖨️). Pressing the <***Change...***> button will display the *Color* dialog box, in which the color of the selected peak classes can be specified.

The peak class properties only apply to the current peak class type. If the selected peak classes are present in other peak class types, they will remain unchanged in those types. If the peak class is copied or moved after adjusting the properties, the peak class will keep the adjusted properties.



The field *Position* is calculated from the spectra during peak matching and cannot be adjusted by the user.

Clicking ***Spectra* > *Remove selected peak classes*** (✖️) will remove the currently selected peak classes. The user will receive a warning before proceeding.

Spectra* > *Invert peak class selection (🔄) will invert the current selection of peak classes. The peak classes that were selected will be unselected and vice versa. If this function is used with no peak classes selected, it

will select all peak classes.

Selecting *Spectra* > *Smart peak class selection* (🔍) will open the *Modify peak class selection* dialog box (see Figure 5.5.6).

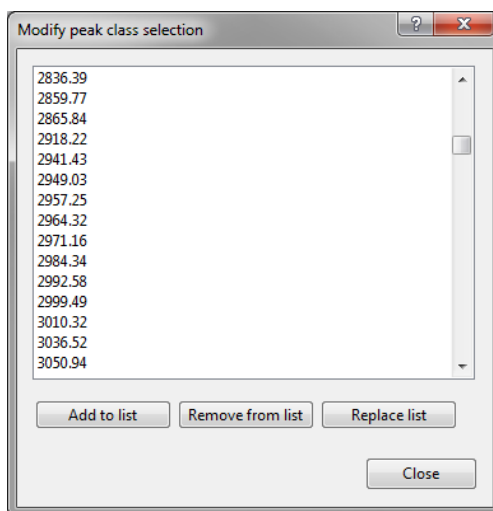


Figure 5.5.6: The *Modify peak class selection* dialog box.

The *Modify peak class selection* dialog box shows a list of all the values present in the peak class display field. For peak classes without value for the display field, the position is shown. The user can select one or more values (using **Shift+click** and **Ctrl+click**). With **<Add to list>** the peak classes meeting the requested criteria will be added to the current selection, with **<Remove from list>** they will be unselected and with **<Replace list>** they will replace the current selection. In this case, the only peaks selected will be those matching the criteria.

For a more elaborate peak selection tool set, we refer to the *Matrix Mining* window (20).

5.5.5 Managing peak class views

Peak class views are a tool to maintain predefined lists of spectrum peak classes (either explicitly enumerated or based on a dynamical query) for specific types of analysis. Similar to object views (see 3.2.2), two types of peak class views can be generated: subset based or query based views.

- Use a **subset based** peak class view for a fixed list of peak classes. The list corresponds to a peak class selection, which could be manually performed by the user or which might be the outcome of a statistical test, e.g. from the matrix mining tool (see 20).
- Use a **query based** peak class view if the information on which to filter is contained in a peak class information field (see 5.5.3) and/or when the list of peak classes is likely to change in the future. If the information in the peak class info field is updated, the list of peak classes returned by the peak class view will be updated as well.

To create a subset based query, first select the peak classes you like to define a subset for in the *Peak Classes* panel of the *Spectrum type* window. Selected peak classes will be highlighted in orange.



Peak class views can also be created from the *Spectrum type* window. To do so, select the peak classes you like to define a subset for in the *Peak Classes* panel. Selected peak classes will be indicated by a check mark in the left-most column. Next, select **Settings** > **Peak class views** > **Manage user defined views...** or choose <**Manage user defined views...**> from the drop-down list in the header of the *Peak Classes* panel. This action opens the *Manage peak class views* dialog box.

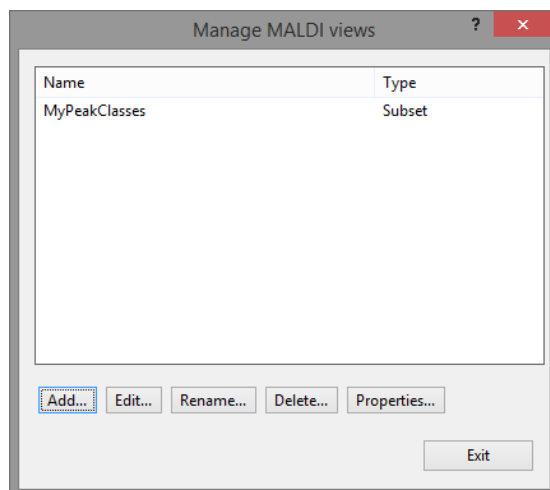


Figure 5.5.7: The *Manage peak class views* dialog box.

The list in the upper part of the dialog box shows all currently defined peak class views (if any), with their Name and Type (the latter will be either Subset or Query).

Press the <**Add**> button to create a new peak class view. The *New character view* dialog box pops up (see Figure 5.5.8).

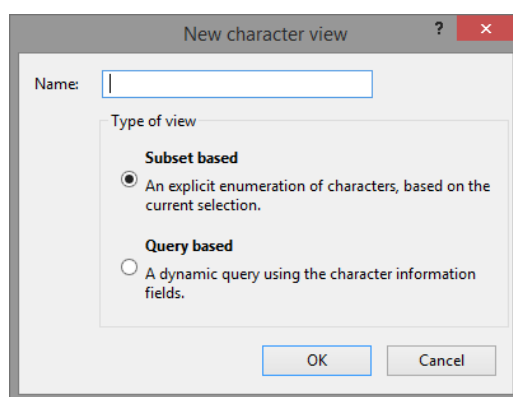


Figure 5.5.8: The *New character view* dialog box.

The dialog prompts you to enter a **Name** for the new peak class view. It also allows to select the **Type of view** to create: either **Subset based** or **Query based**.

For a subset based peak class view, simply enter a name (e.g. “MySubSet”) and press <**OK**>. The *New character view* dialog box can then be closed.



Two minor differences exist between subset based *peak class* views and the more general concept of subset based *object* views:

- Once created, subset based peak class views cannot be edited anymore by adding or removing selected peak classes.
- The peak class order in a subset based peak class view cannot be specified explicitly in the view and is instead governed by the global ordering of the peak classes in the spectrum type.

To create a query based view, select **Spectra > Manage views...** (🔍) in the *Comparison* window. This action opens the *Manage peak class views* dialog box (see Figure 5.5.7).

Enter a name for the peak class view (e.g. “MyQuery”), check the **Query based** option and press <OK>. The *Query view editor* dialog box will open. This dialog allows you to create a query on the peak class names and information stored in any of the peak class information fields (see 5.5.3) that are defined for the spectrum type. The functionality of this dialog box is described in detail in 3.2.2.

Pressing <OK> in the *Query view editor* dialog box will close this dialog and aspect (see 13.2.5) of the spectrum type in the *Experiments* panel will automatically switch to the newly created peak class view.

In the *Manage peak class views* dialog box (see Figure 5.5.7), existing peak class views can be managed. Following commands all work on the highlighted peak class view in the list:

- Query based views can be modified with <Edit>. This action will call the *Query view editor* dialog box again. Note that subset based views cannot be edited; they should be deleted and created again with an updated peak class selection.
- Pressing <Rename> will show the *Rename character view* dialog box, in which a new name for the peak class view can be entered.
- A peak class view can be deleted by pressing the <Delete> button. The software will ask for confirmation before actually deleting the view.
- The object access properties for the peak class view can be edited by pressing the <Properties> button. This action will open the *Object access* dialog box, as discussed in 3.2.3.

5.5.6 Creating a peak matching table

Creating a peak matching table for spectra is identical to creating a band matching table for fingerprints (see 4.3.10). A *composite dataset* that includes only the spectrum experiment type is required. This dataset will contain the character data derived from the peak matching. Inside the *Comparison* window, the composite dataset is synchronized with the current peak class view. If a different peak class view is chosen, the composite dataset will be changed accordingly.

5.5.7 Finding discriminative peaks between entries

Finding discriminative peaks between entries can be done based on the character data derived from a peak matching. This character data can be accessed with a *composite dataset*. Finding discriminative peaks is performed identical to finding discriminative bands (see 4.3.11).

Part 6

Character types

Chapter 6.1

Setting up character type experiments

6.1.1 Defining a new character type

To create a new character type, highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** (+). In the *Create a new experiment type* dialog box, click on **Character type** and press <OK>. This will display the first step of the *New character type* wizard (see Figure 6.1.1).



To be able to work with character type experiments, the Character data module (CH) needs to be present in your BioNumerics configuration.

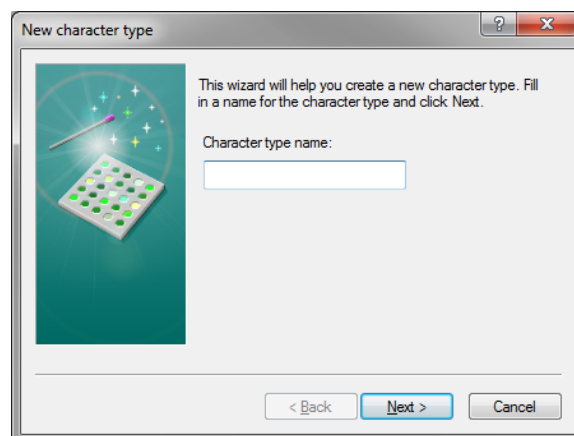


Figure 6.1.1: The first step of the *New character type* wizard.

The wizard prompts you to enter a **Character type name**. Enter a name for the new character type and press <Next> to continue to the next step (see Figure 6.1.2).

In this step of the wizard, the type of dataset should be specified:

- **Binary data:** Check **Binary data** if the output of the tests can only take two possible values.
- **Numerical values:** Check **Numerical values** if the tests can differ in intensity. Specify the number of decimal digits in the input box. If you only want to use integer values enter zero.

Pressing <Next> will display the third step of the *New character type* wizard (see Figure 6.1.3).

In this step, the wizard asks if the character type has an open or closed character set:

- In a *closed character set*, the same number of characters are present for all entries studied. This is the case with commercially available test kits.

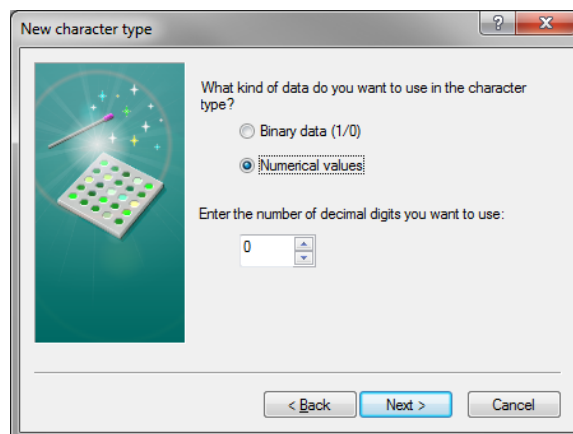


Figure 6.1.2: Step 2 of the *New character type* wizard.

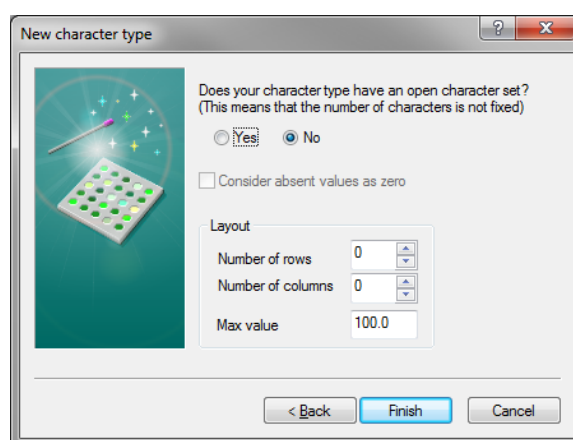


Figure 6.1.3: Step 3 of the *New character type* wizard.

- In an *open character set*, the number of characters is not defined. For example, studying 10 bacterial strains by means of fatty acids can result in a total of 20 fatty acids found, but if some more strains are added, more fatty acids may become present in the list. In such cases, **Consider absent values as zero** should also be checked, because if a fatty acid is not found in a strain it will not be listed in its fatty acid profile, and thus should be considered as zero.

Press <**Finish**> to complete the creation of the new character type, which will appear in the *Experiment types* panel.

6.1.2 Editing a character type

6.1.2.1 The Character type window

All settings that are relevant for a certain character type experiment can be accessed through its *Character type* window (see Figure 6.1.4). This window can be called from the *Main* window by clicking on the character type experiment in the *Experiment types* panel and selecting **Edit > Open highlighted object...** (🖱️, **Enter**) or simply by double-clicking on the character type experiment.

This window consists of five dockable panels:

- The *Characters* panel is the main panel, from which characters can be added or removed (6.1.2.3),

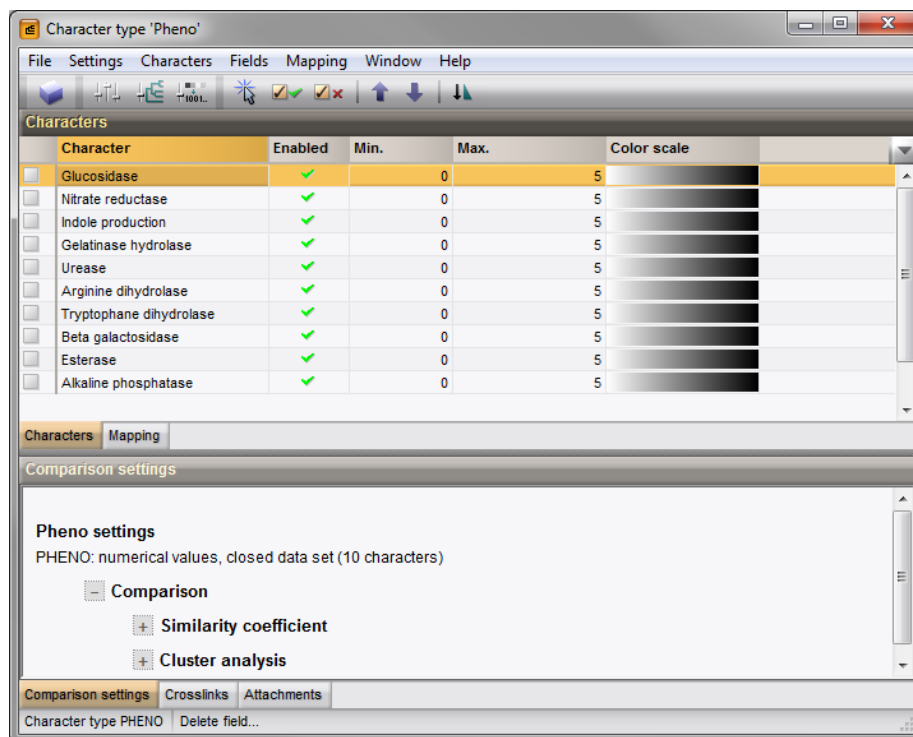


Figure 6.1.4: The *Character type* window with 10 characters defined.

enabled or disabled (6.1.2.4) and rearranged (6.1.2.5). It is also the panel where additional information fields can be added to the character type (6.1.2.9) and where character ranges and color scales can be edited (6.1.2.6).

- In the *Mapping* panel, character mappings can be defined (see 6.1.2.7).
- The character comparison settings (see 6.1.2.8) are listed in the *Comparison settings* panel.
- Cross links (see 3.2.15) from the character type experiment to other database objects can be created from the *Crosslinks* panel.
- Attachments (see 3.2.13) can be added to the character type experiment from the *Attachments* panel.

6.1.2.2 General character type settings

Via *Settings > General settings...* (⚙️), the *Character settings* dialog box is called (see Figure 6.1.5).

Check **Binary data** if the output of the tests can only take two possible values.

Check **Numbers** if the tests can take more than two possible values.

Specify the **Number of decimal digits** in the input box. If you only want to use integer values enter zero.

In case the characters can vary depending on the sample studied (a so-called *open character set*), check the option **Character set grows dynamically**.

Check **Consider absent values as zero** if an absent value should be considered as zero. Unchecking this option will consider an absent value as a missing value ("empty").

Character data can be stored in the relational database (see 21.1) in two different formats: as individual character values or as a single character vector per character experiment. While the former storage format is a little more flexible (i.e., individual values can be queried), the latter type of storage is actually a lot

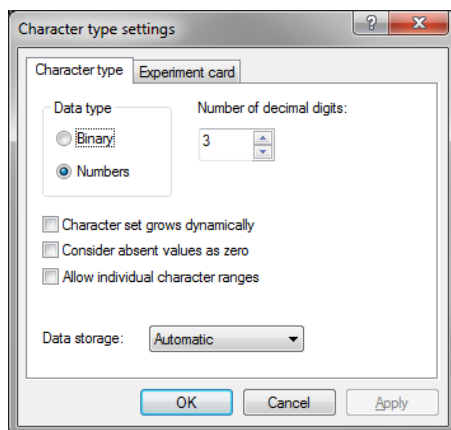


Figure 6.1.5: The *Character settings* dialog box, *Character type* tab.

faster. The **Data storage** drop-down list lets you control the storage type by selecting either “Automatic”, “Individual characters” or “Character vectors”. The default option “Automatic” determines the optimal storage type automatically based on the number of characters in the character type.

When the option **Allow individual character ranges** is disabled, the character ranges and color scales of all characters are changed when choosing **Characters > Change character range...** and **Characters > Change character color scale...** respectively. When this option is enabled, only the character ranges and color scales of the selected characters are updated.

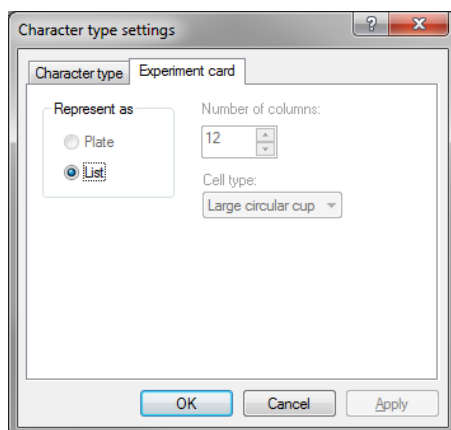


Figure 6.1.6: The *Character settings* dialog box, *Experiment card* tab.

The *Experiment card* tab lets you define some visual attributes of the experiment. These settings apply to the *Experiment card* window, which is explained in 6.1.4.

With **Represent as Plate** and **Represent as List**, you can choose whether the individual tests are shown graphically on a panel, using colors, or as a list of characters with their name and intensities as a numerical value. In case of an open character set (**Character set grows dynamically** checked), only the list type can be chosen.

With **Plate** checked, the **Number of columns** in the test panel can be entered. For example, in case of micro titer plate test kits, one would enter 96 tests and 12 columns.

In order to represent existing commercial kits as truthfully as possible, the **Cell type** drop-down list lets you choose between three different circular cup types and elliptical cups. For blots and microarrays, you can choose between small blot, large micro array spots and small micro array spots.

With **Settings > Binary conversion settings...** (100%), you can specify a binary cutoff value in percent in the

Conversion to binary data (character) dialog box (see Figure 6.1.7).

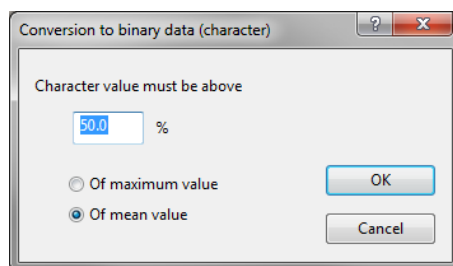


Figure 6.1.7: The *Conversion to binary data (character)* dialog box.

Whenever converting the numbers to binary states, BioNumerics will consider all values above the cutoff value as positive and those below the cutoff value as negative. If you have entered 50% as cutoff value, you can choose the cutoff level to be 50% of the maximum value found in the experiment, or 50% of the average value from the experiment.

6.1.2.3 Adding and removing characters

When a new character type is created (see 6.1.1), it will still be empty because the program does not yet know which, and how many tests it contains.

To manually add a character, select **Characters > Add new character...** (+). This calls the *Add new character* dialog box (see Figure 6.1.8).

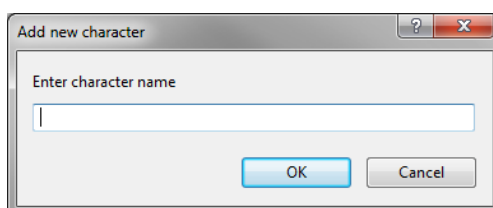


Figure 6.1.8: The *Add new character* dialog box, to add a new character to the experiment.

Enter a name and press <OK>.

The character is now listed in the *Characters* panel.

A quick method to add a complete array of characters at a time, for example a micro plate array, is **Characters > Add array of characters...**

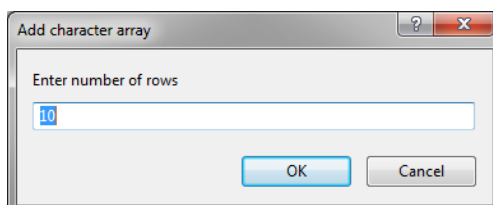


Figure 6.1.9: Specify the number of rows.

The program subsequently asks to enter the number of rows, the number of columns, and the maximum values for the tests. The program automatically assigns names to the tests: A1, A2, A3, ..., B1, B2, B3, ... (see Figure 6.1.12 for an example). These names can be changed into the real test or substrate names afterwards with **Characters > Rename highlighted character...**

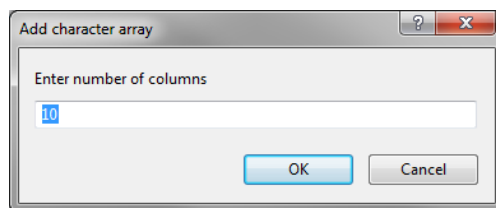


Figure 6.1.10: Specify the number of columns.

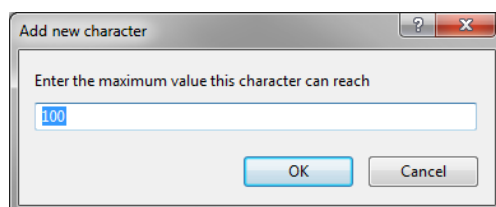


Figure 6.1.11: Specify the maximum value for this character.

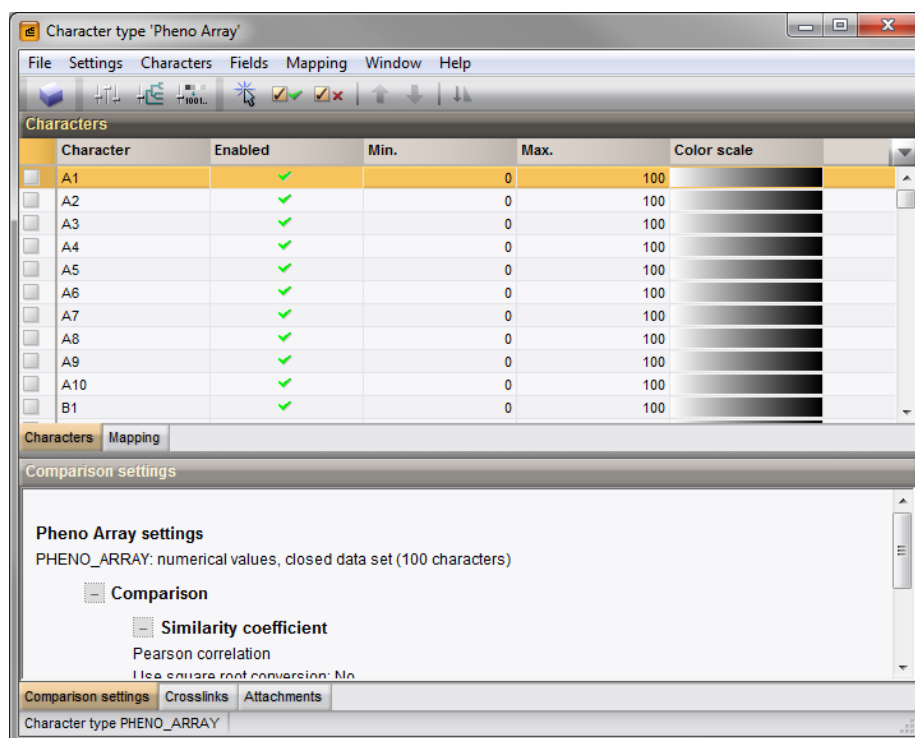


Figure 6.1.12: The *Character type* window.

Characters (and character values) can also be imported from external files using the import functions in BioNumerics. See [6.1.3](#) and [6.1.2.10](#) for more information.

Use **Characters > Rename highlighted character...** if you want to give a character a different name. This opens the *Rename character* dialog box (see [Figure 6.1.13](#)).

Enter the new character name in the text box. Pressing **<OK>** saves the changes to the database.

To delete the selected character(s) from the list, select **Characters > Remove selected characters...** (✖). Note that this action will remove the associated character data from the database. A less drastic way to exclude a character from an analysis is to disable it (see [6.1.2.4](#)).

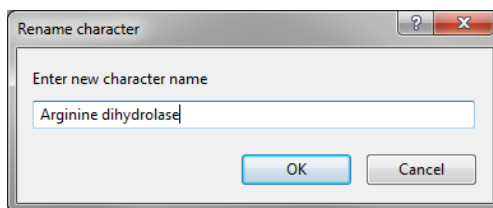


Figure 6.1.13: The *Rename character* dialog box, to rename the selected character.

6.1.2.4 Enabling and disabling characters

Selections in the *Character type* window are indicated with check boxes and provide a convenient tool to enable/disable a group of characters at a time. Characters can be selected or unselected in a similar way as any database object using the **Ctrl-** and **Shift-keys** (see 3.2.4).

Each character that is added to the list of characters in the *Character type* window is default marked with a ✓ sign in the **Enabled** column, which means that it is used in comparisons and identifications (see Figure 6.1.14).

To disable selected characters in the *Character type* window, use **Characters > Disable all selected characters** (✗).

Disabled characters are displayed in gray italic and do not have a ✓ sign in the **Enabled** column (see Figure 6.1.14). Disabling a character might be useful for a blank or control test which is often present in commercial identification kits.

Use **Characters > Enable all selected characters** (✓) to enable all selected characters in the *Character type* window.

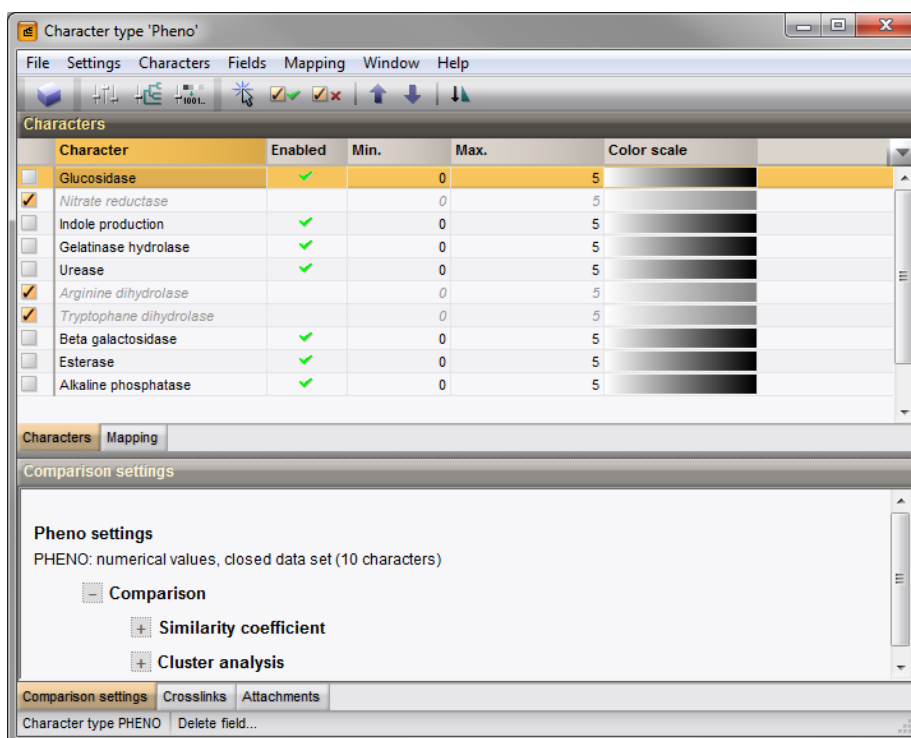


Figure 6.1.14: *Character type* window.

6.1.2.5 Rearranging characters

In the *Character type* window, individual characters can be moved up or down. The characters will be displayed in the same order in the *Comparison* window. To achieve this, highlight the character and use **Characters > Move highlighted character up** (↑, Ctrl+Up) or **Characters > Move highlighted character down** (↓, Ctrl+Down).

Characters can be sorted according to any of the character information fields. The *Arrange characters by field* dialog box is called with **Characters > Arrange characters by field...** (↕) (see Figure 6.1.15).

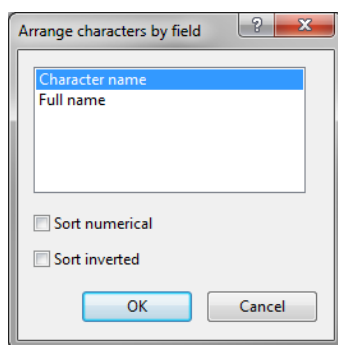


Figure 6.1.15: The *Arrange characters by field* dialog box, to arrange characters by a selected field.

The characters can be sorted according to any of the character information fields listed in the dialog.

When a field contains numerical values, which you want to sort according to increasing number, check **Sort numerical**.

Checking **Sort inverted** will sort the characters in reverse order.

6.1.2.6 Character ranges and color scales

A *color scale* can be defined for all characters, which makes it possible to assess character values at a glance in e.g. the *Comparison* window. The color scale is defined based on a *character range*.

If you want characters to cover another range, select **Characters > Change character range...**. This calls the *Change character range* dialog box (see Figure 6.1.16).

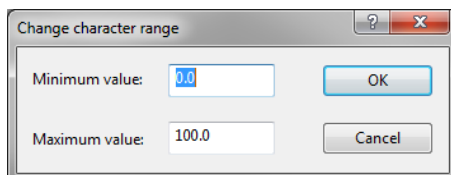


Figure 6.1.16: The *Change character range* dialog box, to change the character range of a selected character.

A **Minimum value** and **Maximum value** can be entered to define the character range. If the option **Allow individual character ranges** is disabled in the *Character settings* dialog box (see Figure 6.1.5), the character ranges of all characters are changed to the new values when pressing <OK>. When this option is enabled, only the character ranges of the selected characters are updated.

The default color scale for each character ranges from white (negative) to black (most positive). To change the color scale, select **Characters > Change character color scale...**. This pops up the *Edit color scale* dialog box (see Figure 6.1.17).

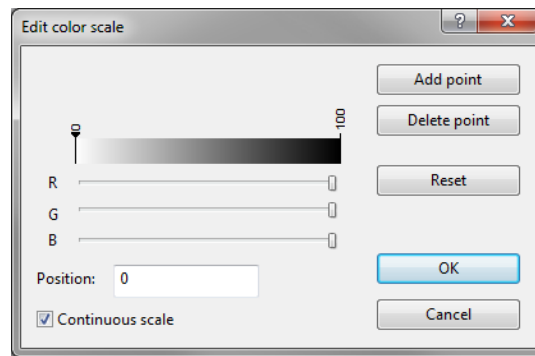


Figure 6.1.17: The *Edit color scale* dialog box.

The **Minimum value** and **Maximum value** of the character range is shown on top of the color scale. The default color scale for all characters ranges from white (minimum value) to black (maximum value).

Initially only two points (marked with a pipe |) are defined on the color scale, corresponding to the minimum and maximum value of the range.

A black triangle marks the point for which the color can be edited. Another point can be selected simply by clicking on it.

The three slider bars below the color range preview represent red, green, and blue, respectively. By adjusting the sliders any desired color can be obtained for the selected point.

If more transition colors are required, use the **<Add point>** button. A new mark appears and can be selected by clicking on it. You can drag the mark to the left or to the right, and adjust its color using the sliders. A selected point can be removed again with **<Delete>**.

The position of the selected mark is shown in the **Position** text box and is automatically updated when dragging the mark to another place on the scale bar. The position can also be entered in the **Position** text box, updating the position of the mark on the scale instantly.

Standard, the color scale is represented as a continuous scale (**Continuous scale** checked). Unchecking the option **Continuous scale** displays a discrete color bar.

With **<Reset>**, the default color range is applied to the character: a linear gradient from white (minimum value in the character range) to black (maximum value).

If the option **Allow individual character ranges** is disabled in the *Character settings* dialog box (see Figure 6.1.5), the color scales of all characters are changed to the new values when pressing **<OK>**. When this option is enabled, only the color scales of the selected characters are updated.

6.1.2.7 Character mappings

In BioNumerics, character values can be mapped to categorical names according to predefined criteria. To be able to work with character mappings, the *Mapping* panel should be displayed in the *Character type* window by clicking the corresponding tab (see Figure 6.1.18).

To add a criterion, select **Mapping > Add new mapping...** This pops up the *Edit character map* dialog box (see Figure 6.1.19).

A **Name** can be entered for the mapping, i.e. the text that will be displayed when the character value falls within the specified mapping range. The latter is defined via a **Range start** and a **Range end**.

As many mappings as needed can be added with the same procedure. An existing mapping can be edited with **Mapping > Edit mapping...**, which will call the *Edit character map* dialog box again. An obsolete

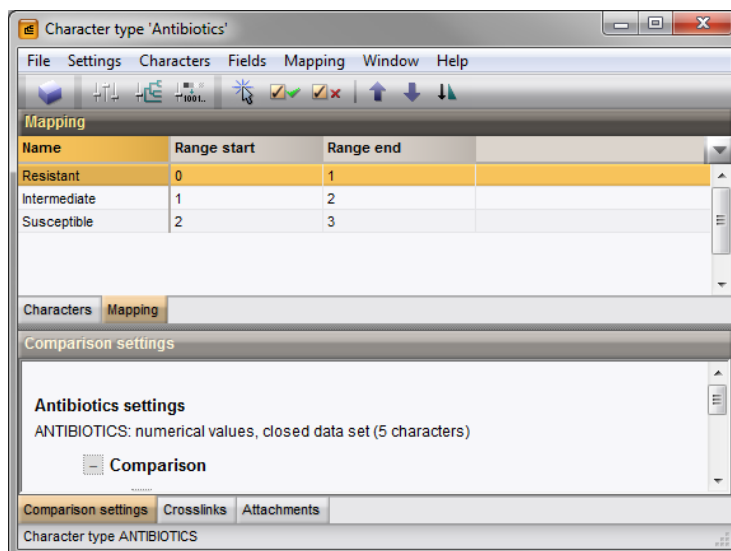


Figure 6.1.18: The *Mapping* panel within the *Character type* window, showing three predefined criteria.

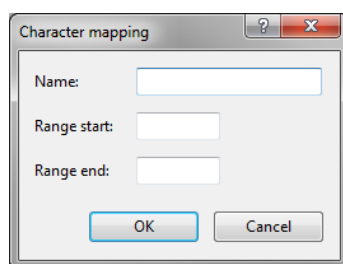


Figure 6.1.19: The *Edit character map* dialog box.

mapping can be removed with **Mapping > Delete mapping...**

The mapped names can be displayed in the *Experiment card* window (see 6.1.4), the *Comparison* window (see 6.2.3), the *Pairwise comparison* window (see 13.3.3), the *Charts and statistics* window (see 14) and in various reports. In case a character value does not fall within the range of the defined mappings, or if no mappings are defined, a "<?>" will be displayed.

When character mappings are present, it becomes possible to define a custom mappings similarity matrix, which determines how similarities are calculated among the mappings. The *Mappings similarity matrix* dialog box can be displayed by selecting **Mapping > Edit mapping similarity matrix...** (see Figure 6.1.20).

This dialog shows all defined character mappings in a matrix format. The default similarity value is "1" or 100% match for self-matches (on the diagonal) and "0" for matches between different mappings. With other words, the default corresponds to a normal categorical matching. All values in the matrix, except for the ones on the diagonal, can be edited. Any value that deviates from the default "0" will be highlighted in green (see Figure 6.1.20).

Pressing <Save> or <OK> will save the mappings similarity matrix to the database. Pressing <Reset to default> will restore the default matrix as described above.

The mappings similarity matrix will be used when similarities are calculated with the *Categorical (mappings)* coefficient in the *Comparison* window (see 6.2.2).

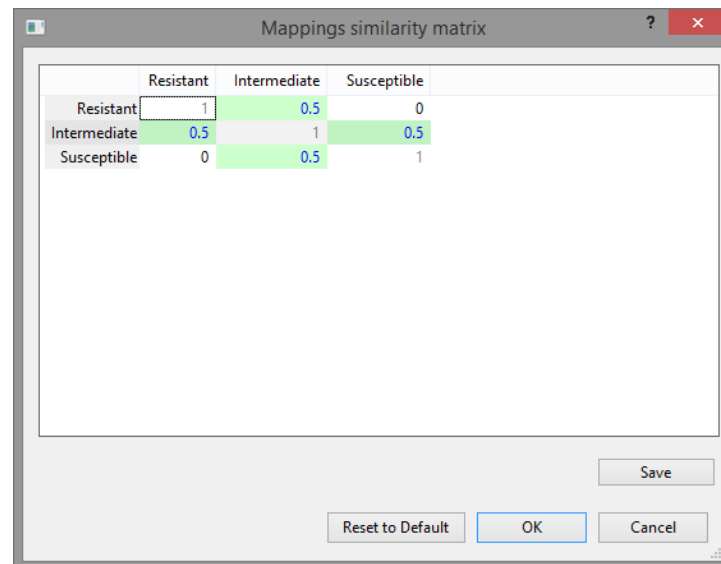


Figure 6.1.20: The *Mappings similarity matrix* dialog box, showing a customized mappings similarity matrix for an antibiotics resistance character experiment.

6.1.2.8 Character comparison settings

In the *Character type* window, the comparison settings defined for the character type are shown in *Comparison settings* panel (see Figure 6.1.14). These settings can be accessed with **Settings > Comparison settings...** (🔧) in the *Character type* window, but also in the *Comparison* window. See 6.2 for a detailed explanation.

6.1.2.9 Character type information fields

In the *Character type* window, additional information fields can be added with **Fields > Add new field...**. This opens the *Add new field* dialog box.

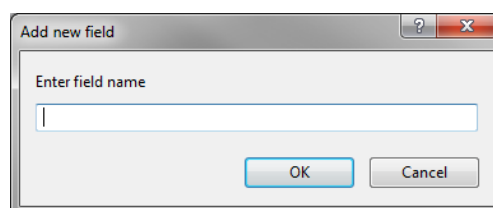


Figure 6.1.21: The *Add new field* dialog box to add a new character field.

Enter the field name in the text box. Press <OK> saves the new field to the database.

A non-default information field is removed from the database with **Fields > Delete field...**

A non-default information field can be renamed with **Fields > Rename field...**. This calls the *Rename field* dialog box.

Specify the new name of the selected character field. Pressing <OK> saves the changes to the database.

Information can be entered for a given character by clicking twice on a field, or by clicking on a field and selecting **Fields > Set field content for selected characters...** (Ctrl+M).

By default, the 'Character' field is displayed in a comparison. You can choose to display another field by clicking on the header of the desired field and selecting **Fields > Use as default field**.

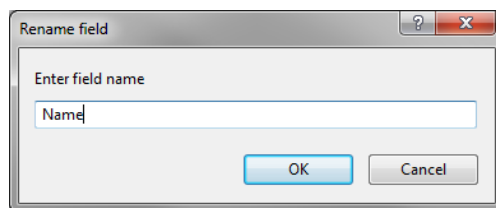


Figure 6.1.22: The *Rename field* dialog box to change the name of the character field.

The column used as default field is highlighted in a pale green color (see Figure 6.1.23).

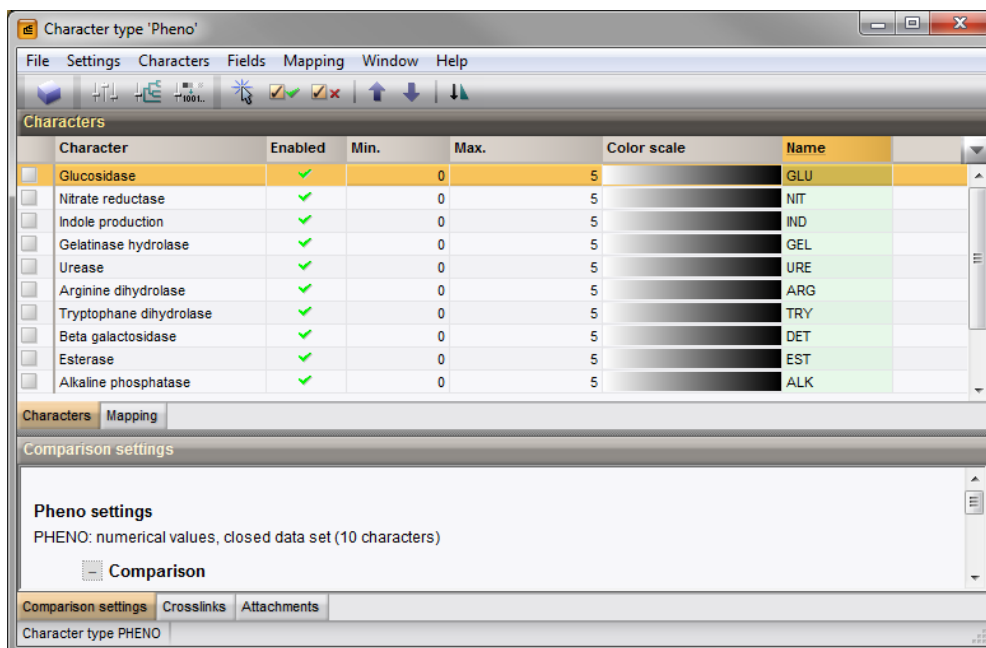


Figure 6.1.23: The *Character type* window with the information field 'Name' set as default field.

6.1.2.10 Importing character information from a text file

Characters and character information, stored in a text file, can be imported in the database using the command **Characters > Import character information**, which can be launched from the *Character type* window.

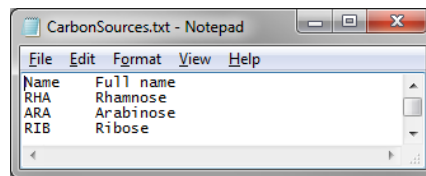
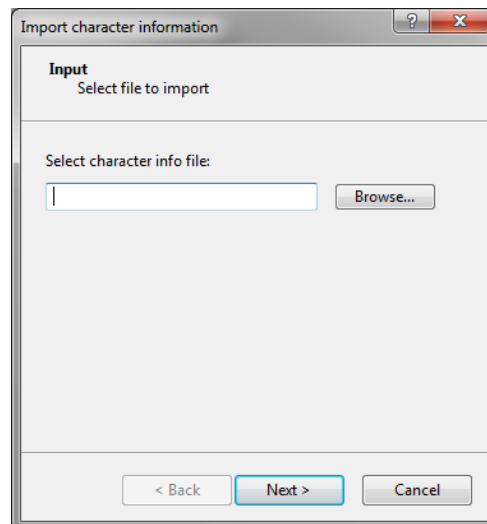
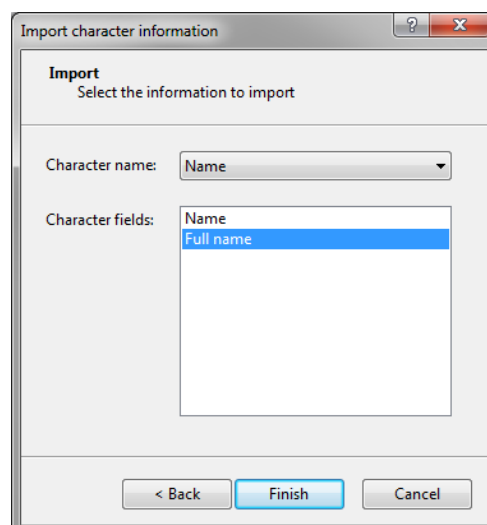
Double-clicking on character type in the *Experiment types* panel in the *Main* window opens the *Character type* window.

The text file should contain a well-defined table with the header of the table containing the character information field names. There should be no extra rows or columns besides the table (see Figure 6.1.24 for an example).

Selecting **Characters > Import character information** calls the *Input* dialog box (see Figure 6.1.25).

This dialog prompts for the text file: pressing the **<Browse>** button allows you to select the file that you want to import, located on your computer, external drive or on a network location.

Browse for the file and press **<OK>**. The next step is displayed (see Figure 6.1.26). If the import routine is unable to open the selected file an error is generated.

**Figure 6.1.24:** Example**Figure 6.1.25:** The *Input* dialog box.**Figure 6.1.26:** The *Import* dialog box.

All column names detected in the selected text file are listed in the **Character name** drop-down list and in the **Character fields** panel.

- **Character name:** Select the column in the text file that holds the unique character names. During import BioNumerics checks if the characters are already present in **Character** field in the *Character type* window, if not the new characters are created.
- **Character fields:** Select the columns that hold the character information you want to import in the database. During import BioNumerics checks if the character information fields are already present

in the *Character type* window, if not the fields are created. To select multiple fields use the **Ctrl-** or **Shift-keys**.

When pressing **<Finish>** a new dialog appears if new character(s) and character field(s) are to be created. Confirm the import action.

In case the **Data type** is set to **Numbers**, all newly created characters will have a default range of 0.0 - 100.0. When working with **Binary** data, the range will be 0.0 - 1.0.

6.1.2.11 Creating and managing character views

Character views are a tool to maintain predefined lists of characters (either explicitly enumerated or based on a dynamical query) for specific types of analysis. Functionally, character views are very similar to object views in BioNumerics (see 3.2.2). Once defined, character views can be quickly selected in the *Comparison* window to perform a specific analysis (i.e. cluster analysis, statistical tests) on the *aspect* of the character experiments they define (see 13.2.5).

Some examples of scenarios where character views prove extremely useful:

- In whole genome MLST (wgMLST), to cluster strains based on sets of loci belonging to different subschemas. See the separate *WGS tools plugin* manual for more information.
- In plant breeding experiment, analyses on large SNP data sets can be limited to e.g. sets of markers associated with a certain phenotypic trait or markers that are residing on a certain chromosome, etc..

Similar to object views, two types of character views can be generated: subset based or query based views.

- Use a **subset based** character view for a fixed list of characters. The list corresponds to a character selection, which could be manually performed by the user or which might be the outcome of a statistical test, e.g. from the matrix mining tool (see 20).
- Use a **query based** character view if the information on which to filter is contained in a character type information field (see 6.1.2.9) and/or when the list of characters is likely to change in the future. If the information in the character type info field is updated, the list of characters returned by the character view will be updated as well.

To create a subset based query, first select the characters you like to define a subset for (see 6.1.2.4 on how to make character selections). Selected characters will be indicated by a check mark in the left-most column.

Next, select **Characters > Character Views > Manage user defined views...** or choose **<Manage user defined views...>** from the drop-down list in the header of the *Characters* panel. This action opens the *Manage character views* dialog box (see Figure 6.1.27).

The list in the upper part of the dialog box shows all currently defined views on the character type (if any), with their Name and Type (the latter will be either Subset or Query).

Press the **<Add>** button to create a new character view. The *New character view* dialog box pops up (see Figure 6.1.28).

The dialog prompts you to enter a **Name** for the new character view. It also allows to select the **Type of view** to create: either **Subset based** or **Query based**.

For a subset based character view, simply enter a name (e.g. "MySubSet") and press **<OK>**. The *New character view* dialog box will close and the *Characters* panel automatically switches to the newly created character view. The *Manage character views* dialog box can then be closed.

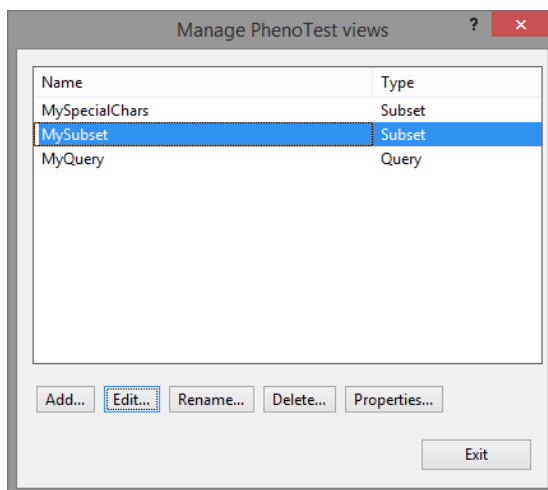


Figure 6.1.27: The *Manage character views* dialog box.

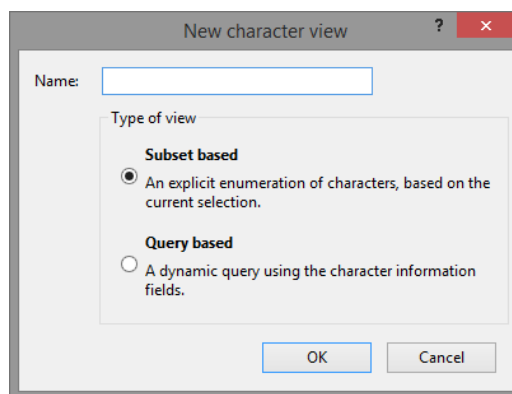


Figure 6.1.28: The *New character view* dialog box.



Two minor differences exist between subset based *character* views and the more general concept of subset based *object* views:

- Once created, subset based character views cannot be edited anymore by adding or removing selected characters.
- The character order in a subset based character view cannot be specified explicitly in the view and is instead governed by the global ordering of the characters in the character type.

To create a query based view, select **Characters > Character Views > Manage user defined views...** or choose **<Manage user defined views...>** from the drop-down list in the header of the *Characters* panel to open the *Manage character views* dialog box again (see Figure 6.1.27).

Enter a name for the character view (e.g. “MyQuery”), check the **Query based** option and press **<OK>**. The *Query view editor* dialog box will open. This dialog allows you to create a query on the character names and information stored in any of the character type information fields (see 6.1.2.9) that are defined for the character type. The functionality of this dialog box is described in detail in 3.2.2.

Pressing **<OK>** in the *Query view editor* dialog box will close this dialog and the *Characters* panel will automatically switch to the newly created character view.

In the *Manage character views* dialog box (see Figure 6.1.27), existing character views can be managed. Following commands all work on the highlighted character view in the list:

- Query based views can be modified with **<Edit>**. This action will call the *Query view editor* dialog box again. Note that subset based views cannot be edited; they should be deleted and created again with an updated character selection.
- Pressing **<Rename>** will show the *Rename character view* dialog box, in which a new name for the character view can be entered.
- A character view can be deleted by pressing the **<Delete>** button. The software will ask for confirmation before actually deleting the view.
- The object access properties for the character view can be edited by pressing the **<Properties>** button. This action will open the *Object access* dialog box, as discussed in 3.2.3.

6.1.3 Importing character data

6.1.3.1 Import options for character data

There are three possibilities for importing or entering data into a character type:

1. Importing character data and database information from text files, Excel files or from an ODBC-compatible data source. Two formats are supported: a table or grid-like format (see 6.1.3.2) and a database style format (see 6.1.3.3).
2. Entering the data via the *Experiment card* window of the database entry (see 6.1.4).
3. Processing and quantification of images scanned as TIFF files (see 6.1.3.4).

6.1.3.2 Importing fields and characters

6.1.3.2.1 Importing fields and characters from a text file

With the *Import fields and characters (text file)* option, listed under the topic *Character type data* in the *Import* dialog box (see Figure 6.1.29), character information and optionally entry information can be imported from text files in the database and linked to new or existing database entries.

Each file should contain a well-defined table with rows corresponding to entries and columns corresponding to characters and entry information fields. The header of the table should contain the character names and the entry information field names (see Figure 6.1.30 for an example). There should be no extra rows or columns besides the table.

6.1.3.2.2 Importing fields and characters from an Excel file

With the *Import fields and characters (Excel file)* option, listed under the topic *Character type data* in the *Import* dialog box (see Figure 6.1.29), character information and optionally entry information can be imported from Excel files in the database and linked to new or existing database entries.

The Excel file should contain a well-defined table with rows corresponding to entries and columns corresponding to characters and entry information fields. The header of the table should contain the character names and the entry information field names (see Figure 6.1.31 for an example). All information or a subset of information present in a particular sheet can be imported.

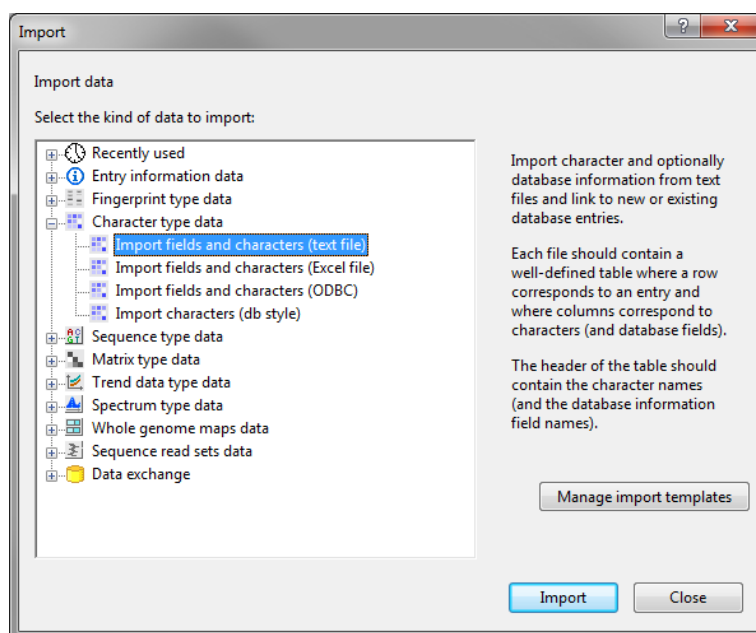


Figure 6.1.29: The *Import fields and characters (text file)* option in the *Import* dialog box.

Strain ID	Genus	Species	Type	Origin	Source	RHA	NAG	RIB	INO	SAC
CL001	Escherichia	coli	O157:H7	Austin	Human	0	0	100	0	0
CL002	Escherichia	coli	O157:H7	Austin	Meat	0	0	0	0	0
CL003	Escherichia	coli	O157:H7	Houston	Human	0	0	0	0	0
CL004	Escherichia	coli	O157:H7	Dallas	Human	0	0	0	0	0
CL005	Escherichia	coli	O157:H7	San Antonio	Human	0	0	100	0	0
CL006	Escherichia	coli	O157:H7	El Paso	Meat	0	0	100	0	0
CL007	Escherichia	coli	O157:H7	Houston	Human	0	0	100	0	0
CL008	Escherichia	coli	O157:H7	Dallas	Human	0	0	100	0	0
CL009	Escherichia	coli	O157:H7	Galveston	Human	0	0	100	0	0
CL010	Escherichia	coli	O157:H7	El Paso	Meat	0	0	100	0	0
CL011	Escherichia	coli	O157:H7	Lubbock	Human	0	0	100	0	0
CL012	Escherichia	coli	O157:H7	Abilene	Meat	0	0	100	0	0

Figure 6.1.30: Import fields and characters from a text file.

6.1.3.2.3 Importing fields and characters from an ODBC-compatible data source

With the *Import fields and characters (ODBC)* option, listed under the topic *Character type data* in the *Import* dialog box (see Figure 6.1.29), character information and optionally entry information can be imported from ODBC-compatible files in the database and linked to new or existing database entries.

The file should contain a well-defined table with rows corresponding to entries and columns corresponding to characters and entry information fields. The header of the table should contain the character names and the entry information field names (see Figure 6.1.32 for an example).

6.1.3.2.4 The Import wizard

Selecting *Import fields and characters (text file)*, *Import fields and characters (Excel file)*, or *Import fields and characters (ODBC)* under *Character type data* in the *Import* dialog box and pressing <Import> starts the import wizard.



Before character information can be imported in the database using these import options, a character type experiment must be defined in the database (see 6.1.1).

If the *Import fields and characters (text file)* option was selected in the *Import* dialog box, the first step of

Strain ID	Genus	Species	Type	Origin	Source	Blank	a-Cyclodextrin	Dextrin	Glycogen	Tween 40	Tween 60
CL001	Escherichia	coli	O157:H7	Austin	Human	9	8	12	13	17	15
CL002	Escherichia	coli	O157:H7	Austin	Meat	7	2	15	9	8	13
CL003	Escherichia	coli	O157:H7	Houston	Human	3	10	20	19	18	23
CL004	Escherichia	coli	O157:H7	Dallas	Human	6	9	25	15	8	17
CL005	Escherichia	coli	O157:H7	San Antonio	Human	5	8	19	11	21	19
CL006	Escherichia	coli	O157:H7	El Paso	Meat	11	4	13	4	11	14
CL007	Escherichia	coli	O157:H7	Houston	Human	10	6	15	12	14	21
CL008	Escherichia	coli	O157:H7	Dallas	Human	7	4	20	5	9	13
CL009	Escherichia	coli	O157:H7	Galveston	Human	8	10	14	7	20	16
CL010	Escherichia	coli	O157:H7	El Paso	Meat	10	5	27	21	15	23
CL011	Escherichia	coli	O157:H7	Lubbock	Human	9	0	13	3	12	16
CL012	Escherichia	coli	O157:H7	Abilene	Meat	10	3	18	15	5	14

Figure 6.1.31: Import fields and characters from an Excel file.

Strain ID	Genus	Species	Type	Origin	Source	Blank	a-Cyclodextrin	Dextrin	Glycogen	Tween 40	Tween 60
CL001	Escherichia	coli	O157:H7	Austin	Human	9	8	12	13	17	15
CL002	Escherichia	coli	O157:H7	Austin	Meat	7	2	15	9	8	13
CL003	Escherichia	coli	O157:H7	Houston	Human	3	10	20	19	18	23
CL004	Escherichia	coli	O157:H7	Dallas	Human	6	9	25	15	8	17
CL005	Escherichia	coli	O157:H7	San Antonio	Human	5	8	19	11	21	19
CL006	Escherichia	coli	O157:H7	El Paso	Meat	11	4	13	4	11	14
CL007	Escherichia	coli	O157:H7	Houston	Human	10	6	15	12	14	21
CL008	Escherichia	coli	O157:H7	Dallas	Human	7	4	20	5	9	13
CL009	Escherichia	coli	O157:H7	Galveston	Human	8	10	14	7	20	16
CL010	Escherichia	coli	O157:H7	El Paso	Meat	10	5	27	21	15	23
CL011	Escherichia	coli	O157:H7	Lubbock	Human	9	0	13	3	12	16
CL012	Escherichia	coli	O157:H7	Abilene	Meat	10	3	18	15	5	14

Figure 6.1.32: Import fields and characters (ODBC).

the wizard prompts for the text file (see Figure 6.1.33). Pressing the **<Browse>** button allows you to select the file that you want to import, located on your computer, external drive or on a network location. Three different text file separators are currently supported: "TAB", "Comma", and "Semicolon".

If the **Import fields and characters (Excel file)** option was selected in the **Import** dialog box, the first step of the wizard prompts for the Excel file and the table name (see Figure 6.1.34):

- Pressing the **<Browse>** button allows you to select the file that you want to import, located on your computer, external drive or on a network location.
- All information present in a particular sheet can be imported by selecting the name of the sheet from the **Data range** drop-down list. If a range of information has been saved in the Excel file and has been assigned a name (i.e. a so-called *named range*), the name of this selection can also be picked from the **Data range** list. If a named range is selected, the import action will only import the information that

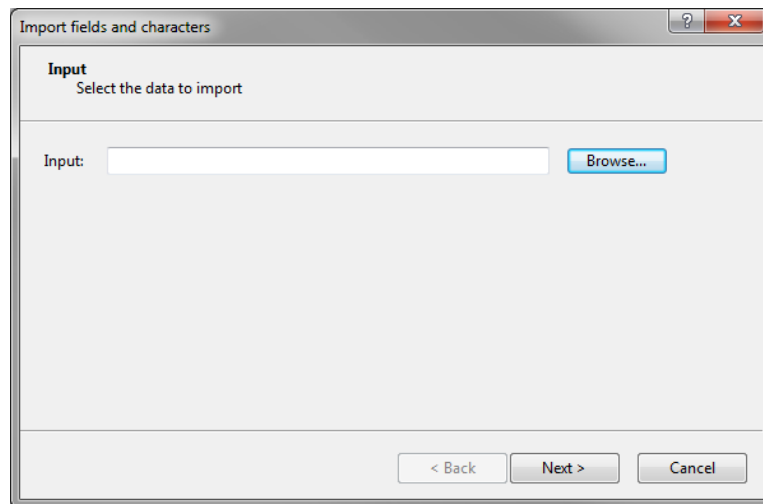


Figure 6.1.33: The *Input* wizard page.

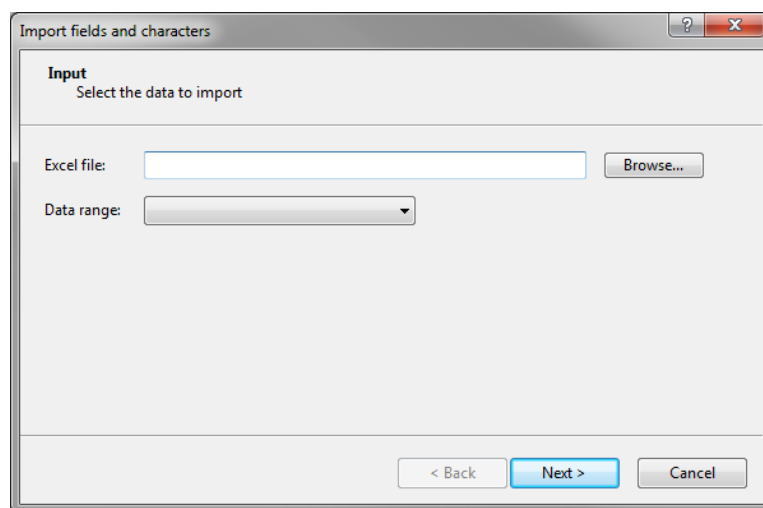


Figure 6.1.34: The *Input* wizard page.

is present in the selection of the named range.

If the **Import fields and characters (ODBC)** option was selected in the *Import* dialog box, the first step of the wizard prompts for the ODBC connection string (i.e. the string that defines the external database) and the table name (see Figure 6.1.35):

- Pressing the **<Build>** button allows you to create the ODBC connection string. The dialog box that pops up is generated by your Windows operating system and may differ depending on the Windows version installed. Select the correct data source from the list (e.g. **MS Access Database** to import information from an Access database). If the data source is not listed, create a new data source. Navigate to the correct path and select the "database" which can be located on your computer, external drive or on a network location. The ODBC string is updated in the **Connection string** input box.
- All information present in a particular **Database table** can be imported by selecting the table name from the drop-down list.

Only when all settings have correctly been specified in the first step of the wizard, pressing **<Next>** will display the next step.

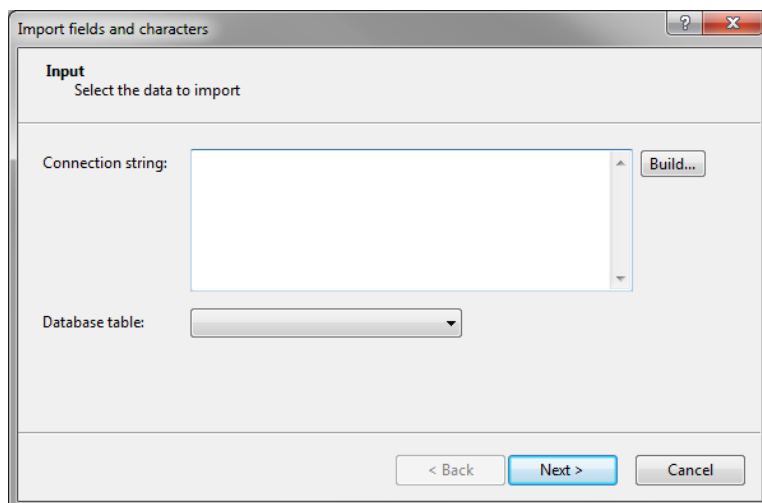


Figure 6.1.35: The *Input* wizard page.

When importing character data for the first time in the database, the *Import rules* dialog box will open (see Figure 6.1.37), otherwise the *Import template* wizard page (see Figure 6.1.36) will open.

If the import routine is unable to open the selected file, an error is generated.

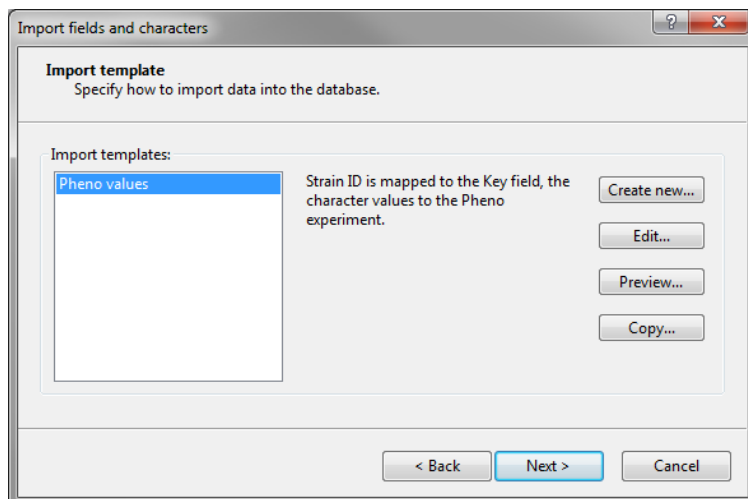


Figure 6.1.36: The *Import template* wizard page.

The way the character and entry information should be imported in the database can be specified with an import template. The *Import templates panel* lists all import fields templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up a new dialog box, allowing you to define a new import template (see Figure 6.1.37).

Each column in the selected file corresponds to a row in the grid (column 1 in the file corresponds to row 1 in the grid, column 2 corresponds to row 2, etc.). The text **File field** is specified in the **Source type** column and the column names are displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields, character type experiments, or character type information fields. Initially the rows are not linked to any information in the database (the **Destination type** and **Destination** for all rows is set to **<None>**).

Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>**

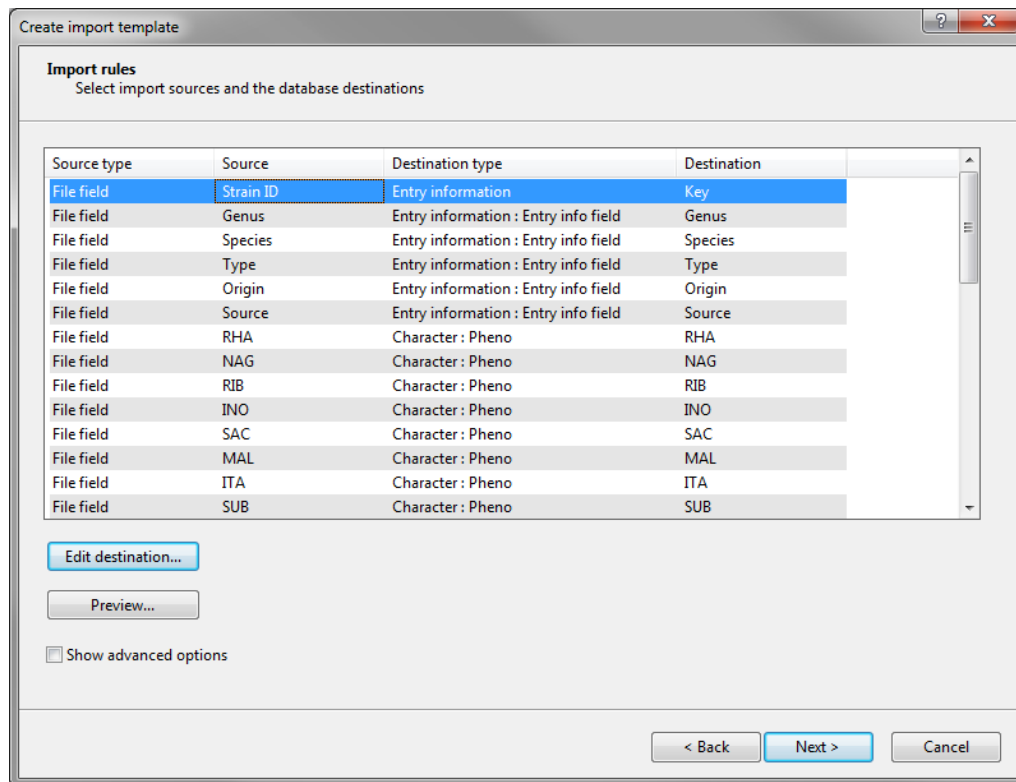


Figure 6.1.37: The *Import rules* dialog box.

button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

When only one row is selected in the grid, the information of this row can be linked to (see Figure 6.1.38):

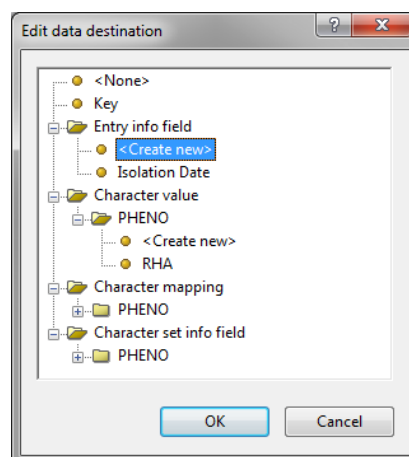


Figure 6.1.38: Edit data destination for a single selected row entry.

- The default information field **Key**.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A value of a new or existing character of an existing character type experiment (select the **<Create new>** option or an existing field under the topic **Character value**, respectively).
- A mapped value of a new or existing character of an existing character type experiment (select the **<Create new>** option or an existing field under the topic **Character mapping**, respectively). In order to import the mapped values correctly, the mapping rules need to be defined in the experiment type before the import. This can be done in the *Character type* window and is described in detail in 6.1.2.7.
- A new or existing character type information field (select **<Create new>** or select an existing field under the topic **Character set info field**, respectively).

If a row is linked to a new entry information field, a new character or a new character type information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

When multiple rows are selected in the grid, the information of these rows can be linked to (see Figure 6.1.39):

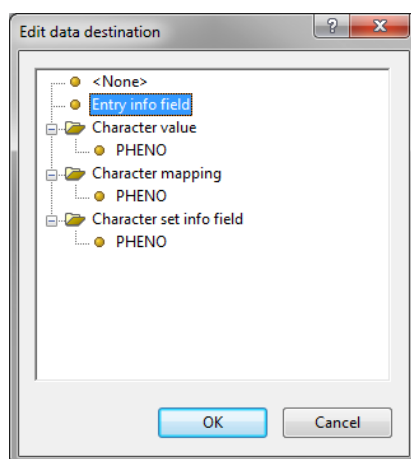


Figure 6.1.39: Edit data destination for multiple selected row entries.

- Non-default entry information fields (select the **Entry info field** option).
- Values of characters of a character type experiment (select the character type experiment under the topic **Character value**).
- Mapped values of characters of a character type experiment (select the character type experiment under the topic **Character mapping**). In order to import the mapped values correctly, the mapping rules need to be defined in the character type experiment before the import. This can be done in the *Character type* window and is described in detail in 6.1.2.7.
- Character information fields of a character type experiment (select the character type experiment under the topic **Character set info field**).

When pressing the **<OK>** button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields, characters, or character information fields in the database. If no entry information fields, characters, or character information fields exist with the same name, a new dialog box pops up prompting for the names.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. Rows linked to characters of a character type experiment hold the name of the character type in the **Destination type** column; the name of the character is displayed in the **Destination** column. When rows are linked to character type information fields, the text **Character set info field**, followed by the name of the character type experiment, is displayed in the **Destination type** column; the name of the character information field is listed in the **Destination** column.

Pressing <**Preview**> opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the <**Close**> button.

When the <**Show advanced options**> check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the <**Cancel**> button cancels the operation and the template settings are not saved to the database.

Pressing the <**Next**> button calls a new dialog where the entry link field needs to be defined.

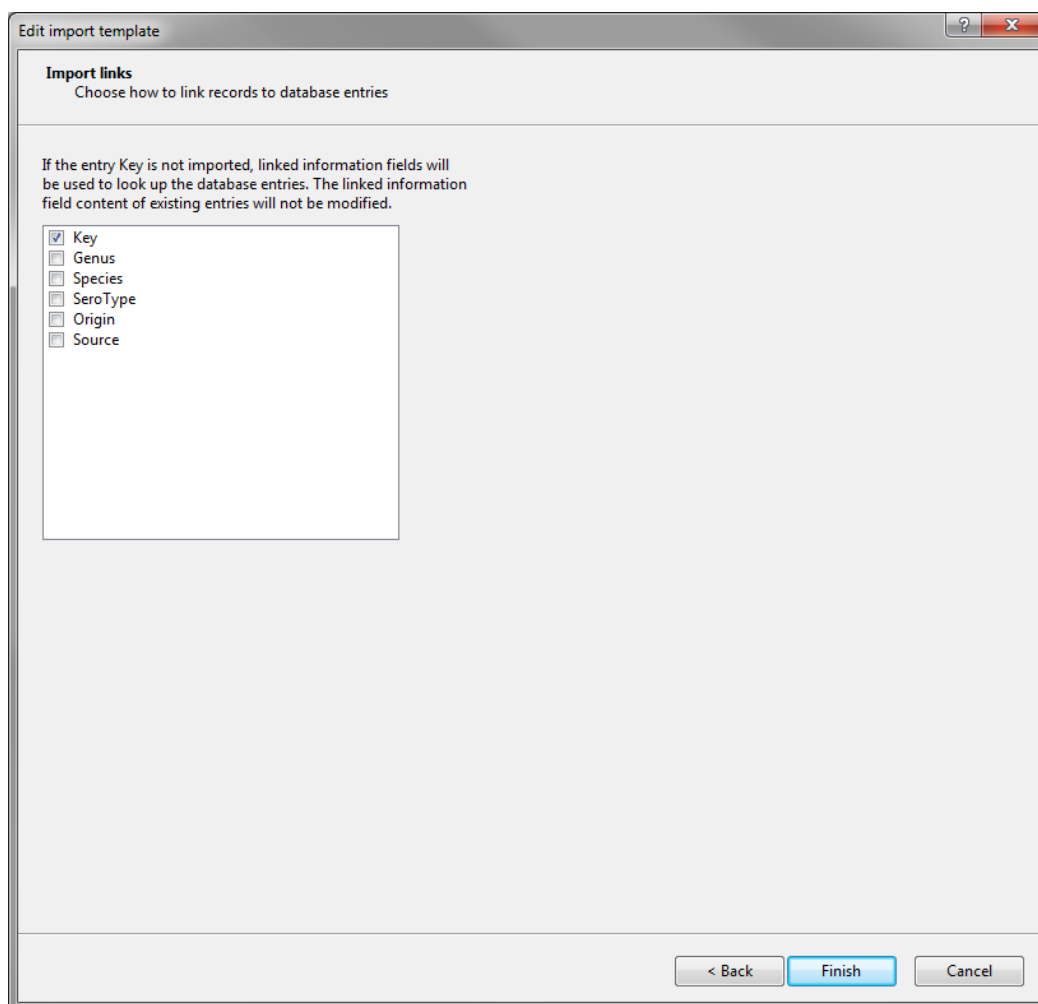


Figure 6.1.40: Specify the entry link field.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.

- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

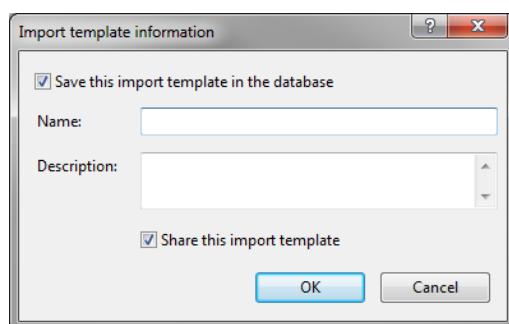


Figure 6.1.41: The *Import template information* dialog box.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

Pressing **<Next>** opens the last step of the wizard, prompting for some final settings.

- When **Create x entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update x entries** if you want the software to be able to update the entry and character information for existing entries.

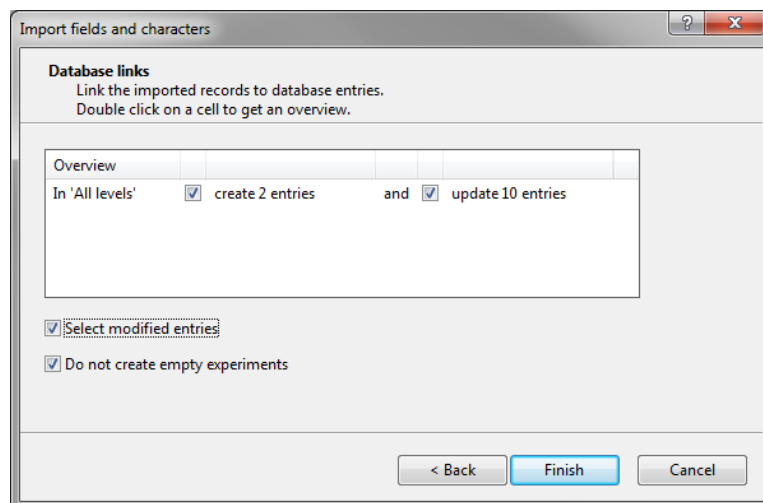


Figure 6.1.42: The *Database links* wizard page.

- If the option *Select modified entries* is checked, entries in the database that were modified during the import routine will be selected after import.

When *Do not create empty experiments* is enabled, the import routine will not create experiment links for entries that are present in the selected file for which no character values are present.

When mapped *Key* information exceeds the maximum number of allowed characters (i.e. 60 characters), the *Create x entries* option will have a red background. The entries with a *Key* exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing *<Finish>* will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.



Mapped character field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

6.1.3.3 Importing characters (db style)

6.1.3.3.1 Introduction

With the *Import characters (db style)* option, listed under the topic *Character type data* in the *Import* dialog box (see Figure 6.1.43), character information stored in a text, Excel or other ODBC-compatible file can be imported in the database and linked to new or existing database entries.

This import option assumes that the character values are placed in a single column. A second column should hold the character names, and a third column should hold the key information (see Figure 6.1.44).

6.1.3.3.2 The Import wizard

Selecting *Import characters (db style)* under *Character type data* in the *Import* dialog box and pressing *<Import>* calls the *Import characters (db style)* dialog box (see Figure 6.1.45).

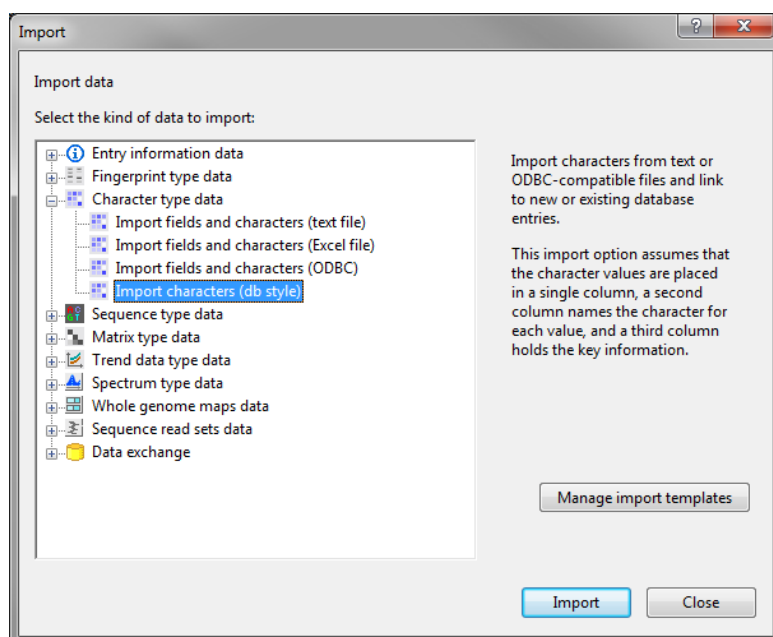


Figure 6.1.43: Import characters (db style).

Figure 6.1.44: Database-style format: (a) text file; (b) Excel file.



Before character information can be imported in the database using this import option, a character type experiment must be defined in the database. To create a new character type highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** (+). In the *Create a new experiment type* dialog box, click on **Character type**.

Characters and character values can be imported from text files (check **Text file**) or from ODBC-compatible files via an ODBC link (check **Database (ODBC link)**).

If the **Text file** option is selected, the first step of the wizard prompts for the text file and the separator (see Figure 6.1.45):

- Pressing the <**Browse**> button allows you to select the file that you want to import, located on your computer, external drive or on a network location.
- Three different text file separators are currently supported and can be selected from the **Separator** drop-down list: "TAB", "Comma", "Semicolon". The separator that corresponds to the selected file should be picked from the list.

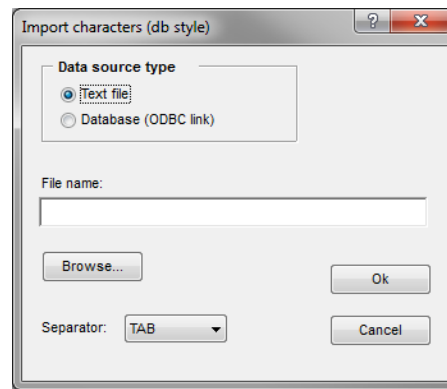


Figure 6.1.45: Import characters from a text file.

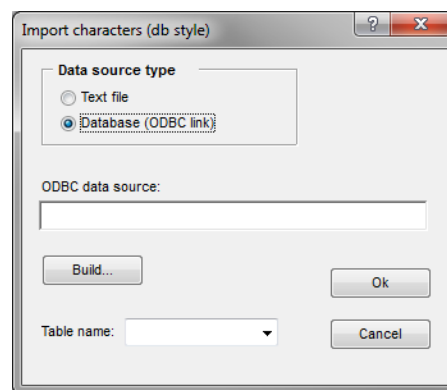


Figure 6.1.46: Import characters from an ODBC compatible file.

If the **Database (ODBC link)** option is selected, the first step of the wizard prompts for the ODBC connection string (i.e. the string that defines the external database) and the table name (see Figure 6.1.46):

- Pressing the **<Build>** button allows you to create the ODBC connection string. The dialog box that pops up is generated by your Windows operating system and may differ depending on the Windows version installed. Select the *Data source tab* and select the correct data source from the list (e.g. **Excel files** in case you want to import information from an Excel file). If the data source is not listed, create a new data source. Navigate to the correct path and select the file which can be located on your computer, external drive or on a network location. After having selected the correct file, the ODBC string is updated in the **ODBC data source** input box.
- When importing information from a spreadsheet program (e.g. Microsoft Excel), all information present in a particular sheet can be imported by specifying the name of the sheet in the **Table name** edit box, followed by a dollar sign (e.g. Sheet1\$). If a range of information has been saved in the file and has been assigned a name (i.e. a so-called *named range*), the name of this selection can also be specified in the **Table name** edit box. If a named range is specified, the import action will only import the information that is present in the selection of the named range.

If all settings have correctly been specified in the first step of the wizard, pressing the **<OK>** button will open a new window (see Figure 6.1.47). If the import plugin is unable to open the selected file, an error is generated.

The window prompts for the columns holding the key information (**Key field**), character names (**Character field**), and character values (**Value field**). The character type experiment that will hold the character values can be chosen from the **Character type** pull-down list.

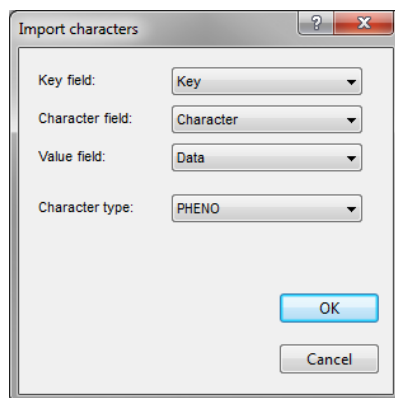


Figure 6.1.47: The *Import characters* dialog box.

Pressing <OK> starts the import of the characters and character values. If during import one or more characters need to be created for the selected character type experiment, a dialog box pops up, prompting for the maximum value of the character(s).

6.1.3.4 Import of character data by quantification of images

6.1.3.4.1 Introduction

Similar as for gel images, BioNumerics can import *character type data* from TIFF images. This happens by quantification of the color intensity and/or color transitions on the TIFF file. Character data from phenotypic test panels often provide color transitions rather than changes in intensity. For example, many test panels have reactions that change from yellow to red, or blue - green - yellow, and hence, quantifying the colors by their intensity would provide no meaningful information. Rather, the program needs to be able to read the files as true RGB images and allow the possibility to define negative colors and positive colors, as well as transition colors. For example, in an acidification reaction with a bromophenol blue dye, non-reactive tests will be blue, whereas weak reactions will show the transition color green, and strongly positive reactions will show up yellow.

Using the same tool, BioNumerics also allows the import of micro-array and gene chip images scanned as TIFF files, offering for each gene or oligonucleotide a quantitative reaction value.

The character import tool is provided as a separate program, BNIMA, that can be started from within BioNumerics. BNIMA only works when the BioNumerics analyze program is running.

6.1.3.4.2 Example 1: import of micro titer plate image

The first example we will use to illustrate the program is Plate1.TIF. This TIFF file is available on the download page of the website (<http://www.applied-maths.com/download/sample-data>, click on "Microplate image"). It is a photograph of a 96 wells micro titer plate with bromophenol blue as reaction indicator dye (see Figure 6.1.48).

- 3.1 Create a new *closed* character type and call it **Microplate**.
- 3.2 Specify a color range from blue to yellow, over green for all characters (see 6.1.2).
- 3.3 Add an array of characters (see 6.1.2.3) and specify 8 rows and 12 columns.
- 3.4 Double-click on an entry to show its *Entry* window.

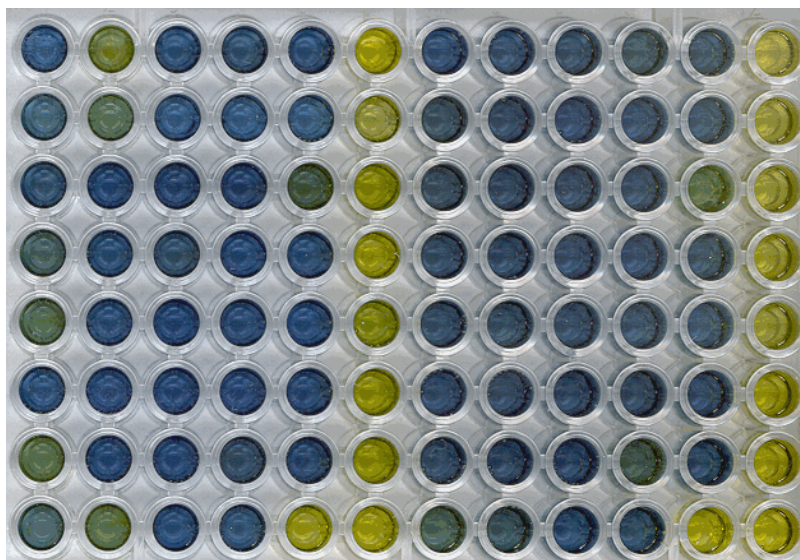


Figure 6.1.48: 96 wells micro titer plate with bromophenol blue as reaction indicator.

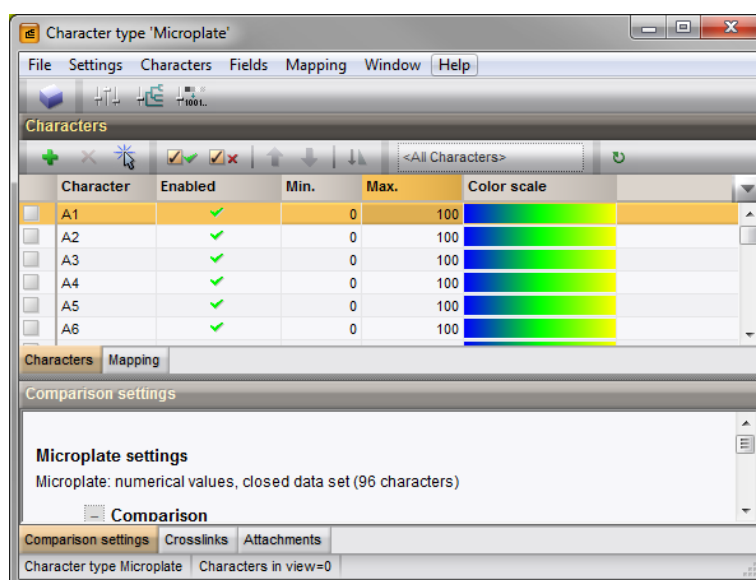


Figure 6.1.49: The character type **Micro plate**.

The experiment type **Micro plate** shows an empty flask.


3.5 Click on the flask button. Since this experiment is not defined for the selected entry, the program asks "Do you want to create a new one?".

3.6 Answer <Yes> to create an *Experiment card* window (see 6.1.4).

An empty micro plate image pops up.

3.7 Right-click on the empty micro plate image and select **Edit image** from the floating menu.

This loads the BNIMA program.

3.8 Select **File > Load image** (or press ) in BNIMA and load the file Plate1.TIF from the downloaded and unzipped folder from the website.

The resulting window looks as in Figure 6.1.50.

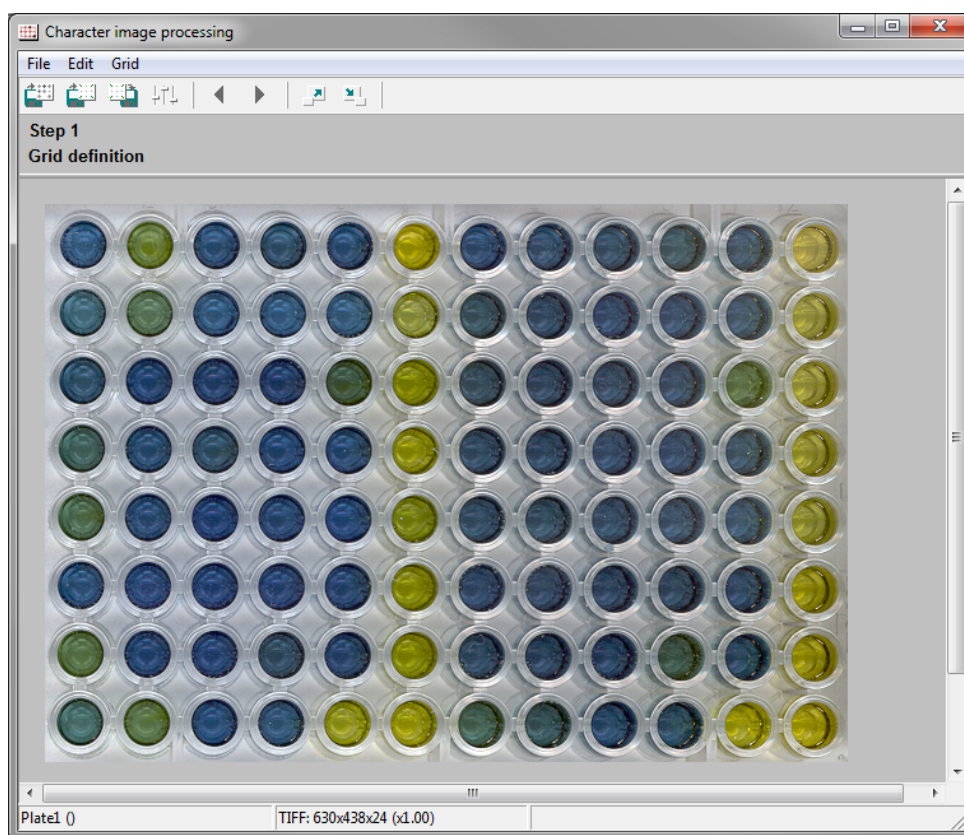



Figure 6.1.50: The BNIMA program with a micro plate image loaded.

3.9 First call the *Settings dialog box* with **Edit > Settings** or .

The *Image tab* offers two choices for the **Image type**: **Densitometric** and **Color scale**.

In case the color reaction can be interpreted as a simple change in intensity (e.g. from light to dark), one should select **Densitometric**. The *Densitometric values panel* offers some additional tools to edit the TIFF file: **Inverted values** is to invert the densitometric values; **Background subtraction** allows a two-dimensional subtraction of the background from the TIFF file, using the rolling ball principle. The **Ball size** can be entered in pixels. **Spot removal** allows all spots and irregularities below a certain size to be removed from the image, whereas larger structures are preserved. The background subtraction and spot removal changes are only seen when **Edit > Show value scale** is enabled in the *Character image processing window*.

In case the reaction causes a change from one color to another color, as in the above example, **Color scale** is the right option. An additional feature, **Hue only**, is particularly useful when the scanned images differ in brightness (illumination) or contrast. If the images do not contain black or white in their color range, it is better to enable this feature.

3.10 Select **Color scale** and **Hue only**.

3.11 Press <OK> to proceed with these settings.

The other settings will be discussed later.

Like the process of normalization of gels, processing a character panel image exists of a number of steps: (1) Grid definition; (2) Cell layout; and (3) Quantification.

In step 1 (Grid definition) we will create a grid that defines the wells of the micro plate.

3.12 Select **Grid > Add new** and enter "8" as **Number of rows** and "12" as **Number of columns**.

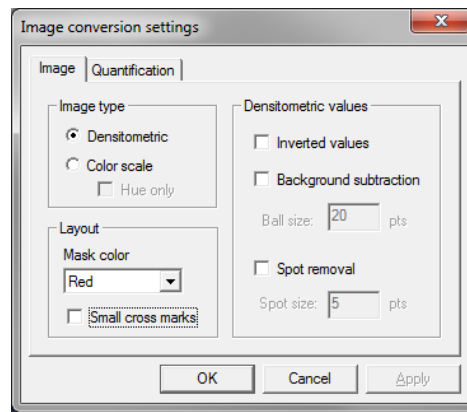


Figure 6.1.51: Image tab.

3.13 Press **<OK>**.

The grid appears. At each edge of the grid, there is a dragging node (green square). The upper left dragging node is to *move* the grid as a whole; the lower right node is to *resize* the grid, and the upper right and lower left nodes are to *distort* the grid in case the image is not perfectly rectangular or not scanned horizontally.


3.14 Drag the nodes until the grid matches with all 96 wells.



Using the **Shift**-key, one can distort the grid locally if needed. The size of the local distortion area is indicated by a circle. It is even possible to reduce or enlarge the size of the distortion area as follows: hold the **Shift**-key and left-click on any cell-marking cross of the grid. The area defining circle appears. Hold the left mouse button down and release the **Shift**-key: the circle is still visible. While holding the left mouse button down, press the **PageDown** or **PageUp** key to reduce or enlarge the area of distortion. The size of the circle will decrease or increase. Using a very small circle, it is possible to correct the grid in any individual cell.

3.15 In case you want to remove the grid and define it again, select one of the cells of the grid (the cell becomes red) and **Grid > Delete**.

In case the image consists of two or more subsets of cells (e.g. some more complex test panels or micro-arrays), it is possible to define more than one grid using the **Grid > Add new** command.

3.16 Move to the next step using **Edit > Next step** or the  button.

In this step, the layout of the cells is defined: the shape and size of the quantification area within each cell. In this step, we also define which cells we want to use for quantification and which cells not. By default, all cells of the grid are used for quantification.

3.17 Click in the upper left corner of the image and while holding the left mouse button down, select the left half of the test panel.

All the selected cells are marked in red.

3.18 Select **Cells > Delete selected**.

The left half of the panel will not be used for quantification, and hence, cannot be used in the resulting character set.

Cross marks of unused cells are smaller than of used cells.

3.19 Select the non-used cells again with **Cells > Add selected**.

Before the program can do the quantification, it needs to know what the averaging area of the cells is. This is done using a *mask* which the user defines. One can define the same mask for all cells, or assign particular

masks to individual (groups of) cells. In the case of a micro plate it is obvious that all cells should have the same mask.

- 3.20 Click in the upper left corner of the image and, while holding the left mouse button down, select all cells in the test panel.

All cells are marked in red.

- 3.21 Add a circular mask to all selected cells with **Cells > Add disk to mask**.

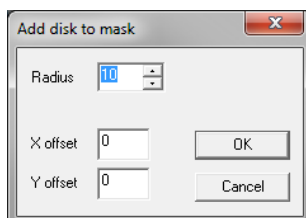



Figure 6.1.52: Add disk to mask.

A dialog box prompts to enter a **Radius** for the disk in pixels, the **X offset** (horizontal shift from the cell marking cross) and the **Y offset** (vertical shift from the cell marking cross). For the offsets, a negative value can be entered.

- 3.22 Enter "8" as radius, and leave the offsets zero. Press <OK> to confirm.

The masks appear on all used cells of the grid as semitransparent red disks. In order to see the masks, it is important that they are in a color that is complementary to the reaction colors of the cells. One can change the color of the masks as follows:

- 3.23 Select **Edit > Settings** or .

- 3.24 Under **Layout**, pull down the **Mask color** menu and select the appropriate color.




In the example micro plate, the most appropriate color is red.



In case of very small cells, e.g. micro plate images, you can select **Small cross marks**, so that they don't overlap most of the cells.

It is possible to add more than one mask to the cells. In case the cells have a more complex layout, i.e. not just circular, one can add two or more disks with different offsets to approach the shape of the cells. By selecting individual cells or groups of cells, it is also possible to change the shape of the masks per cell or per group of cells.



Some more advanced features allow the mask of individual cells to be changed manually: With **Cells > Add pixels to mask** or  the cursor changes into a pencil which you can use to add pixels to the masks manually. When doing so, it is recommended to zoom in on the cell using the **Edit > Zoom in** command or . Similarly, it is possible to remove pixels from the mask with **Cells > Remove pixels from mask** or . Clicking a second time on these buttons or selecting the menu item finishes the pixel editing mode.

If you selected the image type to be **Color scale**, and not **Densitometric** in the settings (see Instruction 3.9 to Instruction 3.10), you can now specify the negative color, the positive color, and any transition colors between negative and positive. For each cell, you can define a unique color scale, which can be necessary for some commercial test panels containing more than one reaction dye.

- 3.25 In the example case, there is only one reaction dye, so select all cells as in Instruction 3.17.

- 3.26 Select **Cells > Edit color scale** or .

This brings up the *Color scale editor* as shown in Figure 6.1.53.

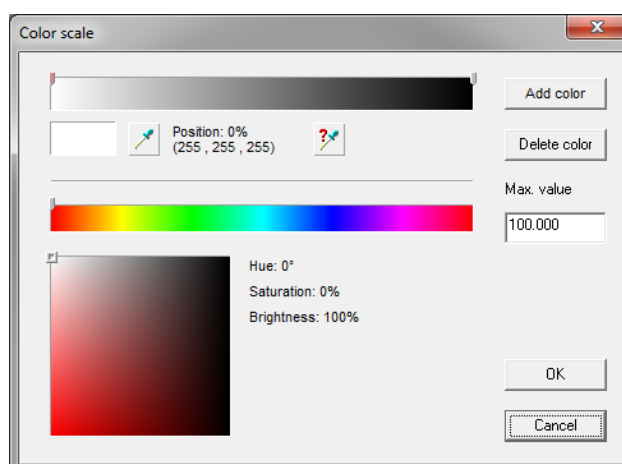


Figure 6.1.53: *Color scale editor* in the Cell layout step of the BNIMA program.

By default, the color scale exists of two colors: white as negative and black as positive. In the case of the example micro titer plate, this scale would obviously not work. Since the scale ranges from blueish (negative) over greenish to yellow (positive), we will add a new intermediate color.

- 3.27 Press <**Add color**>. One new color (gray) is defined in the middle of the scale.
- 3.28 Select the color selector of the negative color (left).
- 3.29 Move the slider on the color scale of predefined colors to blue.
- 3.30 Move the slider in the Saturation/Brightness square to the lower left corner to obtain maximum brightness and saturation.
- 3.31 Repeat Instruction 3.28 to Instruction 3.30 for the intermediate color (middle), assigning green, and for the positive color (right), assigning yellow.





If you selected **Hue only** in the settings (see Instruction 3.9 to Instruction 3.10), changing saturation and brightness has no effect on the obtained color scale. If saturation or brightness transitions within the same color are to be registered, you should disable the **Hue only** feature in the settings.

The upper color scale now should range from blue over green to yellow (Figure 6.1.54).




Figure 6.1.54: Appropriate color scale for the example micro plate image.



One can also pick up colors from the image in order to define the selected color in the upper color scale. To this end, click and hold the left mouse button on the left pipette button . The mouse pointer shape changes into a pipette which you can drag to the most negative cell, e.g. the blank control. The selected color in the color scale automatically changes into the color at the pipette's position. If **Hue only** is enabled, the closest hue color is selected. Once the color scale is defined you can interrogate the reaction of any cell using the right pipette button  as described above. This pipette does not affect the defined color scale, but only shows the position of the pointed cell graphically on the color scale, and the percentage reaction with error indication.

With the **Max. value** field you can enter the maximum value to which all characters will be rescaled.

3.32 Enter 100 as **Max. value** and press <OK> to confirm the color settings.

3.33 Move to the next step using **Edit > Next step** or the  button.

The next and last step involves quantification of the cells. First of all, the cells to be added to the character set need to be defined. In case one or more cells are intended only for calibration purposes, they can be excluded from the resulting character set, but used as calibration marker.

3.34 Select all cells by clicking in the upper left corner of the image and while holding the left mouse button down, select the complete test panel.

3.35 Select **Quantification > Add cells to character set**.

The cells are now numbered from 1 to 96.

3.36 If you click on a particular cell, its quantified value as rescaled according to Instruction 3.32 is given in the status bar as well as the value after calibration (see further).

Quantification is done by integrating the pixels within the defined mask. There are different options for integration:

3.37 Select **Edit > Settings** or  and choose the **Quantification tab**.

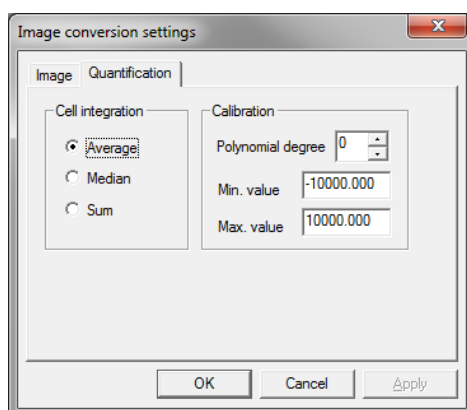


Figure 6.1.55: *Quantification tab*.

Cell integration methods include **Average**, **Median**, and **Sum**. In case the image contains spots that could influence the quantified values, the median option will provide more reliable results than the arithmetic averages.

3.38 Select **Median** integration and press <OK>.

In order to illustrate the calibration feature, we will define one of the cells as negative control (minimum value), and another cell as positive control (maximum value).

3.39 Select cell A1 (negative control) and **Quantification > Define calibration point**. Enter 0 as value and press <OK>.

3.40 Select cell A12 (positive control) and **Quantification > Define calibration point**. Enter "100" as value and press <OK>.

Since only two calibration points are defined now, it is obvious that the program needs to calculate a linear regression through the defined points, in order to re-quantify the other cells according to the negative and positive controls:

3.41 Select **Edit > Settings** or  and choose the **Quantification tab**.

3.42 Under **Calibration**, enter 1 as **Polynomial degree**.


This will result in a first degree regression.

3.43 Press <OK> to close the *Settings dialog box*.


3.44 Select **Quantification** > **View calibration curve**.

This shows a linear regression between the two calibration points, zero and 100.


Finally, there is one more thing to do, i.e. to copy the character values in the micro plate opened in BioNumerics.

3.45 Select **Quantification** > **Export to clipboard** or .

Before closing the BNIMA program, you can save the entire configuration defined for this micro plate system. If you load a next micro plate, you can reload the grid and all other settings such as color scale, disabled cells, quantification parameters etc.

3.46 Select **File** > **Save configuration as** or .

3.47 Enter a name, e.g. "micro plate", and press <OK> to save the configuration.

For next micro plates you can reload the configuration using **File** > **Load configuration** or .

3.48 Close the BNIMA program with **File** > **Exit**.

3.49 Right-click on the *Experiment card* window, and select **Paste from clipboard** from the floating menu.

The micro plate now is filled with data and looks like in Figure 6.1.56.

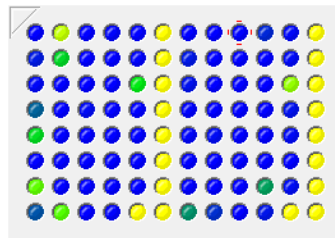


Figure 6.1.56: Example micro plate experiment card after import of character values using BNIMA.

3.50 Click the upper left triangular button to close the experiment card.

3.51 Confirm that you want to save the imported character values.

6.1.3.4.3 Example 2: import of gene-array scanning

The second example we will use to illustrate the BNIMA program is Array.TIF, a fragment of a gene array image, available on the download page of the website (<http://www.applied-maths.com/download/sample-data>, click on "Array image"). The array image was generated by chemiluminescent detection of digoxigenin-labeled cDNA [33]. Each gene is characterized by two spots (horizontally next to each other), which can be considered as a control measure. For this example, we have used a fragment representing two blocks of 14 x 7 genes (the complete array is composed of six blocks of 14x7 genes, totaling 588 characters). The left and right half are separated by one blank column, and the two bottom rows contain calibration and reference spots (see Figure 6.1.57).

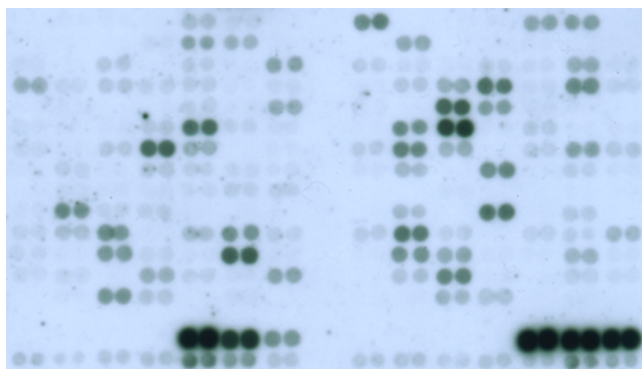


Figure 6.1.57: Fragment of gene array scanned as TIFF image.

3.52 Create a new *closed* character type as described in 6.1.1, and call it **Gene array**.

3.53 Add an array of characters (see 6.1.2.3) and specify 14 rows and 14 columns.

3.54 Select **Settings** > **General settings**, and click the *Experiment card* tab.

3.55 Specify 14 columns and under **Cell type**, select **Small blot**, which makes it possible to show large data sets in the *Experiment card* window (see 6.1.4).

3.56 Click <OK> and close the *Character type* window.

3.57 Double-click on an entry to show its *Entry* window.

The experiment type **Gene array** shows an empty flask.

3.58 Click on the flask button. Since this experiment is not defined for the selected entry, the program asks "Do you want to create a new one?".

3.59 Answer <Yes> to create an *Experiment card* window.


An empty 14 by 14 array image pops up.

3.60 Right-click on the empty array image and select **Edit image** from the floating menu.

This loads the BNIMA program.

3.61 Select **File** > **Load image** (or press  in BNIMA and load the file `Array.TIF` from the downloaded and unzipped folder from the website.

The resulting window looks as in Figure 6.1.58.

3.62 First call the *Settings dialog box* with **Edit** > **Settings** or .

The *Image tab* offers two choices for the **Image type**: **Densitometric** and **Color scale**.

Unlike the first micro plate image, the color reaction of this gene array can be interpreted as a simple change in intensity (e.g. from light to dark), hence one should select **Densitometric**.

3.63 Select **Densitometric** under **Image type**.

The *Densitometric values panel* offers some additional tools to edit the TIFF file: **Inverted values** is to invert the densitometric values; **Background subtraction** allows a two-dimensional subtraction of the background from the TIFF file, using the rolling ball principle. The **Ball size** can be entered in pixels. Background subtraction is only necessary if the illumination of the image is not uniform, which is not the case in the

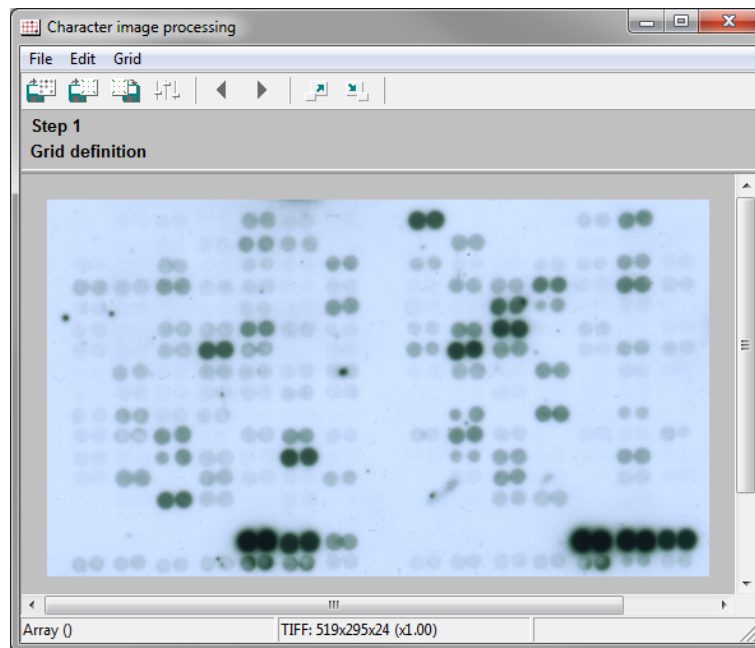


Figure 6.1.58: The BNIMA program with a gene array image (fragment) loaded.

example image. **Spot removal** allows all spots and irregularities below a certain size to be removed from the image, whereas larger structures are preserved.

3.64 Leave **Background subtraction** disabled, and enable **Spot removal**, with a maximal **Spot size** of 3 pixels.

3.65 Press <OK> to quit the *Settings dialog box*.

The background subtraction and spot removal changes are only seen when **Edit > Show value scale** is enabled in the *Character image processing window*.

3.66 Check **Edit > Show value scale**.

The image now looks cleaner: spots are removed and the image is shown in gray scale rather than as 24 bit true color image.

In step 1 (grid definition) we will create a grid that defines the cells of the array.

3.67 Select **Grid > Add new** and enter "17" as **Number of rows** and "15" as **Number of columns**.

Choosing 17 and 15 rather than 14 by 14 is to allow the calibration spots to be included, and to take account of the blank column.


3.68 Press <OK> and the grid appears.

3.69 Move the upper left dragging node until the grid crosses match the middle of each double spot in the upper left area of the array.

3.70 Next, move the lower right dragging node until the grid crosses match the middle of each double spot in the lower right area of the array.

3.71 Then, move the lower left and upper right dragging nodes of the grid to distort the rectangle so that the grid crosses in the lower left and upper right areas, respectively, match with the double spots.

The grid on the image should now look as in Figure 6.1.59.

3.72 Move to the next step using **Edit > Next step** or the  button.

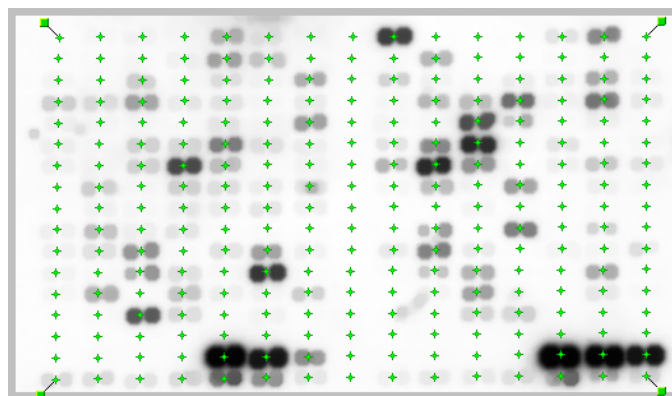


Figure 6.1.59: Correctly aligned grid on example gene array.

In this step, the layout of the cells is defined: the shape and size of the quantification area within each cell. In this step, we also define which cells we want to use for quantification and which cells not. By default, all cells of the grid are used for quantification.

3.73 Select the cells in the blank column of the image and **Cells > Delete selected**.

3.74 Similarly, select the three lowest rows and **Cells > Delete selected**.

Two cells of the second last row represent 0 and 100% hybridization respectively: the fourth and the fifth cell. We will include these cells for calibration, hence we have to include them again:

3.75 Select the fourth and fifth cell of the second last row and **Cells > Add selected**.

Before the program can do the quantification, it needs to know what the averaging area of the cells is. This is done using a *mask* which the user defines. In this case, it is clear that we will have to define two masks per cell, in order to cover the duplicate spots.

3.76 Select all cells as in Instruction 3.17.

3.77 Add a circular mask to all selected cells with **Cells > Add disk to mask**.

A dialog box prompts to enter a **Radius** for the disk in pixels, the **X offset** (horizontal shift from the cell marking cross) and the **Y offset** (vertical shift from the cell marking cross). For the offsets, a negative value can be entered.

3.78 Enter 6 as **Radius**, and -6 as **X offset**. Press <OK> to confirm.


The masks appear on all used cells of the grid as semitransparent red disks.

3.79 Add a second mask to all selected cells with **Cells > Add disk to mask**.

3.80 Enter 6 as **Radius**, and 6 as **X offset**. Press <OK> to confirm.

After these steps, the *BNIMA window* should look like in Figure 6.1.60.

3.81 If this is the case, move to the next step using **Edit > Next step** or the  button.

3.82 In the Quantification step, first call the *Settings dialog box* with **Edit > Settings** or .

3.83 Select the *Quantification tab*, specify a **First degree polynomial fit** and click <OK>.

3.84 Select the fourth cell in the second last row and **Quantification > Define calibration point**.

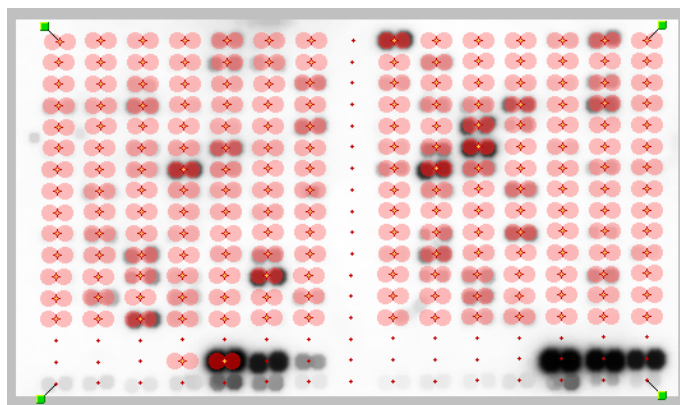


Figure 6.1.60: Array editing in BNIMA, with included and excluded cells, and masks defined.

3.85 Enter "0" (zero).


3.86 Select the fifth cell in the second last row and *Quantification > Define calibration point*.

3.87 Enter "100".


All cells are now quantified between the zero and 100% hybridization control, and we now need to specify which cells to add to the character set. Since the calibration cells (second last row) are not part of the character set, these should not be included.

3.88 Select all but the three last rows and select *Quantification > Add cells to character set*.

The cells to be used in the character set are now numbered 1 to 196.

3.89 Copy the quantified cells to the clipboard with *Quantification > Export to clipboard* or .

Before closing the BNIMA program, you can save the entire configuration defined for this gene array system:

3.90 Select *File > Save configuration* as or .

3.91 Enter a name e.g. "Gene array", and press <OK> to save the configuration.

3.92 Close the BNIMA program.

3.93 Right-click on the *Experiment card* window (see also 6.1.4), and select *Paste from clipboard* from the floating menu.

The experiment card now is filled with data and looks like in Figure 6.1.62.

3.94 Click the upper left triangular button to close the experiment card and save the imported character values.

6.1.4 Character experiment card

Clicking on the colored dot of a character type experiment in the *Database entries* panel pops up the character *Experiment card* window. Alternatively, open the *Entry* window of the entry and click on the flask next the experiment name in the *Experiments* panel.

Right-clicking in the *Experiment card* window of a character experiment pops up a floating menu, from which you can call the character image import program BNIMA (*Edit image*) if applicable. You can copy the data set to the clipboard (*Copy to clipboard*), or paste data from the clipboard into the experiment (*Paste*

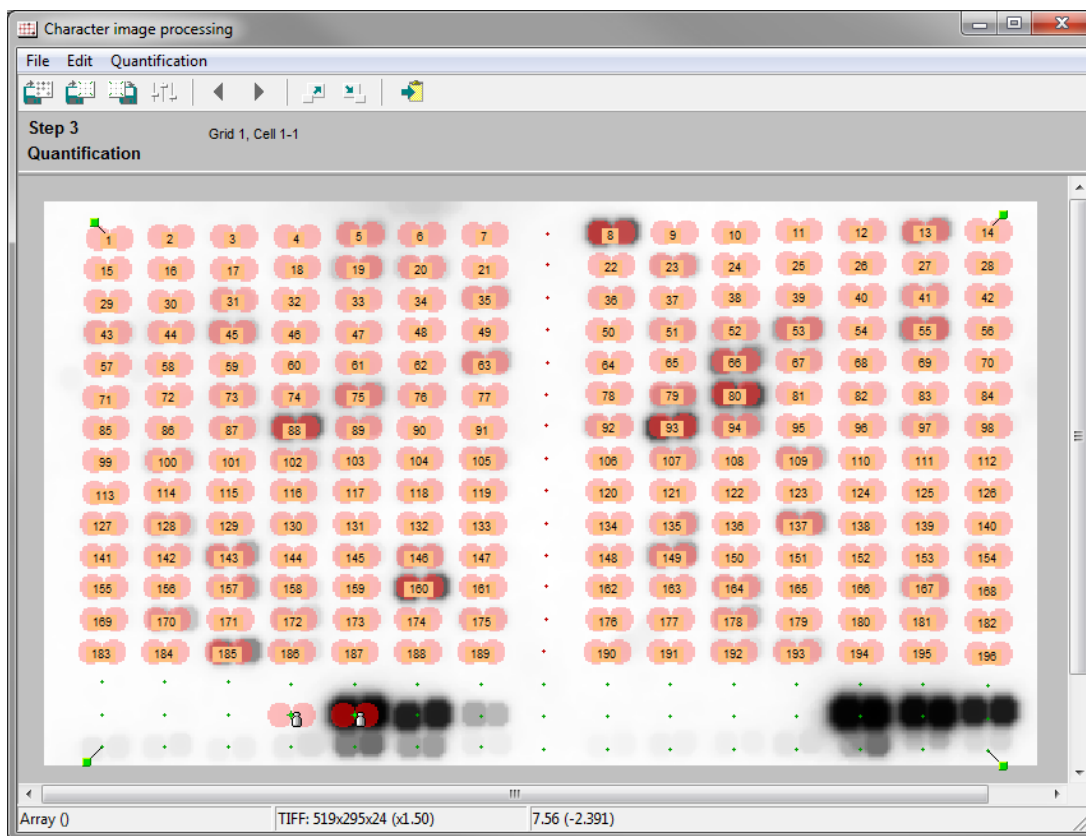


Figure 6.1.61: Quantification step.

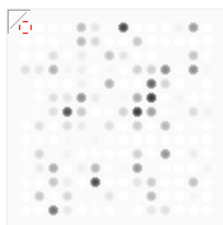


Figure 6.1.62: Example gene array experiment card after import of character values using BNIMA.

from clipboard). *Export character values* creates a similar output, but provides the names of the characters in case of an open character set (see 6.1.1). With the option *Remove this experiment*, the character set can be deleted from the database. This is an irreversible operation.

It is possible to enter or edit the character data directly in the *Experiment card* window. To do so, open the *Entry* window for an entry which has no character data available for a character type experiment. For experiments that are not available for the entry, an empty flask is shown. Clicking on the empty flask button of an empty character type displays the following message: "The experiment "XXX" is not defined for this entry. Do you wish to create a new one?". An empty *Experiment card* window appears when confirming the action. Depending on whether the character type is displayed as list or as plate (see 6.1.2), the input method is different:

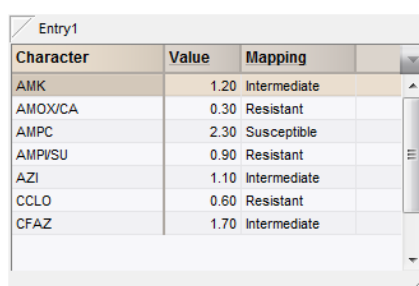
Plate presentation:

In case of binary (plus or minus) data, the values can be entered in the card using the numerical + and - keys. The cursor automatically jumps to the next test after have entered a value. The cursor can be moved using the left and right arrow keys.

In case of non-binary values (real or integer values), each test can be varied continuously between the minimum and the maximum using the **PageUp** key (increase intensity) and the **PageDown** key (decrease intensity). When using the + and - keys to enter non-binary data, the defined maximum for the character type is used if + is entered.


List presentation:

A new character is added to the list with the **<Insert>** button. A dialog box shows all available characters for this character type that have not yet been used for this entry. A character can be selected from this list. A new character is added when pressing the **<Create new>** button. A value can be entered/changed for a character by clicking in the **Value** column next to the character name. A **Mapping** column is shown next to the **Value** column (see Figure 6.1.63). If character values fall within the ranges of defined mappings (see 6.1.2.7) for this character type, the name of the mapping is displayed in the **Mapping** column.




Character	Value	Mapping
AMK	1.20	Intermediate
AMOX/CA	0.30	Resistant
AMPC	2.30	Susceptible
AMPVSU	0.90	Resistant
AZI	1.10	Intermediate
CCLO	0.60	Resistant
CFAZ	1.70	Intermediate

Figure 6.1.63: The 'Mapping' column in the *Experiment card* window of a character type.

The character *Experiment card* window is closed by pressing the close button  in the upper left corner of the card. The program asks to save the changes made.

6.1.5 Exporting character data

With the **Export fields and characters** option, listed under the topic **Character type data** in the *Export* dialog box (see Figure 6.1.64), character information and optionally entry information can be exported to a Comma Separated Values (CSV) file.

In the *Database entries* panel of the *Main* window, select the entries to export. A single entry can be selected by holding the **Ctrl**-key and left-clicking (**CTRL+click**). Check boxes for selected entries are indicated as . In order to select a group of entries, hold the **Shift**-key and click on another entry. All the entries in the database can be selected using the keyboard shortcut **Ctrl+A** or with *Edit* > **Select all**.

Selecting **Export fields and characters** under **Character type data** in the *Export* dialog box and pressing **<Export>** starts the export wizard (see Figure 6.1.65).

In the *Export* dialog box, all database fields are listed in the upper panel, all character types defined in the database are displayed in the lower panel.

Select the **Fields** and **Character types** to export. To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.

Pressing **<Next>** brings up the last step, the *Settings* dialog box.

All character types selected in the previous step (*Export* dialog box) are listed in the grid.

For each character type following settings can be specified:

- **Absent value:** Absent values are default denoted as "???" in the output file. This can be changed to any

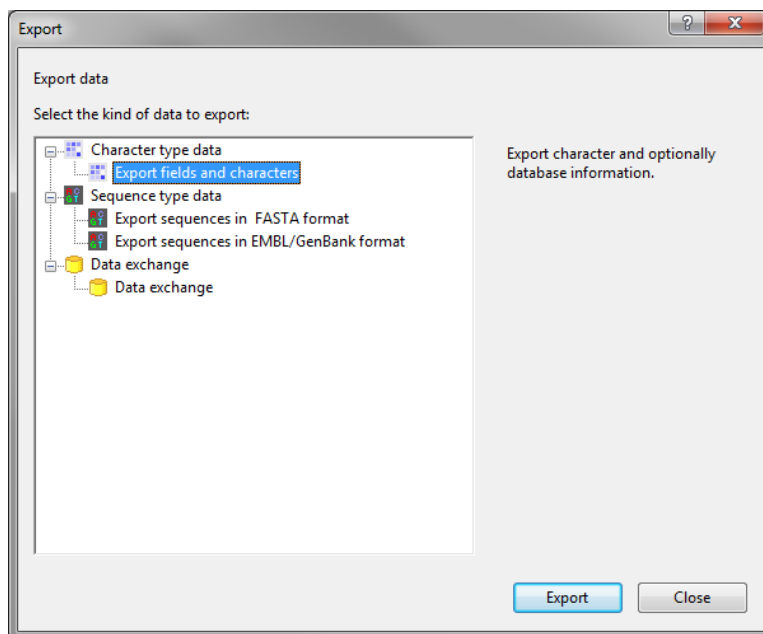


Figure 6.1.64: The *Export fields and characters* option in the *Export* dialog box.

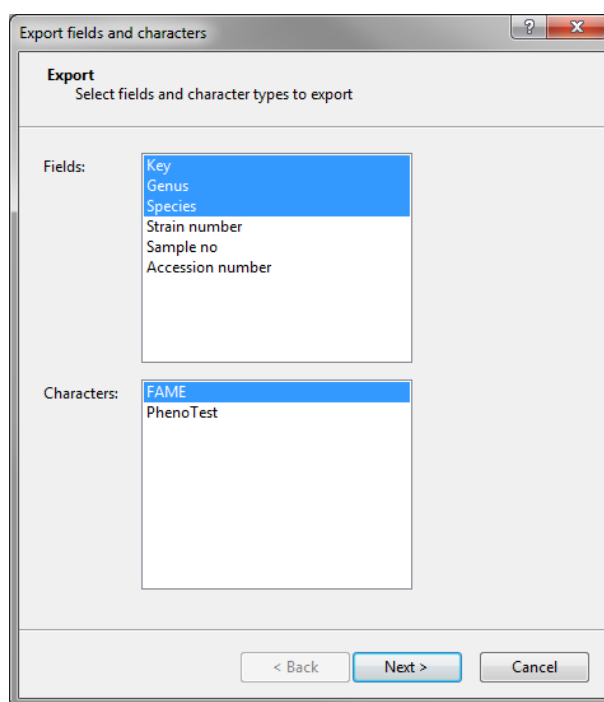


Figure 6.1.65: The *Export* dialog box.

other symbol or text.

- **Active only:** When this option is checked, only the active characters will be included in export (see 6.1.2.4 on how to enable/disable characters in the *Character type* window). Unchecking this option will export all characters.
- **Use mapping:** Default the character values are exported. To export the mapping states instead of the character values, check this option (see 6.1.2.7 on how to create mappings).

Pressing <*Finish*> exports the information in tabular format to a Comma Separated Values (CSV) file, that

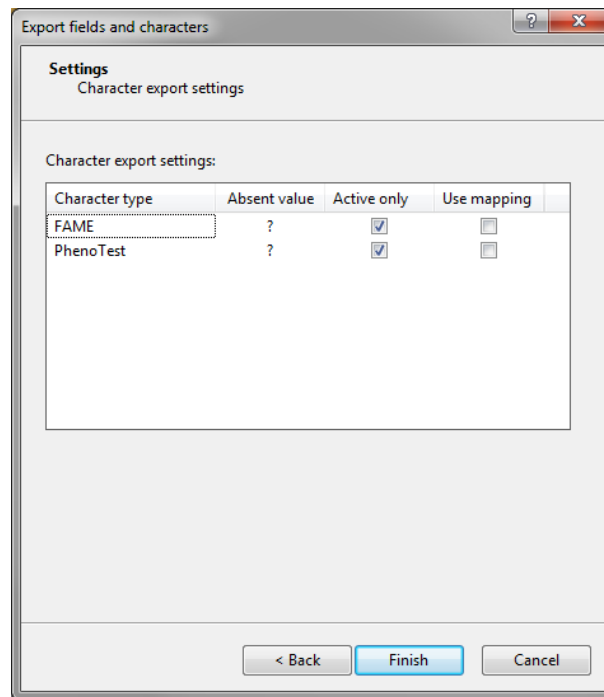


Figure 6.1.66: The *Settings* dialog box.

will be opened by the default CSV editor on your computer (often MS Excel).

Chapter 6.2

Cluster analysis of characters

6.2.1 Selecting characters for comparison

Character experiment types in BioNumerics are very flexible in the sense that an analysis in the *Comparison* window can not only be performed on the complete data set, but also on any subset thereof. This is achieved using *character views* (see 6.1.2.11). Character views can be query or subset-based and are defined within the character type. In the *Comparison* window, the desired character view is selected from the 'Aspect' drop-down list in the *Experiments* panel (see Figure 6.2.1).

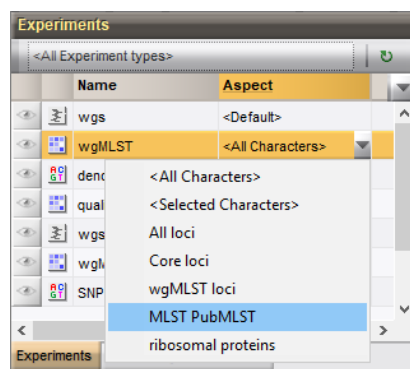


Figure 6.2.1: The 'Aspect' drop-down list for a character type in the *Experiments* panel.

One of the default character views is the <Selected Characters> view, which allows you to change the input data for e.g. a cluster analysis almost on-the-fly: with this view enabled, any analysis will be executed on the currently selected characters only.

The selection state of a single character can be toggled with **Ctrl+click** in the header of the *Experiment data* panel: an orange rectangle (▲) in the header indicates that a character is currently selected. A continuous range of characters is selected with **Shift+click**. To invert the character selection (i.e., the selected characters will become unselected and the unselected ones will be selected) use **Characters > Invert character selection**.

With **Characters > Select all characters**, all characters in a character experiment type can be selected at once. Use **Characters > Clear character selection** to unselect all characters.

Character selections can be made based on character absence / presence. This is useful e.g. to limit the number of missing character data from a character matrix.

To select characters based on missing values, use **Characters > Filter characters > Select absent characters....** This action displays the *Select absent characters* dialog box (see Figure 6.2.2).

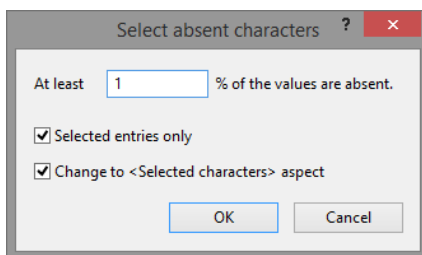


Figure 6.2.2: The *Select absent characters* dialog box.

A percentage can be specified in the text box *At least ... % of the values are absent*. The software will determine for each character what the percentage absent values is ($R_{\text{absent}} = 100 * \frac{N_{\text{absent}}}{N_{\text{total}}}$ with N_{absent} the number of entries for which the character value is missing and N_{total} the total number of entries) and will select the character only if the percentage is larger or equal to this threshold.

The percentage that was specified in the text box above is by default calculated on *all* entries in the comparison. With *Selected entries only*, this percentage is determined on the *selected* entries in the comparison.

With the option *Change to <Selected characters> aspect* checked, the software will automatically switch to the <Selected characters> character view. This will be indicated in the 'Aspect' field in the *Experiments* panel.

Conversely, to select characters based on the percentage of character values that are present, use *Characters > Filter characters > Select present characters....* This action will show the *Select present characters* dialog box (see Figure 6.2.3).

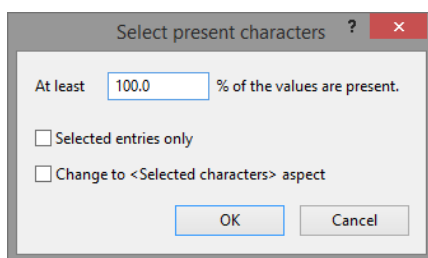


Figure 6.2.3: The *Select present characters* dialog box.

A percentage can be specified in the text box *At least ... % of the values are present*. The software will determine for each character what the percentage present values is ($R_{\text{present}} = 100 * \frac{N_{\text{present}}}{N_{\text{total}}}$ with N_{present} the number of entries for which a character value is present and N_{total} the total number of entries) and will select the character if the percentage is larger or equal to this threshold.

The percentage that was specified in the text box above is by default calculated on *all* entries in the comparison. With *Selected entries only*, this percentage is determined on the *selected* entries in the comparison.

With the option *Change to <Selected characters> aspect* checked, the software will automatically switch to the <Selected characters> character view. This will be indicated in the 'Aspect' field in the *Experiments* panel.

Character selections can be made based on monomorphic or polymorphic characters. This is useful for categorical characters to filter out characters that show little or no variation.

To select characters based on the percentage of monomorphic (i.e. same state) characters, use *Characters > Filter characters > Select monomorphic characters....* This action will show the *Select monomorphic characters* dialog box (see Figure 6.2.4).

A percentage can be specified in the text box *At least ... % of the values are the same*. The software will

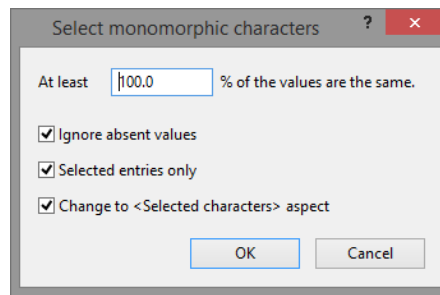


Figure 6.2.4: The *Select monomorphic characters* dialog box.

determine for each character what the percentage monomorphic values is ($R_{mono} = 100 * \frac{N_{mono}}{N_{total}}$ with N_{mono} the maximum number of entries for which the character value is found to be the same and N_{total} the total number of entries) and will select the character if the percentage is larger or equal to this threshold.

By default, absent character values will be included in the calculation of the percentage, unless **Ignore absent values** is checked.

The percentage that was specified in the text box above is by default calculated on *all* entries in the comparison. With **Selected entries only**, this percentage is determined on the *selected* entries in the comparison.

With the option **Change to <Selected characters> aspect** checked, the software will automatically switch to the <Selected characters> character view. This will be indicated in the 'Aspect' field in the *Experiments* panel.

Conversely, to select characters based on the percentage of polymorphic (i.e. different state) characters, use **Characters > Filter characters > Select polymorphic characters...** show the *Select polymorphic characters* dialog box (see Figure 6.2.5).

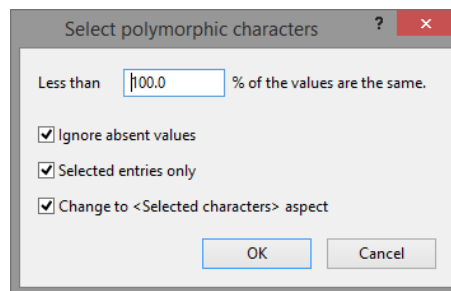


Figure 6.2.5: The *Select polymorphic characters* dialog box.

A percentage can be specified in the text box **Less than ... % of the values are the same**. The software will determine for each character what the percentage monomorphic values is ($R_{mono} = 100 * \frac{N_{mono}}{N_{total}}$ with N_{mono} the maximum number of entries for which the character value is found to be the same and N_{total} the total number of entries) and will select the character if the percentage is smaller than this threshold.

By default, absent character values will be included in the calculation of the percentage, unless **Ignore absent values** is checked.

The percentage that was specified in the text box above is by default calculated on *all* entries in the comparison. With **Selected entries only**, this percentage is determined on the *selected* entries in the comparison.

With the option **Change to <Selected characters> aspect** checked, the software will automatically switch to the <Selected characters> character view. This will be indicated in the 'Aspect' field in the *Experiments* panel.

6.2.2 Character comparison settings

Please note that cluster analysis of characters requires both the Character data module (**CH**) and the Tree and network inference module (**TN**) to be present in your BioNumerics configuration.

In terms of parameter settings, character sets are the simplest class of data to analyze. The various types of character sets that exist, however, require a large number of coefficients to be available for analyzing character tables.

To calculate a cluster analysis on a character type, highlight the character type to analyze in the *Experiments* panel of the *Comparison* window. Optionally, select an *aspect* of the character type from the drop-down list in the *Experiments* panel (see 13.2.5). Next, use **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**... The *Comparison settings* wizard appears (Figure 6.2.6).

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient.

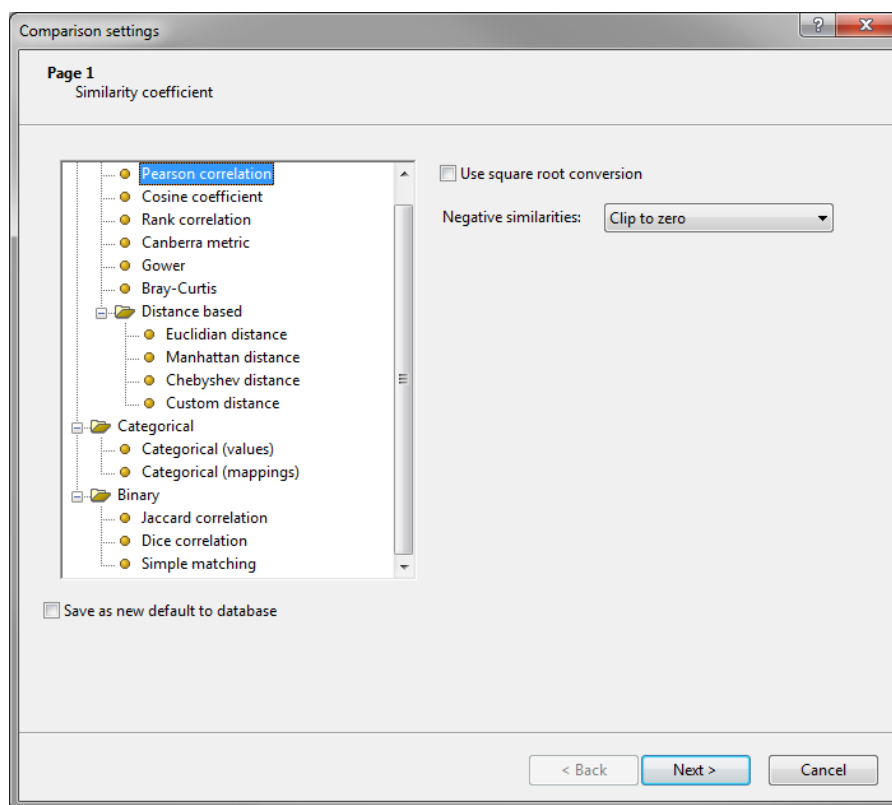


Figure 6.2.6: The *Similarity coefficient* wizard page, which deals with the choice of the similarity coefficient.

The hierarchical representation on the left provides an overview of the available coefficients. Depending on the selected coefficient, the relevant settings are displayed on the right. The coefficients are subdivided in three categories: **Numerical**, **Categorical**, and **Binary**. The **Numerical** category has another subdivision, which is **Distance based**. Each of the categories can be collapsed by clicking on the small "-" (minus) sign that precedes the category name.

Numerical coefficients treat the character values as numbers.

The **Pearson correlation** (or Pearson product-moment correlation) is calculated as:

$$r_{j,k} = \frac{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k)}{\sqrt{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2 \sum_{i=1}^n (x_{i,k} - \bar{x}_k)^2}}$$

with

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

and

$$\bar{x}_k = \frac{1}{n} \sum_{i=1}^n x_{i,k}$$

Hereby, n denotes the number of characters in the character set and $x_{i,j}$ and $x_{i,k}$ the i^{th} character value of entries j and k , respectively. r_i is the character range of character i .

The related **Cosine coefficient** is calculated as:

$$C_{j,k} = \frac{\sum_{i=1}^n x_{i,j} x_{i,k}}{\sqrt{\sum_{i=1}^n x_{i,j}^2 \sum_{i=1}^n x_{i,k}^2}}$$

The **Rank correlation** (or Spearman rank-order correlation) coefficient first transforms an array of characters into an array of ranks according to the magnitude (intensity) of the character values. The rank arrays are then compared using the Pearson product-moment correlation coefficient. The **Rank correlation** is known to be a very robust coefficient, but with low sensitivity.

The **Canberra metric** is calculated as:

$$D_{CANB(i,j)} = \frac{1}{n} \sum_{i=1}^n \frac{|x_{i,j} - x_{i,k}|}{|x_{i,j} + x_{i,k}|}$$

The **Gower** coefficient is calculated as:

$$D_{G(j,k)} = 1 - \sum_{i=1}^n \frac{|x_{i,j} - x_{i,k}|}{r_i}$$

The **Bray-Curtis** metric is calculated as:

$$D_{BC(j,k)} = \frac{\sum_{i=1}^n |x_{i,j} - x_{i,k}|}{\sum_{i=1}^n |x_{i,j} + x_{i,k}|}$$

The **Morisita-Horn** overlap index [27] [28] [19] is a statistical measure of dispersion of individuals in a population. It is used to compare overlap among samples. This formula is based on the assumption that increasing the size of the samples will increase the diversity because it will include different habitats (i.e. different faunas). The index is given by:

$$C_H = \frac{2 \sum_{i=1}^S x_i y_i}{(\frac{\sum_{i=1}^S x_i^2}{X^2} + \frac{\sum_{i=1}^S y_i^2}{Y^2})XY}$$

where x_i are the character values of the first entry, and y_i are the character values of the second entry. X resp. Y is the sum of all character values for the first resp. second entry. The resulting values are reported as similarities between 0 and 100.

The quantitative **Kulczynski** index uses for each character the minimum of the two character values as an estimate of similarity, and normalizes the sum of these minima by dividing by the sum of all observations in the first sample. To obtain symmetry, the same is done after swapping the order of the samples and the two results are averaged out. The index takes values in the range [0, 1], and are reported as similarities between 0 and 100.

The **Smith theta** similarity index (or Smith's community Jaccard index) [36] calculates the probability that, given a randomly selected character is present in at least one of the entries, is present in both of the entries.

The **Yue & Clayton** similarity measure [42] is a non-parametric version of the **Smith theta** similarity index.

Following parameters are available for numerical coefficients:

- Checking **Use square root conversion** can be particularly useful when comparing highly related organisms. This has the effect that narrow branches on a dendrogram are stretched out relatively more than distant links.
- **Negative similarities** can be dealt with in different ways. If **Clip to zero** is selected, negative similarity values will be replaced by zero (no correlation). When **Unchanged** is set, the program will calculate with the negative values. **Absolute value** will treat negative and positive similarity values in the same way. **Negative similarities** values can only be obtained with the **Pearson correlation** coefficient, therefore the option is not available when any of the other coefficients is selected.

Distance based is a subcategory of the **Numerical** coefficients. Instead of a matrix of similarity values, application of any these coefficients will result in a matrix of distances.

The **Euclidean distance** coefficient is calculated as:

$$\Delta_{j,k} = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_{i,j} - X_{i,k})^2}$$

with $X_{i,j}$ and $X_{i,k}$ the i^{th} scaled character value of entries j and k , respectively. Character values are scaled by dividing them by the character range or a fixed distance factor.

The **Manhattan distance** coefficient is calculated as:

$$M_{j,k} = \frac{1}{n} \sum_{i=1}^n |X_{i,j} - X_{i,k}|$$

The **Chebyshev distance** coefficient is calculated as:

$$C_{j,k} = \text{MAX} (|X_{i,j} - X_{i,k}|)$$

The **Hellinger distance** [32] [18] is a distance used to quantify the similarity between two probability distributions. For two discrete probability distributions

$$P = (p_1 \dots p_k) \text{ and } Q = (q_1 \dots q_k)$$

the Hellinger distance is given by:

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^k (\sqrt{p_i} - \sqrt{q_i})^2}$$

The resulting distance takes values between 0 and 1. In BioNumerics, character values are translated to a discrete probability distribution by dividing each character value by the sum of all character values. The resulting values are reported as distances between 0 and 100.

The **Soergel distance** is a rescaled version of the Manhattan distance ("city block metric"), where for each pair of character values, the absolute value is divided by the absolute value of the maximum value of both values.

$$d_{soergel}(x_i, x_j) = \frac{\sum_{k=1}^3 |x_{ik} - x_{jk}|}{\sum_{k=1}^3 |\max(x_{ik}, x_{jk})|}$$

The **Profile distance** calculates an Euclidean distance between the discrete probability distributions derived from the character values.

$$d(i, i') = \sqrt{\sum_{j=1}^p \left(\frac{f_{ij}}{f_{i.}} - \frac{f_{i'j}}{f_{i'.}} \right)^2}$$

Character values are translated to a discrete probability distribution by dividing each character value by the sum of all character values.

The **Chi-square distance** calculates a weighted profile distance between character values. The weight of each character is defined as the frequency of the character $f_{.j}$ over all samples in the comparison.

$$d(i, i') = \sqrt{\sum_{j=1}^p \left(\frac{f_{ij}}{f_{i.}} - \frac{f_{i'j}}{f_{i'.}} \right)^2 \cdot \frac{1}{f_{.j}}}$$

The **Chord distance** calculates the Euclidean distance between two sets of character values that have been normalized by dividing by the Euclidean norm of each of the sets. As such, the chord distance calculates the length of the chord between two points on a multidimensional sphere.

In addition to the **Use square root conversion** option, for distance based numerical coefficients the **Distance scaling** can be chosen. This can be either the **Character range** or **Fixed**. If **Fixed** is selected, a **Distance factor** can be specified.

The **Custom distance** is an advanced feature that allows you to custom create any distance coefficient. The formula used to calculate this coefficient is:

$$Cu_{j,k} = \sqrt[p]{\frac{1}{n} \sum_{i=1}^n [f(|X_{i,j} - X_{i,k}|)]^p}$$

The user can enter the **Power** p and determine the custom function $f()$ by specifying **Points**. These **Points** should be entered as series of x y value pairs, separated by semicolons (e.g. "x1;y1;x2;y2"). By default, the point 0;0 is always used. Furthermore, one can specify whether or not to **Divide by number of characters**. The function $f()$ should be regarded as specifying a penalty for each possible difference between character values. When no points are entered, or for any value outside of the range of points entered, $f(|X_{i,j} - X_{i,k}|) = |X_{i,j} - X_{i,k}|$. Figure 6.2.7 shows two example functions and the points that were entered to specify these functions. The first function (Figure 6.2.7 a) does not penalize differences smaller than 1 (from 0;0 to 1;0), the penalty for larger differences increases proportionally (no points specified). The second function (Figure 6.2.7 b) penalizes differences up to 1 proportionally (from 0;0 to 1;1). After this threshold, the penalty does not increase any more (achieved via entering an x y value pair (100;1) outside the expected range).

Categorical coefficients are neither binary nor numerical, since they treat each different value as a different *state*. These coefficients are useful for analyzing *multi-state* character sets, for example colors (red, green, blue etc.) represent each a categorical state. Typical multi-state characters used in typing, taxonomy and phylogeny are phage typing, (whole genome) Multi Locus Sequence Typing (MLST or wgMLST), Variable Number Tandem Repeats (VNTR) typing, and Single Nucleotide Polymorphisms. The types or categories

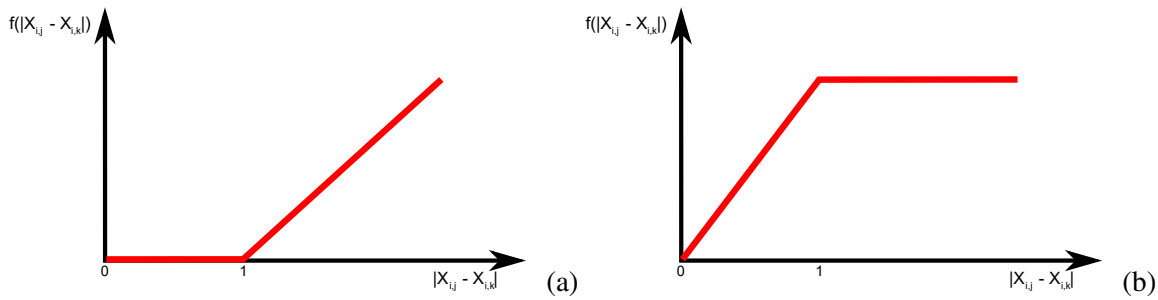


Figure 6.2.7: Examples of functions for custom distance coefficients, with following points entered: "1;0" (a) and "1;1;100;1" (b).

assigned to the different phage reactions, allele numbers, repeat numbers and base calls, respectively, are good examples of categorical or multi state data which can be analyzed using a categorical coefficient.

The *Categorical (values)* coefficient works directly on the character values. It has following parameters:

- With *Calculate as distance* checked, the distance between two entries is reported. With this option unchecked, the software calculates similarity values. Note that there is a maximum distance of 200; distance values that exceed this maximum will be clipped to 200. The distance D is calculated with this coefficient as:

$$D = \frac{N\Delta_{AB}}{N_{AB}}$$

with N the total number of characters in the comparison, Δ_{AB} the number of characters different between entry A and B , and N_{AB} the number of characters present in both entries.

- *Ignore zero values* will leave any character out of the pairwise comparison for which at least one entry in the pair has a zero (0) value. With other words, zeros in a character set will be treated the same way as missing values if this option is checked.
- With *Fuzzy logic* checked, the coefficient will score each character match decreasingly with increasing distance between the values, between full match (zero distance) and no match (distance = tolerance).
- A certain *Tolerance* can be specified for values to be considered as belonging to the same category. This makes it possible to treat non-discrete (non-integer) values as categorical. By default, the *Tolerance* value is set to zero, which means that no tolerance is allowed, i.e. the values must be identical to be considered the same category.

The *Categorical (differences)* coefficient also works on the character values and calculates by default a distance matrix. Hence, the corresponding option is grayed out. The *Categorical (differences)* coefficient differs from *Categorical (values)* in that it uses the absolute number of differences instead of a normalized distance, i.e. $D = \Delta_{AB}$.

The coefficient has an additional parameter called *Scaling factor*, to deal with the hard-coded maximum of 200 that can be calculated for a distance value. Sensible values for the *Scaling factor* are "1", "10" and "100", allowing the correct visualization of maximally 200, 2000 and 20000 different character values, respectively, in a cluster analysis. To trace back the number of different character values from the dendrogram branches or distance matrix, the displayed values needs to be multiplied with the *Scaling factor* used.

The *Categorical (mappings)* coefficient does not work on the character values directly, but takes instead the *character mapping* into account. This coefficient allows you to work with a customized mappings similarity matrix (see 6.1.2).

For a *Binary* coefficient, a character can only have two states: positive or negative (0 or 1).

The **Jaccard** coefficient is calculated as:

$$S_J = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

with N the total number of characters. N_A , N_B and N_{AB} are the number of characters that are positive for entry A , entry B and both A and B , respectively. N_{ab} is the number of characters that is negative for both A and B .

The **Dice** is calculated as:

$$S_D = \frac{2N_{AB}}{N_A + N_B}$$

The **Simple matching** is calculated as:

$$S_{SM} = \frac{N_{AB} + N_{ab}}{N}$$

Jaccard correlation and **Dice correlation** are very related to each other whereas **Simple matching** is more fundamentally different. The Jaccard and Dice coefficients only consider "scoring characters" being two positive characters in both data sets, whereas the simple matching coefficient also considers two negative characters as scoring.

The **Shared observations** similarity coefficient counts the number of characters that are *present* in both sets. This is in contrast to the **Simple matching** coefficient, which also counts the characters that are *absent* in both sets.

When dealing with a non-binary (numerical) data set, a **Binary conversion** needs to be done from numerical values to binary values (positive or negative), before one of these coefficients can be applied. Any value character value above the **Binary conversion limit** will be converted to positive. Under **Limit type**, one can specify that the **Binary conversion limit** should be expressed as a certain percentage of either the **Mean value** or **Maximum value** from the experiment.

If a similarity matrix already exists for the selected experiment, an option **Keep existing similarity matrix** appears. When checked, the previously calculated similarity matrix will be used and all coefficient options will appear gray (disabled).

Pressing <Next> opens the *Cluster analysis* wizard page, which deals with the calculation of a dendrogram from the similarity matrix and is discussed in [13.2.6](#).


Pressing <Next> again in the *Cluster analysis* wizard page starts the cluster analysis. When finished, a dendrogram and similarity matrix are shown for the character type.

Different weights can be defined for individual characters and these weights can be taken into account when calculating the similarities based on the set of characters:

First, a character information field called **Weight** needs to be added in the *Character type* window (select **Fields > Add new field...**). The weights can either be entered manually or imported from a text file. In the *Comparison* window, make sure the character type experiment is highlighted in the *Experiments* panel and choose **Clustering > Weighted categorical clustering**.


This action calculates a similarity matrix and dendrogram using a weighted categorical coefficient. The weight is taken from the character information field named **Weight**. The analysis that is created with this command will be listed with the character type name (the name cannot be changed). Once the similarity matrix is calculated, the similarity values can be used as input data for the calculation of other trees or networks (e.g. minimum spanning tree): select **Clustering > Calculate > Advanced cluster analysis...** and make sure to start from the **Similarity matrix** and not from the **Character data**.


6.2.3 Character display functions


Different options are available to visualize character data in a comparison. Before using these character display functions, make sure that the character data are shown in the *Experiment data* panel by pressing the eye button () next to the experiment name in the *Experiments* panel.

By default, the characters are named in the header of the *Experiment data* panel with their character names. Using the drop-down list in the toolbar of this panel, any character type information field (if available, see 6.1.2.9 on how to create character type information fields) can be selected as display name. This action affects only the display of the characters in the current comparison, not in other parts of the software.



The character values are initially displayed as colors according to the color scale defined for each character (see 6.1.2).


Select **Characters > Show bar graphs** () to display the character values as colored bar graphs. The colors used for the bar graphs are those as defined in the color scale for each character, while the bar heights are proportional to the character values, expressed as a percentage of the character range (see 6.1.2).

Select **Characters > Show values** () to show the corresponding character values for all entries in the comparison.

If a mapping was defined for the character type (see 6.1.2), the mapped names for each character value can be displayed in the *Experiment data* panel with **Characters > Show mappings** (). In case no mapping is present or when the character values do not fall within the ranges of the predefined criteria, a "<?>" is displayed.

The character values can be displayed as colors again with **Characters > Show colors** ()

The colors can also be shown in overlay with the values or mappings with **Characters > Show values+colors** () or **Characters > Show mappings+colors** (), respectively.

A convenient option to quickly check the behavior of an individual character is to list the entries according to the values of this character. Entries are sorted by increasing value of a selected character with **Characters > Sort by character value** (). A dendrogram is not displayed any more, since it would impose a different order on the entries. If a dendrogram was calculated previously, it can be called again from the *Analyses* panel.

6.2.4 Calculating the diversity of characters

In ecology, a number of coefficients describing species diversity, evenness and richness were developed to quantitate the diversity of an ecosystem based on species counts. Calculating such diversity indices on categorical character data can provide insights in the variability of each character within a set of samples.

To calculate one or more diversity indices on a character experiment type in the *Comparison* window, highlight the character type in the *Experiments* panel and select the desired character aspect from the 'Aspect' drop-down list. Next, select **Statistics > Character diversity indices...** to open the *Index of diversity* dialog box (see Figure 6.2.8).

From the **Diversity index** list, tick the check boxes of the diversity indices you want to calculate.

Checking **Ignore absent values** leaves the missing character values out of the calculations. When this option is unchecked, absent values will be considered as a separate state.

If **Selected entries only** is checked, the analysis will be performed on the selected entries in the comparison. When unchecked, all entries in the comparison will be used.

Pressing <OK> will open a *Charts and statistics* window with the selected diversity indices pre-calculated. Only the values calculated with the first diversity index is displayed in the *Chart area* panel. By right-

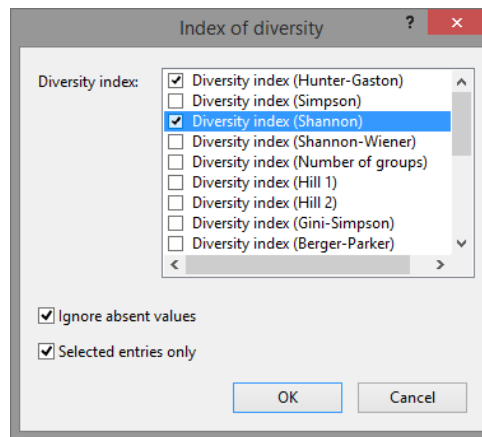


Figure 6.2.8: The *Index of diversity* dialog box.

clicking on another diversity index and selecting *Use as Chart value*, the results from this index are plotted on the chart. See 14.5 for more instructions on how to use the *Charts and statistics* window.

6.2.5 Exporting character information

To export the character values of all entries in the comparison, select *Characters > Export character table*.

The export file, `export.csv` or `export.txt` (depending on the preferences set, see 2.3.3), is a comma or tab-delimited text file which contains the character names on the first line, followed on each of the next lines by the database key and corresponding character values of each of the entries in the comparison. Entries are listed in the export file in the same order as they appear in the comparison.

6.2.6 Character bar graphs

For categorical (multi-state) character experiment types, a bar graph shows the frequency of each character state over a set of entries. This provides a convenient tool e.g. to quickly show the percentage of resistant isolates against a series of antibiotics.

A bar graph can be created from the active character aspect by selecting *Statistics > Character bar graph...*. This action opens the *Character summary* dialog box (see Figure 6.2.9).

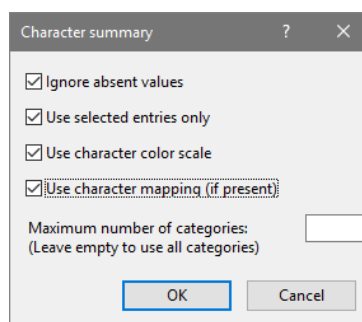


Figure 6.2.9: The *Character summary* dialog box.

It is possible to *Ignore absent values*. If this option is unchecked, absent character values will be grouped in a separate category 'None'.

With *Use selected entries only* checked, only character data for the selected entries in the comparison will be used for the bar graph. Otherwise, character data from all comparison entries will be included.

If *Use character color scale* is checked, the same color scale as specified in the character experiment type (see 6.1.2.6) will be used in the bar graph. Otherwise, arbitrary colors will be assigned to the different categories in the bar graph.

Checking *Use character mapping (if present)* will use the mappings (see 6.1.2.7) instead of the actual values.

The *Maximum number of categories* used in the bar graph can be specified manually. If this option is not specified, all categories found within the data set will be used in the bar graph.

When the <OK> button in the *Character summary* dialog box is pressed, the *Charts and statistics* window appears with the bar graph displayed.

Part 7

Trend data types

Chapter 7.1

Setting up trend data type experiments

7.1.1 Introduction

Reactions to certain substrates or conditions are sometimes recorded as multiple readings in function of time, as *kinetic* readings. The kinetic reading of enzymatic or metabolic activity is thought to be both more informative and more reliable than measuring the degree of activity at one point in time. Examples are the kinetic analysis of metabolic and enzymatic reactions, real-time PCR, or time-course experiments using microarrays. Although multiple readings per experiment are mostly done in function of time, they can also depend on another factor, e.g. measurements in function of different concentrations.

These different data types have in common that they measure a trend of one parameter in function of another. We therefore call them *trend type* data. Analysis is usually done by fitting a *curve* through the measurement points using a *fit model*. To use a fit model, we have to assume that the biological data that are being studied behave according to a certain predictable pattern. A model is a function that fits the biological data as closely as possible. Specific parameters can be deduced from the model function and comparing the samples can therefore also be done using the parameters of the curves rather than the original measurement points. Bacterial growth or activity is usually analyzed using a *Logistic Growth* fit. A number of parameters can be calculated from the curve fit (Figure 7.1.1), including the times at 5% growth increase (T_{05}), 50% growth increase (T_{50}) and 95% growth increase (T_{95}), the maximum slope (S_{max}), the time at maximum slope (TS_{max}), the initial value (MIN), the final value (MAX), the initial exponential growth rate (r), and the initial doubling time (T_{doubt}).

Depending on the data type, other fit models may be used, such as linear, logarithmic, exponential, hyperbolic, Gaussian, Gompertz, Michaelis-Menten, power function, etc., each resulting into specific parameters that describe the fit.

In BioNumerics, analysis and comparison of curve type data can be done on one or more parameters derived from the curve fit. For example, if one uses S_{max} and MAX , each curve is translated into two character values. Figure 7.1.2 illustrates in a schematic way how a hypothetical test panel (in the example containing 6 tests) is processed into a data matrix in BioNumerics. Each test results in 5 readings (1), through which a curve is fitted, using an appropriate model (2). The *Logistic Growth* model is used in the example. For a given model, one or more characteristic parameters can be derived from the curves. In the example, the *maximum slope* S_{max} and the *final value* MAX are calculated (3). This leads to two data matrices, each containing one value per test and per organism or sample (4).

For taxonomy or typing purposes, one might be interested in combining the data from multiple parameters into one clustering or identification. In BioNumerics, it is possible to specify a comparison coefficient for each used parameter separately. The software then averages the respective similarity values into one similarity value per pair of entries compared. An important issue is that the parameters used can have different ranges, as is the case in the example in Figure 7.1.2. If a coefficient is chosen that has no inherent

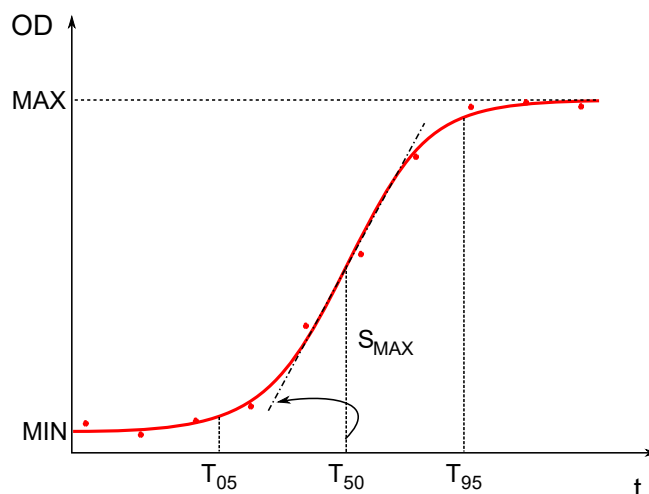


Figure 7.1.1: Trend curve that follows the logistic growth model and some derived parameters. T_{05} , T_{50} and T_{95} are the times at 5%, 50% and 95% growth increase, respectively. S_{max} is the maximum slope, MIN is the initial value, and MAX is the final value.

scaling, e.g. Euclidean distance, an appropriate range should be specified for each parameter, so that the weights of the different parameters are standardized when they are combined by averaging.

7.1.2 Defining a new trend data type

To create a new trend data type, highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** (➕). In the *Create a new experiment type* dialog box, click on **Trend data type** and press <OK>. This will display the *New trend data type* wizard (see Figure 7.1.3).



To be able to work with trend data type experiments, the Trend data module (TD) needs to be present in your BioNumerics configuration.

The wizard prompts you to enter a **Trend data type name**. Enter a name for the new trend data type and press <Finish> to complete the setup of the new trend data type. The new trend data experiment will appear in the *Experiment types* panel of the *Main* window.

7.1.3 Editing a trend data type

7.1.3.1 The Trend data type window

All settings that are relevant for a certain trend data type experiment can be accessed through its *Trend type* window. This window can be called from the *Main* window by clicking on the trend data type experiment in the *Experiment types* panel and selecting **Edit > Open highlighted object...** (🔍, Enter) or simply by double-clicking on the trend data type experiment.

The *Add new trend curve* dialog box prompts for a trend curve name. Enter a name for the new curve and press <Finish> to add the curve to the database.

The new trend curve is displayed in the *Curves* panel of the *Trend type* window (Figure 7.1.5). Optionally, a description for the trend curve can be entered in the **Description** field.

A selected trend curve can be renamed with **TrendCurves > Rename trend curve...**. This calls the *Rename*

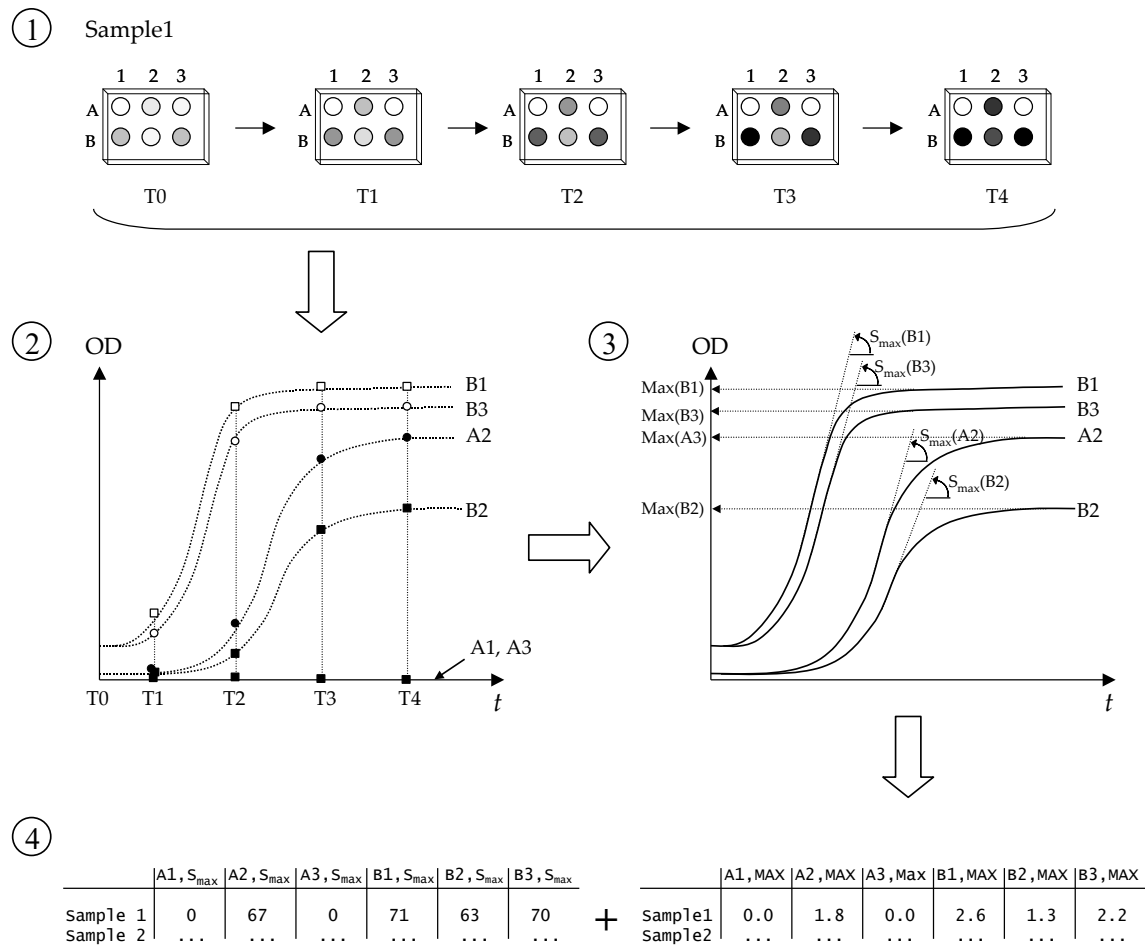


Figure 7.1.2: Example of the processing of kinetic readings of a phenotypic test panel. (1) Readings are done at different times $T_0 \dots T_4$; (2) A curve model is fit through the values obtained for each well in the test panel (in the example, Logistic Growth); (3) One or more specific parameters are derived from the curves (in the example, the final value MAX and the maximum slope S_{max}); (4) A data matrix is constructed from a curve parameter obtained for each well, including all the samples analyzed. In the example, two data matrices are generated as two parameters were chosen.

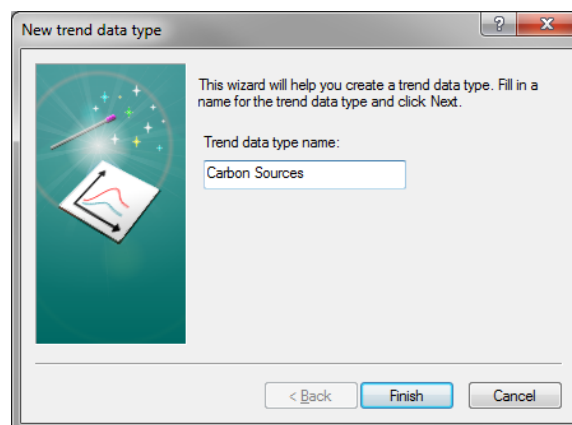


Figure 7.1.3: The *New trend data type* wizard.

trend curve dialog box (see Figure 7.1.6).

Once a trend data type is created, a *trend curve* has to be defined for each character that results into multiple

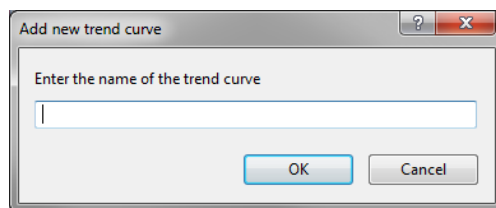


Figure 7.1.4: The *Add new trend curve* dialog box to add a new trend curve.

readings. Figure 7.1.5 illustrates an example of an experiment consisting of 6 characters so that 6 trend curves will be calculated. In reality, a trend data experiment might as well exist of 96 characters, for example if a micro titer plate with 96 tests is read as kinetic data.

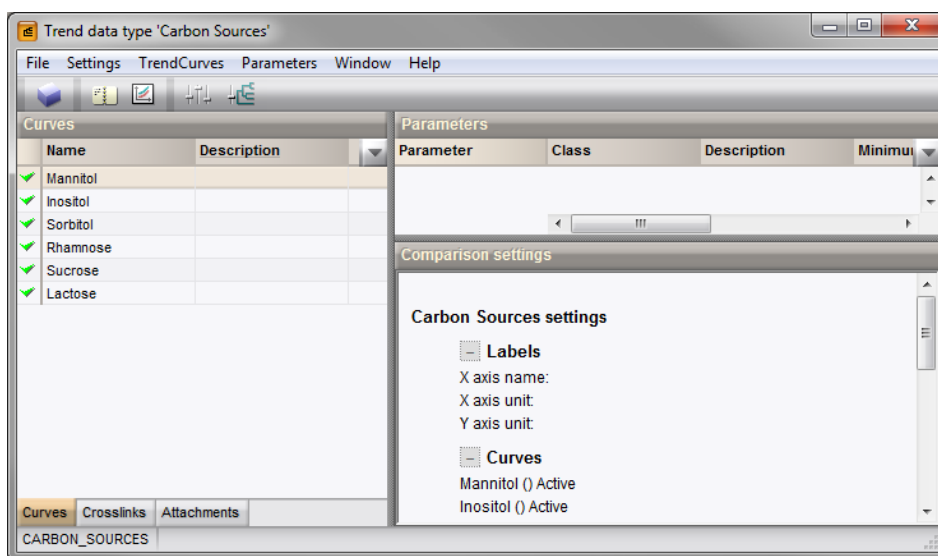


Figure 7.1.5: The *Trend* type window, with six trend curves defined.

A new trend curve is created with **TrendCurves > Add new trend curve....** This calls the *Add new trend curve* dialog box (see Figure 7.1.4).

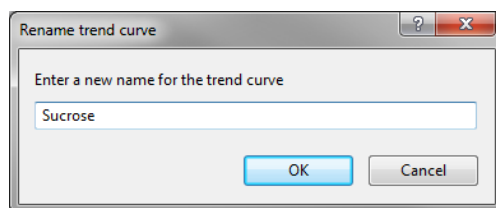


Figure 7.1.6: The *Rename trend curve* dialog box.

Enter a new name for the new curve and press **<OK>** to save the changes to the database.

Before any analysis can be done, parameters have to be defined, derived from the appropriate model curves. The *Trend curve parameters* dialog box is called with **Parameters > Model parameters...** (see Figure 7.1.7).

All available models are displayed in the left panel of the *Trend curve parameters* dialog box. A check box **Use model** allows the model to be included or not. If a model is selected, a number of parameters, listed under **Active parameters**, can be chosen to be included in the analysis. Additionally, one or more choices (**Model choices**) can be specified for each model (see 7.1.4 for an overview of the models and parameters). The *Trend curve parameters* dialog box is closed with the **<Exit>** button. The parameters appear in the

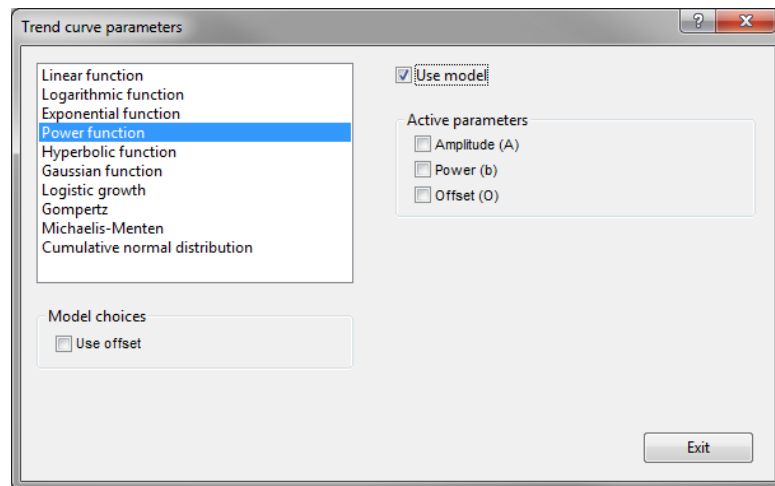


Figure 7.1.7: Select the models to use and the associated parameters to include.

Parameters panel of the *Trend* type window.

A selected parameter is removed from the list with ***Parameters* > Remove parameter...**

For a selected parameter, the range can be specified with ***Parameters* > Parameter properties...** This action calls the *Parameter properties* dialog box (see Figure 7.1.8).

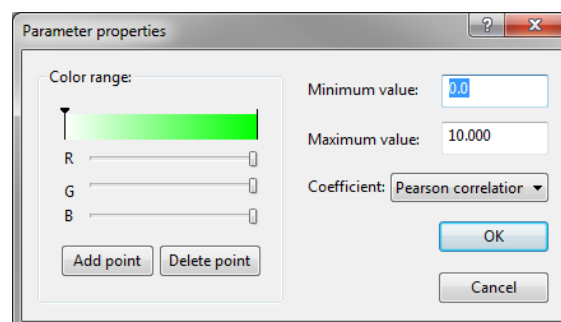


Figure 7.1.8: The *Parameter properties* dialog box.

Under ***Color range***, the left and right ends of the color scale can be selected and a color can be assigned. R, G, and B represent the red, green and blue component, respectively. Using **<Add point>**, intermediate nodes can be added and assigned a color. The color range specified will be used in the *Experiment data* panel of the *Comparison* window (see 7.2).

For each parameter, a similarity coefficient can be selected from the drop-down menu (***Pearson correlation***, ***Cosine coefficient***, ***Euclidean distance***, ***Canberra metric***). This coefficient will be used to calculate the associated data matrix when the option ***Parameter similarity*** is selected in the *Comparison settings* wizard (see 7.2 for a detailed explanation).

The ***Minimum value*** and ***Maximum value*** (range) of the parameter are important if the ***Euclidean distance*** coefficient is chosen, which has no inherent scaling. Using the ranges, the data values are normalized so that distance values from different parameters have equal weights.

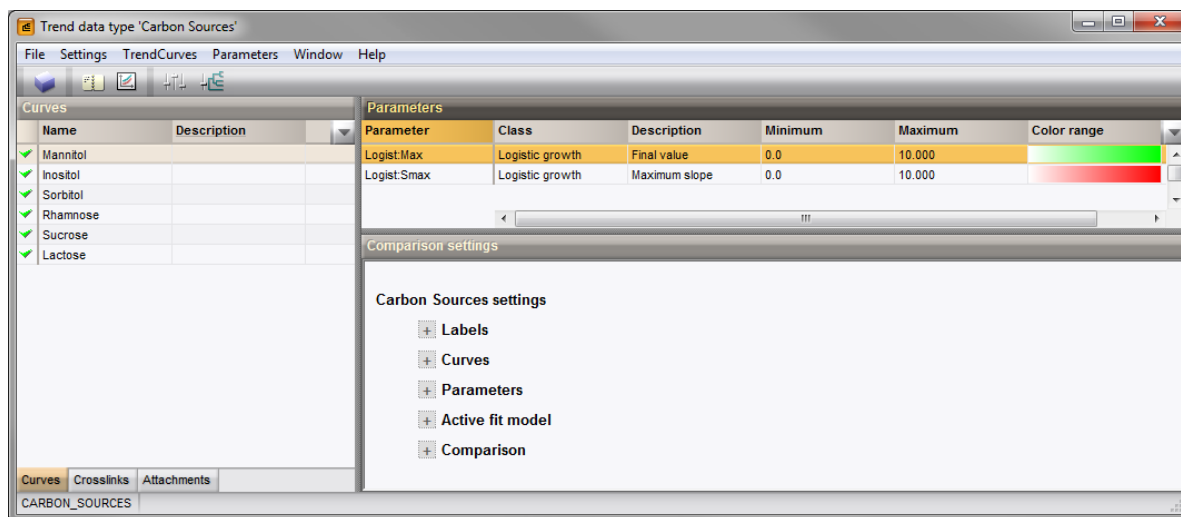


Figure 7.1.9: The *Trend* type window with two model parameters defined.

7.1.4 Trend curve models and parameters

7.1.4.1 Linear function

$$y = A + Bx$$

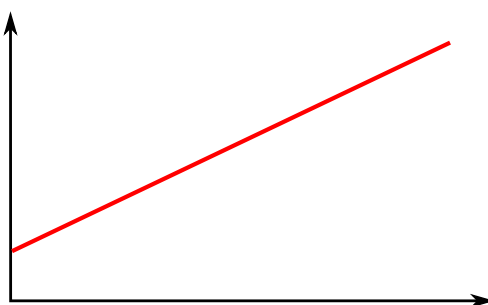


Figure 7.1.10: Linear function.

Available parameters are the *Intercept* (A) and the *Slope* (B). The function can be forced to pass through zero, in which case the intercept A is always zero.

7.1.4.2 Logarithmic function

$$y = A + B \log x$$

Similar as for a linear function, the available parameters are the *Intercept* (A) and the *Slope* (B).

7.1.4.3 Exponential function

$$y = O + Ae^{rx}$$

The function offers the *Amplitude* (A) and the *Exponential* (r) as parameters. If the model choice *Use offset* is checked, the *Offset* (O) is a parameter as well.

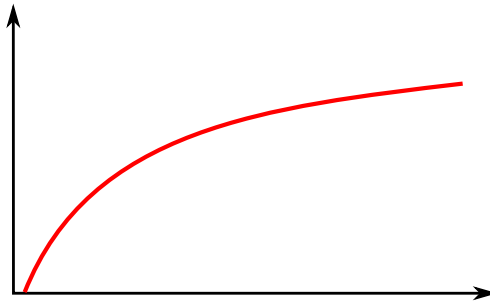


Figure 7.1.11: Logarithmic function.

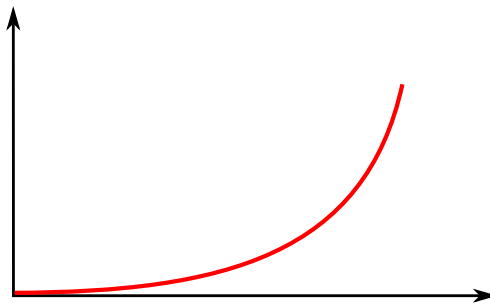


Figure 7.1.12: Exponential function.

7.1.4.4 Power function

$$y = O + Ax^B$$

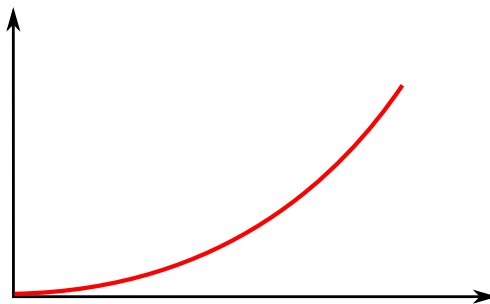


Figure 7.1.13: Power function.

The function offers the **Amplitude** (A) and the **Power** (B) as parameters. If the model choice *Use offset* is checked, the **Offset** (O) is a parameter as well.

7.1.4.5 Hyperbolic function

$$y = A + \frac{B}{x - C}$$

This model offers the **Offset** (A) and the **Amplitude** (B) as parameters. As a choice, an asymptote can be fitted (*Fit asymptote*). In this case, a **Pole** (C) is a parameter as well, with $C \neq 0$.

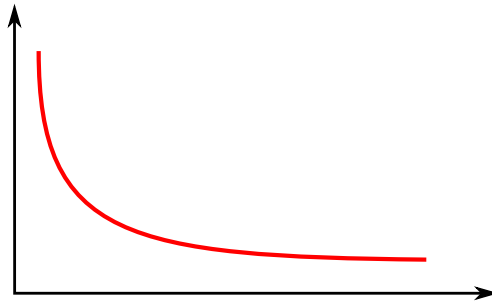


Figure 7.1.14: Hyperbolic function.

7.1.4.6 Gaussian function

$$y = O + Ae^{-\left(\frac{x-M}{S}\right)^2}$$

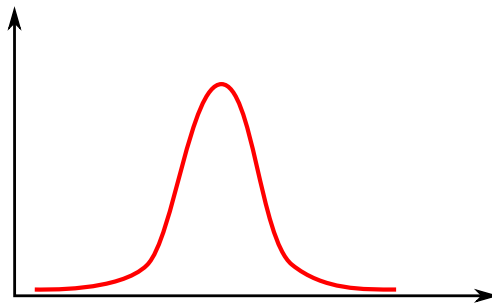


Figure 7.1.15: Gaussian function.

The Gaussian model offers the *Amplitude* (A), the *Position of center* (M), the *Width of the Gaussian* (S) and the *Offset* (O) as parameters.

7.1.4.7 Logistic growth function

$$y = A + \frac{C}{\left[1 + e^{Q-B(X-M)}\right]^{\frac{1}{\theta}}}$$

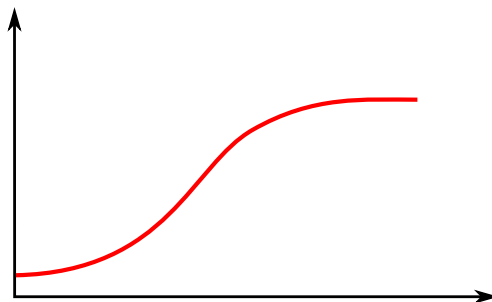


Figure 7.1.16: Logistic growth function.

Following are the parameters for *Logistic growth*:

- The **Initial value (Min)**, i.e. the minimum value derived from the curve
- The **Final value (Max)**, i.e. the maximum value derived from the curve
- The **Initial exponential growth rate (r)**
- The **Initial doubling time (Tdoubl)**, which is the time needed for y to double
- The **Maximum slope (Smax)**, the maximum growth rate of y
- The **Time at maximum slope (TSmax)**, i.e. the x-value at maximum slope
- **Time at 5%, 50% and 95% growth** are the x values at 5%, 50% and 95% growth of the y value, respectively.

If the model choice *Use offset* is not checked, the A value becomes zero in all cases. If *Use generalized formula* is not checked, the value Q becomes zero in all cases.

7.1.4.8 Gompertz function

$$y = A + Ce^{-[e^{B(X-M)}]}$$

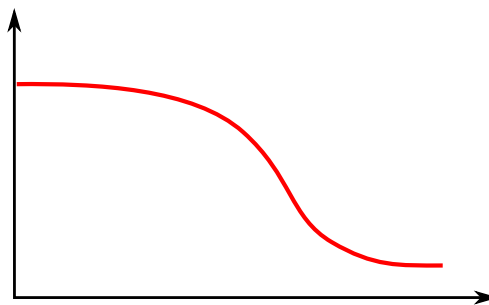


Figure 7.1.17: Gompertz function.

Following are the parameters for **Gompertz**:

- The **Initial value (Min)**, i.e. the minimum value derived from the curve
- The **Final value (Max)**, i.e. the maximum value derived from the curve
- The **Maximum slope (Smax)**, the maximum growth rate of y
- The **Time at maximum slope (TSmax)**, i.e. the x-value at maximum slope
- **Time at 5%, 50% and 95% growth** are the x values at 5%, 50% and 95% growth of the y value, respectively.

If the model choice *Use offset* is not checked, the A value becomes zero in all cases.

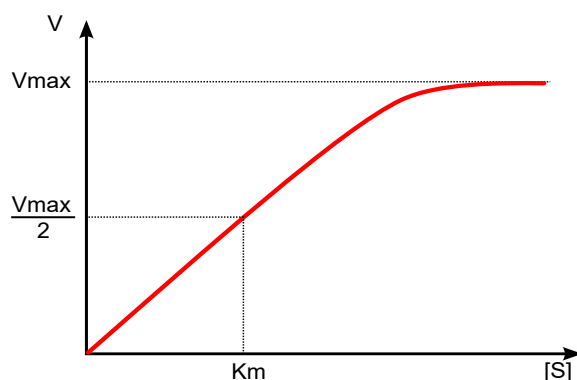


Figure 7.1.18: Michaelis-Menten function.

7.1.4.9 Michaelis–Menten function

$$V_0 = A + \frac{V_{max}[S]}{K_m + [S]}$$

Following are the parameters for *Michaelis-Menten*:

- The **Maximum** (*Vmax*), i.e. the maximum reaction rate in case of non-limiting substrate concentration ([S]).
- The **Michaelis constant** (*Km*)

If the model choice *Use offset* is not checked, the *A* value becomes zero in all cases.

7.1.5 Importing trend data

7.1.5.1 Introduction

With the *Import trend data* import routine, listed under the topic *Trend data type data* in the *Import* dialog box (see Figure 7.1.19), data from text and csv files containing a set of measurements taken under a series of conditions, such as time or concentration can be imported in the database and linked to new or existing database entries.

Each file should contain a well-defined table. The first row should be the header row describing the curve (character) names, and one of the columns should contain the X-values. The other columns should contain the Y-values for the curve named in the column header (see Table 7.1.1).

X name	Curve 1	Curve 2	...
X value 1	Y value	Y value	...
X value 2	Y value	Y value	...
...

Table 7.1.1: Trend data format.

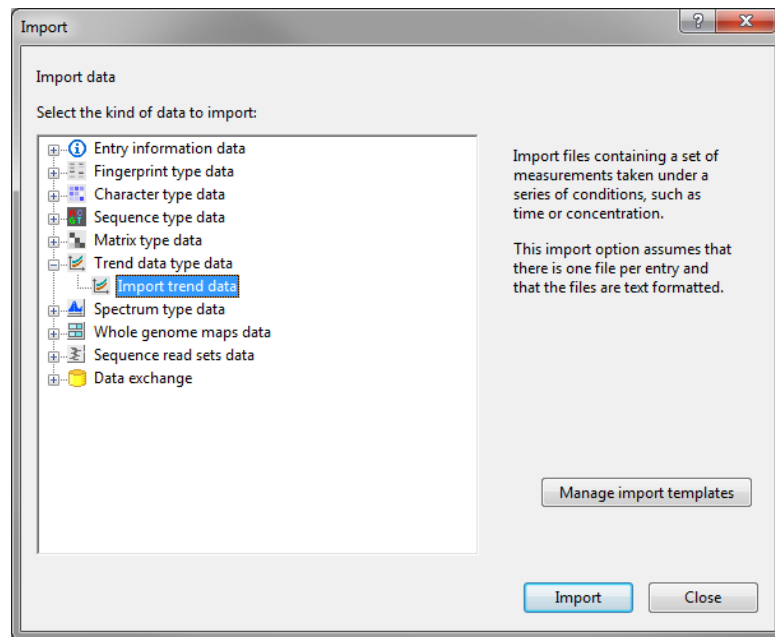
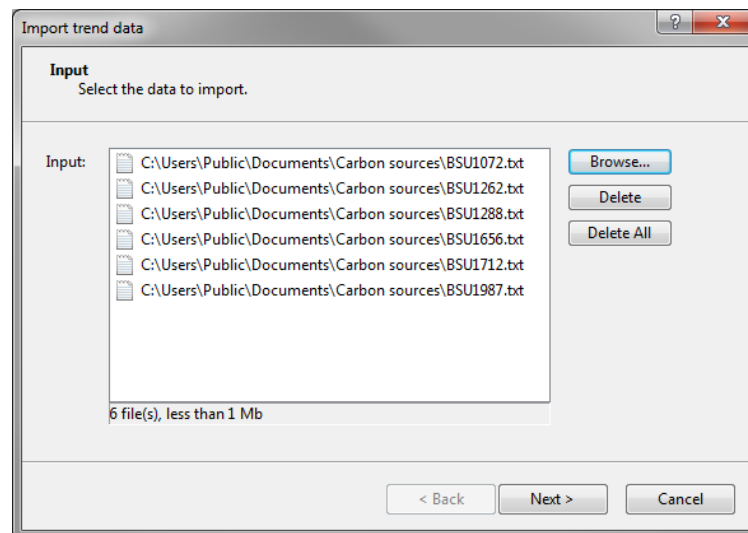


Figure 7.1.19: Importing trend data.

7.1.5.2 The Import wizard

Selecting *Import trend data* under *Trend data type data* in the *Import* dialog box and pressing <*Import*> opens the *Input* wizard page.

Figure 7.1.20: The *Input* wizard page.

Pressing the <*Browse*> button allows you to select the file(s) that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. The number of files and total size is displayed below the list.

With the <*Delete*> button all selected files are removed from the import list. All files are deleted at once from the import list when pressing <*Delete All*>.

Pressing <*Next*> will display the next step.

When importing trend data for the first time in the database, the *Import rules* dialog box will open, otherwise

the *Import template* wizard page will open.

If the import routine is unable to open the selected file(s), an error is generated.

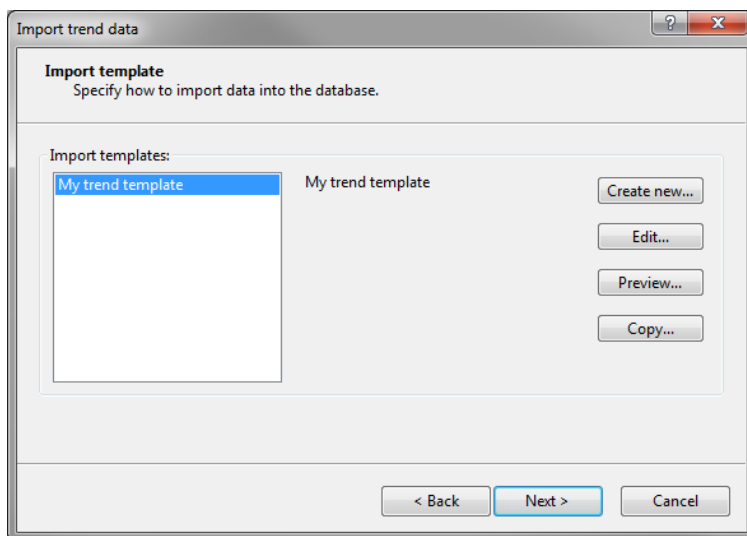


Figure 7.1.21: The *Import template* wizard page.

The way the trend information should be imported in the database can be specified with an import template. The *Import templates panel* lists all import trend data templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up a new dialog box, allowing you to define a new import template.

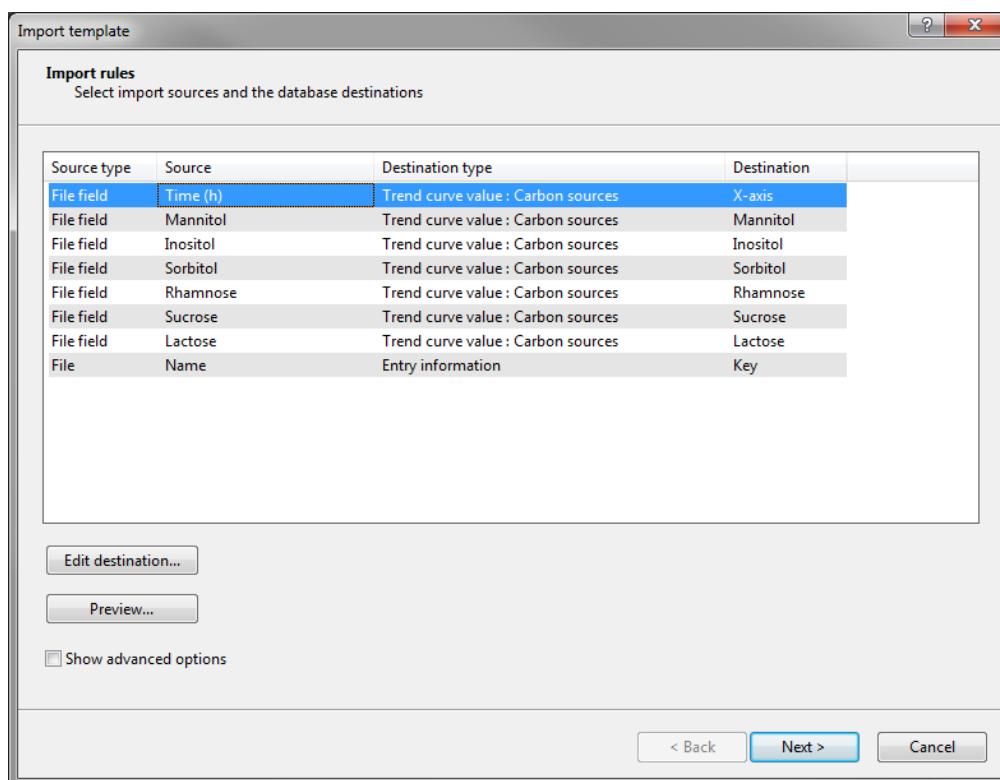


Figure 7.1.22: The *Import rules* dialog box.

Each column found in the selected file(s) corresponds to a row in the grid (column 1 in the file corresponds to row 1 in the grid, column 2 corresponds to row 2, etc.). The text **File field** is specified in the **Source type** column and the column names are displayed in the **Source** column.

Using the last row in the grid, the (parsed) file name of the selected files can be stored in the database. The text **File** is specified in the **Source type** column and the text **Name** is displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields, trend curve X or Y values or trend data information fields. Initially the rows are not linked to any information in the database (the **Destination type** and **Destination** for all rows is set to **<None>**).

Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

Pressing the **<Preview>** button opens a new dialog box displays the parsed information using the template settings.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

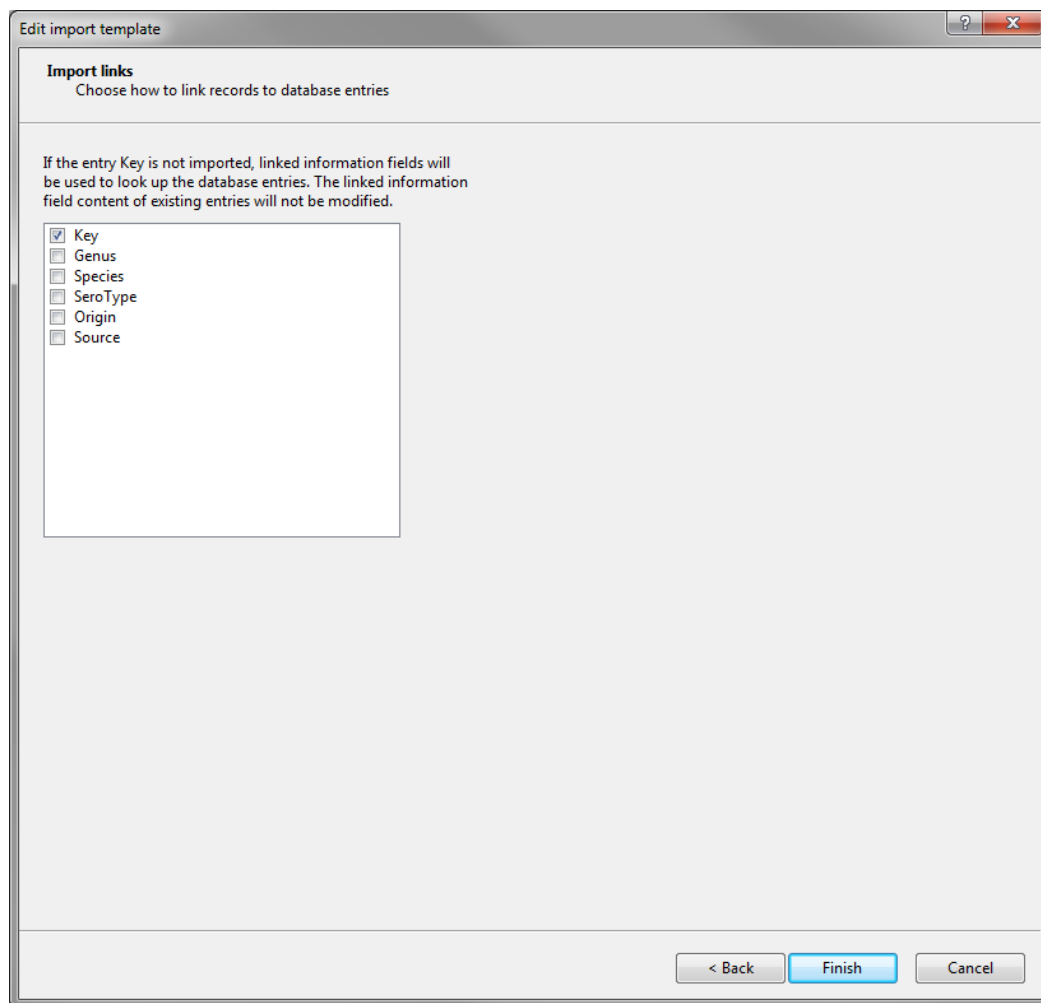


Figure 7.1.23: Specify the entry link field.

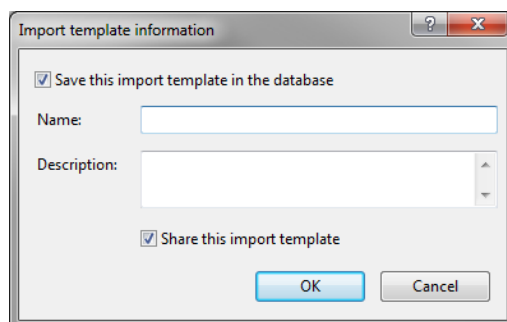


Figure 7.1.24: The *Import template information* dialog box.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template

can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

Pressing **<Next>** opens the last step of the wizard, prompting for some final settings.

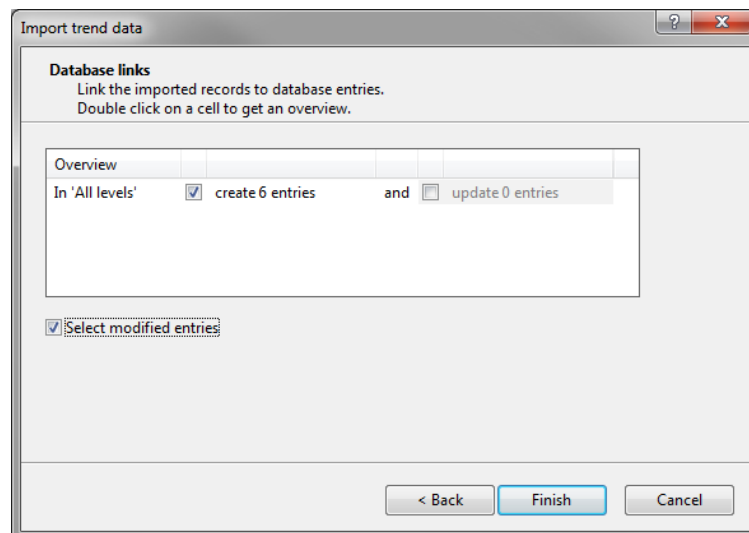


Figure 7.1.25: The *Database links* wizard page.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the entry and character information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database.

Pressing **<Finish>** will start the import.

7.1.6 Displaying trend data

For visualization and comparison purposes, a default *curve fit model* will have to be chosen for a particular trend data type. A default curve model can be chosen for the trend data type by selecting **Settings > Default trend curve model...** in the *Trend type* window. This calls the *Trend curve fit model* dialog box (see Figure 7.1.26).

The *Trend curve fit model* dialog box lists the available models and regressions (left) and their additional parameters (right). Press **<OK>** to set the parameters. The selected model and parameter(s) are displayed in the *Comparison settings* panel under **Active fit model**.



This choice determines the fit model used for visualization of trend curves, and also the fit model used for curve comparisons, in case the fit model is used instead of the raw data values (see 7.2.1).

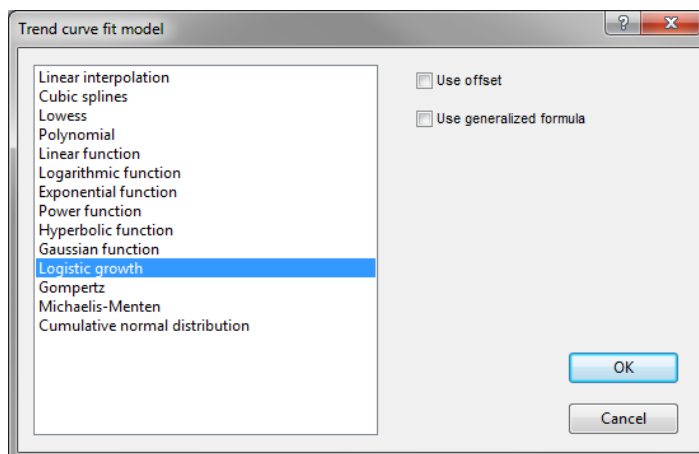


Figure 7.1.26: The *Trend curve fit model* dialog box.

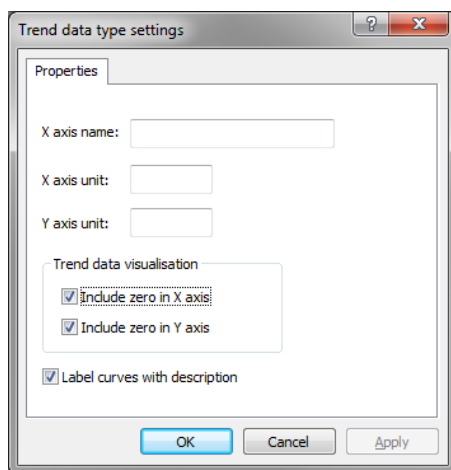



Figure 7.1.27: The *Curve settings* dialog box.

Selecting **Settings > General settings...** () calls the *Curve settings* dialog box (see Figure 7.1.27).

An X axis denominator can be provided in the **X axis name** text box and units for X and Y axes entered in the **X axis unit** and **Y axis unit** text boxes, respectively.

Additional visualization settings can be specified: **Include zero in X axis** and **Include zero in Y axis**. If these settings are checked (enabled), the zero on the X axis and the Y axis, respectively, will always be shown on the plot, irrespective of the ranges of the components.

With **Label curves with description** checked, the information in the "Description" field (if provided) will be used to label the trend curve instead of its name.

Pressing <OK> saves the settings to the database. The labels are displayed in the *Comparison settings* panel under the first topic.

When clicking on a colored dot in the *Experiment presence* panel that represents the trend data type for a particular entry, the curves for the selected entry are displayed in a *Experiment card* window (Figure 7.1.28).

The name of the entry to which the curves belong (i.e., the key) is written in the status bar. The box can be resized in the bottom right corner. The box can be moved by clicking and holding down the left mouse button anywhere inside the borders.

Upper left in the *Experiment card* window is a pull-down list where you can choose which curves to display. The default setting is <All>. You can select any particular curve by clicking inside the list box and selecting

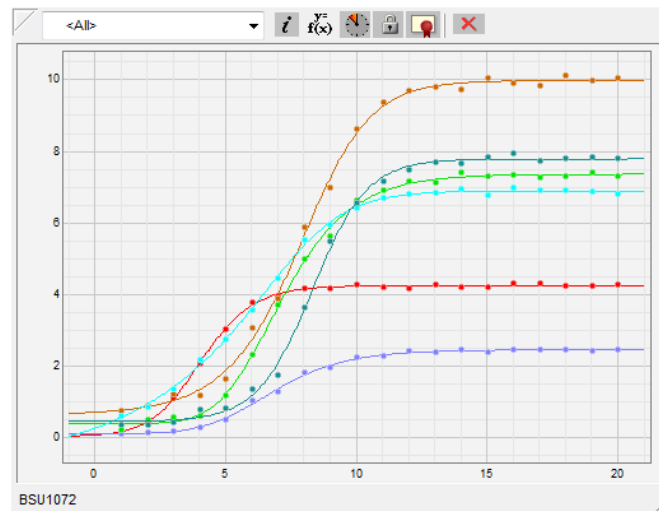





Figure 7.1.28: The *Experiment card* window, showing trend data curves for one entry.


one of the curves present in the data type.


Using the  button, you can toggle between the curve view as depicted in Figure 7.1.28 and the info view, which contains detailed information about:

- The fit model chosen for visualization, the standard deviation and the parameters derived from the formulas (see 7.1.2) are indicated.
- The curve parameters selected for comparison (see 7.1.2).

With the  button, another regression or curve fit can be specified for the present set of curves. This choice only applies to the currently open *Experiment card* window, and will not influence the default curve fit defined with *Settings > Default trend curve model...*

Using the button , it is possible to remove the curves for the selected entry from the database.

The *Experiment card* window can be closed by clicking on the triangular button in the upper left corner (.

To compare curves between different entries, trend curves can be displayed for multiple entries at a time in the same window. Make a selection of entries in the database for which trend curves are present, open the *Trend type* window and choose *File > Create trend data window* (.

The resulting *Trend data* window (see Figure 7.1.29) contains three dockable panels: the *Curves panel*, the *Entries panel* and the *Trend curve details panel*.

By default, the *Curves panel* displays all curves for all selected entries in a single plot, labeled by trend curve (*View > Label by trend curve*).

When choosing the option *View > Label by entry* the curves have different colors according to the entries, as indicated in the legend box (see Figure 7.1.29).

With *View > Label by selection* the curves are labeled based on the entry selection status. Curves of selected entries are shown in blue, curves of unselected entries in black.

Inside the window, a legend shows the colors and the names (or description) of the corresponding curves. This legend is a box that can be moved inside the *Trend data* window.

Similar as in the *Experiment card* window, a pull-down list box allows either all trend curves or one particular curve to be displayed (see Figure 7.1.29).

A data point and its corresponding curve can be selected by clicking it with the mouse. The selected dot

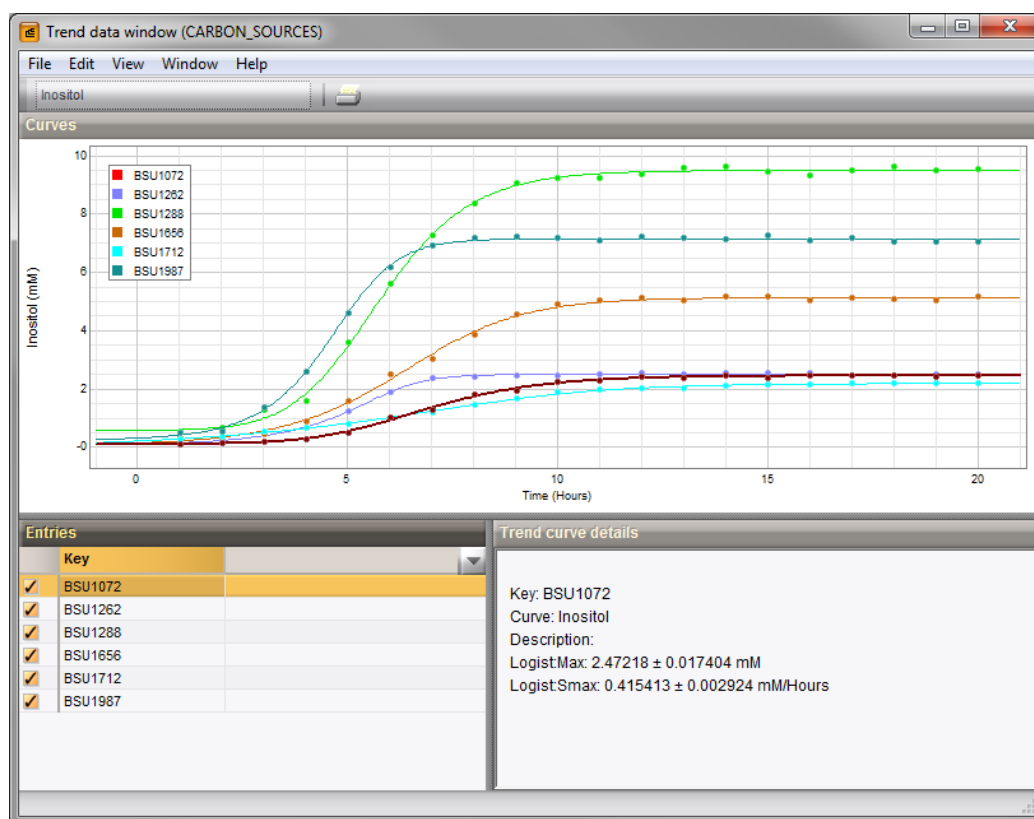


Figure 7.1.29: The *Trend data* window, displaying trend curves for multiple entries.

will be displayed within a colored square and the selected curve in a thicker, darker line. The corresponding entry will be highlighted in the *Entries* panel.

The entries can also be queried interactively from this window:

- By double-clicking on a dot (data point) of a curve, the *Entry* window of the entry to which the curve belongs will pop up.
- By holding down the **Ctrl**-key and clicking on a dot (**Ctrl+click**), the entry will be selected or unselected in the database.

With **View > Use colors**, one can toggle between the color view and a black-and-white view, in which the data points of the different curves are represented by different symbols such as circles, squares, and triangles.

In the **Label by selection** view in black-and-white, selected entries are represented by a filled circle, whereas non-selected entries are represented by an open circle.

A selected data point can be edited in the *Trend data* window with **Edit > Edit selected point...** (**Ctrl+ENTER**). This pops up the *Edit trend point* dialog box (see Figure 7.1.30).

A new *X value*, *Y value* and/or *Error value* can be entered in the respective text boxes. Pressing **<OK>** will update the *Curves* panel with the new values for the edited data point.

To add a data point to a curve, make sure the curve is selected and choose **Edit > Add new point...** (**INSERT**). This will open the *Edit trend point* dialog box (see Figure 7.1.30), in which the new values can be entered.

An aberrant data point can be deleted with **Edit > Delete selected point** (**DEL**). A confirmation dialog box will appear.

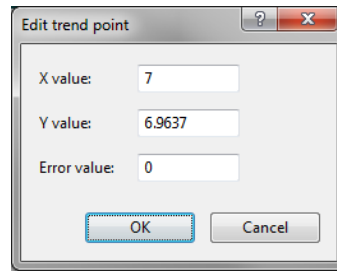


Figure 7.1.30: The *Edit trend point* dialog box.

The modified curve data is saved to the database upon selecting **File > Save changes**. When curve data has been modified and you attempt to close the *Trend data* window, a warning will appear, asking you if the modifications should be saved or not.

The image can be copied to the clipboard with **File > Copy to clipboard**.

7.1.7 Comparison settings

The comparison settings for a trend data type can be accessed with **Settings > Comparison settings...** (🔗) in the *Trend type* window, but also in the *Comparison* window. See 7.2 for a detailed explanation.

7.1.8 Additional trend data comparison parameters

In the *Trend type* window (Figure 7.1.5), it is possible to define additional comparison parameters. Selecting **Parameters > Statistics parameters...** calls the *Statistics parameters* dialog box (see Figure 7.1.31).

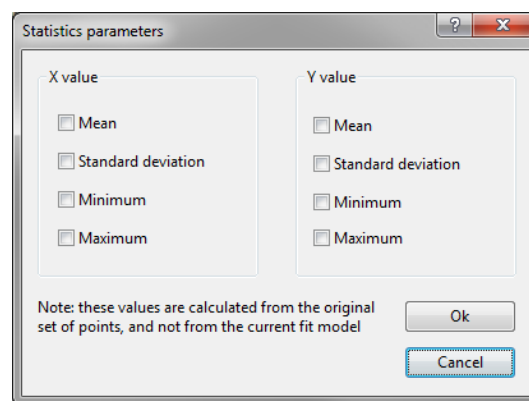


Figure 7.1.31: The *Statistics parameters* dialog box.

From this dialog box, the **Mean**, **Standard deviation**, **Minimum** and **Maximum** value of the X and Y component can be added as comparison parameters. The statistics parameters are not model-based, but are instead calculated on the original data points.

With **Parameters > Add value...**, a Y value can be included that corresponds to a fixed X value. The *Add model value parameter* dialog box opens (see Figure 7.1.32).

The *Add model value parameter* dialog box prompts to enter an X value. An Y value will be included that corresponds to this fixed X value.

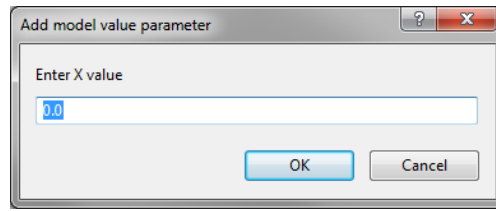


Figure 7.1.32: The *Add model value parameter* dialog box.

Selecting **Parameters > Add inverted value...** calls the *Invert curve parameter* dialog box (see Figure 7.1.33).

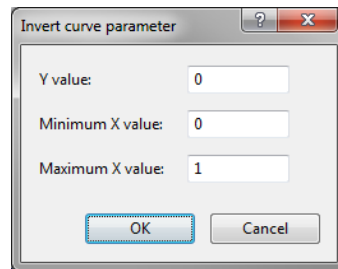


Figure 7.1.33: The *Invert curve parameter* dialog box.

The dialog prompts for the **Y value**. In addition to the Y value, the input box prompts you for a **Minimum X value** and **Maximum X value** to be reported. Pressing <OK> will include the X value that corresponds to the Y value.

A minimum or maximum Y value within a range of X values can be added as parameter with **Parameters > Add minimum value in range...** or **Parameters > Add maximum value in range...**, respectively.

Similarly, the X value that corresponds to the minimum or maximum Y value found within a range of X values can be added with **Parameters > Add minimum position in range...** or **Parameters > Add maximum position in range...**, respectively.

All these actions call the *Prompt curve X range* dialog box (see Figure 7.1.34).

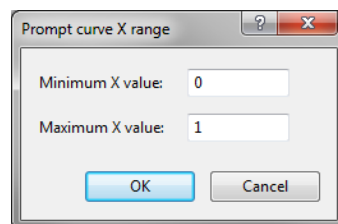


Figure 7.1.34: The *Prompt curve X range* dialog box.

A minimum and a maximum value should be specified in the input boxes.

A slope can be calculated that corresponds to a fixed X value with **Parameters > Add slope...**. This calls the *Add model slope parameter* dialog box (see Figure 7.1.35).

An input box prompts to enter an X value. A slope will be calculated that corresponds to this fixed X value.

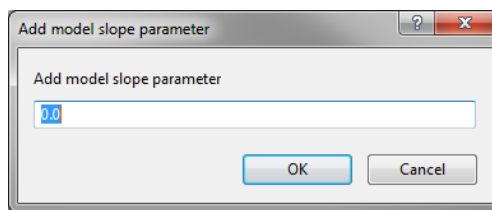


Figure 7.1.35: The *Add model slope parameter* dialog box.

Chapter 7.2

Cluster analysis of trend data

7.2.1 Trend data comparison settings

Please note that to perform a cluster analysis on trend data, the Trend data module (TD) and the Tree and network inference module (TN) need to be present in your BioNumerics configuration.

The comparison settings for a trend data type include the coefficients and clustering methods used for creating dendrograms from trend data. These settings can be changed from the *Trend type* window (see 7.1.7), or in the *Comparison* window, by clicking on the trend data type in the *Experiments* panel and selecting **Clustering > Calculate > Cluster analysis (similarity matrix)...**

As a result, the *Comparison settings* wizard for trend data types pops up (Figure 7.2.1). The *Comparison settings* wizard allows one to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient.

The hierarchical representation on the left provides an overview of the available coefficients for trend data types. The coefficients are subdivided in two categories: **Curve based** and **Parameter based**. Each of the categories can be collapsed by clicking on the small "-" (minus) sign that precedes the category name.

The **Curve based** category lists coefficients that calculate similarity on the curves directly.

The **Pearson correlation** coefficient is calculated as:

$$r = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \sqrt{\sum_{i=1}^n y_i^2 - \frac{1}{n} (\sum_{i=1}^n y_i)^2}}$$

with n the number of points in the trend data curve and x_i and y_i the i^{th} trend data point in the curve of entries x and y , respectively.

The **Cosine correlation** coefficient is calculated as:

$$r = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

The **Euclidean distance** coefficient is calculated as:

$$D = \frac{1}{1 + d}$$

with

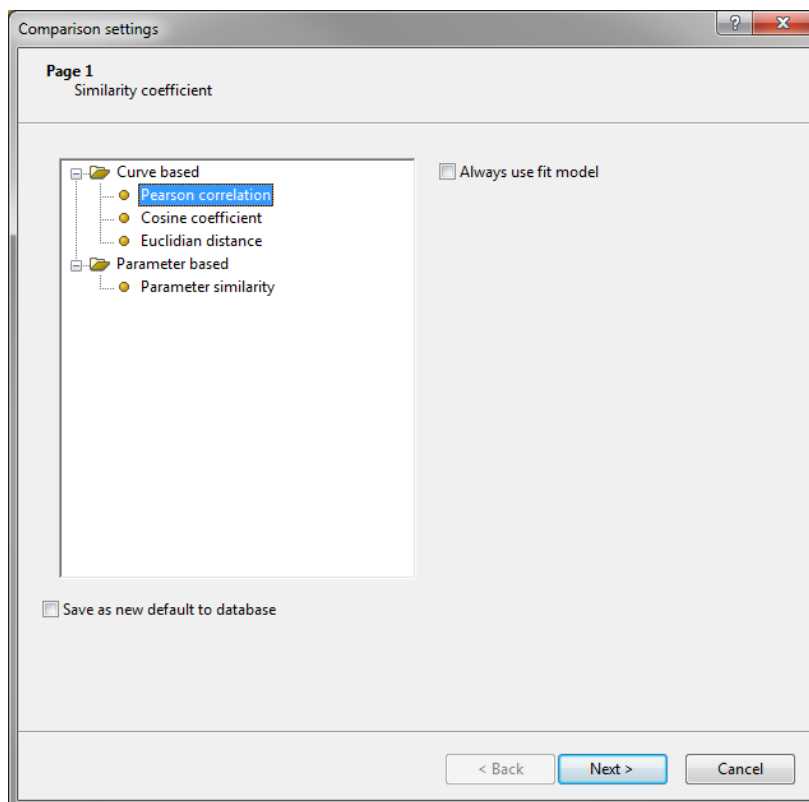


Figure 7.2.1: The *Similarity coefficient* wizard page for trend data, which deals with the choice of the similarity coefficient.

$$d = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2}$$

By default, similarity values are calculated on the raw curve, i.e. on the original input data values. When the option *Always use fit model* is checked, the fit curve as produced from the default trend curve fit model used (see 7.1.6).

The *Parameter based* category lists only *Parameter similarity*. Using this coefficient, the similarity or distance is calculated from the parameters defined for the experiment type (see 7.1.2 and 7.1.8). The way parameter values are processed into data matrices is illustrated in Figure 7.1.2: each parameter defined leads to a data matrix with the number of characters (values per entry) defined by the number of curves defined for the experiment type. In case two parameters are defined, two data matrices are generated. For each parameter, a separate coefficient can be chosen to analyze the associated data matrix (Figure 7.2.2). The obtained similarity matrices are averaged and a dendrogram is calculated.

Check *Save as new default to database* if you want the specified comparison settings to be saved in the database as default settings.

If a similarity matrix already exists for the selected experiment, an option *Keep existing similarity matrix* appears. When checked, the previously calculated similarity matrix will be used and all coefficient options will appear gray (disabled).

The *Cluster analysis* wizard page deals with the calculation of a dendrogram from the similarity matrix and is discussed in 13.2.6.

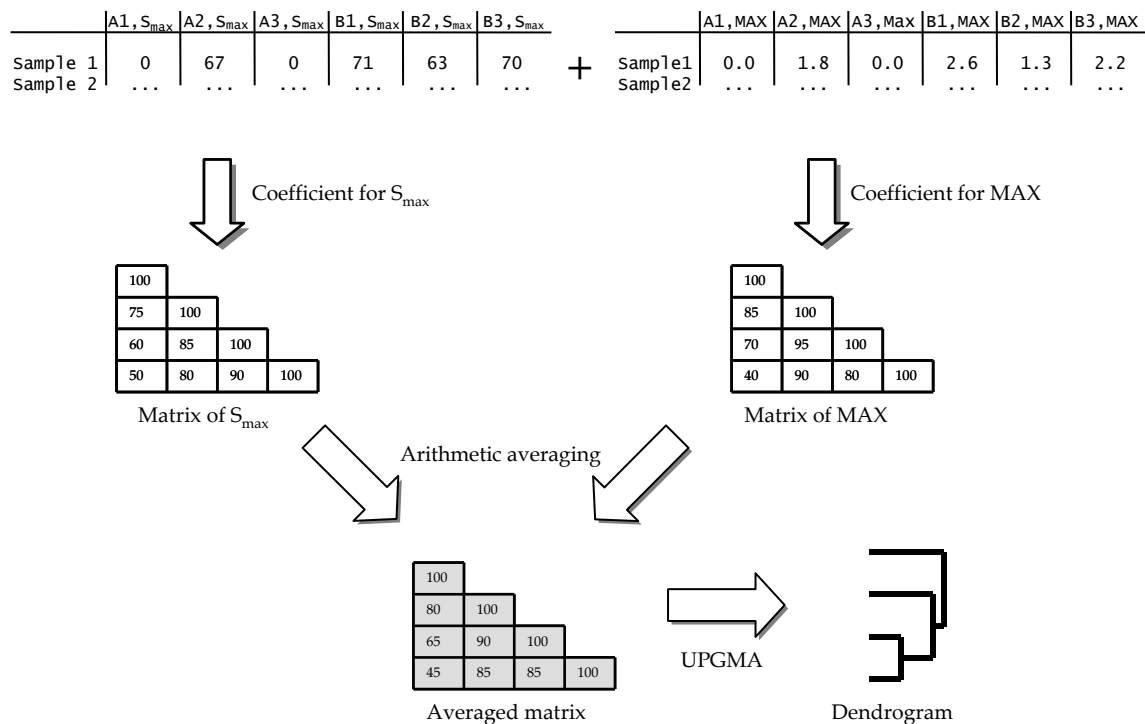


Figure 7.2.2: Schematic representation of parameter-based cluster analysis of trend data. This example, where two parameters were defined, is a continuation of the processing scheme presented in Figure 7.1.2.

7.2.2 Display options for trend data

A number of options related to trend data are categorized under the menu **Trend Data** in the *Comparison* window.

With **TrendData > Show parameter colors** , you can have the values of the parameters displayed as colors, as defined in the *Trend type* window (see 7.1.2).

With **TrendData > Show parameter values** , the values of the parameters are displayed as numerical values.

A combination of the two above options is obtained with **TrendData > Show parameter values & colors** .

With **TrendData > Order by parameter**, the parameter list can be ordered by grouping the same parameters from the different curves together.

Alternatively, with **TrendData > Order by trend curve**, the parameter list is ordered by grouping the different parameters from the same curves together.

For a selected parameter, the entries can be sorted according to increasing value using **TrendData > Sort entries by parameter value** .



The separator bar between the parameter names and the values can be dragged down if the names are not completely visible.

A *Trend data* window can be created from the entries contained in the comparison with **TrendData > Create trend data window** . To calculate correlation and regression on trend data, open a *Trend analysis* window with **TrendData > Perform trend analysis** . For more information about trend analysis, see 7.3.

7.2.3 Exporting trend data

A tab-delimited text file of the entries and trend data values contained in the current comparison can be exported with *TrendData* > *Export character table*.

Chapter 7.3

Trend analysis

7.3.1 Introduction

7.3.1.1 Exploring trends among trend data

In a number of experimental setups, several parameters of a study object are recorded over a certain period of time. Possible study objects can be very diverse in nature, such as an ecosystem (e.g. a soil mesocosm, a pond or a greenhouse), a patient or patient group, an individual cell, etc. In this kind of study, researchers often are looking for trends among the different parameters that are recorded, i.e. whether the parameters are *correlated* in some way. Since all parameters are monitored in function of time, they can be treated as *trend data* in BioNumerics (see 7.1).



Although monitoring in function of time might be by far the most common case, monitoring could also be done in function of any other continuous parameter, such as temperature, pH, light intensity, etc.

The easiest way to visually examine if any trends exist among trend curves, is to simply plot all curves on the same axes (coordinate system). However, a number of situations can be envisaged that hamper such a visual observation of trends. For example, one parameter, which might be measured more frequently than another, could display an additional trend with a smaller period that obscures an underlying trend (e.g. a circadian rhythm superposed on seasonal fluctuations). In other cases, there can be a certain lag time before a parameter reacts to an environmental change (e.g. rainfall in a desertified ecosystem will cause soil humidity to rise almost instantaneously, while a parameter such as the number of new seedlings will only increase after a few days). For the above reasons, it might be needed to apply a *function* to a trend curve (see 7.3.2.4), which will transform the curve to one that can be better compared with others.

Mathematically, trends are quantified by calculating the *correlation* and by performing a *regression analysis*.

Please note that the functionality discussed in this chapter requires the presence of the Trend data module (TD) and the Dimensioning and Statistics module (DI) in your BioNumerics configuration.

7.3.1.2 The Trend analysis window

The *Trend analysis* window (see Figure 7.3.1) is called from the *Comparison* window with **TrendData > Perform trend analysis** (🔗).

The window consists of three dockable panels:

- The *Data tree* panel contains a hierarchical tree representation of all trend data experiments in the comparison and all analyses and functions calculated by the user with their respective parameters.

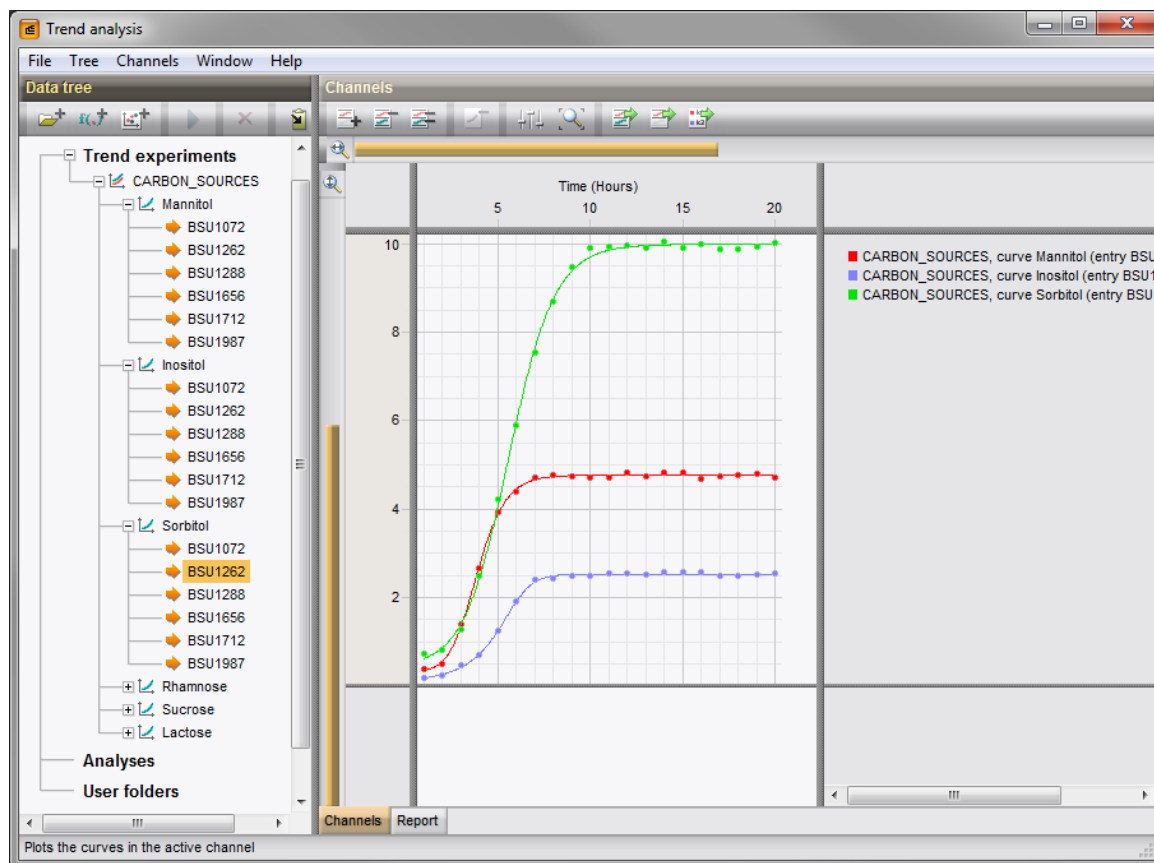


Figure 7.3.1: The *Trend analysis* window.

- In the *Channels* panel, trend data curves and analyses are plotted in one or more channels.
- The *Report* panel displays a formatted text report for each item in the data tree.

7.3.2 Analysis of trend data

7.3.2.1 Using the data tree

The *Data tree* panel displays the trend data experiments, available for the entries in the comparison, in a hierarchical tree-like representation. Branches of the data tree can be collapsed by clicking on the small "minus" sign that precedes the branch name. Clicking on the "plus" sign will expand the branch again.

The selection state of the corresponding entries in the database is indicated with colored arrows: ➤ indicates a selected entry, ➤ indicates an unselected entry. The selection state can be changed by clicking on the arrow icon.

A number of commands (see below) work on highlighted items in the data tree. Individual items can be highlighted, just by clicking on the name of the item. To highlight more than one item, hold the Ctrl-key and click on the items with the left mouse button (**Ctrl+click**). A range of items can be highlighted at once with **Shift+click**.



Only ranges of items within the same node can be selected with **Shift+click**. When attempting to select a range that is spread out over more than one node, the error message "Inconsistent items for selection" appears.

The data tree contains a node "User folders", in which custom folders can be created by selecting **Tree >**

New folder... (📁). The *Add new user folder* dialog box appears.

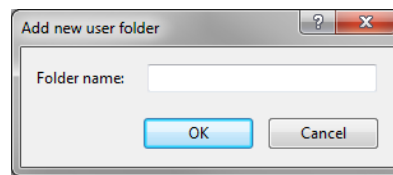


Figure 7.3.2: The *Add new user folder* dialog box to add new folder.

The dialog box prompts for the name of the folder to be created.

Highlighted items in the data tree can be copied with **Tree > Copy items** (📋, **Ctrl+C**) and pasted at a different location in the tree with **Tree > Paste items** (📋, **Ctrl+V**). This functionality can be used to copy items to custom folders, but also to provide arguments for functions (see 7.3.2.4) and variables for analyses (see 7.3.2.5), that are calculated on trend curves.

Highlighted items can be duplicated with **Tree > Duplicate** or removed from the data tree with **Tree > Remove items** (🗑️, **Del**).

Some items with non-default names (e.g. user-created analyses) can be renamed: Highlight the item and select **Tree > Rename** or just click the item twice to make the item name editable.

7.3.2.2 Reports

Each item in the data tree has some information associated with it. When an item is highlighted in the data tree, this information is shown in the *Report* panel of the *Trend analysis* window (see Figure 7.3.3). The amount of available information ranges from just the name of the item (e.g. for a user folder; see 7.3.2.1) to a detailed statistical report in the case of a trend analysis (see 7.3.2.5).

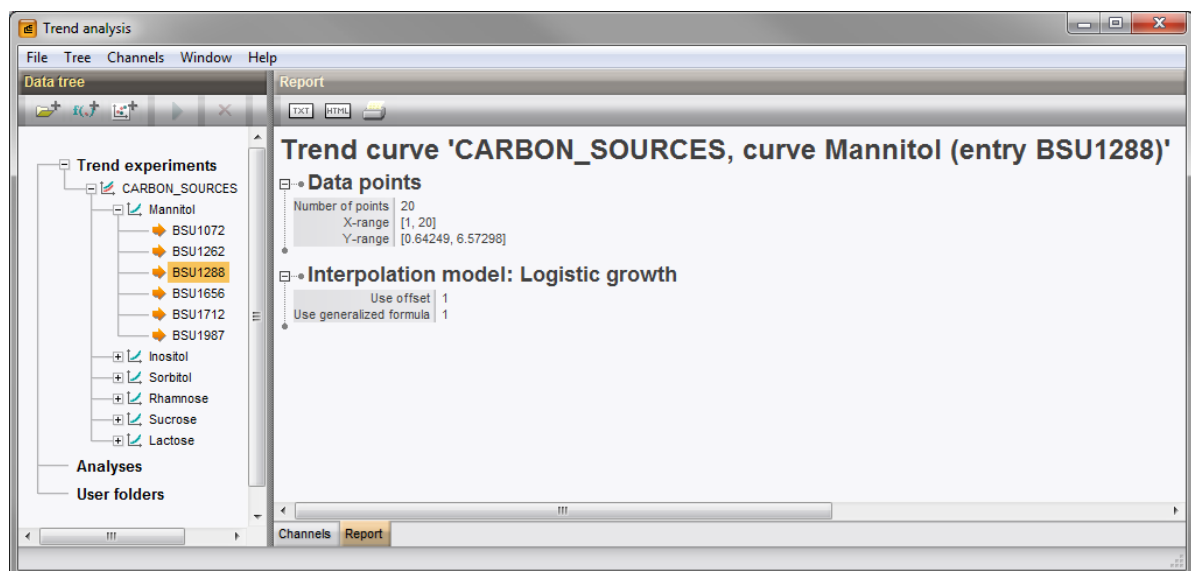


Figure 7.3.3: The *Trend analysis* window, with the *Report* panel displayed.



When multiple items are highlighted in the data tree, no information is displayed in the *Report* panel.

The information in a report can be exported as plain text with **File > Export report as text** (📄). The resulting `export.txt` file is saved in the BioNumerics home directory (see 3.1.2) and opened in the default

text editor (e.g. Notepad). Alternatively, a report can be exported as HTML with **File > Export report as html** (HTML). In this case, an export.htm file is generated in the home directory and opened in the default browser.

A report can be printed with **File > Print report...** (Printer).

In statistical reports of trend analyses, histograms are displayed as thumbnail images. When you click on such a thumbnail image, the histogram appears in its own *Histogram* window (see Figure 7.3.4).

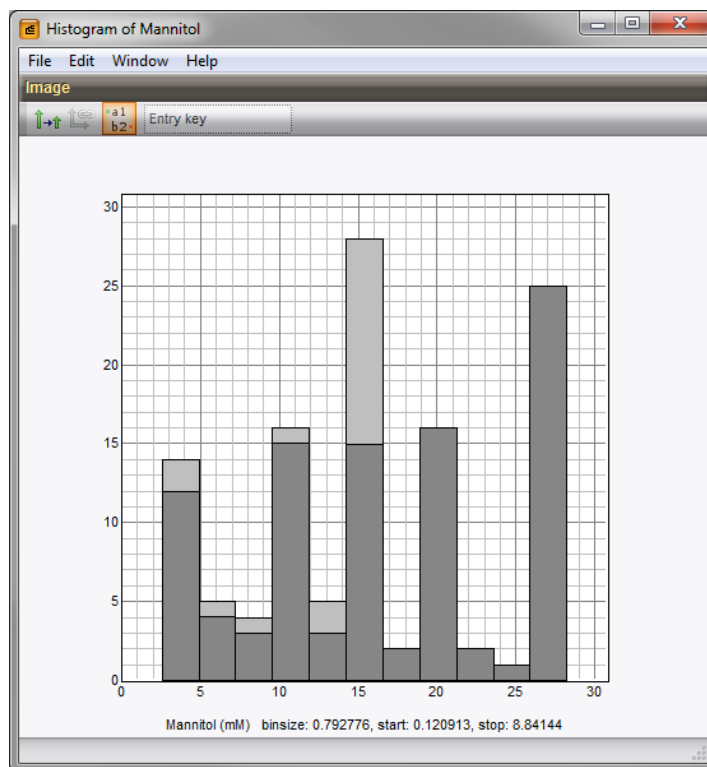


Figure 7.3.4: The *Histogram* window.

This window displays a histogram of a binned quantitative variables. The bin size is automatically selected to create an optimal result.

With **Edit > Use global/local scale**, it is possible to toggle between a scale that would allow to fit any data set from the underlying comparison (global scale) and a scale that is optimized for the currently displayed data set (local scale).



Commands that are not applicable will be inactive (grayed out).

A bar in the histogram can be selected by holding the **Shift**-key and drawing a rectangle with the mouse. Entries that fall in the bin represented by that bar are selected in the database. Conversely, a selection of entries in the database is reflected in the histogram; selected entries are displayed in a darker shade. Pressing **F4** will unselect all entries.

The histogram can be printed with **File > Print image**.

Selecting **Edit > Copy image** calls the *Trend copy image* dialog box.

For the **Image format**, the choice is offered between *metafile* (for import in Windows applications) and *bitmap* (for other applications). In case *bitmap* is checked, the **Maximum size** and the **Resolution** can be set.

Likewise, scatter plots may be displayed in the *Report* panel. When the corresponding thumbnail is clicked, the scatter plot appears in its own *Scatterplot* window (see Figure 7.3.6).

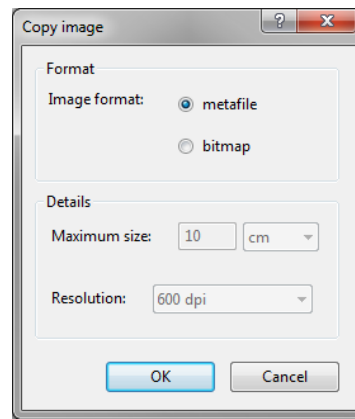


Figure 7.3.5: The *Trend* copy image dialog box, to select the format of the exported image.

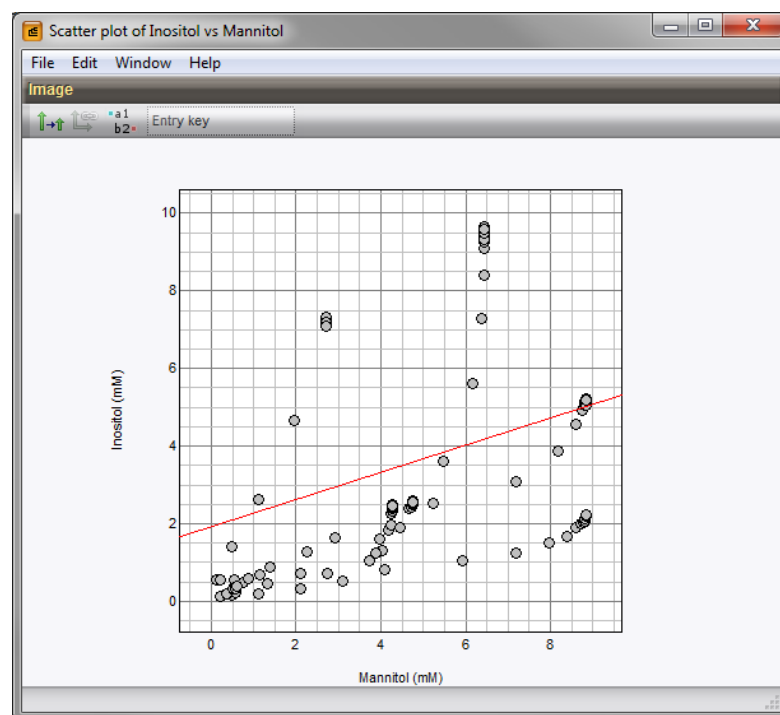


Figure 7.3.6: The *Scatterplot* window.

This window displays a scatter plot of two quantitative variables.

With **Edit > Use global/local scale**, it is possible to toggle between a scale that would allow to fit any data set from the underlying comparison (global scale) and a scale that is optimized for the currently displayed data set (local scale).

Select **Edit > Maintain aspect ratio** to switch between a coordinate system where a unit on the X-axis corresponds to a unit on the Y-axis (aspect ratio maintained) or a coordinate system in which X- and Y-axis units were calculated independently for an optimal spread (aspect ratio not maintained).

Labels can be displayed or hidden with **Edit > Show/hide labels**. If labels are shown, the label that will be used can be selected from the drop-down list in the toolbar or via the menu with **Edit > Choose labels**. When labels are hidden from display, these options will be unavailable.



Commands that are not applicable will be inactive (grayed out).

Dots in the scatter plot can be selected by holding the **Shift**-key and drawing a rectangle around them with the mouse. The corresponding entries are selected in the database. Conversely, a selection of entries in the database is indicated in the scatter plot with an orange circle around the dot. Pressing **F4** will unselect all entries.

The histogram can be printed with **File > Print image**.

Selecting **Edit > Copy image** calls the *Trend copy image* dialog box.

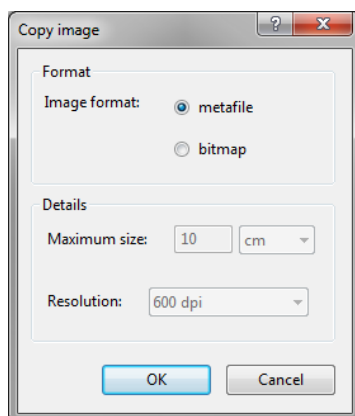


Figure 7.3.7: The *Trend copy image* dialog box, to select the format of the exported image.

For the **Image format**, the choice is offered between *metafile* (for import in Windows applications) and *bitmap* (for other applications). In case *bitmap* is checked, the **Maximum size** and the **Resolution** can be set.

7.3.2.3 Displaying trend data in channels

The trend curves of one or more highlighted items in the data tree (see 7.3.2.1) are plotted in a newly created channel with **Tree > Plot in new channel** (🖨️). Selecting **Tree > Entry selection > Plot in new channel** will plot only the trend curves for which the entries are selected in the database. The trend curve(s) will appear in a plot on the left-hand side of the *Channels panel* and a legend will be displayed on the right-hand side.

Using the 📏 and 📏 zoom sliders, it is possible to zoom in and out on the plots horizontally and vertically, respectively. Additionally, the plots can be auto-fitted in the available width of the left part of the *Channels panel* with **Channels > Zoom to fit** (📏).

If **Tree > Plot in channel set** (🖨️) is selected (not available when a single trend curve is highlighted), each trend curve from the highlighted item(s) in the data tree will be plotted in its own newly created channel. Similar to the above, selecting **Tree > Entry selection > Plot in channel set** will plot only the trend curves for the selected entries in their own channel. All channels share the same X-axis to make the curves comparable with each other.

A new, empty channel is displayed with **Channels > Add channel** (📄+).

When several channels are displayed simultaneously in the *Channels panel*, one of these channels can be highlighted, i.e. is the *active* channel. A channel can be made active by clicking it with the mouse.

A selection of trend curves can be added to the active channel with **Tree > Entry selection > Plot in active channel**. Again, selecting **Tree > Entry selection > Plot in active channel** will plot only the trend curves for the selected entries in the active channel.

With **Channels > Channel settings...** (⚙️), the *Channel settings* dialog box is called (see Figure 7.3.8).

The **Curve window**, i.e. the X-axis on which the trend data will be plotted in the channel can be either

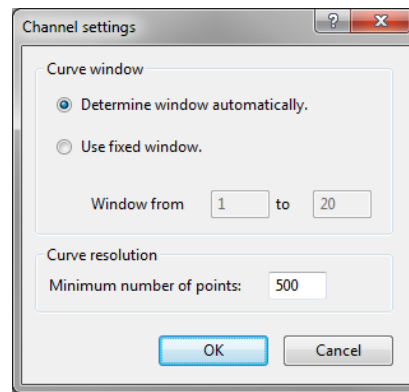


Figure 7.3.8: The *Channel settings* dialog box.

determined automatically from the data (***Determine window automatically***) or the start and stop values can be entered if ***Use fixed window*** is checked. By default, the lowest and the highest X-values found are displayed in the corresponding text boxes.

The ***Curve resolution*** determines the visual smoothness of the interpolated curves and is entered as the ***Minimum number of points*** used in the interpolation model. This parameter affects the visual representation both on screen and when the curves are exported.

Clicking on a curve in a channel activates the curve: it will be displayed in a darker and thicker line than original and the corresponding item in the legend will be highlighted. This can make it easier to identify a curve. Trend curves can also be activated by clicking on the legend item. A number of commands work on the active curve:

With ***Channels > Select tree item***, the item in the data tree that corresponds to the active trend curve will be highlighted.

Using ***Channels > Open entry window***, the *Entry* window (see 3.3.4) of the corresponding database entry appears.

The active trend curve is removed from the channel with ***Channels > Remove selected curve*** (🗑️).

A number of options are available to export images from the *Channels* panel:

All channels can be exported at once by selecting ***File > Export all channels*** (🖨️). The *Trend copy image* dialog box pops up.

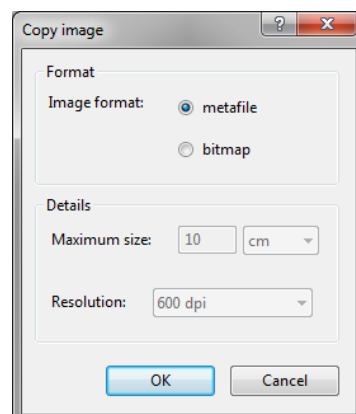


Figure 7.3.9: The *Trend copy image* dialog box, to select the format of the exported image.

For the ***Image format***, the choice is offered between ***metafile*** (for import in Windows applications) and

bitmap (for other applications). In case **bitmap** is checked, the **Maximum size** and the **Resolution** can be set.

The active channel can be exported with **File > Export active channel** (📁) and the legend of the active channel with **File > Export channel legend** (📁). In both cases, the *Trend copy image* dialog box will appear, from which the image format can be chosen.

7.3.2.4 Calculating functions on trend curves

7.3.2.4.1 Adding a function

A function can transform a trend curve (or a set of trend curves) in a specific way. Functions are always calculated on the data points of the trend curve, not on the interpolation model.

Select **Tree > New function...** (🔧) to call the *Add function* dialog box (see Figure 7.3.10).

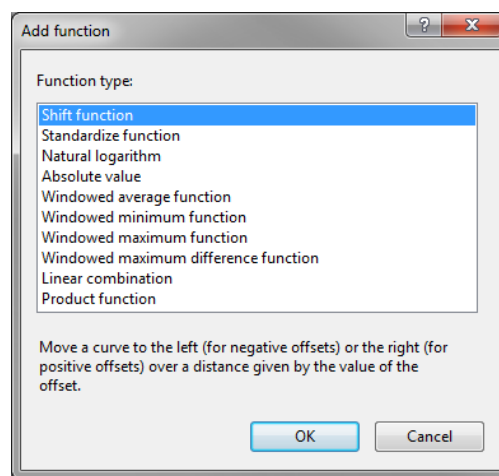




Figure 7.3.10: The *Add function* dialog box.

From this dialog box, a function to apply on a trend curve or a set of trend curves, can be selected:

- **Shift function:** Moves a curve to the left (when a negative offset is entered) or to the right (when a positive offset is entered) over a distance given by the value of the offset.
- **Standardize function:** Standardizes the values in a curve by subtracting the average and dividing by the standard deviation.
- **Natural logarithm:** Takes the natural logarithm of the values in the curve.
- **Absolute value:** Takes the absolute value of the values in the curve.
- **Windowed average function:** Calculates the average of the values in a certain window.
- **Windowed minimum function:** Calculates the minimum of the values in a certain window.
- **Windowed maximum function:** Calculates the maximum of the values in a certain window.
- **Windowed maximum difference function:** Calculates the maximum difference of the values in a certain window. The difference will be positive when the minimum precedes the maximum and negative when the maximum precedes the minimum.
- **Linear combination:** Makes a linear combination of two curves.

- **Product function:** Makes the product of two curves.

Each function takes one or more **Parameters**, one or two **Arguments** and can produce a **Result** when calculated. When a function is added, these items appear in the data tree as sub-nodes from the function node. In case trend curves or items that contain trend curves were highlighted in the data tree when **Tree > New function...**  was selected, the curves will be used as arguments for the function wherever possible. Alternatively, you can copy and paste items as sub nodes of the argument nodes (see 7.3.2.1 for more information about data tree functionality).

When all required arguments are supplied, the function can be calculated. Highlight the function in the data tree and select **Tree > Calculate...** .

7.3.2.4.2 Calculating a Shift function

Will prompt for the **Shift distance**, the distance on the X-axis over which the curve(s) will be shifted left or right.

7.3.2.4.3 Calculating a Standardize function

The trend curve will be standardized according to:

$$y = \frac{x - \langle x \rangle}{S_D}$$

with the average ($\langle x \rangle$)

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^N x_i$$

and the standard deviation (S_D)

$$S_D = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \langle x \rangle)^2}$$

7.3.2.4.4 Calculating a Natural logarithm function

Will take the natural logarithm ($y = \ln(x)$) of the trend curve.

7.3.2.4.5 Calculating an Absolute value function

Will take the absolute value ($y = |x|$) of the trend curve.

7.3.2.4.6 Calculating a Windowed average function

Lets you specify a window by defining a minimum (**from**) and a maximum distance (**to**). For each point in the curve, all points that fall within this window will be averaged.

7.3.2.4.7 Calculating a Windowed minimum function

Lets you specify a window by defining a minimum (*from*) and a maximum distance (*to*). For each point in the curve, the minimum value of all points that fall within this window will be used.

7.3.2.4.8 Calculating a Windowed maximum function

Lets you specify a window by defining a minimum (*from*) and a maximum distance (*to*). For each point in the curve, the maximum difference between any two points that fall within this window will be used.

7.3.2.4.9 Calculating a Windowed maximum difference function

Lets you specify a window by defining a minimum (*from*) and a maximum distance (*to*). For each point in the curve, the minimum and maximum of all points that fall within this window are determined and the difference is calculated.

7.3.2.4.10 Calculating a Linear combination function

Prompts for the *First coefficient* (*a*) and *Second coefficient* (*b*) to use in the formula $y = ax_1 + bx_2$, with x_1 a point in the trend curve specified as Argument 1 and x_2 the corresponding point in the trend curve specified as Argument 2.

7.3.2.4.11 Calculating a Product function

Will calculate the product ($y = x_1 \cdot x_2$) of the two trend curves that were specified as arguments.


7.3.2.4.12 Resetting a function

When a function has been calculated, the Results sub-folder will be populated. The resulting trend curves can be plotted in channels as described in [7.3.2.3](#).

In order to re-calculate a function with other parameters or arguments, it should be reset first. This can be achieved by highlighting the function in the data tree and selecting **Tree > Reset**. Only functions that are not in use, i.e. of which no results are used as arguments of other functions, as variables in analyses (see [7.3.2.5](#)) or plotted in a channel) can be reset.

7.3.2.5 Regression and correlation analysis

7.3.2.5.1 Adding an analysis

A new analysis can be added via **Tree > New analysis...** . This command calls the *Add statistical analysis* dialog box (see Figure [7.3.11](#)).

The dialog box prompts for an *Analysis name*.

One of following *Analysis types* can be chosen:

- **Trend regression:** Is used to model a response curve as a linear combination of a number of explanatory curves.
- **Trend correlation analysis:** Calculates the Pearson and Spearman correlation on trend curves.

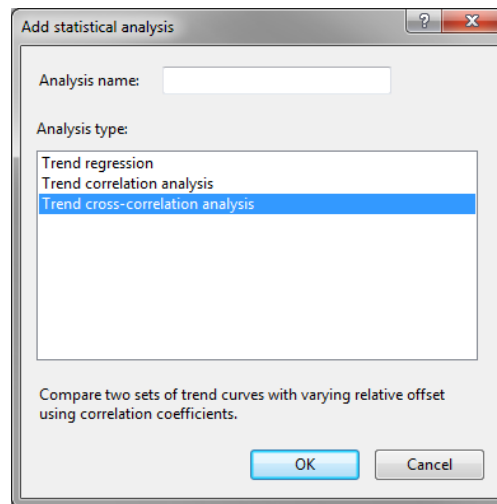


Figure 7.3.11: The *Add statistical analysis* dialog box.

- ***Trend cross-correlation analysis:*** Calculates the Pearson and Spearman correlation on trend curves, within a certain offset window.

After an analysis is created, it will appear in the data tree under Analyses. In case trend curves were highlighted when **Tree > New analysis...** (🖨️) was selected, these trend curves will be used as variables for the analysis. Additionally, required variables can be copied and pasted into the corresponding sub-folders (see 7.3.2.1 for data tree functionality).

7.3.2.5.2 Regression analysis

In a regression analysis, a linear combination of the explanatory variables (= trend curves) is calculated, to model the response variables. Both explanatory variables and response variables are required to calculate a regression analysis.

With a regression analysis highlighted in the data tree, select **Tree > Calculate...** (▶️). The *Trend regression analysis* dialog box pops up for a regression analysis (see Figure 7.3.12).

For the calculation of the regression model, one can specify to **Use all entries** or to only use the current selection of entries in the database (**Use Current selection**). The latter option will be grayed out if no entry selection is present.

In case different groups are present in the trend experiments, one of the available information fields or the comparison groups can be specified, which contains a **Categorical variable** that defines the groups. In case the trend curves should be considered as a single group, "No categorical" should be selected from the drop-down list.

The **Covariance type** can be selected:

- ***Use full covariance matrix*** considers all covariance values; response variables as well as the relations between response variables will be taken into account.
- ***Use only individual variances*** will calculate linear regression models for each response variable individually.
- With ***Use PCA to determine significant variability***, principal component analysis is used to reduce the space of response curves to a subspace with the bulk of variance. When the latter option is checked, the **Significance threshold (relative to total variance)** can be entered as a percentage.

Figure 7.3.12: The *Trend regression analysis* dialog box, prompting for the parameters to calculate a regression analysis.

The ***Significance level for confidence intervals*** can be entered as a percentage, e.g. when "5" is entered, significance intervals will be calculated at a confidence level of 95%.

Under ***Value type***, one can use either the actual measured values (*Use true values*) or use the values as calculated by the trend curve fit model (*Use interpolated values*).

The ***Analysis range*** can be set from the corresponding drop-down list:

- With "Use full range" selected, a regression analysis will be performed on the complete range of the trend curves.
- When "Use specific window" is selected, the window on which to perform the regression analysis can be specified as start and end point of the range (*Perform analysis on points between ... and ...*).
- "Use running window" will calculate a regression analyses for a number of windows in a certain range (specified as *Perform analysis on points between ... and ...*). The size of the running window can be entered, as well as the step size.



The **Analysis range** options "Use full range" and "Use specific window" result in a single regression model to be calculated, i.e. individual *values* will be obtained for intercept and slope. However, with "Use running window" selected, multiple (the exact number is determined by the range and step size) regression models will be calculated, so the intercepts and slopes again form *trend curves*.

When the regression analysis is performed, a detailed statistical report for each specified trend curve window will appear in the *Report* panel (see 7.3.2.2). This report has following sections:

- **Exploratory data analysis:** A number of histograms and scatter plots, calculated on the explanatory and response variables.
- **Regression model:** The actual linear regression model that was calculated, with confidence intervals indicated on slope and intercept.
- **Regression model assessment:** An assessment of the quality of the obtained regression model.
- **Fitted values scatter plots:** Scatter plots calculated on the fitted values, i.e. the variation that is explained by the model.
- **Residual scatter plots:** Scatter plots calculated on the residuals, i.e. the variation which is *not* explained by the model.
- **Explanatory variable importance assessment:** Importance of explanatory variables, assessed by omitting the variable from the model and then comparing this reduced model with the complete model.

Furthermore, a number of items appear under the Fitted values, Residuals, Intercepts and Slopes nodes in the data tree. These items can be plotted in channels, as described in 7.3.2.3.

7.3.2.5.3 Correlation analysis

In a correlation analysis, the degree of relationship between two variables is described. Therefore, trend curves should be provided in the Variables 1 and Variables 2 sub-nodes of the correlation analysis in the data tree.

With a correlation analysis highlighted in the data tree, select **Tree > Calculate...** (▶). The *Trend correlation analysis* dialog box pops up for a correlation analysis (see Figure 7.3.13).

The correlation between all entries will be calculated if **Use all entries** is specified, while only the current selection of entries in the database will be used with **Use Current selection** specified. The latter option will be grayed out if no entry selection is present.

The **Significance level for confidence intervals** can be entered as a percentage, e.g. when "5" is entered, significance intervals will be calculated at a confidence level of 95%.

Under **Value type**, one can use either the actual measured values (**Use true values**) or use the values as calculated by the trend curve fit model (**Use interpolated values**).

The **Analysis range** can be set from the corresponding drop-down list:

- With "Use full range" selected, a correlation analysis will be performed on the complete range of the trend curves.
- When "Use specific window" is selected, the window on which to perform the correlation analysis can be specified as start and end point of the range (**Perform analysis on points between ... and ...**).

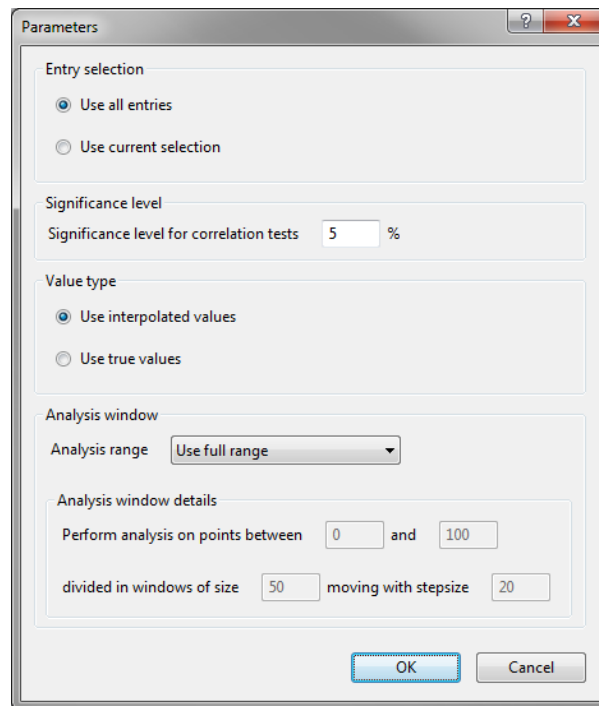


Figure 7.3.13: The *Trend correlation analysis* dialog box, prompting for the parameters to calculate a correlation analysis.

- "Use running window" will calculate a correlation analyses for a number of windows in a certain range (specified as *Perform analysis on points between ... and ...*). The size of the running window can be entered, as well as the step size.



The **Analysis range** options "Use full range" and "Use specific window" result in a single correlation analysis to be performed, i.e. individual *values* will be obtained for correlation coefficients and their corresponding *p*-values. However, with "Use running window" selected, multiple (the exact number is determined by the range and step size) correlation analyses will be calculated, so the coefficients and their *p*-values again form *trend curves*.

When the correlation analysis is performed, a detailed statistical report for each specified trend curve window will appear in the *Report* panel. This report has following sections:

- **Global correlations:** The Pearson and Spearman correlation and their corresponding *p*-values, calculated for all entries in the analysis.
- **Correlations by entry:** The Pearson and Spearman correlation and their corresponding *p*-values, calculated for entries individually.

Furthermore, a number of items appear under the Pearson correlation, Pearson *p*-values, Global Pearson correlation, Global Pearson *p*-values, Spearman correlation, Spearman *p*-values, Global Spearman correlation, and Global Spearman *p*-values nodes in the data tree. These items can be plotted in channels, as described in 7.3.2.3.

7.3.2.5.4 Cross-correlation analysis

In a cross-correlation analysis, a number of correlation analyses are performed, each by applying a certain offset of one trend curve versus the other. As for a normal correlation analysis, trend curves should be provided in its Variables 1 and Variables 2 sub-nodes.

With a cross-correlation analysis highlighted in the data tree, select **Tree > Calculate...** (▶). The *Trend cross-correlation analysis* dialog box pops up for a cross-correlation analysis (see Figure 7.3.14).

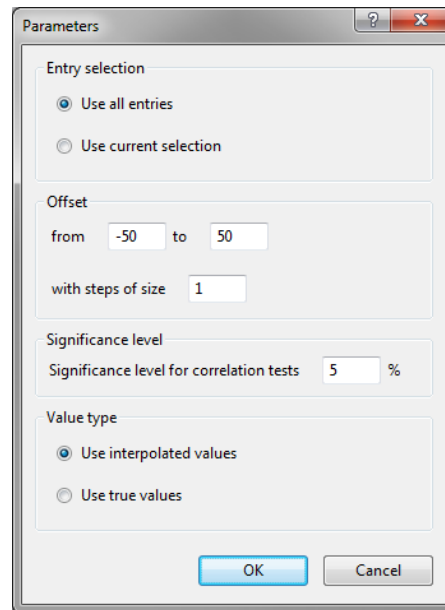


Figure 7.3.14: The *Trend cross-correlation analysis* dialog box, prompting for the parameters to calculate a cross-correlation analysis.

The correlation between all entries will be calculated if *Use all entries* is specified, while only the current selection of entries in the database will be used with *Use Current selection* specified. The latter option will be grayed out if no entry selection is present.

The *Offset*, i.e. the relative shift of trend curves specified in Variables 1 against trend curves specified as Variables 2, that will be applied before performing a correlation analysis can be specified as a window (*from ... to ...*) and a step size. The number of correlation analyses actually performed will therefore be $(WindowSize \times StepSize) + 1$.

The *Significance level for confidence intervals* can be entered as a percentage, e.g. when "5" is entered, significance intervals will be calculated at a confidence level of 95%.

Under *Value type*, one can use either the actual measured values (*Use true values*) or use the values as calculated by the trend curve fit model (*Use interpolated values*).

When the cross-correlation analysis is performed, a detailed statistical report will appear in the *Report* panel. This report has following sections:

- **Global cross-correlation analysis (Pearson):** Scatter plots of the Pearson correlation coefficients and their corresponding *p*-values in function of offset, as calculated for all entries in the analysis.
- **Global cross-correlation analysis (Spearman):** Scatter plots of the Spearman correlation coefficients and their corresponding *p*-values in function of offset, as calculated for all entries in the analysis.
- **Cross-correlation analysis by entry:** Scatter plots of the Pearson and Spearman correlation coefficients and their corresponding *p*-values in function of offset, as calculated for entries individually.

7.3.2.5.5 Resetting an analysis

In order to re-calculate a trend analysis with different parameters, or to calculate the same analysis on a different set of variables (by copying and pasting trend curves), the analysis should be reset first. This can

be achieved by highlighting the analysis in the data tree and selecting **Tree > Reset**. Only analyses that are not in use, i.e. of which no results are used as variables in other analyses, as arguments of functions or plotted in a channel) can be reset.

Part 8

Sequence types

Chapter 8.1

Setting up sequence type experiments

8.1.1 Defining a new sequence type

Please note that, to be able to work with sequences, the Sequence data module (SQ) needs to be present in your BioNumerics configuration.

To create a new sequence type, click on the *Experiment types* panel to activate it and select **Edit > Create new object...** (+). From the *Create a new experiment type* dialog box that pops up, select **Sequence type** to start the *New sequence type* wizard (see Figure 8.1.1).

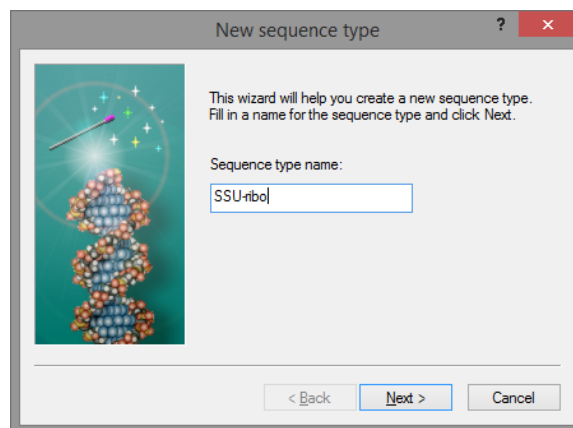


Figure 8.1.1: The *New sequence type* wizard, page 1.

The first page prompts you to enter a name for the new sequence type. Pressing **<Next>** will bring you to the second page of the *New sequence type* wizard (see Figure 8.1.2).

Check the radio button that corresponds to the kind of the sequences: **Nucleic acid sequences** or **Amino acid sequences**.

The option **Use reference sequence as mapping template (required for SNP analysis)** should be checked to enable SNP analyses (see 8.10). Checking this option has a few important consequences:

- The first sequence that is imported into this sequence type will automatically become the reference sequence (see 8.1.2.3 for more information).
- For any future sequence imported, its length will be checked and the sequence will only be accepted if its length equals the length of the reference sequence.

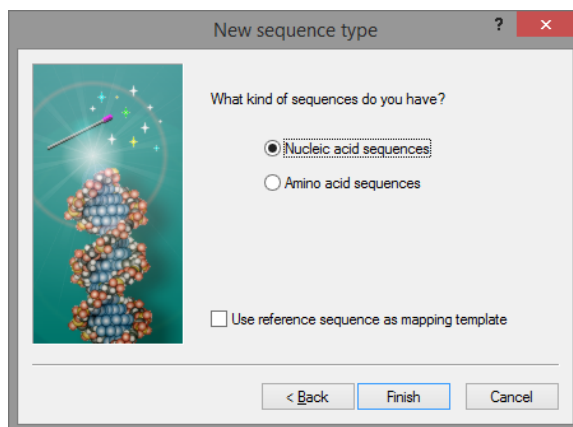


Figure 8.1.2: The *New sequence type* wizard, page 2.

This ensures that all sequences are in the same frame as the reference sequence. Consequentially, a sequence comparison (being either a similarity calculation or a SNP search) from a "reference mapped" sequence type does not require a prior – and computationally intensive – sequence alignment.



A "regular" sequence type can still be converted to a reference mapped sequence type after creation, see [8.1.2.2](#).

Press the **<Finish>** button to complete the setup of the new sequence type. The sequence type is now listed in the *Experiment types* panel of the *Main* window.

8.1.2 Editing a sequence type

8.1.2.1 Sequence type settings

A number of settings related to a sequence type are stored as initial settings. These include display and storage settings as well as alignment, clustering, and conversion settings. These settings can be changed in the *Sequence type* window (see Figure [8.1.3](#)).

To open the *Sequence type* window, highlight the sequence type in the *Experiment types* panel and select **Edit > Open highlighted object...** (🔗, **Enter**) (see Figure [8.1.3](#)). Alternatively, simply double-click on the sequence type.

Via **Settings > General settings...** (⚙️), the *Sequence settings* dialog box is called (see Figure [8.1.4](#)).

The radio button checked in the *Type panel* corresponds to the option that was selected in the second step of the *New sequence type* wizard: **Nucleotide sequence** or **Amino acid sequence**.

When the option **Store in database** is checked (the default setting), the sequence will be stored as a record in the relational database. When the option **Store in database** is unchecked, the sequence will be stored in the *Sourcefiles* subdirectory of the database folder (see [3.7.2](#)). Storing sequences outside the database is to be preferred when working with large sequences.

From the *Sequence type* window, other relevant settings for the sequence type can be accessed:

With **Settings > Comparison settings...** (🔍), you can edit the pairwise comparison settings as explained in [8.3.2](#) and the settings for the calculation of a cluster analysis from a multiple alignment as explained in [8.3.13](#).

Using **Settings > Multiple alignment settings...** (🔍), the settings for calculating a multiple alignment (see [8.3.3](#)) can be edited.

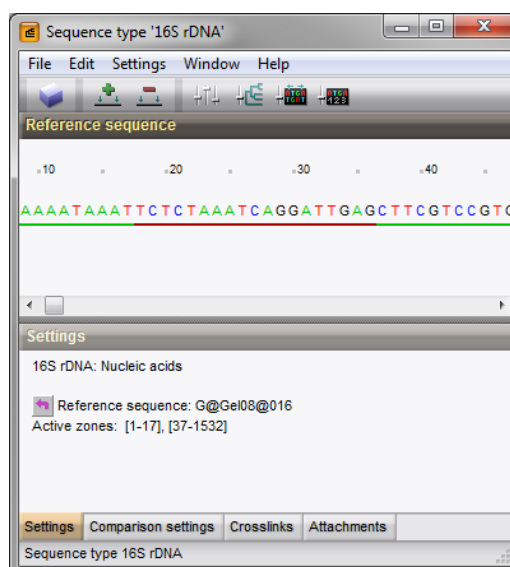


Figure 8.1.3: The *Sequence type* window with a reference sequence defined and region excluded.

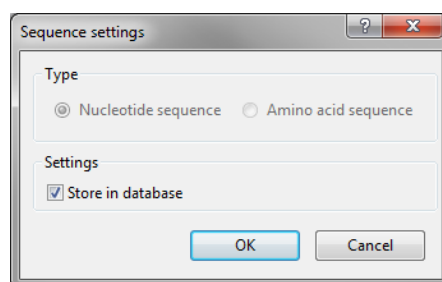


Figure 8.1.4: The *Sequence settings* dialog box.

All comparison settings defined for the sequence type are shown in the *Comparison settings* panel.

The command **Settings > Character conversion settings...** (🔧) allows the parameters to be set for converting bases into categorical characters (see 8.3.15).

Settings > Display settings... (🖨️) allows the color and viewing settings in the multiple alignment editor to be specified in the *Sequence display settings* dialog box (Figure 8.1.5).

The dialog box provides two defaults for color settings: the **White default**, which corresponds to the most widely used colors for the bases on a white background, and the **Black default**, which uses a black background in the multiple alignment editor, using the base color scheme of earlier versions of BioNumerics. Apart from the two defaults, every item can be assigned a specific color using the slider bars for the **Red**, **Green** and **Blue** components. Characters can be chosen to indicate gaps (**Gap indicator**) and consensus positions (**Consensus indicator**).

The assembly settings (accessible via **Settings > Assembly settings...**) and the assembly trimming settings (via **Settings > Assembly trimming settings...**) are used when assembling sequences in batch (see 8.1.3.2).

8.1.2.2 Converting to a reference mapped sequence type

In order to perform a SNP analysis (see 8.10), a *reference mapped* sequence type is required. In case the option **Use reference sequence as mapping template (required for SNP analysis)** was not checked during creation of the sequence type (see 8.1.1), a regular sequence experiment type can still be converted into a

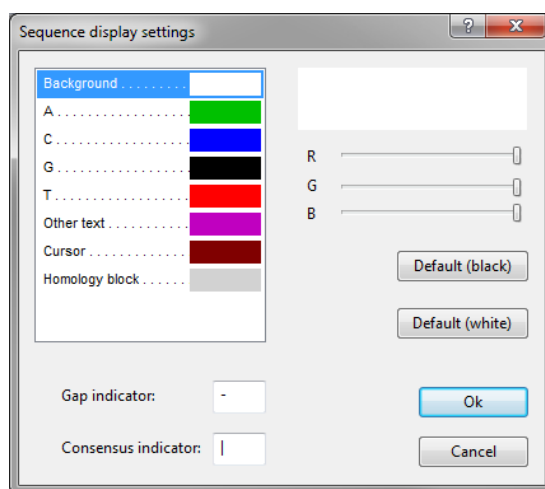


Figure 8.1.5: The *Sequence display settings* dialog box.

reference mapped sequence experiment type.



It can be seen in the *Settings* panel of the *Sequence type* window whether a sequence type is reference mapped or not: the message "This is a reference controlled sequence type with fixed length, typically used for SNP analysis." will be displayed only for a reference mapped sequence type.

To convert a sequence experiment type, open its *Sequence type* window and select **Edit > Convert to reference mapped sequence type**. Note that this command is only active if actual sequence data is present (see 8.1.3 for import options) and a reference sequence is defined (see 8.1.2.3).

After confirmation, the software will first check if all sequences are of equal length. If this is indeed the case, the sequence experiment type will be converted. An error message will be generated if not all sequences have the same length, in which case the sequence type cannot be converted.


8.1.2.3 The reference sequence and excluding regions

Similar as for the comparison of fingerprints, it is possible to exclude regions from the sequences to be clustered. First, one needs to define a reference sequence, and next, one can indicate the zones to be excluded and included on the reference sequence. The exclusion of regions is only possible when calculating a cluster analysis based upon globally aligned sequences (multiple alignment) and when the reference sequence is included in the multiple alignment. Only then, the program can introduce a consistent base numbering based on the reference sequence, which makes it possible to specify the same exclude/include settings for different multiple alignments within the sequence type.

In the *Main* window, open the *Sequence type* window by double-clicking on the sequence type in the *Experiment types* panel.

Initially, there is no reference sequence present. A *link arrow* (↗) allows you to link a reference sequence to a database entry, by clicking on the arrow and dragging it onto a database entry, and then releasing the mouse button. When the experiment is linked, its link arrow is purple (↗) instead of gray and the reference sequence is shown in the *Sequence type* window (Figure 8.1.3). If the original reference sequence experiment needs to be accessed later on, use **Edit > Open reference sequence** to open it in the *Sequence editor* window.

To exclude a region for comparison, select **Edit > Exclude active region** (🔍) in the *Sequence type* window. Enter start and end base number of the region to be excluded, and press the <OK> button. The included regions are marked with a green line whereas the excluded regions are marked with a red line (see Figure 8.1.3).

In order to remove all excluded regions at a time, select **Edit > Include active region** (). Enter “1” as **From** number, and enter the length of the sequence, or a number which certainly exceeds the sequence’s length, as **To** number.

If now a *Comparison* window is opened with aligned sequences, the reference sequence and an indication of the active zones can be displayed on top of the alignment as follows: show the image of the aligned sequences, select the branch tip of the reference sequence in the *Dendrogram* panel and use **Sequence > Create consensus of branch**.

By creating a consensus of a single sequence, you can display the reference sequence in the consensus sequence line. At the same time, the excluded and included regions are indicated, and the base numbering (according to the reference sequence), appears. In order to see the base numbering it may be necessary to drag the horizontal line that separates the header from the *Experiment data* panel downwards.

8.1.3 Importing sequence data

8.1.3.1 Import options for sequence data

Sequence data can be imported in BioNumerics in several ways:

1. Importing sequences in GenBank, EMBL (see [8.1.3.5](#)) and FASTA formats (see [8.1.3.4](#)) from a text file.
2. Importing sequences from online repositories (see [8.1.3.6](#)).
3. Importing sequence assemblies from BAM or SAM files (see [8.1.3.3](#)).
4. Entering or pasting the sequences via the *Experiment card* window (see [8.1.5](#)) or *Sequence editor* window (see [8.1.6](#)) of the database entry.
5. Importing and assembling sequences using BioNumerics’ own Assembler program, either contig by contig (see [8.1.3.7](#)) or in batch (see [8.1.3.2](#)).
6. Assembling reads from next-generation sequencers into consensus sequences using BioNumerics’ own Power Assembler program (see [18](#)).

8.1.3.2 Importing and assembling sequences in batch

8.1.3.2.1 Introduction

With the **Import and assemble trace files** option, listed under the topic **Sequence type data** in the *Import* dialog box (see Figure [8.1.6](#)) the Assembler program can run within BioNumerics in batch mode. The batch import routine accepts binary chromatogram files from Applied Biosystems (**ABI**), as well as Beckman (**SCF**), and Amersham (**MGB**) automated sequencers.

With the **Import and assemble trace files from FASTA text files** option, listed under the topic **Sequence type data** in the *Import* dialog box (see Figure [8.1.7](#)) the Assembler program can run within BioNumerics in batch mode. The batch import routine accepts text files containing FASTA formatted sequences.

8.1.3.2.2 Name parsing

Name parsing for sequence trace files:

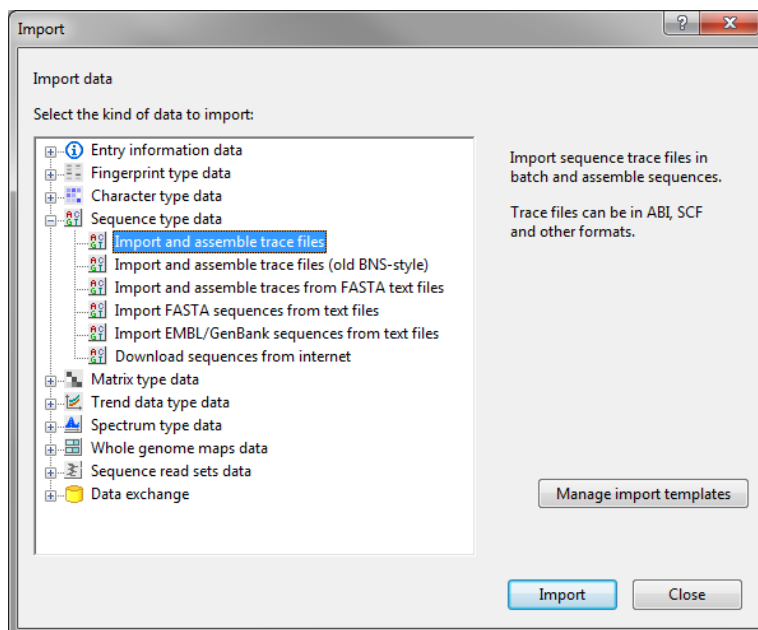


Figure 8.1.6: Import and assemble trace files in batch.

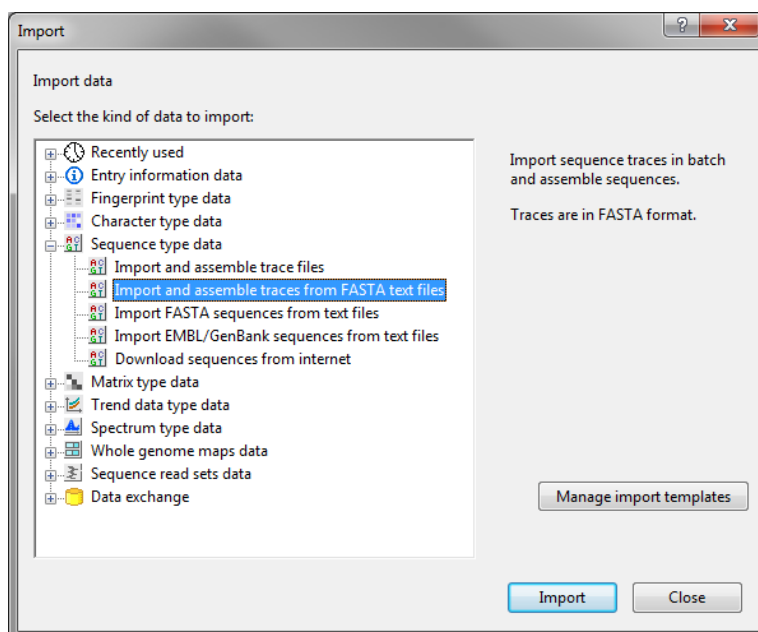


Figure 8.1.7: Import and assemble traces in FASTA format.

Since the *Import and assemble trace files* import routine allows one to assemble a number of trace files into *multiple* contig projects, the program needs to know to which contig project a given trace file belongs. A contig project is unambiguously defined only if both the unique sample information and the experiment (a sequence type) is known for each individual trace file.

The name parsing features of the import routine are very flexible and accommodate for a variety of different situations. These situations can be divided in four different cases:

1. Only the sample identifiers can be parsed from the trace file names.
2. The sample identifiers AND the experiments can be parsed from the trace file names.

3. Only the sample identifiers are contained in a template file.
4. The sample identifiers AND the experiments are contained in a template file.

In case 1 and 2, parts of the trace file name will be used as a unique sample information for the corresponding database entry.

Case 3 and 4 occur when the trace file name is a unique identifier that does not contain a string that you wish to import in the database. In those cases, an external tab-delimited text file (= *Template file*) provides the "translation" between the trace file names and the sample identifiers they belong to.

In case 1 and 3, the experiment is selected by the user. Therefore, all trace files in a batch should belong to the same experiment.

In case 2 and 4, the experiment name is contained in the trace file name or in the template file, respectively, and this experiment can be automatically created by the software. The trace files in a single batch can therefore correspond to multiple experiments.

For all four cases, step-by-step instructions are given in 8.1.3.2.7 to assemble the trace files of an example dataset in batch.

Name parsing for FASTA files:

When sequences are stored in FASTA format, each sequence begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than (">") symbol. The description line contains the *FASTA tags*, separated by a pipe ("|") symbol (see Figure 8.1.8 for an example).

Since the *Import and assemble traces from FASTA text file* import routine allows one to assemble a number of FASTA formatted sequences into *multiple* contig projects, the program needs to know to which contig project a given FASTA sequence belongs. A contig project is unambiguously defined only if both the unique sample information and the experiment (a sequence type) is known for each individual FASTA sequence.

The name parsing features of the import routine are very flexible and accommodate for a variety of different situations. These situations can be divided in following cases:

1. Only the sample identifiers can be parsed from the FASTA tags or file names (see Figure 8.1.8 for an example). As a consequence all trace files in the batch should belong to the same experiment.
2. The sample identifiers AND the experiments can be parsed from the FASTA tags and/or the file names (see Figure 8.1.9 and Figure 8.1.10 for an example).

8.1.3.2.3 Import wizard: assembling sequence trace files

Selecting *Import and assemble trace files* under *Sequence type data* in the *Import* dialog box and pressing <Import> opens the *Import sequence traces* wizard page (see Figure 8.1.11).

Pressing the <Browse> button allows you to select the file(s) that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. The number of files and total size is displayed below the list.

With the <Delete> button all selected files are removed from the import list. All files are deleted at once from the import list when pressing <Delete All>.

When the trace file names do not contain a string that you wish to use as the entry key, an external tab-delimited text file (= *Template file*) can be used, holding the "translation" between the trace file names and the entry keys they belong to. Click on the <Browse> button and browse for the template file.

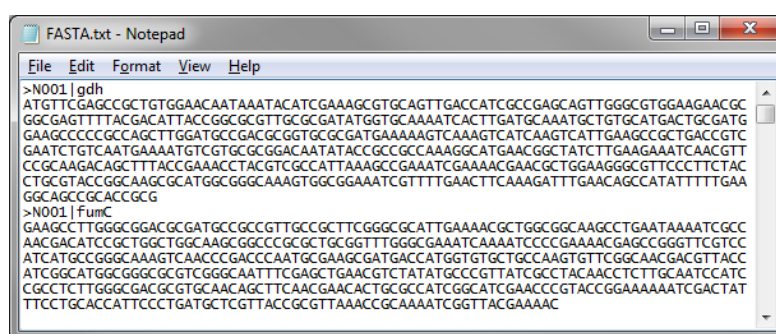


```

>N001 | 2013-04-21
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCGGGCAGTT
TTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGATGGAAGCCCCGCCAGCT
TGGATGCCGACGGCGGTGCGGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCGTCAATCTGTCAATGAAAATGTCGTG
CGCGGACAAATACCGCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAACGTTTCCGCAAGACAGCTTTACCGAAACCTACGTCGC
CATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCTTCTACCTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGGAAA
TCGTTTGAACCTCAAGATTGTAACAGCCATATTTTGAAGGACGCCGACCCGCG
>N002 | 2013-04-21
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCGGGCAGTT
TTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGATGGAAGCCCCGCCAGCT
TGGATGCCGACGGCGGTGCGGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCGTCAATCTGTCAATGAAAATGTCGTG
CGCGGACAAATACCGCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAACGTTTCCGCAAGACAGCTTTACCGAAACCTACGTCGC
CATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCTTCTACCTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGGAAA
TCGTTTGAACCTCAAGATTGTAACAGCCATATTTTGAAGGACGCCGACCCGCG
>N003 | 2013-04-25
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCGGGCAGTT
TTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGATGGAAGCCCCGCCAGCT
TGGATGCCGACGGCGGTGCGGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCGTCAATCTGTCAATGAAAATGTCGTG
CGCGGACAAATACCGCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAACGTTTCCGCAAGACAGCTTTACCGAAACCTACGTCGC
CATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCTTCTACCTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGGAAA
TCGTTTGAACCTCAAGATTGTAACAGCCATATTTTGAAGGACGCCGACCCGCG

```

Figure 8.1.8: FASTA formatted text file: the first FASTA tag corresponds to the Sample information.

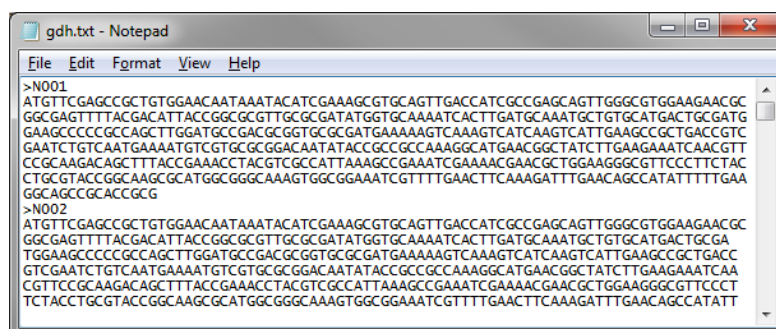


```

>N001 | gdh
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCG
GGCGAGTTTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGATGCGGATG
GAAGCCCCGCCAGCTTGGATGCCGACGGTGCAGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCGTCA
GAATCTGTCAATGAAAATGTCGTGCGGGACAATATACCGCCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAACGTT
CCGCAAGACAGCTTTACCGAAACCTACGTCGCCATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCTTCTAC
CTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGAAATCGTTTGAACCTCAAGATTGTAACAGCCATATTTTGAAG
GGCAGCCGCCCGCG
>N001 | fumc
GAAGCCCTTGGGCGGACGCGATGCCGCCGTTGCCGCTTGGGCGCATTTGAAAACGCTGGCGGCAAGCGCTGAATAAAATGCC
AAGGACATCCGCTGGCTGGCAAGCGGCCGCGCTGCGGTTTGGGCGAAATCAAAATCCCCGAAACGAGCCGGGTCTGTC
ATCATGCCGGCAAGGTCAAACCGAACCAATGCGGAAGCGATGACCATGGTGTGCTGCAAGTGTGTCGCAACGAGCTTACC
ATCGGATGGCGGGCGCTCGGGCAATTTGAGCGTAACGTCATATAGCCGCTTATCGCTCAACCTCTTGAATCCATC
GCGCTCTTGGGCGACGCGTGCAACAGCTTCAACGAACACTGCGCATCGGCAATCGGCAATCGGCAATCGGCAATCGGCAAT
TTCTGCAACCATTCCTGATGCTGTTACCGCGTTAAACCGCAAAATCGGTTACGAAAC

```

Figure 8.1.9: FASTA formatted text file: the first FASTA tag corresponds to the Sample information, the second tag holds the name of the sequence type experiment.



```

>N001
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCG
GGCGAGTTTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGGATG
GAAGCCCCGCCAGCTTGGATGCCGACGGTGCAGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCGTCA
GAATCTGTCAATGAAAATGTCGTGCGGGACAATATACCGCCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAACGTT
CCGCAAGACAGCTTTACCGAAACCTACGTCGCCATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCTTCTAC
CTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGAAATCGTTTGAACCTCAAGATTGTAACAGCCATATTTTGAAG
GGCAGCCGCCCGCG
>N002
ATGTTTCGAGCCGCTGTGGAACAATAAATACATCGAAAGCGTGCAAGTTGACCATCGCCGAGCAGTTGGGCGTGGAAGAAGCG
GGCGAGTTTACGACATTACCGGCGCTTGCAGCATATGGTGCAAAATCACTTGATGCAAAATGCTGTGATGACTGCGGATG
TGAAGCCCCGCCAGCTTGGATGCCGACGGTGCAGATGAAAAAGTCAAAAGTCATCAAGTCATTGAAGCCGCTGACCG
GTCGAATCTGTCAATGAAAATGTCGTGCGGGACAATATACCGCCGCCAAAGGCGATGAACGGCTATCTTGAAGAAATCAA
GTTTCCGCAAGACAGCTTTACCGAAACCTACGTCGCCATTAAAGCCGAAATCGAAACGAACGCTGGAAGGGCGTTCCCT
TCTACCTGCGTACCGGCAAGCGCATGGCGGGCAAGTGCGGAAATCGTTTGAACCTCAAGATTGTAACAGCCATATTT

```

Figure 8.1.10: FASTA formatted text file: the file name holds the name of the sequence type experiment, the FASTA tag corresponds to the Sample information.

A *Template file* needs to have a specific format (see Figure 8.1.12 for an example). This tab-delimited text file must contain exactly one header row containing the columns names, and at least two columns: one column with the trace file names without the file extension and one column with the strain information that will be imported and stored in the database. When the trace files of a single batch correspond to multiple experiments a third column should be present containing the sequence type names.



The first column in the template file should ALWAYS contain the link to the trace files.

Only when all settings have correctly been specified, pressing **<Next>** will display the next step.

The way the sequence information should be imported in the database can be specified with an import

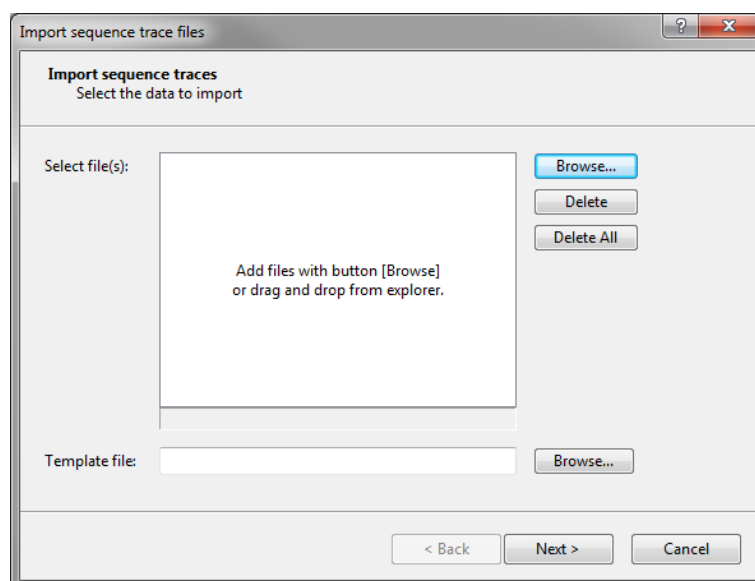
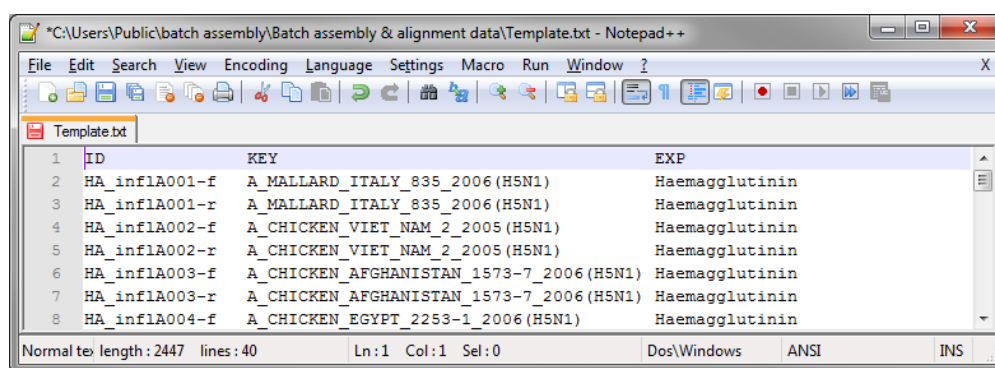
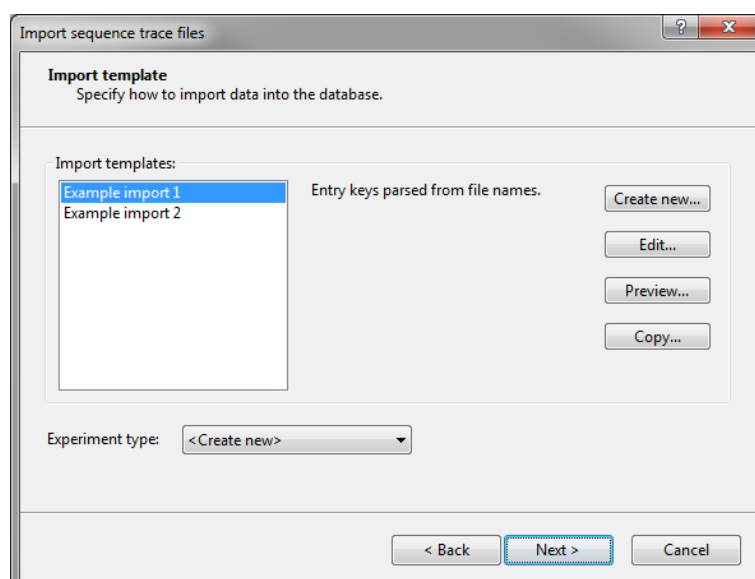
Figure 8.1.11: The *Import sequence traces* wizard page.

Figure 8.1.12: An example template file.

Figure 8.1.13: The *Import template* wizard page.

template. The *Import templates panel* lists all templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up the *Import rules* dialog box allowing you to define a new import template.

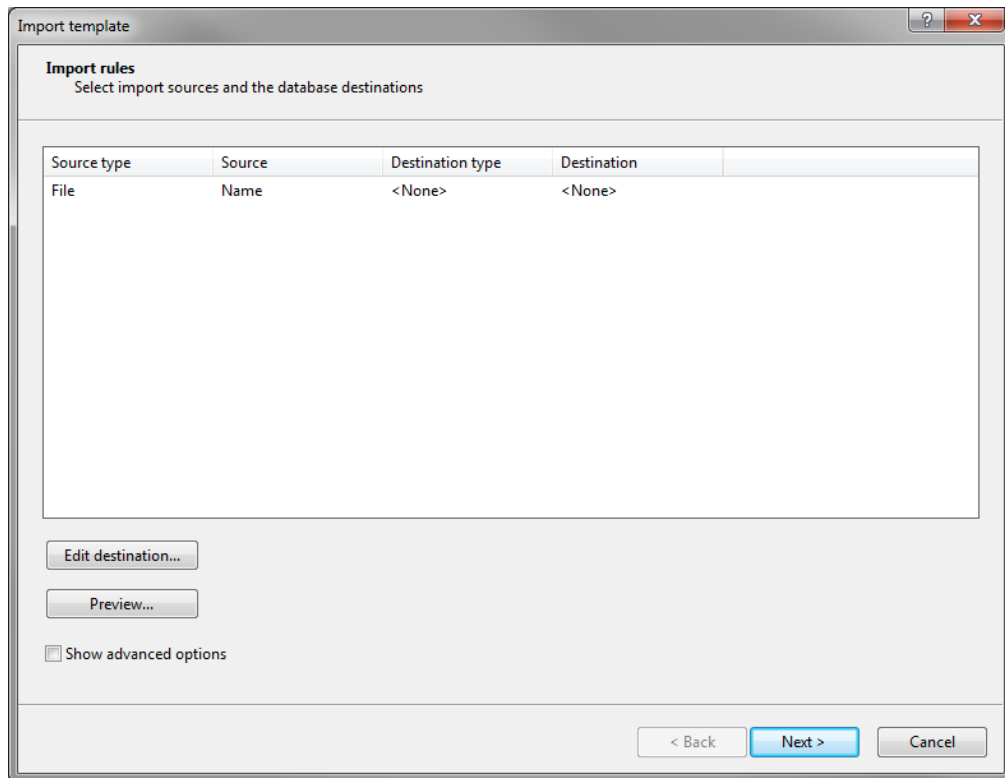


Figure 8.1.14: The *Import rules* dialog box.

When no **template file** is used, the only source of information available in the newly created import template is the file name. The text **File** is specified in the **Source type** column and the text **Name** is displayed in the **Source** column.

When a template file was selected in the first step of the import routine, the file name information in the first column of the template file will be used as link field. The other columns in the template file are displayed in the grid. The text **File field** is specified in the **Source type** column and the column names are displayed in the **Source** column.

The row(s) in the grid can be associated with new or existing entry information fields, sequence information fields and sequence types. Initially the row is not linked to any information (the **Destination type** and **Destination** for the is set to **<None>**). Specifying a *destination* can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row.

The information of the selected row can be linked to:

- The default information field **Key**.
- A **Sequence type** name. The (parsed) information will hold the sequence type name.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select **<Create new>** or select an existing field under the topic **Sequence info field**, respectively).

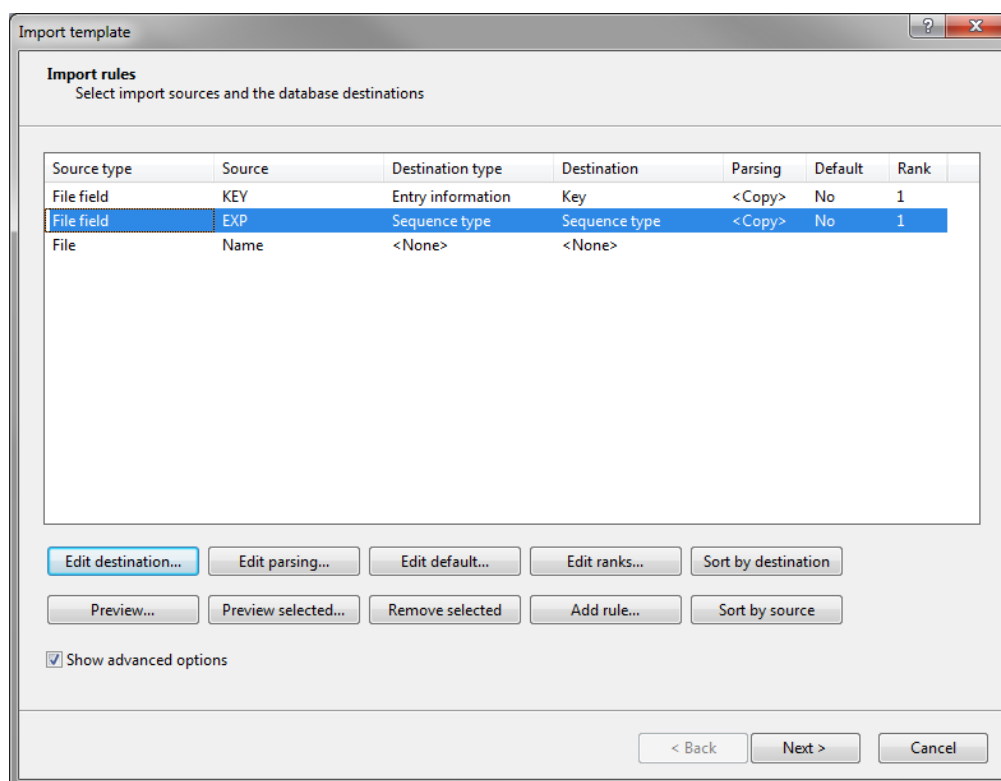


Figure 8.1.15: The *Import rules* dialog box with a template file.

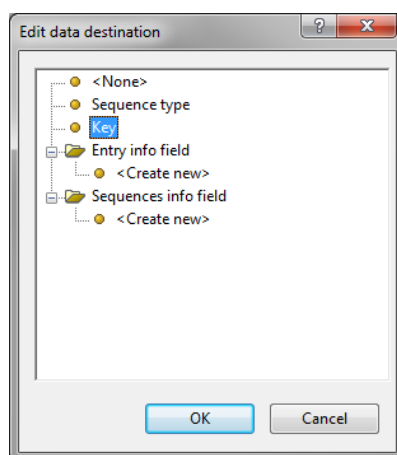


Figure 8.1.16: Edit data destination for a single selected row entry.

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. When a row is linked to a sequence type name, the **Destination type** and **Destination** columns display the text **Sequence type**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to sequence information fields, the text **Sequence info field** is displayed in the **Destination type** column; the name of the field is listed in the **Destination** column.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5, they will be needed in most cases when importing and assembling trace files, as the file name will contain several pieces of useful information.

When the entry keys can be parsed from the trace file names, link the **File name** row in the grid to the **Key** field, and specify the parsing string (**<Edit parsing>**) to get the correct part of information out of the filename (see Figure 8.1.17 for an example). This string should at least contain the [DATA] component, in which case the complete information is retained. The asterisk (*) can be used as a wildcard to omit characters from the information. In the parsing string ***_[DATA]-*** for example, all characters before the underscore and after the hyphen will be ignored.

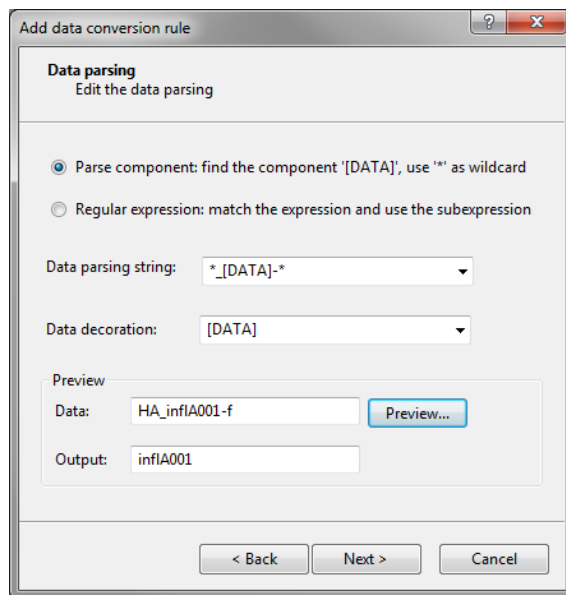


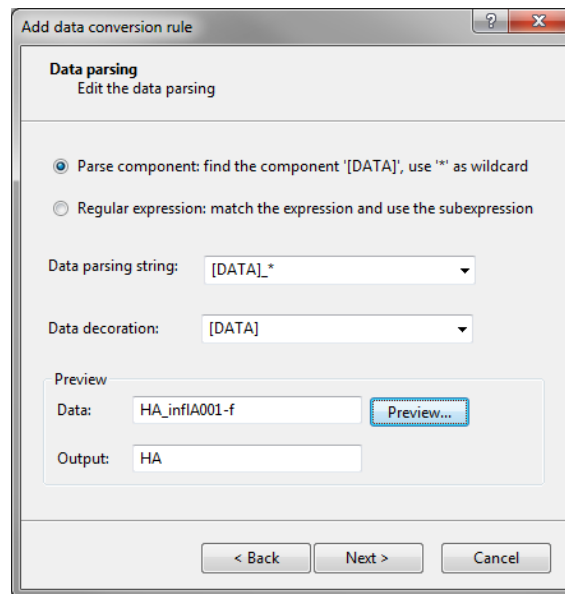
Figure 8.1.17: Parsing the key information from the file name.

When the experiment name can be parsed from the trace file names, add a new rule (**<Add rule>**), select the file name (**Name**) as data source and the **Sequence type** option as as data destination. Specify the parsing string (**<Edit parsing>**) to get the correct part of information out of the filename (see Figure 8.1.18 for an example).

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

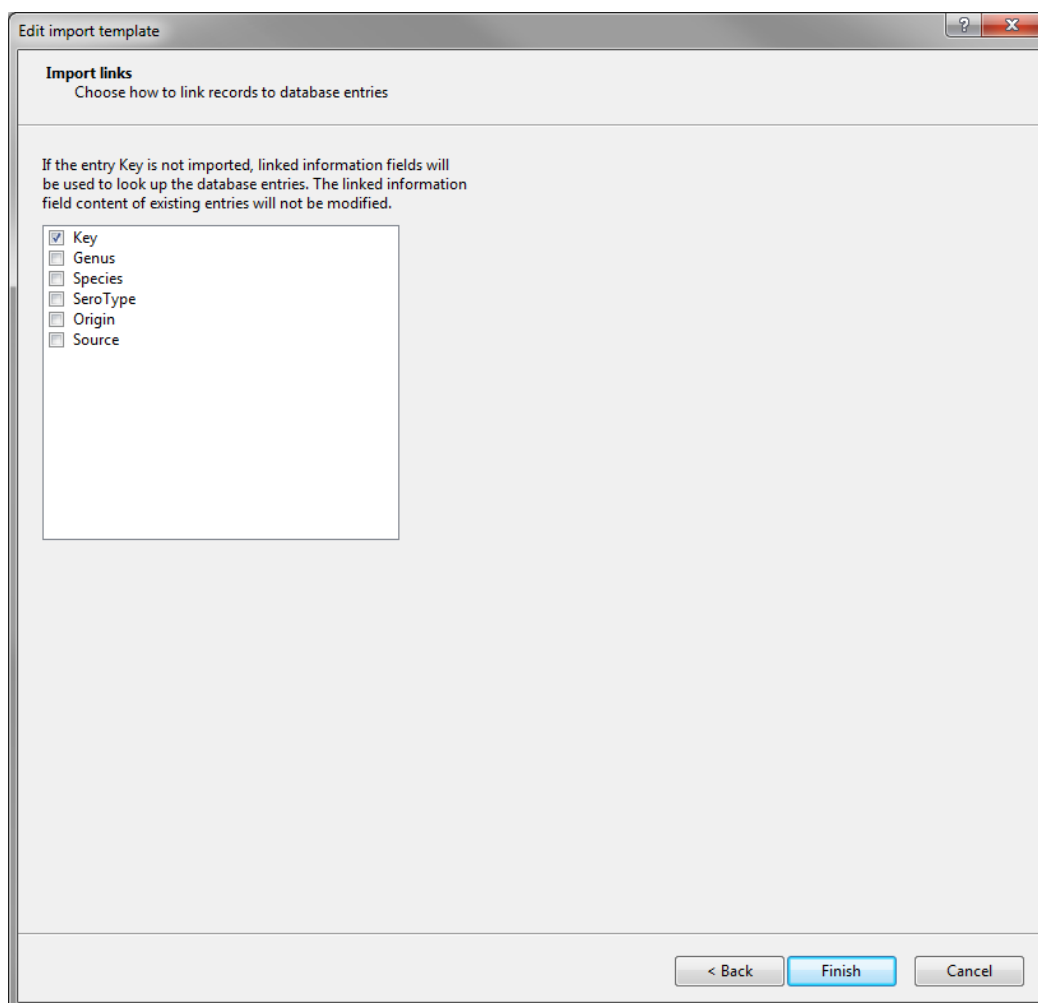
Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.



The dialog box is titled "Add data conversion rule" and contains a "Data parsing" section with the instruction "Edit the data parsing". It has two radio buttons: "Parse component: find the component '[DATA]', use '*' as wildcard" (selected) and "Regular expression: match the expression and use the subexpression". Below these are two dropdown menus: "Data parsing string:" with "[DATA]_" selected and "Data decoration:" with "[DATA]" selected. A "Preview" section shows "Data:" as "HA_inflA001-f" and "Output:" as "HA", with a "Preview..." button. At the bottom are "< Back", "Next >", and "Cancel" buttons.

Figure 8.1.18: Parsing the experiment name from the file name.



The dialog box is titled "Edit import template" and contains an "Import links" section with the instruction "Choose how to link records to database entries". A text block explains: "If the entry Key is not imported, linked information fields will be used to look up the database entries. The linked information field content of existing entries will not be modified." Below this is a list of fields with checkboxes: "Key" (checked), "Genus", "Species", "SeroType", "Origin", and "Source". At the bottom are "< Back", "Finish", and "Cancel" buttons.

Figure 8.1.19: Specify the entry link field.

Pressing **<Finish>** brings up the last step of the wizard.

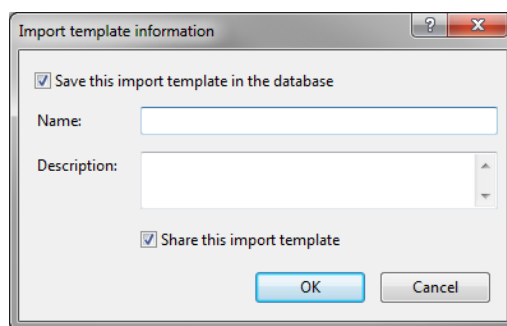


Figure 8.1.20: The *Import template information* dialog box.

Each import template has its own unique *Name*.

Optionally, a descriptive text string can be entered in the *Description* input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option *Save this import template in the database* is checked.

Check or uncheck the option *Share this import template* when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template *Name* is shown in the *Import templates panel* and is automatically selected. The template *Description* is shown in the right panel (see Figure 8.1.21).

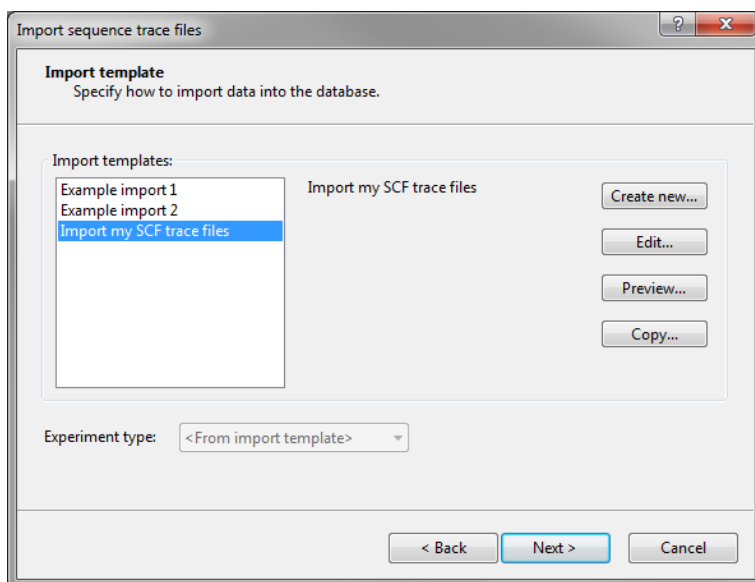


Figure 8.1.21: Import template added to the list.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the *Source* column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

If no row entry in the grid is linked to the *Sequence type name* destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (**Create New**). When sequences are linked to a new sequence type experiment, a dialog box pops up when pressing the **<Next>** button, prompting for the sequence type name.

If a row in the grid is linked to the *Sequence type name* destination, the text **From import template** is automatically selected in the *Experiment type* text box. The import tool will link the sequences to the corresponding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the **<Next>** button, prompting for the sequence type names.

The *Database links* wizard page prompts for some final settings.

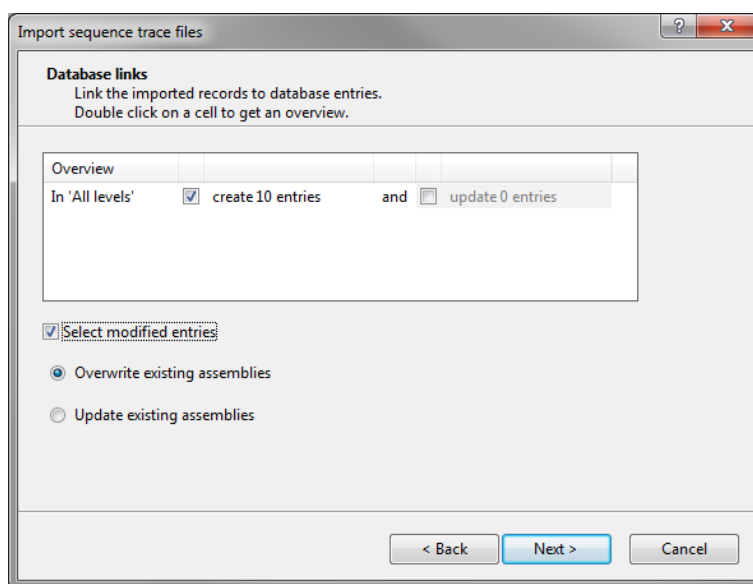


Figure 8.1.22: The *Database links* wizard page.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

With the **Existing assemblies** options one can specify which action should be undertaken in case an assembly is already present for any key/experiment pair to be imported:

- **Update existing assemblies:** Assemblies with the same name are replaced and new trace files added. A new alignment is created (any manual editing will be lost).
- **Overwrite existing assemblies:** The existing assemblies are completely removed and new assemblies are created.



The program does **NOT** check if a key/experiment pair has a sequence experiment without assembly, i.e. when a nucleic acid sequence was imported in another way than via the Assembler program. These key/experiment pairs will be overwritten without warning!

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create x entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing <Next> will bring up the *Processing* wizard page, the final step of the wizard (see 8.1.3.2.5).

8.1.3.2.4 Import wizard: assembling FASTA files

Selecting **Import and assemble trace files** under **Sequence type data** in the *Import* dialog box and pressing <Import> opens the *Import sequence traces* wizard page (see Figure 8.1.23).

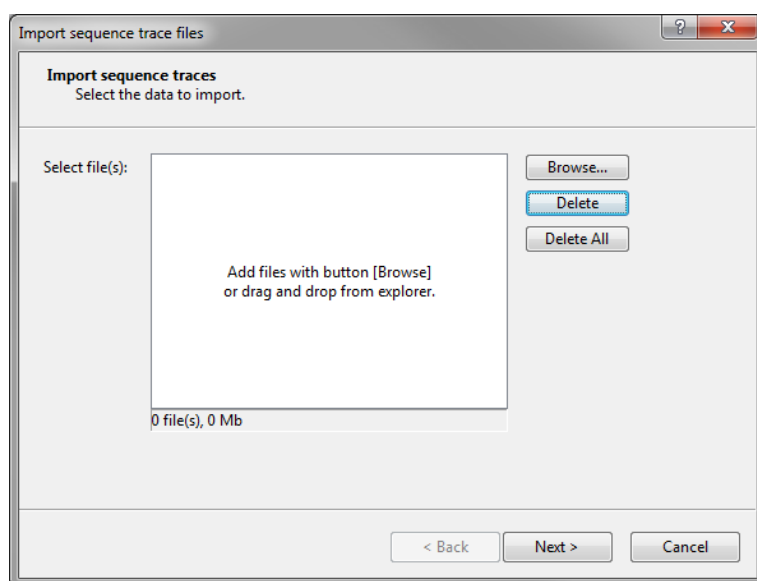


Figure 8.1.23: The *Import sequence traces* wizard page.

Pressing the <Browse> button allows you to select the file(s) that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. The number of files and total size is displayed below the list.

With the <Delete> button all selected files are removed from the import list. All files are deleted at once from the import list when pressing <Delete All>.

Pressing <Next> will display the next step.

When sequences are stored in FASTA format, each sequence begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than (>) symbol. The description line contains the *FASTA tags*, separated by a pipe (|) symbol. Each *FASTA tag* corresponds to a row in the grid (maximum 20 FASTA tags can be parsed from the description line). The text **FASTA field** is specified in the **Source type** column and the position of the tags in the description line is displayed in the **Source** column.

Using the last row in the grid, the (parsed) file name of the selected file can be stored in the database. You might need to scroll down the list to view the last row entry. The text **File** is specified in the **Source type** column and the text **Name** is displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields, sequence information fields, or sequence types. Initially the rows are not linked to any information (the **Destination type** and **Destination** for all rows is set to <None>). Specifying a *destination* for one or more selected rows can be done by pressing the <Edit destination> button or by double-clicking. This action pops up a new dialog

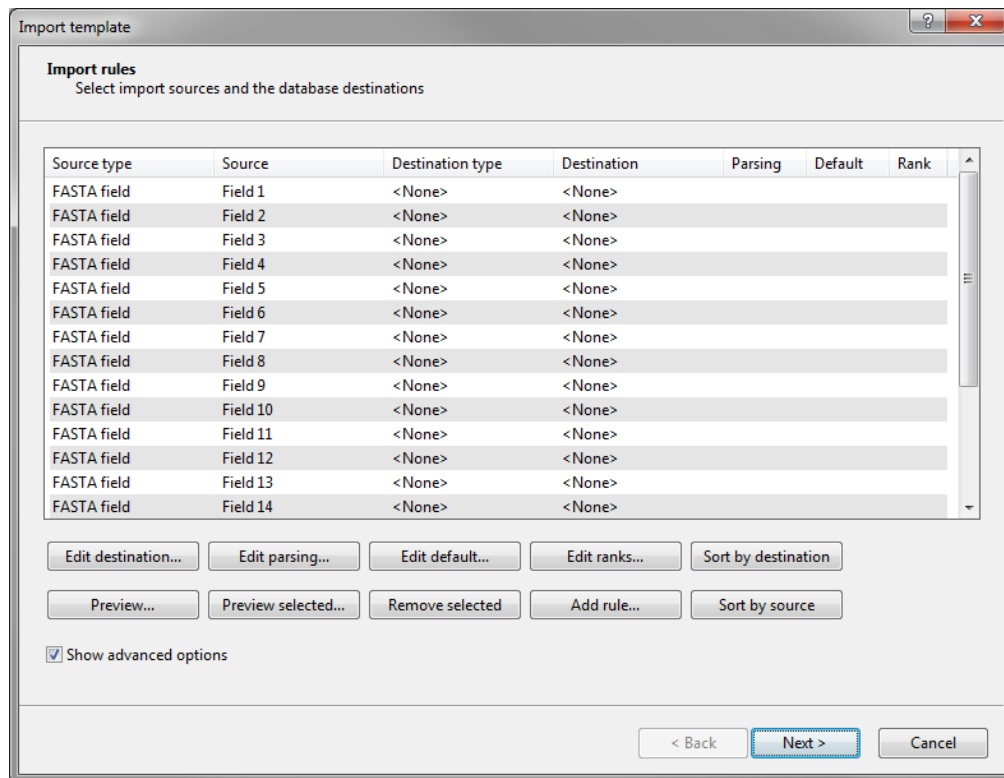


Figure 8.1.24: The *Import rules* dialog box.

box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

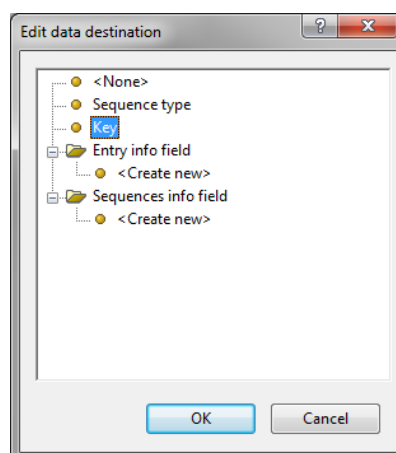


Figure 8.1.25: Edit data destination for a single selected row entry.

The information of the selected row can be linked to:

- The default information field **Key**.

- A **Sequence type** name. The (parsed) information will hold the sequence type name.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select **<Create new>** or select an existing field under the topic **Sequence info field**, respectively).

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. When a row is linked to a sequence type name, the **Destination type** and **Destination** columns display the text **Sequence type**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to sequence information fields, the text **Sequence info field** is displayed in the **Destination type** column; the name of the field is listed in the **Destination** column.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in [3.3.5.5](#).

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

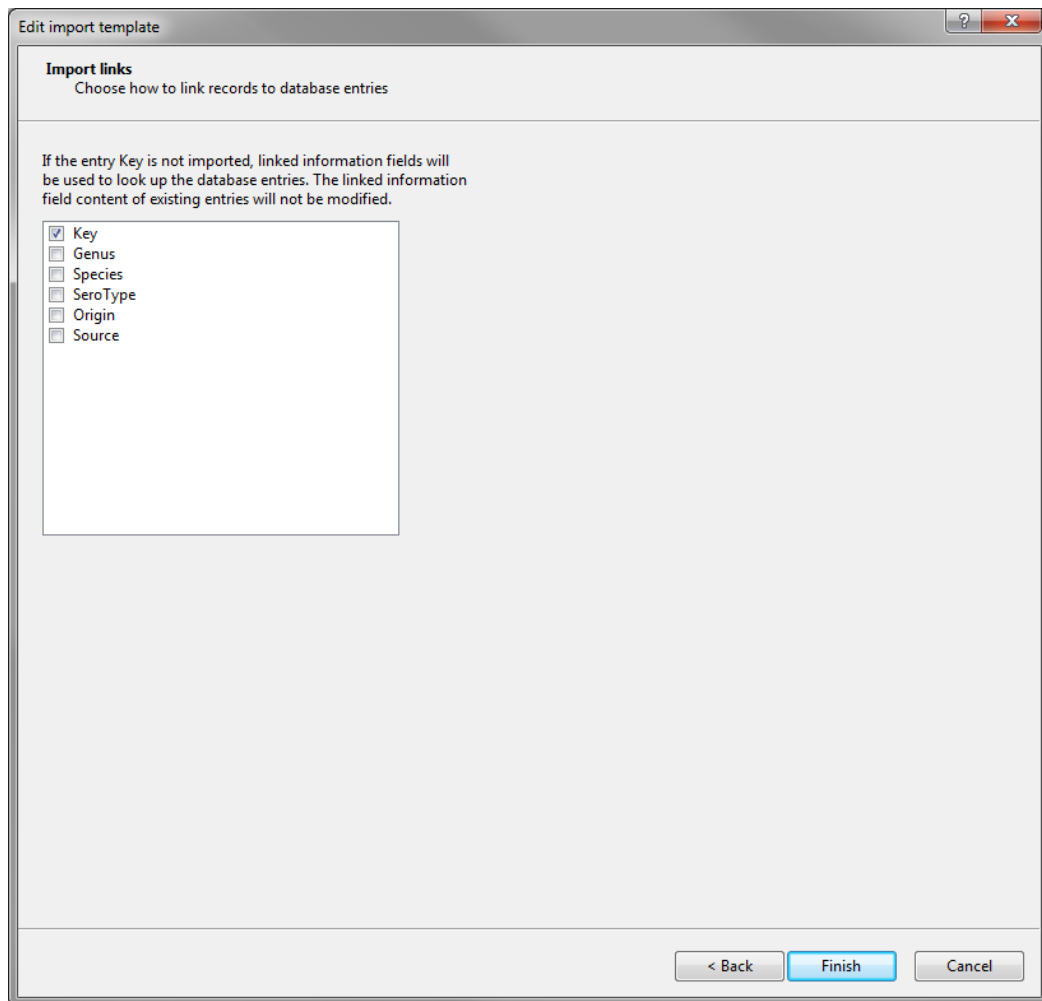


Figure 8.1.26: Specify the entry link field.

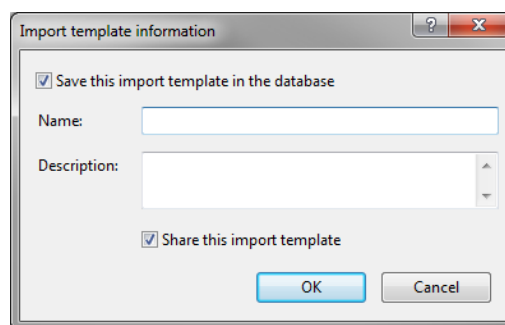


Figure 8.1.27: The *Import template information* dialog box.

The *Import templates panel* lists all FASTA templates that have been created and stored in the database.

Pressing the **<Create new>** button brings up the *Import rules* dialog box again allowing you to define a new import template.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not

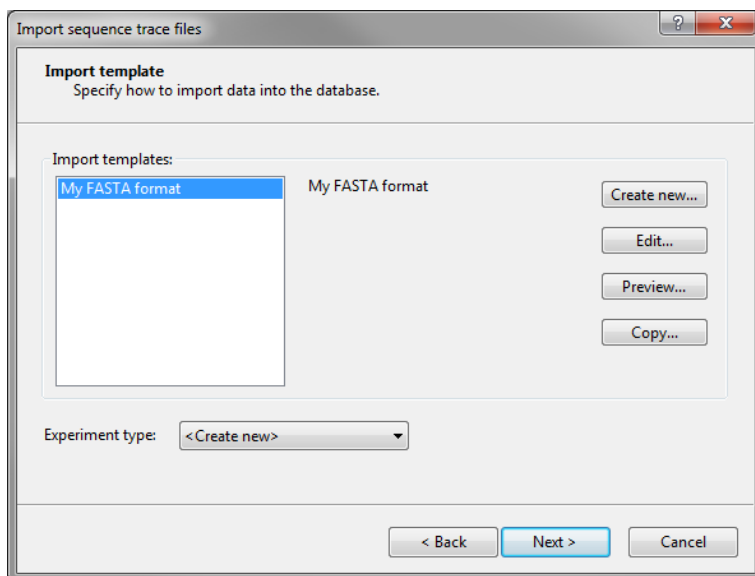


Figure 8.1.28: The *Import template* wizard page.

be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

If no row entry in the grid is linked to the **Sequence type name** destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (**Create New**). When sequences are linked to a new sequence type experiment, a dialog box pops up when pressing the **<Next>** button, prompting for the sequence type name.

If a row in the grid is linked to the **Sequence type name** destination, the text **From import template** is automatically selected in the **Experiment type** text box. The import tool will link the sequences to the corresponding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the **<Next>** button, prompting for the names.

The *Database links* wizard page prompts for some additional settings.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

With the **Existing assemblies** options one can specify which action should be undertaken in case an assembly is already present for any key/experiment pair to be imported:

- **Update existing assemblies:** Assemblies with the same name are replaced and new trace files added. A new alignment is created (any manual editing will be lost).
- **Overwrite existing assemblies:** The existing assemblies are completely removed and new assemblies are created.

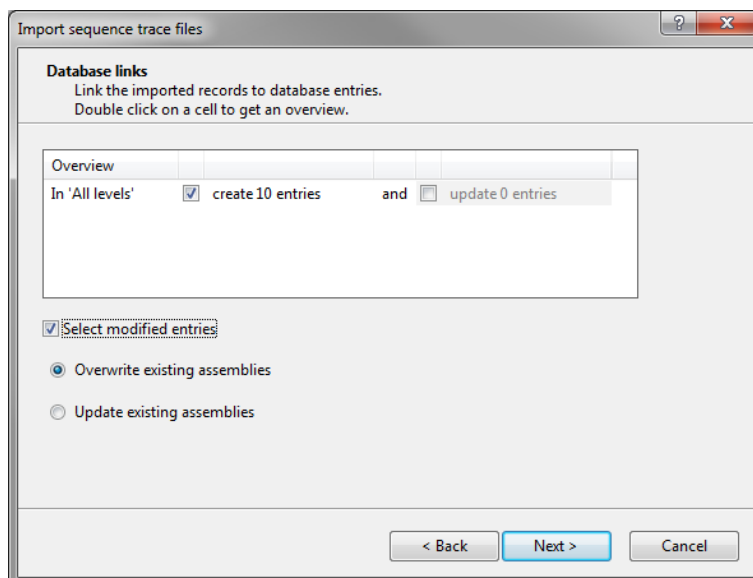


Figure 8.1.29: The *Database links* wizard page.



The program does **NOT** check if a key/experiment pair has a sequence experiment without assembly, i.e. when a nucleic acid sequence was imported in another way than via the Assembler program. These key/experiment pairs will be overwritten without warning!

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing <*Next*> will bring up the *Processing* wizard page, the final step of the wizard (see 8.1.3.2.5).

8.1.3.2.5 Processing

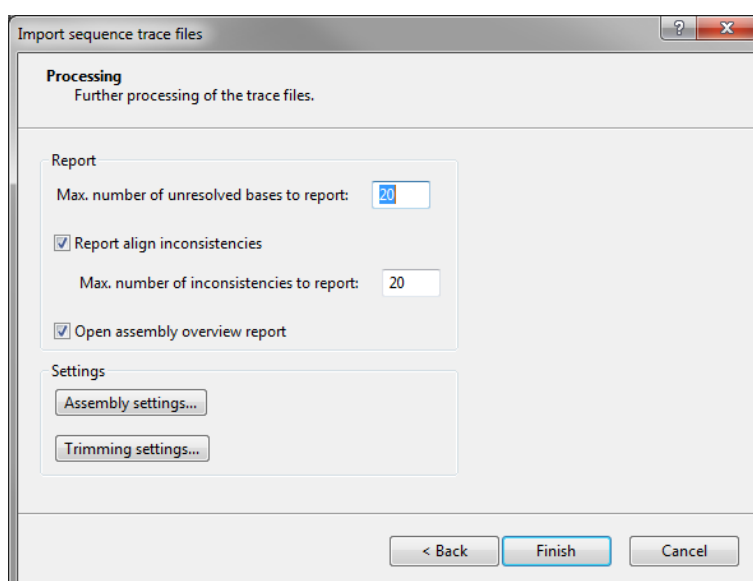


Figure 8.1.30: The *Processing* wizard page.

Under **Reports**, the **Maximum# of unresolved bases reported** can be specified (default value 20). Likewise,

the **Maximum #of align inconsistencies reported** can be entered (default value 20). Align inconsistencies are positions where the consensus is resolved, but where one or more trace sequences are different from the consensus sequence. Using the check box, one can also choose not to display these inconsistencies at all.

A number of settings can be saved with an experiment type. These are divided in *Assembly settings* and *Trimming settings*.

Assembly settings:

In case the sequences are linked to different sequence type experiments, the *Assembly settings* dialog box appears when pressing the <Assembly settings> button, displaying all sequence type experiments.

When all sequences are linked to the same sequence type experiment, the *Assembly settings* dialog box is called when pressing the <Assembly settings> button.

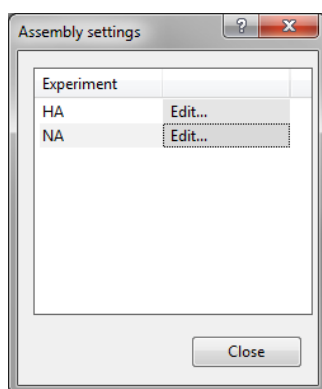


Figure 8.1.31: The *Assembly settings* dialog box.

All sequence type experiments that are included in the import routine are displayed in the *Assembly settings* dialog box. The individual assembly settings for a sequence type experiment can be called by double-clicking <Edit...> next to the **Experiment** name. This action calls the *Assembly settings* dialog box.

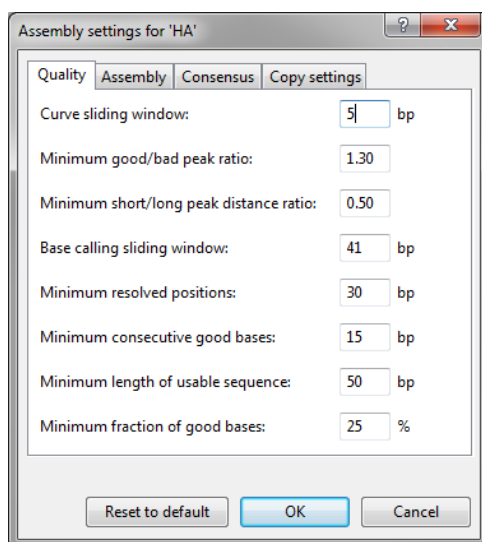


Figure 8.1.32: The *Assembly settings* dialog box.

The name of the experiment type to which the assembly settings apply is displayed in the title of the dialog.

The Assembly settings are grouped in tabs per settings dialog box in *Assembler*: **Quality** assignment, **Assembly** and **Consensus** determination (see 8.1.3.7 for more information about these parameters).

Check **Remove gaps from consensus** in the *Consensus* tab if you want Assembler to automatically remove gaps from the consensus sequence. Any removed gap will be reported as a warning. This setting is only applicable in the *Contig assembly* window, since gaps are always removed when a sequence is saved to the *Experiment card* window.

The default assembly settings are displayed. In most cases, these default settings will work fine. It is, however, possible to modify any of the settings listed in the dialog box and save them to the experiment type. If you have made modifications to the default assembly settings, press **<OK>** to save the modified settings to the experiment type and close the *Assembly settings* dialog box.

Pressing **<Reset to default>** will display the default assembly settings in all tabs.

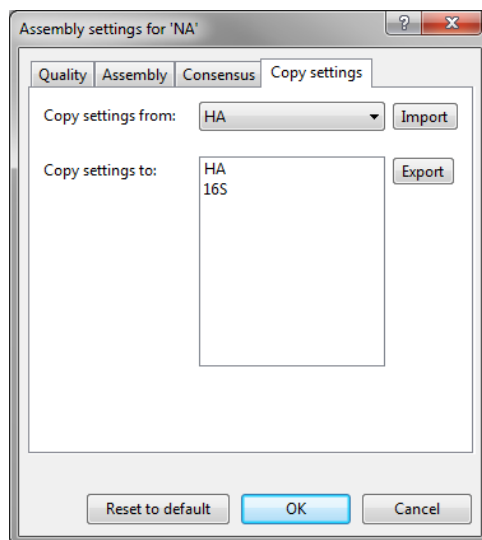


Figure 8.1.33: The *Consensus* tab.

In the *Copy settings* tab assembly settings can be copied from and to another sequence experiment that is present in the database (see Figure 8.1.33):

- From the pull-down list, select the experiment from which you want to import the assembly settings and press **<Import>**. The assembly settings are imported from the selected experiment.
- To export the settings to another sequence experiment, select the experiment(s) in the grid and press the **<Export>** button.

Trimming settings:

In case the sequences are linked to different sequence type experiments, the *Assembly trimming settings* dialog box appears when pressing the **<Trimming settings>** button, displaying all sequence type experiments.

When all sequences are linked to the same sequence type experiment, the *Assembly trimming settings* dialog box is called when pressing the **<Trimming settings>** button.

All sequence type experiments that are included in the import routine are displayed in the *Assembly trimming settings* dialog box. The trimming patterns defined for each experiment are displayed in the **Start pattern** and **Stop pattern** columns. The individual trimming settings for a sequence type experiment can be called by double-clicking **<Edit...>** next to the **Experiment** name. This action calls the *Assembly trimming settings* dialog box.

The name of the experiment type for which the trimming settings apply is displayed in the title of the dialog. In contrast to the assembly settings, which mainly depend on the quality of the trace files, the trimming settings are likely to be specific for each experiment.

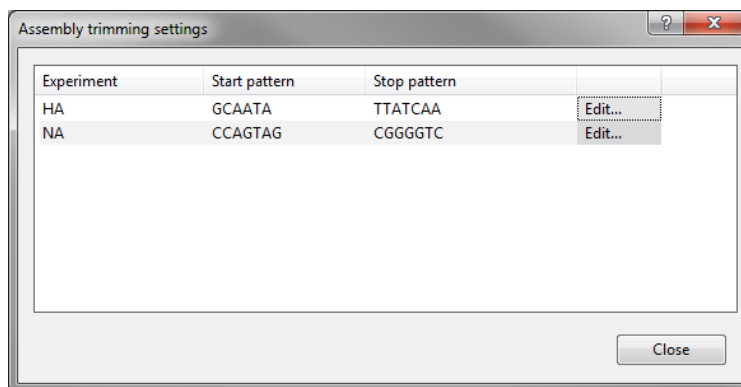


Figure 8.1.34: The *Assembly trimming settings* dialog box.

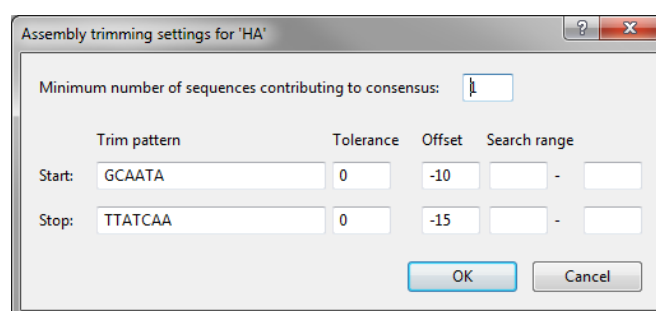


Figure 8.1.35: The *Assembly trimming settings* dialog box.

Minimum number of sequences contributing to consensus (default value 1) specifies a minimum number of trace sequences that should be contributing to the subsequence on the consensus that matches the trimming targets. For example, if “2” is entered, a trimming target will only be set if the matching region on the consensus is *fully* defined by at least 2 sequences.

For both the **Start** and **Stop** position, a **Trim pattern** can be entered as a sequence of bases. The use of IUPAC code for ambiguous positions is supported.

The **Tolerance** is the number of mismatches allowed for a sequence still to be recognized as a trim pattern.

With the **Offset**, one can specify that the consensus is trimmed at a certain offset from the start and end trimming target positions. If no offset is specified (zero), the trimming targets are *included* in the trimmed consensus.

The **Search range** is to restrict the search to certain regions on the consensus, e.g. to prevent incidental matches inside the targeted consensus sequence.

The entered trim patterns will be searched on the consensus sequence in both directions, i.e. on the consensus as it appears as well as on its complementary strand. In case the trim patterns match the complementary strand of the consensus, it will be automatically invert-complemented. If the **Trim pattern** text boxes are left empty, no preference sense is available.

Pressing <OK> saves the settings to the experiment type and closes the *Assembly trimming settings* dialog box.

Pressing <Finish> in the *Processing* wizard page will assemble the selected trace files.

When the assemblies are processed, the interactive *Batch sequence assembly report* window will appear when the option **Open assembly overview report** is checked (see 8.1.3.2.6). This window can also be displayed from the *Main* window with *Analysis* > *Sequence types* > *Batch assembly reports...*

8.1.3.2.6 Batch assembly reports

For every contig project (key/experiment pair) assembled using the *Import and assemble trace files* and *Import and assemble traces from FASTA text files* import routines some information is stored in the connected database, which can be shown in reports. These reports can display the contig status, list unresolved bases, align inconsistencies or other errors that occurred during assembly of a batch. The reports also form an interactive link with the contig project in the Assembler program, allowing the user to correct assembly errors and update the reports accordingly.

By default, an overview report is displayed each time a batch of trace files has been assembled (the option *Open assembly overview report* is default checked in the *Processing* wizard page).

Overview reports can also be displayed from the *Main* window, either for all entries or a selection, or for each batch of assembled sequences. The *Analysis* > *Sequence types* > *Batch assembly reports...* action calls the *Batch assembly reports* dialog box (see Figure 8.1.36).

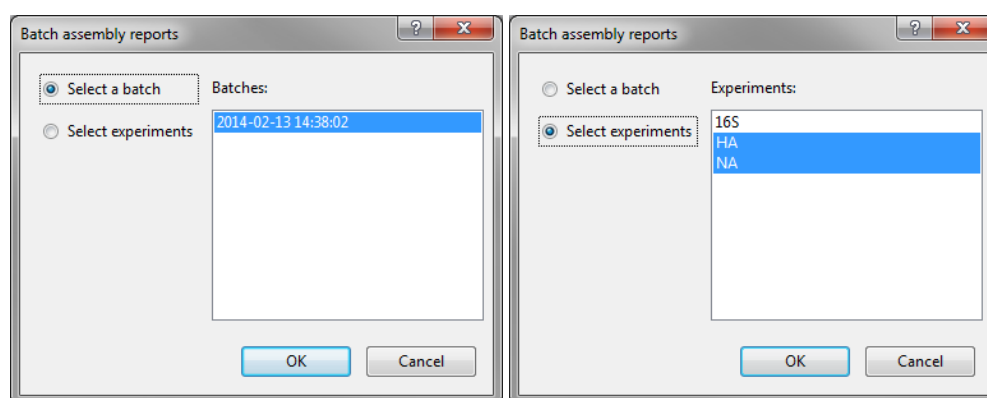


Figure 8.1.36: Select a batch or experiment(s).

When *Select a batch* is checked, all available batches in the database are displayed on the right-hand side of the dialog. Select a batch from the list for which you want to generate a report and press <OK> to display the report window.

When the option *Select experiments* is checked, all available sequence type experiments are listed. All highlighted experiments will be included in the report. Multiple experiments can be selected using the **Shift**- and **Ctrl**- keys. Pressing <OK> will include all selected entry/experiment pairs in the report window. When no selection is present, BioNumerics will ask to confirm the creation of a report including all entries displayed in the *Database entries* panel (see Figure 8.1.37).

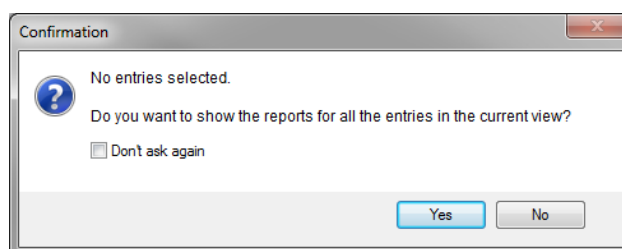


Figure 8.1.37: Confirmation message.

In the *Batch sequence assembly report* window, the *Overview* panel displays the entries (keys) as rows and the experiments as columns. Each cell in the grid, corresponding to a key/experiment pair, provides information about the current status of the contig project. This information can be:

- **N/A**: No such experiment exists with this key.

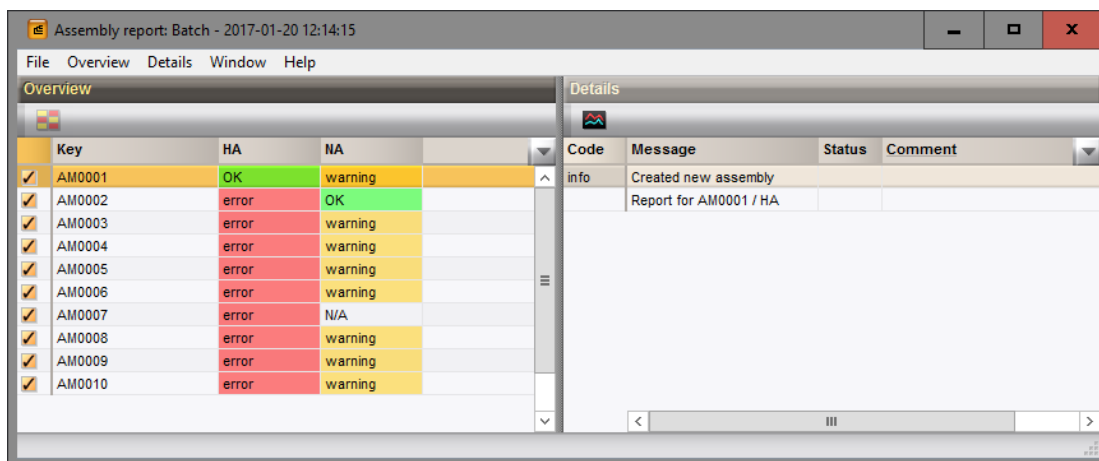


Figure 8.1.38: The *Batch sequence assembly report* window.

- **N/B**: An experiment with this key exists, but (a) the assembly was not created from this batch; or (b) no assembly is present for this sequence.
- **OK** (green): A contig was assembled without any problems.
- **Warning** (orange): Align inconsistencies occurred that were resolved under the applied consensus determination settings.
- **Error** (red): At least one of several possible assembly errors occurred, e.g. a trace sequence did not meet the quality criteria, more than one contig was created, the trimming positions were not found or unresolved bases are present in the consensus.
- **Read** (red): A warning or error that was read by the user, but not solved yet.
- **Solved** (green): A warning or error that was solved by the user (see below).

In the *Overview* panel, individual entries can be selected or unselected using the check boxes. Check boxes for selected entries are indicated as ☒. Ranges of entries can be selected by selecting the first entry in the range and, while holding the **Shift**-key, clicking the last entry in the range.

Select **Overview** > **Hide solved entries** (🔍) if you only want to display those entries for which at least one warning or error is reported. To display the complete report again, including the "solved" entries, select **Overview** > **Hide solved entries** (🔍) again.

Clicking on a cell in the *Overview* panel, updates the information in the *Details* panel. The *Details* panel is organized in message rows with four columns.

- The first column displays a message **Code**, which can be either "info", "warning" or "error".
- The second column shows the actual **Message**. Double-clicking on this cell opens the *Contig assembly* window, with the corresponding position highlighted.
- The third column displays the **Status** of the message, which can be "new", "read" or "solved". The status can be changed by the user.
- The fourth column is a **Comment** field. A comment can be entered by the user.

Double-click on a message cell in the *Details* panel. Alternatively, click on a message cell and select **Details** > **Open in assembler** (🔍, **Enter**). This will open the sequence in the *Contig assembly* window, with the

corresponding position in focus (see Figure 8.1.39). The position can now be examined and - if needed - the base calling can be changed manually.

In case of the unresolved base highlighted in Figure 8.1.39, the "T" needs to be changed into a "y". The base will be resolved under the default assembly settings and is no longer highlighted in red.

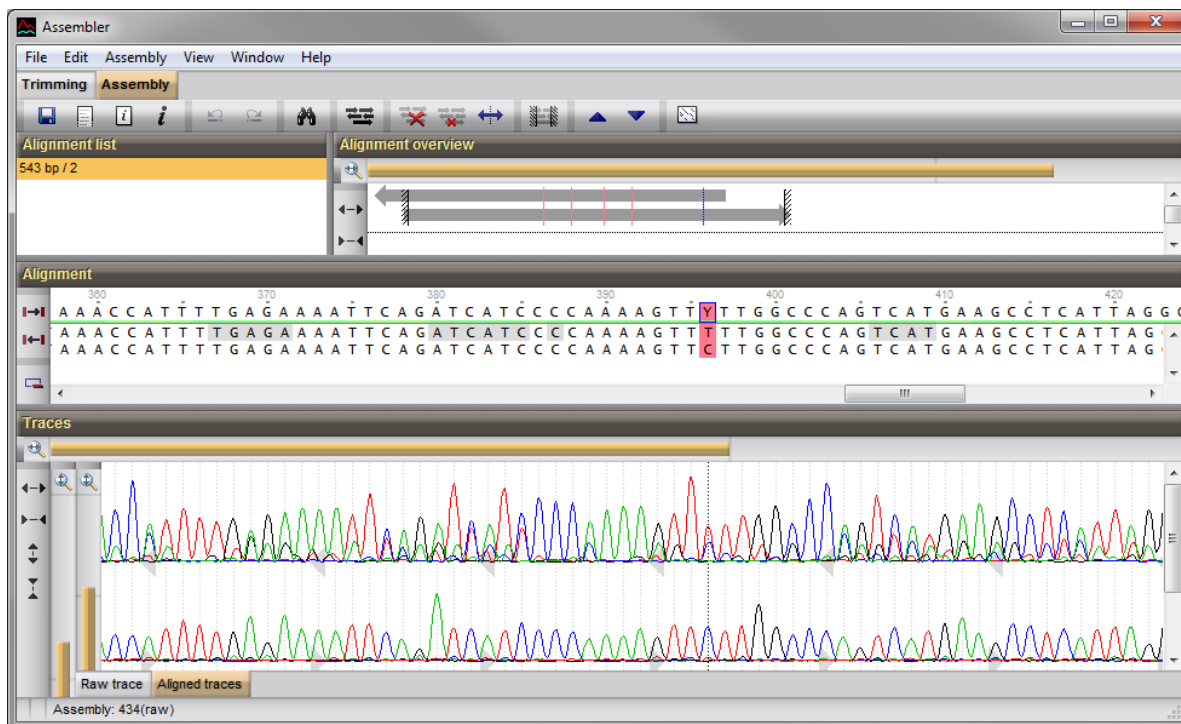


Figure 8.1.39: The *Contig assembly* window as called from the *Details* panel by double-clicking an error message. The window shows the contig project with the unresolved base in focus.

A comment can be specified for each message in the *Details* panel by clicking twice in the **Comment** field. The text box will appear highlighted and a comment can be added.

The status of a message in the *Details* panel can be set to "solved" by clicking on the message and selecting **Details > Set message to solved (S)**. To set all messages in the *Details* panel to "solved" select **Details > Set all messages to solved (Ctrl+S)**.

Individual messages or all messages at once can be set to "read" by selecting **Details > Set message to read (R)** or **Details > Set all messages to read (Ctrl+R)**, respectively.

As an alternative, select **Batch sequence assembly > Set report to solved** in the *Contig assembly* window. Select **File > Save (Ctrl+S)** to save the contig project and then close the *Contig assembly* window.



Assembly errors can be reviewed in Assembler as well, without selecting the errors in the *Details* panel. To make the cursor jump to the next unresolved position, select **View > Next unresolved problem (Ctrl+Right)**. To make the cursor jump to the previous unresolved position, select **View > Previous unresolved problem (Ctrl+Left)**. However, it is important to note that align inconsistencies (reported as "warning") will not be highlighted this way since, by definition, these positions are resolved under the current consensus determination rules.

Conveniently, the step described above could also be combined in a single action: **Batch sequence assembly > Set report to solved, save and close (Ctrl+Shift+S)**.

When all warning and error messages in the *Details* panel are set to solved, the corresponding key/experiment cell in the *Overview* panel is updated and displayed in green. However, instead of "OK", the field says "solved" to remind the user that some modifications have been made after the batch sequence assembly has

completed.

The *Batch sequence assembly report* window is closed with **File** > **Exit**. All *Contig assembly* window that were opened from the overview report are closed simultaneously.

In the *Batch sequence assembly report* window it is possible to remove all contig projects included in the report and **re-assemble** the traces files using other (or the same) assembly and trimming settings. Since new alignments will be created using this action any manual editing will be lost.

Select **Overview** > **Re-assemble...** to call the *Re-assemble trace files* dialog box (see Figure 8.1.40). This dialog provides access to the trimming and assembly settings and report options as present in the *Processing* wizard page.

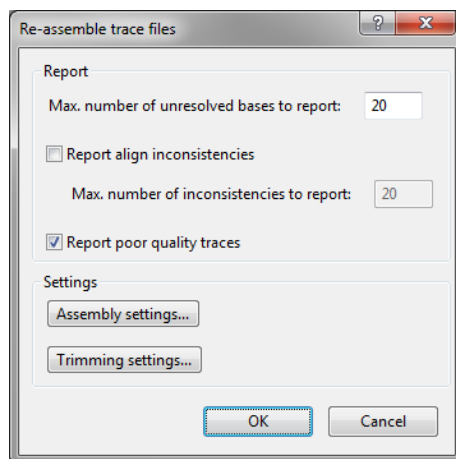


Figure 8.1.40: The *Re-assemble trace files* dialog box.

The *Maximum# of unresolved bases reported* can be specified (default value 20).

Likewise, the *Maximum #of align inconsistencies reported* can be entered (default value 20). Align inconsistencies are positions where the consensus is resolved, but where one or more trace sequences are different from the consensus sequence. Using the check box, one can also choose not to display these inconsistencies at all.

When poor quality traces are present, it is possible to report this as an error by checking the option **Report poor quality traces as errors**.

Pressing the <**Assembly settings**> button calls the *Assembly settings* dialog box when only one sequence type is included in the report, otherwise the *Assembly settings* dialog box will appear. In the *Assembly settings* dialog box the assembly settings can be modified and saved with each individual sequence type.

Pressing the <**Trimming settings**> button calls the *Assembly trimming settings* dialog box when only one sequence type is included in the report, otherwise the *Assembly trimming settings* dialog box will appear. In the *Assembly trimming settings* dialog box the trimming settings can be modified and saved with each individual sequence type.

Pressing <**OK**> removes all existing assemblies, re-assembles the trace files using the defined settings and updates the reports.



When an assembly has been re-assembled, this will be indicated with "info" messages in the *Details* panel, including the date when the re-assembly was done and previous status.

In the *Batch sequence assembly report* window it is possible to **re-trim** all projects using the same or different trimming settings. Re-trimming the contigs will not remove any manual editing done on the contigs.

Select **Overview** > **Re-trim...** to call the *Re-trim assemblies* dialog box (see Figure 8.1.41). This dialog provides access to the report and trimming settings as present in the *Processing* wizard page.

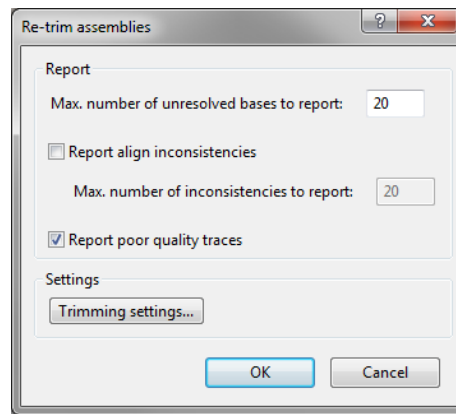


Figure 8.1.41: The *Re-trim assemblies* dialog box.

The *Maximum# of unresolved bases reported* can be specified (default value 20).

Likewise, the *Maximum #of align inconsistencies reported* can be entered (default value 20). Align inconsistencies are positions where the consensus is resolved, but where one or more trace sequences are different from the consensus sequence. Using the check box, one can also choose not to display these inconsistencies at all.

When poor quality traces are present, it is possible to report this as an error by checking the option **Report poor quality traces as errors**.

Pressing the <**Trimming settings**> button calls the *Assembly trimming settings* dialog box when only one sequence type is included in the report, otherwise the *Assembly trimming settings* dialog box will appear. In the *Assembly trimming settings* dialog box the trimming settings can be modified and saved with each individual experiment type.

Pressing <**OK**> re-trims all assemblies in the report window and updates the reports.



When an assembly has been re-trimmed, this will be indicated with "info" messages in the *Details* panel, including the date when the re-trimming was done and previous status.

In case some or all of the assembly errors and alignment inconsistencies have been solved in Assembler, a useful option is to **re-create the assembly reports** for the current status of the contig projects:

Select **Overview > Re-evaluate....** The *Re-evaluate assemblies* dialog box will pop up, providing access to the report settings as present in the *Processing* wizard page (see Figure 8.1.42).

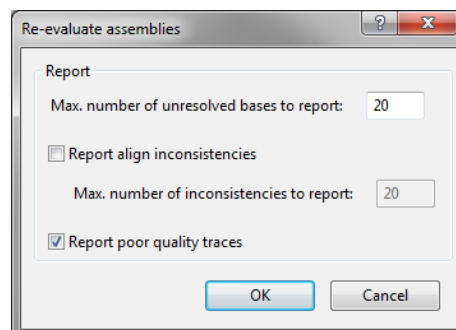


Figure 8.1.42: The *Re-evaluate assemblies* dialog box.

The *Maximum# of unresolved bases reported* can be specified (default value 20).

Likewise, the *Maximum #of align inconsistencies reported* can be entered (default value 20). Align inconsistencies are positions where the consensus is resolved, but where one or more trace sequences are different

from the consensus sequence. Using the check box, one can also choose not to display these inconsistencies at all.

When poor quality traces are present, it is possible to report this as an error by checking the option **Report poor quality traces as errors**.

Pressing <OK> re-evaluates the assemblies and updates the reports.



When an assembly has been re-evaluated, this will be indicated with "info" messages in the *Details* panel, including the date when the re-evaluation was done and previous status.

8.1.3.2.7 Data parsing tutorials

The example dataset for the four tutorials (see further) can be downloaded from the Applied Maths website (<http://www.applied-maths.com>, click on "Batch assembly & alignment data"). The trace files originate from influenza A virus strains and represent partial sequences of the haemagglutinin (HA) and neuraminidase (NA) genes. These publicly available trace files were downloaded from the NCBI Trace Archive (<http://0-www.ncbi.nlm.nih.gov.catalog.llu.edu/Traces/trace.cgi?>).

Depending on your own data, one of these four tutorials will agree best with your own situation. In the tutorials, new experiments will be created, but obviously, batch sequence assembly is also possible for sequences belonging to already existing experiments.

Case 1: Only the sample identifiers can be parsed from the trace file names

We will assemble the partial HA sequences from the example dataset in batch.

- 3.1 Select **File > Import...** (📁, **Ctrl+I**) to call the *Import* dialog box.
- 3.2 Select **Import and assemble trace files** under *Sequence type data* and press <Import> to start the batch import routine.
- 3.3 Browse for the correct folder, select the 20 HA sequence trace files and press <Open>.

All selected trace files are displayed in the next step (see Figure 8.1.43).

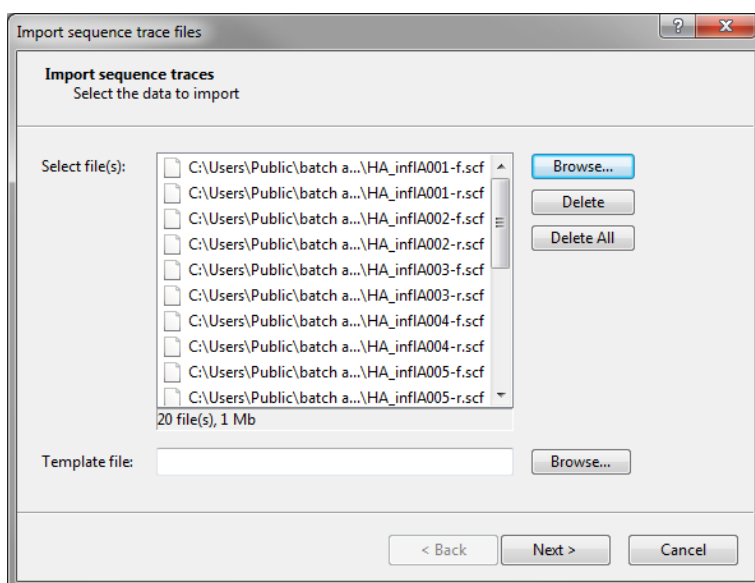


Figure 8.1.43: Selected trace files.

- 3.4 Press <Next> to go to the next step.

The way the information should be imported in the database can be specified with an import template. In this tutorial, the **Key** will be parsed from the trace file name. A new import template needs to be defined:

3.5 Press the **Create new** button to call the *Import rules* dialog box.

The only source of information available in the newly created import template is the file name.

3.6 Double-click on the **Name** row or select the row and press <**Edit destination**>. Select **Key** as destination and press <**OK**> (see Figure 8.1.44).

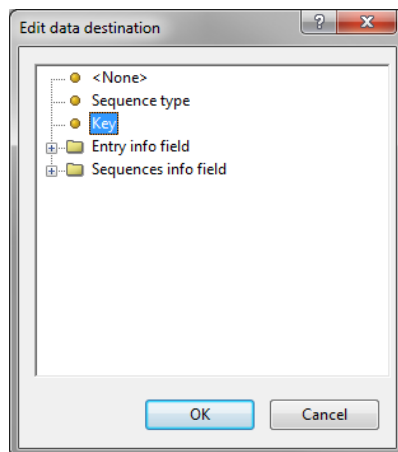


Figure 8.1.44: The *Edit data destination* dialog box.

The import rule in the *Import rules* dialog box is updated.

3.7 Check the option **Show advanced options** and press the <**Edit parsing**> button.

3.8 In the *Data parsing* dialog box, fill in following data parsing string: “*_[DATA]-*”.

This parsing string will use the string composed of any character occurring after the underscore (_) and before the hyphen (-) as an entry key. The asterisk (*) serves as a wildcard to omit characters. In the parsing string used above, any characters before the underscore and after the hyphen will be ignored.

3.9 Press the <**Preview**> button and press <**OK**> when the parsing is correct (see Figure 8.1.45).

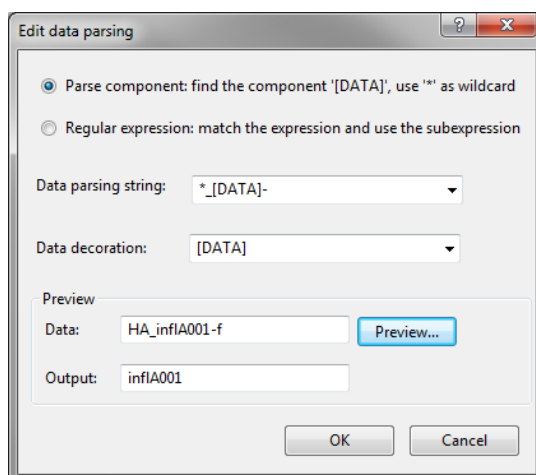


Figure 8.1.45: Data parsing string.

3.10 Press <**Next**> and <**Finish**>.

- 3.11 Specify a template name, e.g. **Import SCF trace files, case 1**, optionally enter a description (e.g. "Only parse entry keys from file names") and press **<OK>**.

The template is added to the template list (see Figure 8.1.46).

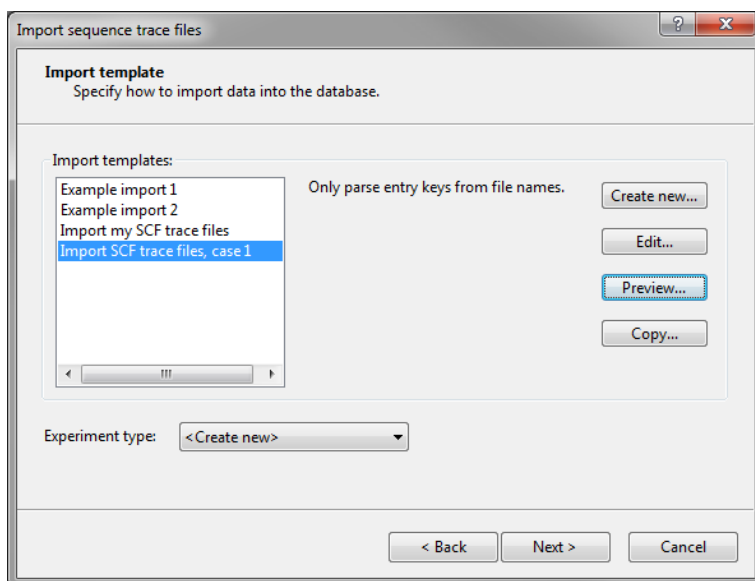


Figure 8.1.46: New import template.

- 3.12 Make sure the newly created template is selected and press the **<Preview>** button.

The preview should now look like Figure 8.1.47.

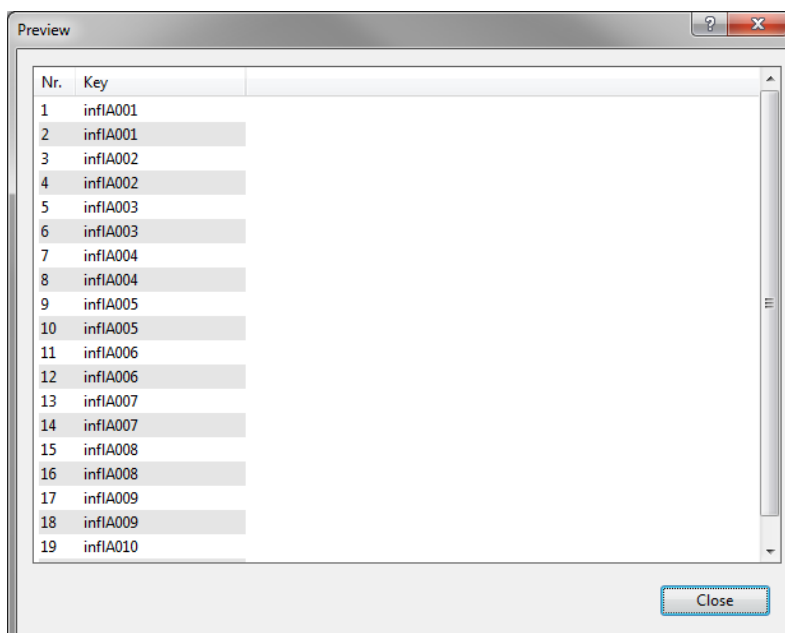


Figure 8.1.47: Preview of the parsing.

- 3.13 Close the preview.

- 3.14 Make sure the newly created template is selected, select **Create new** as **Experiment type** and press **<Next>**.

3.15 Specify an experiment name, e.g. "HA", and press **<OK>** and confirm the creation of the new sequence type in the database.

3.16 Press **<Next>** to confirm the creation of 10 new entries (see Figure 8.1.48).

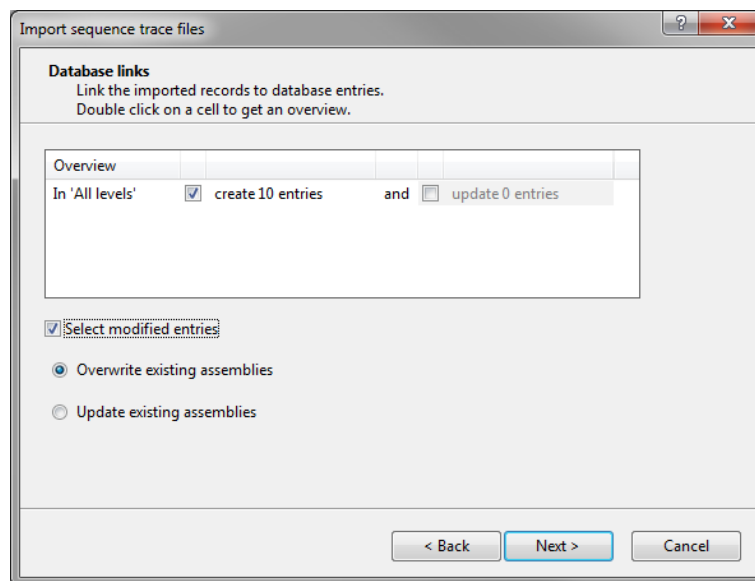


Figure 8.1.48: Database links.

The *Processing* wizard page opens.

3.17 Continue with the steps described in 8.1.3.2.8.

Case 2: The sample identifiers AND the experiments are parsed from the trace file names

We will assemble the partial HA sequences and the partial NA sequences from the example dataset in batch.

3.18 Select **File > Import...** (📁, **Ctrl+I**) to call the *Import* dialog box.

3.19 Select **Import and assemble trace files** under *Sequence type data* and press **<Import>** to start the batch import routine.

3.20 Browse for the correct folder, select all HA and NA sequence trace files and press **<Open>**.

All selected trace files are displayed in the next step (see Figure 8.1.49).

3.21 Press **<Next>** to go to the next step.

The way the information should be imported in the database can be specified with an import template. In this tutorial, the **Key** and **Experiment names** will be parsed from the trace file name. A new import template needs to be defined:

3.22 Press the **Create new** button to call the *Import rules* dialog box.

The only source of information available in the newly created import template is the file name.

3.23 Double-click on the **Name** row or select the row and press **<Edit destination>**. Select **Key** as destination and press **<OK>** (see Figure 8.1.50).

The import rule in the *Import rules* dialog box is updated.

3.24 Check the option **Show advanced options** and press the **<Edit parsing>** button.

3.25 In the *Data parsing* dialog box, fill in following data parsing string: **"*-[DATA]-*"**.

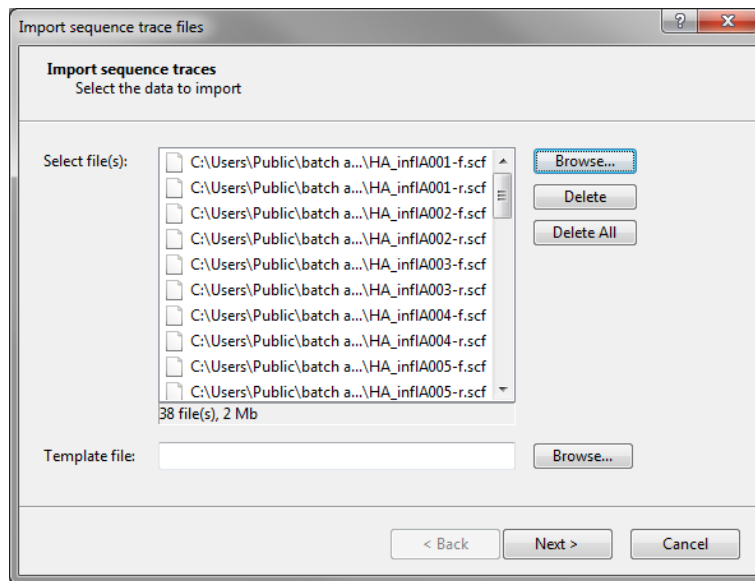


Figure 8.1.49: Selected trace files.

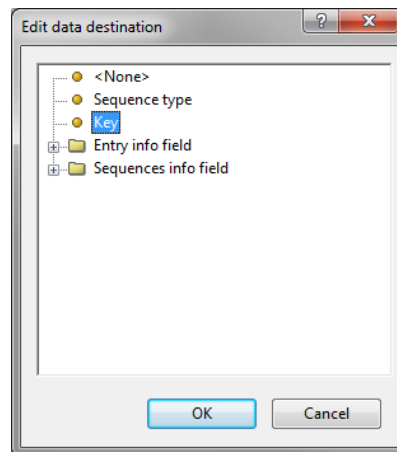


Figure 8.1.50: The *Edit data destination* dialog box.

This parsing string will use the string composed of any character occurring after the underscore (.) and before the hyphen (-) as an entry key. The asterisk (*) serves as a wildcard to omit characters. In the parsing string used above, any characters before the underscore and after the hyphen will be ignored.

3.26 Press the **<Preview>** button and press **<OK>** when the parsing is correct (see Figure 8.1.51).

Next, we will specify a new rule that links the part of the file name appearing before the underscore (.) to the *Sequence type name*.

3.27 Press **<Add rule>** and select the file **<Name>** as data source and press **<Next>** (see Figure 8.1.52).

3.28 Choose

3.29 Sequence type

as destination and press **<Next>** (see Figure 8.1.53).

3.30 In the *Data parsing* dialog box, fill in following data parsing string: “[DATA]_*”.

This parsing string will only take into account the text occurring before the first underscore (.). The asterisk

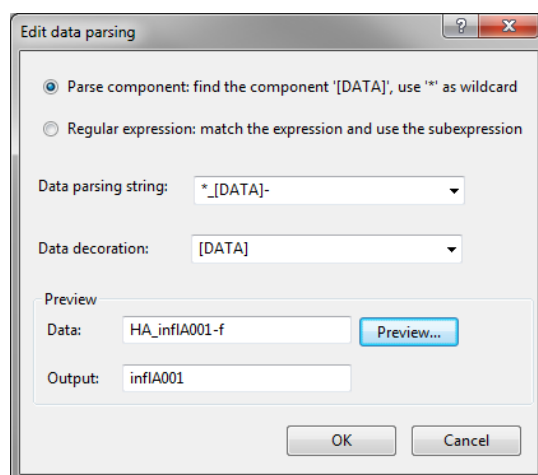


Figure 8.1.51: Data parsing string.

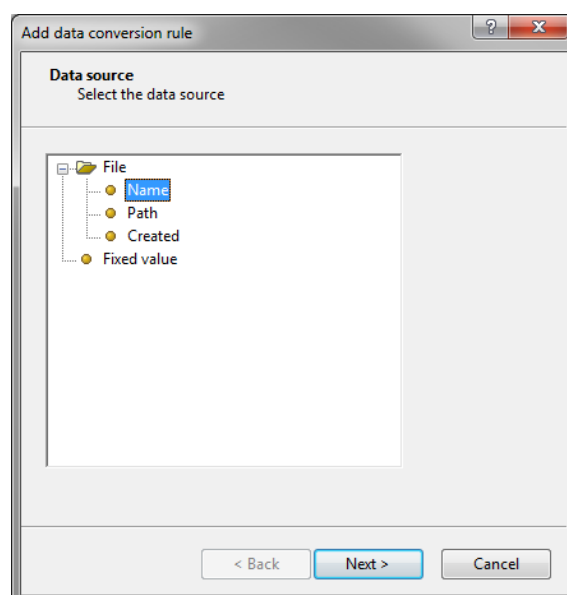


Figure 8.1.52: Data source.

(*) serves as a wildcard, meaning that all characters after the underscore will be ignored.

- 3.31 Press the **<Preview>** button and press **<Next>** when the parsing is correct (see Figure 8.1.54). Press **<Finish>** to add the rule to the *Import rules* dialog box.

The *Import rules* dialog box should now look like Figure 8.1.55.

- 3.32 Press **<Next>** and **<Finish>**.

- 3.33 Specify a template name, e.g. **Import SCF trace files, case 2**, optionally enter a description (e.g. "Parse entry keys and experiment names from file names") and press **<OK>**.

The template is added to the template list (see Figure 8.1.56).

- 3.34 Make sure the newly created template is selected and press the **<Preview>** button.

The preview should now look like Figure 8.1.57.

- 3.35 Close the preview.

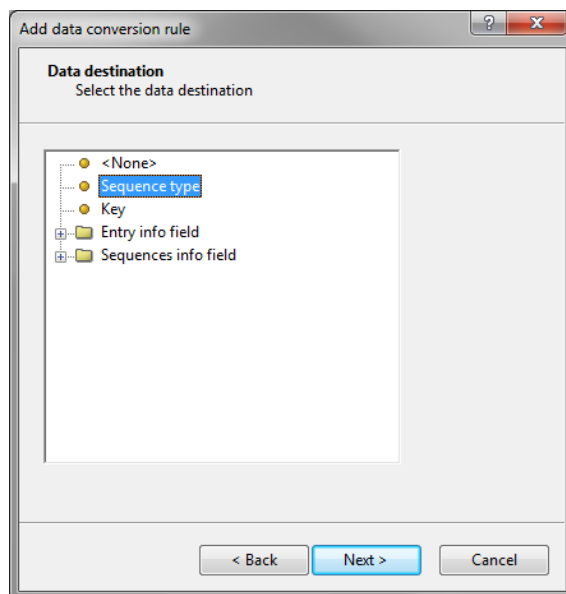


Figure 8.1.53: Data destination.

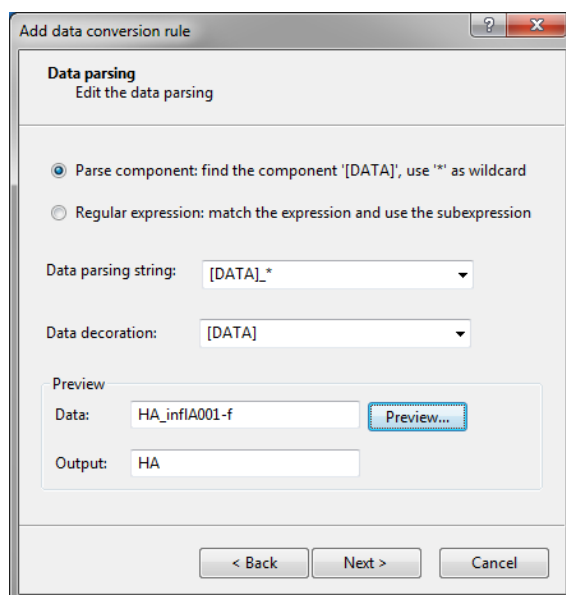


Figure 8.1.54: Data parsing string.

3.36 Make sure the newly created template is selected and press **<Next>**.

3.37 Click **<Yes>** twice to confirm the creation of the new sequence types in the database.

3.38 Press **<Next>** to confirm the creation of 10 new entries (see Figure 8.1.58).

The *Processing* wizard page opens.

3.39 Continue with the steps described in 8.1.3.2.8.

Case 3: Only the sample identifiers are contained in a template file

We will assemble the partial HA sequences from the example dataset in batch.

3.40 Select **File > Import...** (📁, **Ctrl+I**) to call the *Import* dialog box.

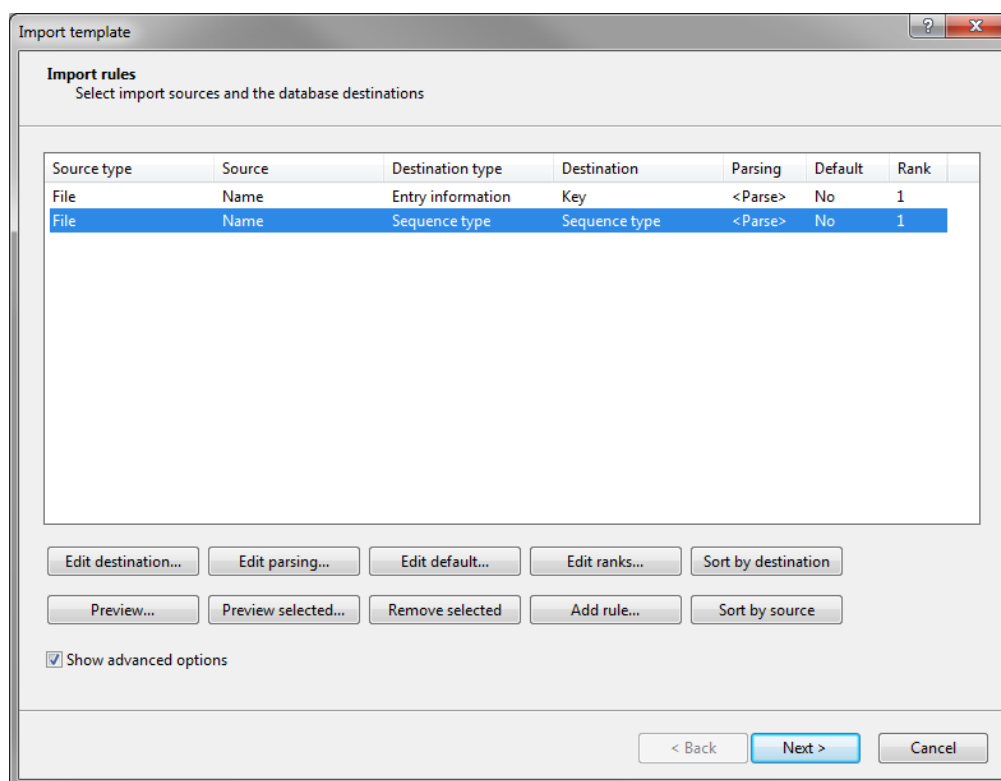


Figure 8.1.55: Import rules.

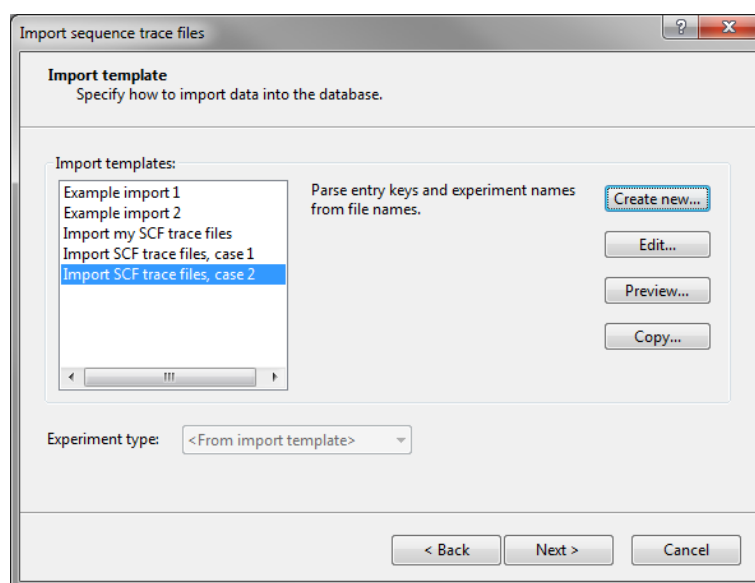
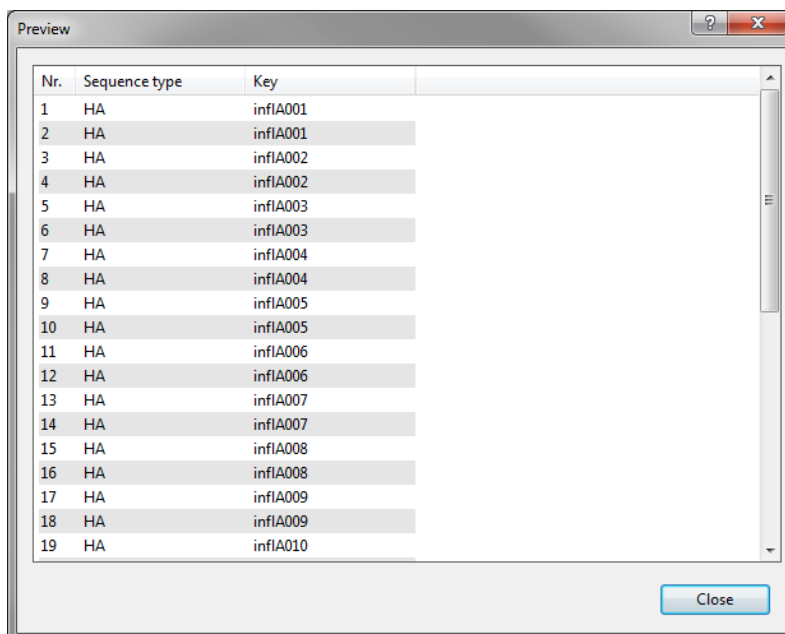


Figure 8.1.56: New import template.

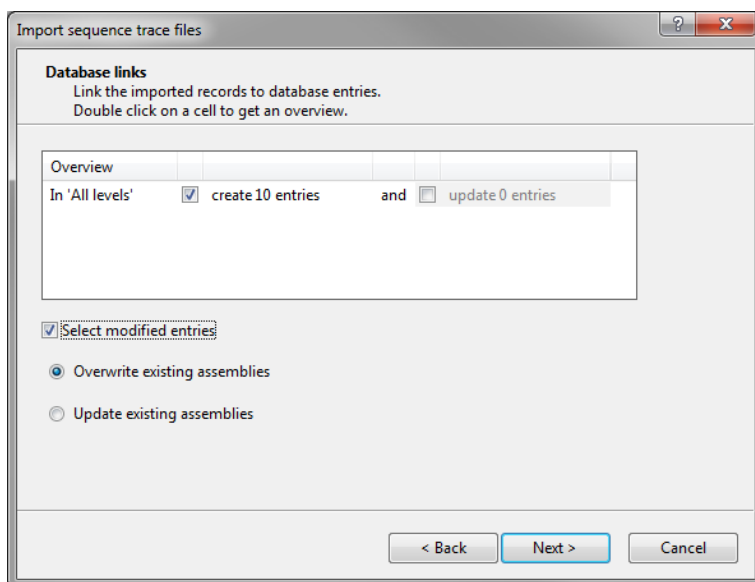
- 3.41 Select **Import and assemble trace files** under *Sequence type data* and press **<Import>** to start the batch import routine.
- 3.42 Browse for the correct folder, select all 20 HA sequence trace files and press **<Open>**.
- 3.43 Browse for the correct folder and select the `Template_HA_only.txt` template file.

The *Import sequence traces* wizard page is updated (see Figure 8.1.59).



Nr.	Sequence type	Key
1	HA	inflA001
2	HA	inflA001
3	HA	inflA002
4	HA	inflA002
5	HA	inflA003
6	HA	inflA003
7	HA	inflA004
8	HA	inflA004
9	HA	inflA005
10	HA	inflA005
11	HA	inflA006
12	HA	inflA006
13	HA	inflA007
14	HA	inflA007
15	HA	inflA008
16	HA	inflA008
17	HA	inflA009
18	HA	inflA009
19	HA	inflA010

Figure 8.1.57: Preview of the parsing.



Database links
Link the imported records to database entries.
Double click on a cell to get an overview.

Overview

In 'All levels' ☒ create 10 entries and ☐ update 0 entries

☒ Select modified entries

☒ Overwrite existing assemblies

☐ Update existing assemblies

< Back Next > Cancel

Figure 8.1.58: Database links.

3.44 Press **<Next>** to go to the next step.

The way the information should be imported in the database can be specified with an import template. In this tutorial, the file names and strain information are contained in the template file. A new import template needs to be defined:

3.45 Press the **Create new** button to call the *Import rules* dialog box.

Since a template file was selected, BioNumerics will use the file name information in the first column of the template file as link field. The text in the column "KEY" of our template file can be linked to any information field. In this tutorial we will link the information in the "KEY" column to the **Key** field in our

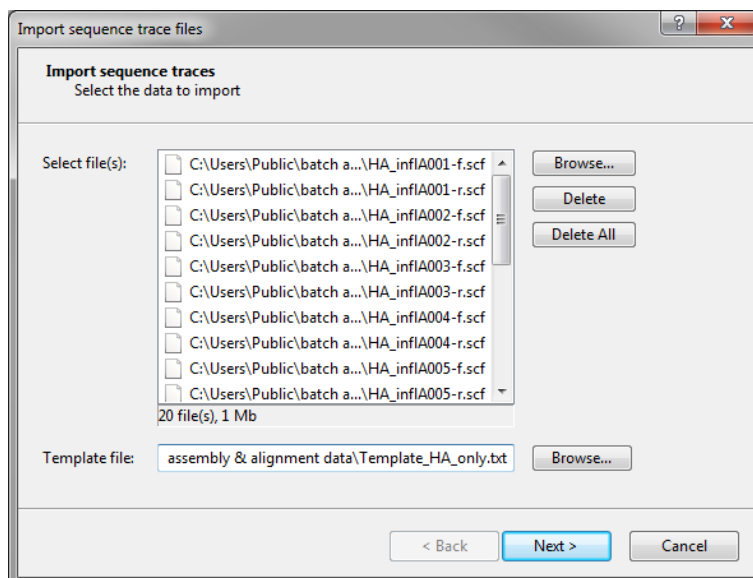


Figure 8.1.59: Selected trace and template files.

database:

- 3.46 Double-click on the **KEY** row or select the row and press **<Edit destination>**. Select **Key** as destination and press **<OK>** (see Figure 8.1.60).

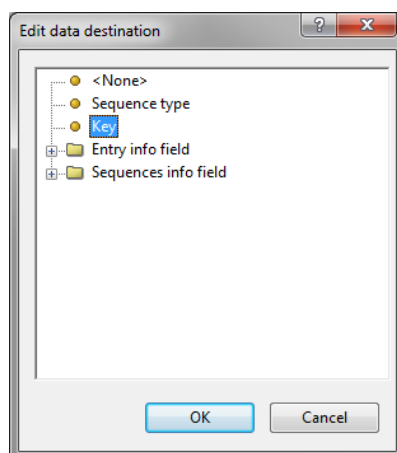


Figure 8.1.60: The *Edit data destination* dialog box.

The import rule in the *Import rules* dialog box is updated (see Figure 8.1.61).

- 3.47 Press the **<Preview>** button and press **<OK>** when the parsing is correct.
- 3.48 Press **<Next>** and **<Finish>**.
- 3.49 Specify a template name, e.g. **Import SCF trace files, case 3**, optionally enter a description (e.g. "Entry keys contained in template file") and press **<OK>**.

The template is added to the template list (see Figure 8.1.62).

- 3.50 Make sure the newly created template is selected and press the **<Preview>** button.
- 3.51 Close the preview.

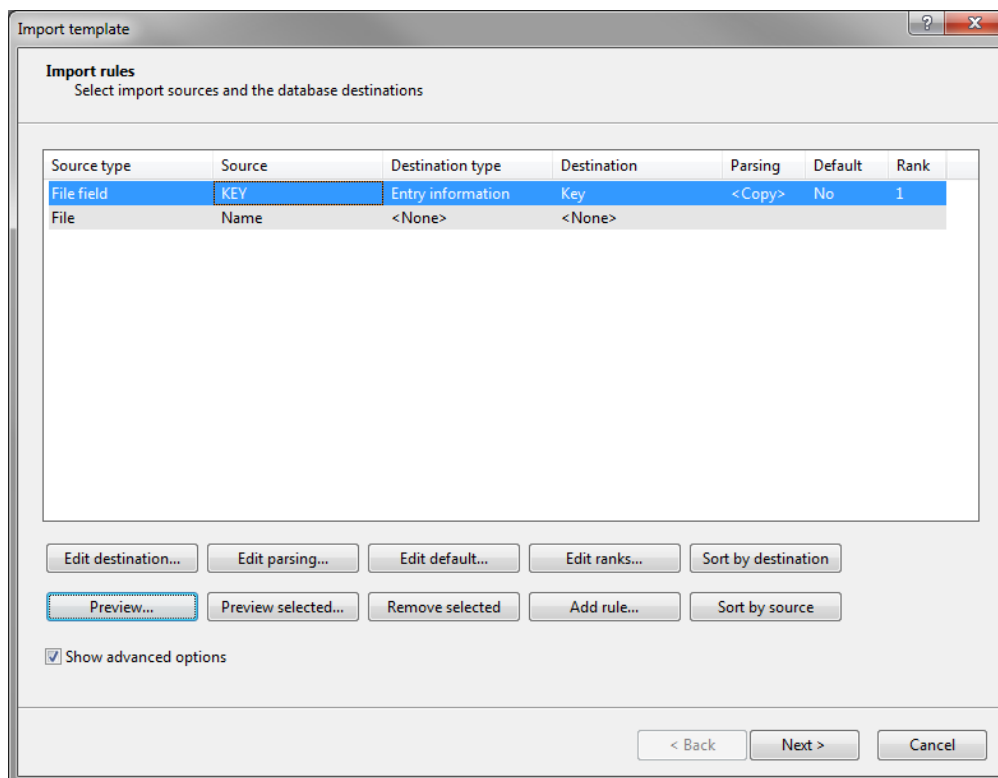


Figure 8.1.61: Import rules.

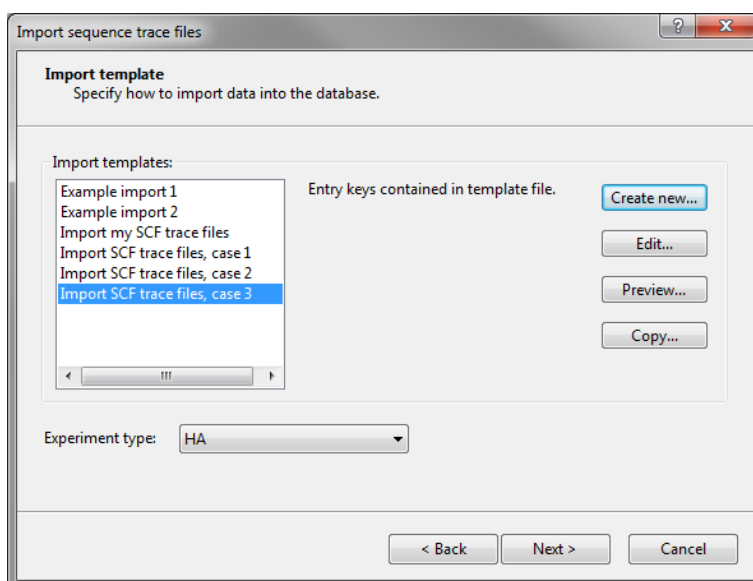


Figure 8.1.62: New import template.

3.52 Make sure the newly created template is selected, select *Create new* as *Experiment type* and press *<Next>*.

3.53 Specify an experiment name, e.g. "HA", and press *<OK>* and confirm the creation of the new sequence type in the database.

3.54 Press *<Next>* to confirm the creation of 10 new entries (see Figure 8.1.63).

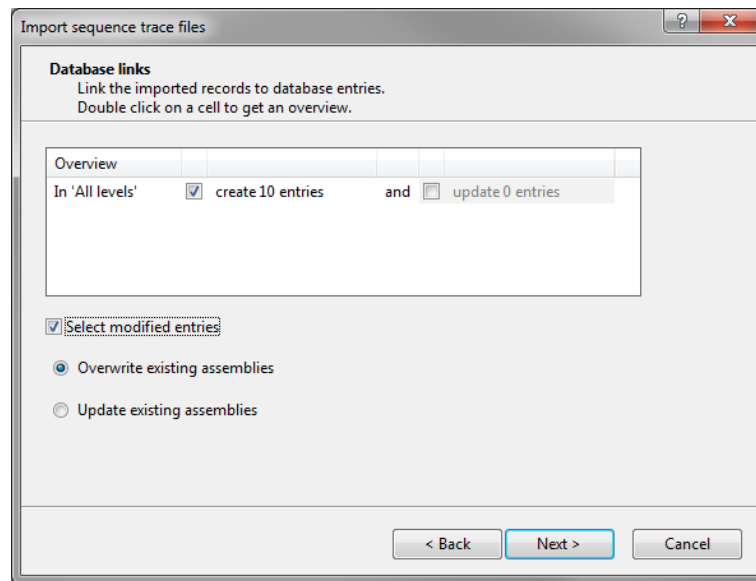


Figure 8.1.63: Database links.

The *Processing* wizard page opens.

3.55 Continue with the steps described in 8.1.3.2.8.

Case 4: The sample identifiers AND the experiments are contained in a template file

We will assemble the partial HA sequences and the partial NA sequences from the example dataset in batch.

3.56 Select **File > Import...** (📁, **Ctrl+I**) to call the *Import* dialog box.

3.57 Select **Import and assemble trace files** under **Sequence type data** and press **<Import>** to start the batch import routine.

3.58 Browse for the correct folder, select all HA and NA sequence trace files and press **<Open>**.

3.59 Browse for the correct folder and select the `Template.txt` template file.

The *Import sequence traces* wizard page is updated (see Figure 8.1.64).

3.60 Press **<Next>** to go to the next step.

The way the information should be imported in the database can be specified with an import template. In this tutorial, the file names, strain information and experiment names are contained in the template file. A new import template needs to be defined:

3.61 Press the **Create new** button to call the *Import rules* dialog box.

Since a template file was selected, BioNumerics will use the file name information in the first column of the template file as link field. The text in the columns "KEY" and "EXP" of our template file can be linked to any new or existing information field or sequence experiment type in our database. In this tutorial we will link the information in the "KEY" column to the **Key** field in our database, the information in the "EXP" column will be linked to sequence type experiments:

3.62 Double-click on the **KEY** row or select the row and press **<Edit destination>**. Select **Key** as destination and press **<OK>** (see Figure 8.1.65).

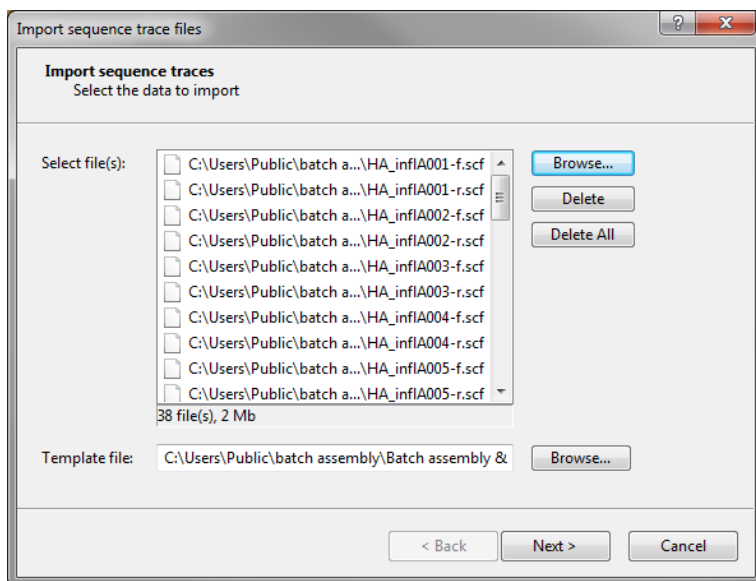


Figure 8.1.64: Selected trace and template files.

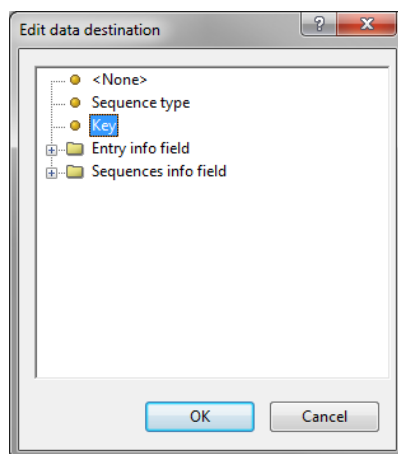


Figure 8.1.65: The *Edit data destination* dialog box.

The import rule in the *Import rules* dialog box is updated.

- 3.63 Double-click on the **EXP** row or select the row and press <Edit destination>. Select **Sequence type** as destination and press <OK> (see Figure 8.1.66).

The import rule in the *Import rules* dialog box is updated (see Figure 8.1.67).

- 3.64 Press the <Preview> button and press <Close> when the parsing is correct (see Figure 8.1.68).

- 3.65 Press <Next> and <Finish>.

- 3.66 Specify a template name, e.g. **Import SCF trace files, case 4**, optionally enter a description (e.g. "Entry keys and experiment names contained in template file") and press <OK>.

The template is added to the template list (see Figure 8.1.69).

- 3.67 Make sure the newly created template is selected and press the <Preview> button.

- 3.68 Close the preview.

- 3.69 Make sure the newly created template is selected and press <Next>.

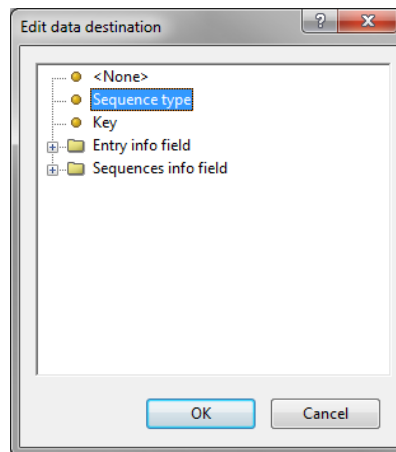


Figure 8.1.66: The *Edit data destination* dialog box.

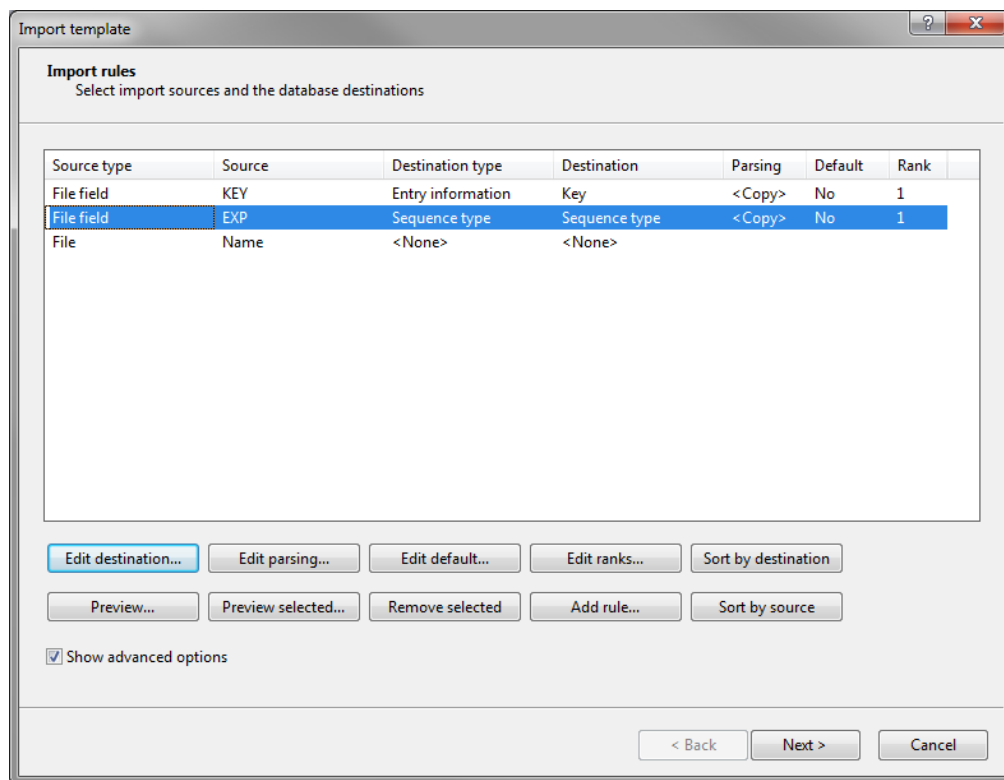


Figure 8.1.67: Import rules.

3.70 Click *<Yes>* twice to confirm the creation of the new sequence types in the database.

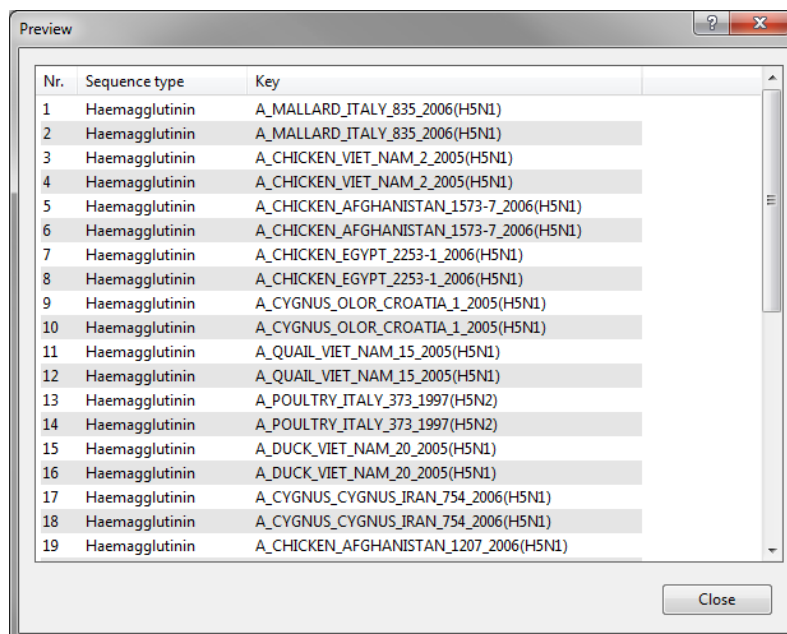
3.71 Press *<Next>* to confirm the creation of 10 new entries (see Figure 8.1.70).

The *Processing* wizard page opens.

3.72 Continue with the steps described in 8.1.3.2.8.

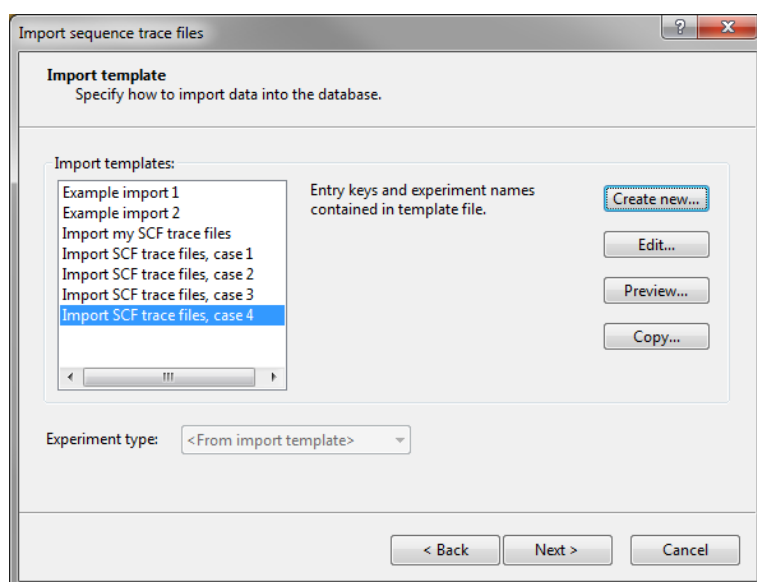
8.1.3.2.8 Processing step

Processing step



Nr.	Sequence type	Key
1	Haemagglutinin	A_MALLARD_ITALY_835_2006(H5N1)
2	Haemagglutinin	A_MALLARD_ITALY_835_2006(H5N1)
3	Haemagglutinin	A_CHICKEN_VIET_NAM_2_2005(H5N1)
4	Haemagglutinin	A_CHICKEN_VIET_NAM_2_2005(H5N1)
5	Haemagglutinin	A_CHICKEN_AFGHANISTAN_1573-7_2006(H5N1)
6	Haemagglutinin	A_CHICKEN_AFGHANISTAN_1573-7_2006(H5N1)
7	Haemagglutinin	A_CHICKEN_EGYPT_2253-1_2006(H5N1)
8	Haemagglutinin	A_CHICKEN_EGYPT_2253-1_2006(H5N1)
9	Haemagglutinin	A_CYGNUS_OLOR_CROATIA_1_2005(H5N1)
10	Haemagglutinin	A_CYGNUS_OLOR_CROATIA_1_2005(H5N1)
11	Haemagglutinin	A_QUAIL_VIET_NAM_15_2005(H5N1)
12	Haemagglutinin	A_QUAIL_VIET_NAM_15_2005(H5N1)
13	Haemagglutinin	A_POULTRY_ITALY_373_1997(H5N2)
14	Haemagglutinin	A_POULTRY_ITALY_373_1997(H5N2)
15	Haemagglutinin	A_DUCK_VIET_NAM_20_2005(H5N1)
16	Haemagglutinin	A_DUCK_VIET_NAM_20_2005(H5N1)
17	Haemagglutinin	A_CYGNUS_CYGNUS_IRAN_754_2006(H5N1)
18	Haemagglutinin	A_CYGNUS_CYGNUS_IRAN_754_2006(H5N1)
19	Haemagglutinin	A_CHICKEN_AFGHANISTAN_1207_2006(H5N1)

Figure 8.1.68: Preview.



Import sequence trace files

Import template
Specify how to import data into the database.

Import templates:

- Example import 1
- Example import 2
- Import my SCF trace files
- Import SCF trace files, case 1
- Import SCF trace files, case 2
- Import SCF trace files, case 3
- Import SCF trace files, case 4**

Entry keys and experiment names contained in template file.

Create new... Edit... Preview... Copy...

Experiment type: <From import template>

< Back Next > Cancel

Figure 8.1.69: New import template.

3.73 Press the **<Assembly settings>** button.

In case the sequences are linked to different sequence type experiments, the *Assembly settings* dialog box appears when pressing the **<Assembly settings>** button, displaying all sequence type experiments.

When all sequences are linked to the same sequence type experiment, the *Assembly settings* dialog box is called when pressing the **<Assembly settings>** button.

3.74 When the *Assembly settings* dialog box appears, double-click on an **<Edit>** button to call the *Assembly settings* dialog box (see Figure 8.1.72).

The name of the experiment type to which the assembly settings apply is displayed in the title of the dialog. The Assembly settings are grouped in tabs per settings dialog box in *Assembler: Quality* assignment,

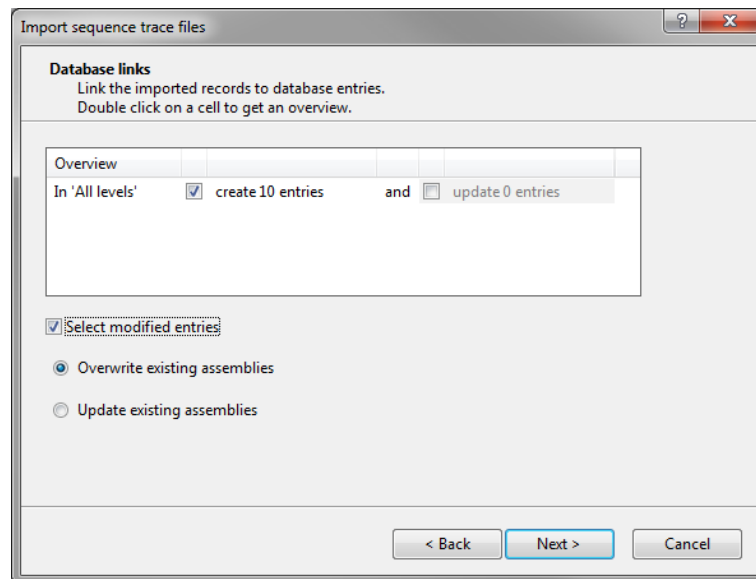
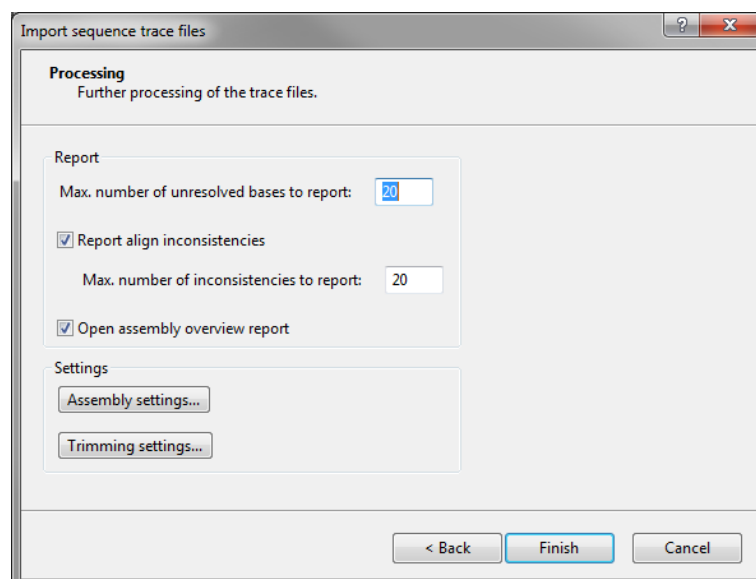


Figure 8.1.70: Database links.

Figure 8.1.71: The *Processing* wizard page.

Assembly and *Consensus* determination (see 8.1.3.7 for more information about these parameters).

3.75 Do not change to settings for any sequence type experiment in this tutorial and close the assembly dialogs.

3.76 Press the **<Trimming settings>** button.

In case the sequences are linked to different sequence type experiments, the *Assembly trimming settings* dialog box appears when pressing the **<Trimming settings>** button, displaying all sequence type experiments.

When all sequences are linked to the same sequence type experiment, the *Assembly trimming settings* dialog box is called when pressing the **<Trimming settings>** button.

3.77 When the *Assembly trimming settings* dialog box appears, double-click on the **<Edit>** button for the haemagglutinin (*HA*) experiment to call the *Assembly trimming settings* dialog box.

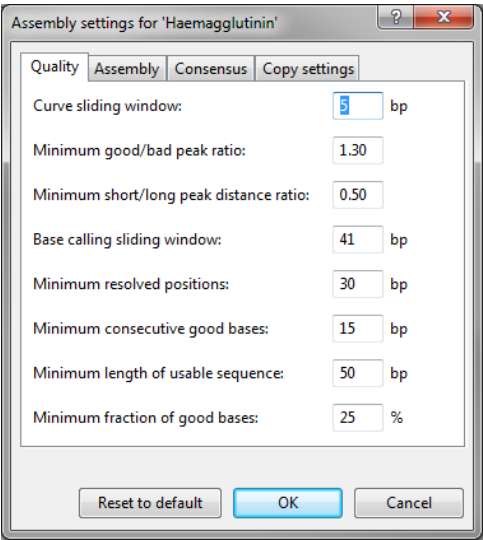


Figure 8.1.72: Assembly settings.

3.78 For the HA sequences in the example data set, enter the trimming settings as specified in Figure 8.1.73 and press <OK>.

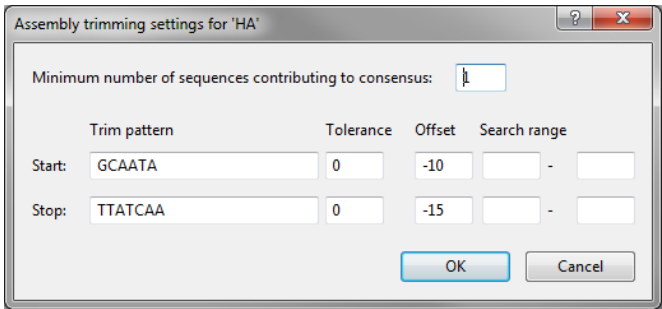


Figure 8.1.73: The *Assembly trimming settings* dialog box for haemagglutinin.

3.79 When the neuraminidase (NA) experiment is included in the import routine, double-click on the <Edit> button for the neuraminidase experiment to call the *Assembly trimming settings* dialog box.

3.80 For the NA sequences in the example dataset, enter following trimming settings: start trim pattern “CCAGTAG”, start position offset “0”, stop trim pattern “CGGGGTC” and stop position offset “-20” (see Figure 8.1.74). Press <OK>.

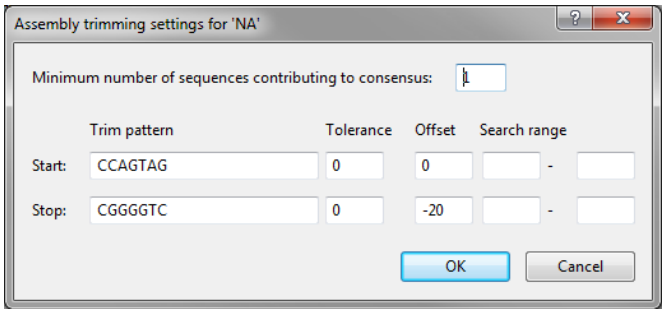


Figure 8.1.74: The *Assembly trimming settings* dialog box for neuraminidase.

3.81 Close the trimming dialogs.

- 3.82 Press <**Finish**> to assemble the selected trace files from the example data set into separate contig projects, representing HA and/or NA partial sequences of ten different influenza A strains.

When the assemblies are processed, an interactive report window appears (see Figure 8.1.75).

Key	HA	NA
✓ AM0001	OK	warning
✓ AM0002	error	OK
✓ AM0003	error	warning
✓ AM0004	error	warning
✓ AM0005	error	warning
✓ AM0006	error	warning
✓ AM0007	error	N/A
✓ AM0008	error	warning
✓ AM0009	error	warning
✓ AM0010	error	warning

Code	Message	Status	Comment
info	Created new assembly		
	Report for AM0001 / HA		

Figure 8.1.75: The *Batch sequence assembly report* window.

- 3.83 Click on a cell in the *Batch sequence assembly report* window to update the information in the *Details* panel.
- 3.84 Double-click on the cells with a warning or error message in the *Details* panel, solve the issues and update the status of the cells.

8.1.3.3 Importing sequence assemblies from BAM or SAM files

8.1.3.3.1 Introduction

With the **Import sequence assemblies from BAM files** option, listed under the topic *Sequence type data* in the *Import* dialog box (see Figure 8.1.76) a sequence assembly can be imported in BAM or SAM format.

A BAM file (file extension `.bam`) is the binary version of a SAM (Sequence Alignment/Map) file. A SAM file (file extension `.sam`) is a tab-delimited text file that contains large nucleotide sequence alignment data. Detailed format descriptions can be found on the SAM Tools web site: <http://samtools.sourceforge.net>.

Upon import of BAM or SAM files in BioNumerics, the reads in the assembly file needs to be sorted on position, if not already pre-calculated and available from the files, and the assembly file is indexed by genomic position to efficiently retrieve all reads aligning to a specific region. Both file manipulations use the SAM Tools.

Once the import is completed, the consensus sequence, defined by the default coverage and base calling settings, is saved to the database together with the coverage information and can be viewed from the *Sequence editor* window.

8.1.3.3.2 The Import wizard

Selecting **Import sequence assemblies from BAM files** under *Sequence type data* in the *Import* dialog box and pressing <**Import**> opens the *Input* wizard page (see Figure 8.1.77).

Pressing the <**Browse**> button allows you to select the BAM or SAM file(s) that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import

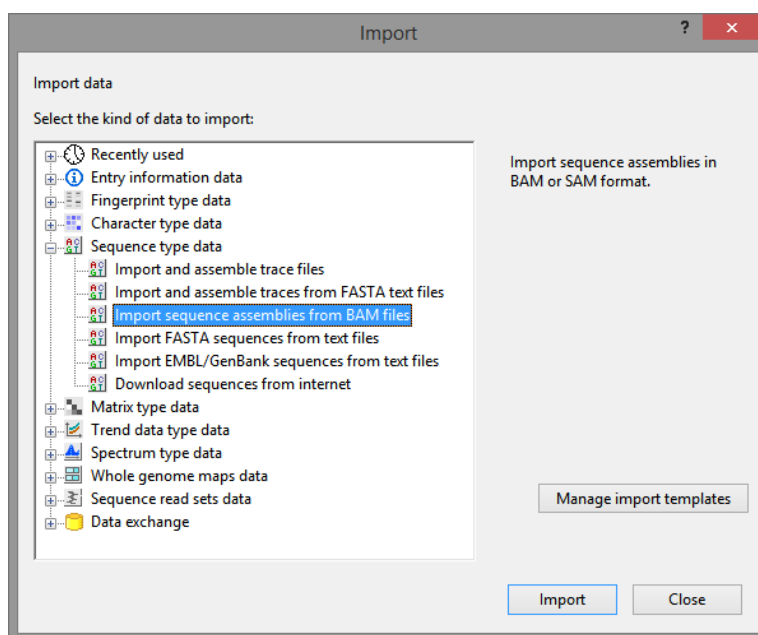


Figure 8.1.76: Import sequence assemblies from BAM files.

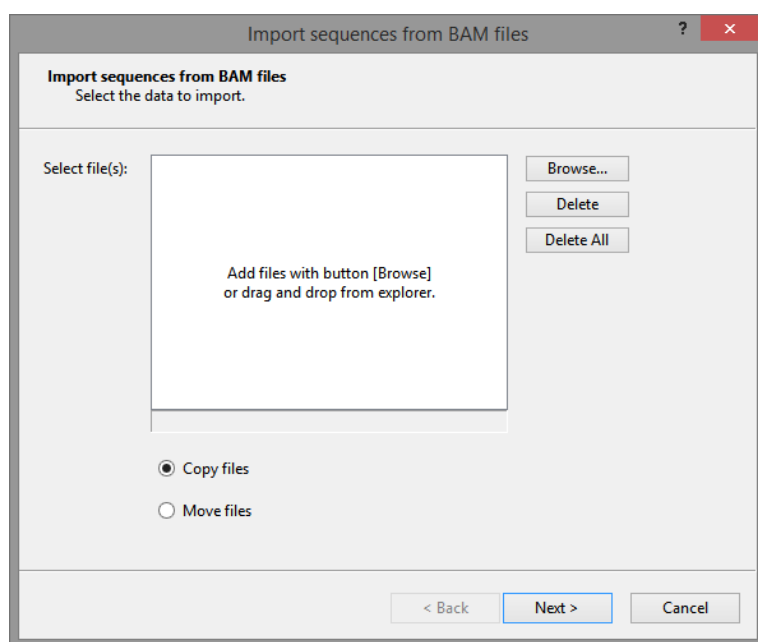


Figure 8.1.77: The *Input* wizard page.

list through drag and drop. The number of files and total size is displayed below the list.

With the **<Delete>** button all selected files are removed from the import list. All files are deleted at once from the import list when pressing **<Delete All>**.

Checking the option **Copy files** copies the imported file(s) to the source files location of the database without altering the original files, whereas the option **Move files** will remove the file(s) from their original file location and store them in the source files location of the database. Pressing **<Next>** displays the *Import template* wizard page.

The way the sequence information should be imported in the database can be specified with an import template. The *Import templates panel* lists all templates that have been created and stored in the database to

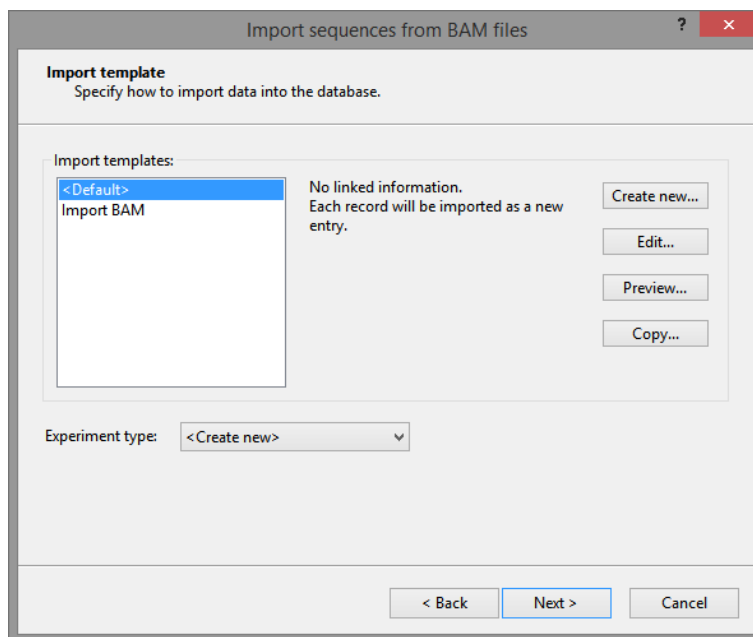


Figure 8.1.78: The *Import template* wizard page.

import BAM and/or SAM files.

The **Default** template will import the sequence assemblies in the database and links the sequences to new entries in the database (if the option **Create *x* entries** is checked in the final step). The keys are automatically created by the import routine.

Pressing the **<Create new>** button brings up the *Import rules* dialog box allowing you to define a new import template.

The only data source available for this type of sequence import is the file name. This information is present under the **Source type File** where the **Name** is specified in the **Source** column.

The row in the grid can be associated with new or existing entry information fields, sequence information fields, or sequence type names. Initially the row is not linked to any information (the **Destination type** and **Destination** is set to **<None>**). Specifying a *destination* can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row.

The information of this row can be linked to (see Figure 8.1.79):

- The default information field **Key**.
- A **Sequence type** name. The (parsed) information will hold the sequence type name.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select **<Create new>** or select an existing field under the topic **Sequence info field**, respectively).

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

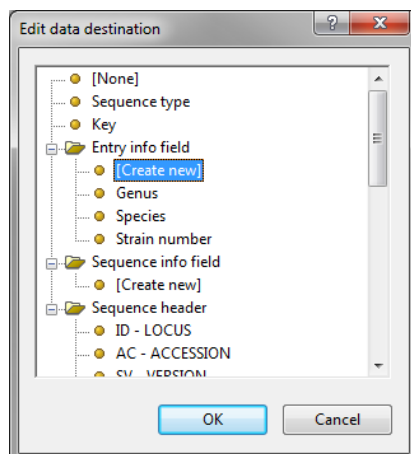


Figure 8.1.79: Edit data destination for a single selected row entry.

When the *<Show advanced options>* check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the *<Cancel>* button cancels the operation and the template settings are not saved to the database.

Pressing the *<Next>* button calls a new dialog where the entry link field needs to be defined.

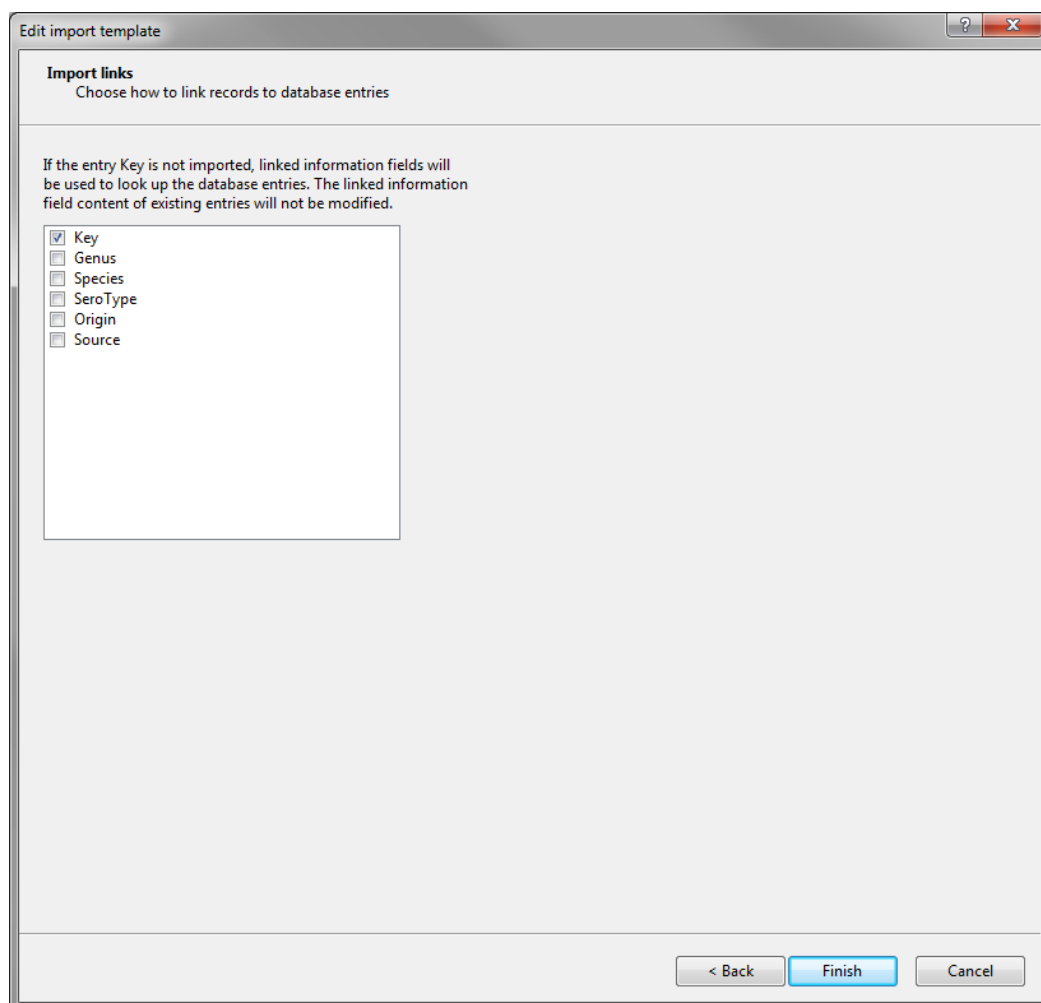


Figure 8.1.80: Specify the entry link field.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing <**Finish**> brings up the last step of the wizard.

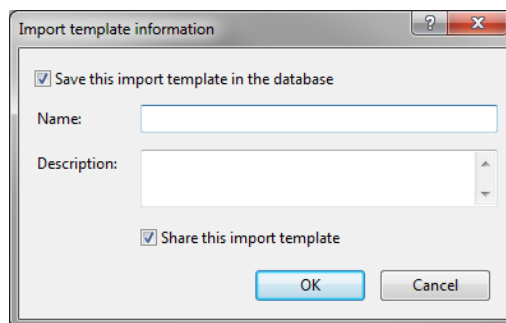


Figure 8.1.81: The *Import template information* dialog box.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the <**OK**> button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel (see Figure 8.1.82).

A highlighted import template can be copied and saved under a different name by pressing <**Copy**>.

Pressing the <**Edit**> button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag <**Absent**> is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the <**Preview**> button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the <**Close**> button.

If no row entry in the grid is linked to the **Sequence type name** destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (**Create New**). When sequences

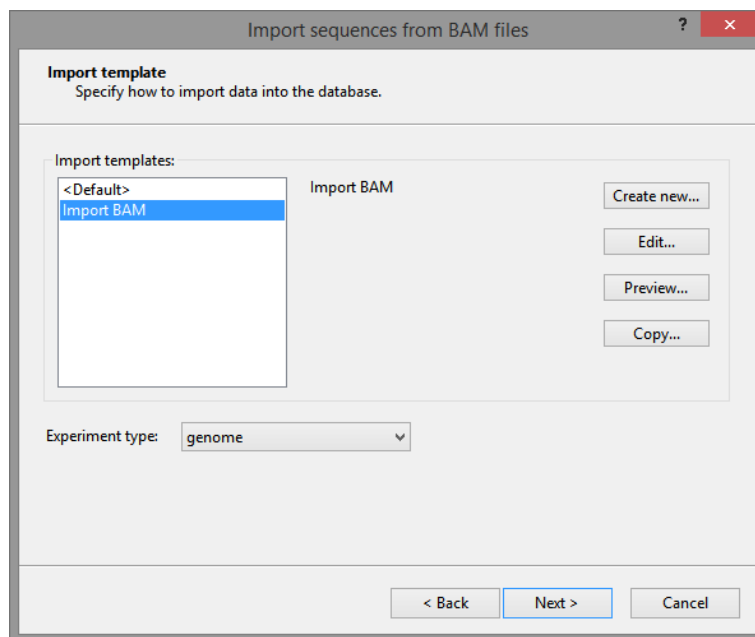


Figure 8.1.82: Import template added to the list.

are linked to a new sequence type experiment, a dialog box pops up when pressing the *<Next>* button, prompting for the sequence type name.

If a row in the grid is linked to the *Sequence type name* destination, the text *From import template* is automatically selected in the *Experiment type* text box. The import tool will link the sequences to the corresponding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the *<Next>* button, prompting for the names.

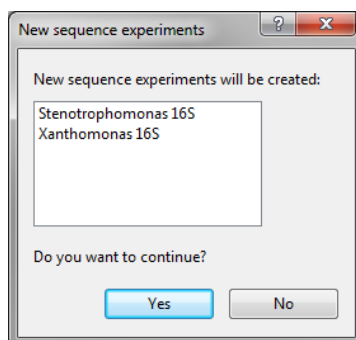


Figure 8.1.83: The *New experiment types* dialog box.

This dialog asks you to confirm the creation of the sequence type(s).

The *Database links* wizard page prompts for some final settings.

- When *Create x entries* is checked, the import tool is allowed to create the new entries in the database.
- Check the option *Update x entries* if you want the software to be able to update the information for existing entries.
- If the option *Select modified entries* is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create x entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing <Next> will start the import.



Import of the BAM or SAM files might take some time as the file sorting and indexing needs to be calculated upon import.

8.1.3.3.3 Inspecting BAM sequence assemblies

Sequence assemblies imported from BAM or SAM files can be viewed in the *BAM viewer* window. This window can be launched from the *Entry* window by clicking on the flask button next to the name of the sequence type in the *Experiments* panel, or by directly clicking the sequence experiment dot for that entry. In the *Sequence editor* window that opens, selecting **File > Open assembler** (🖼️) will launch the *BAM viewer* window to access the imported BAM or SAM information.

The *BAM viewer* window is used for visualizing large amounts of sequence reads which are aligned against a reference genome sequence (see Figure 8.1.84). This window consists of three different panels:

- The *Assembly view* panel displays the sequence scale on top, together with the reference and the consensus sequence, and visualizes the aligned reads towards the reference sequence.
- The *Overview* panel visualizes the different tracks e.g. total overall coverage, forward coverage, reverse coverage, median insert size of paired-end reads ...
- The *Tracks* panel manages which tracks are visualized in the *Overview* panel.

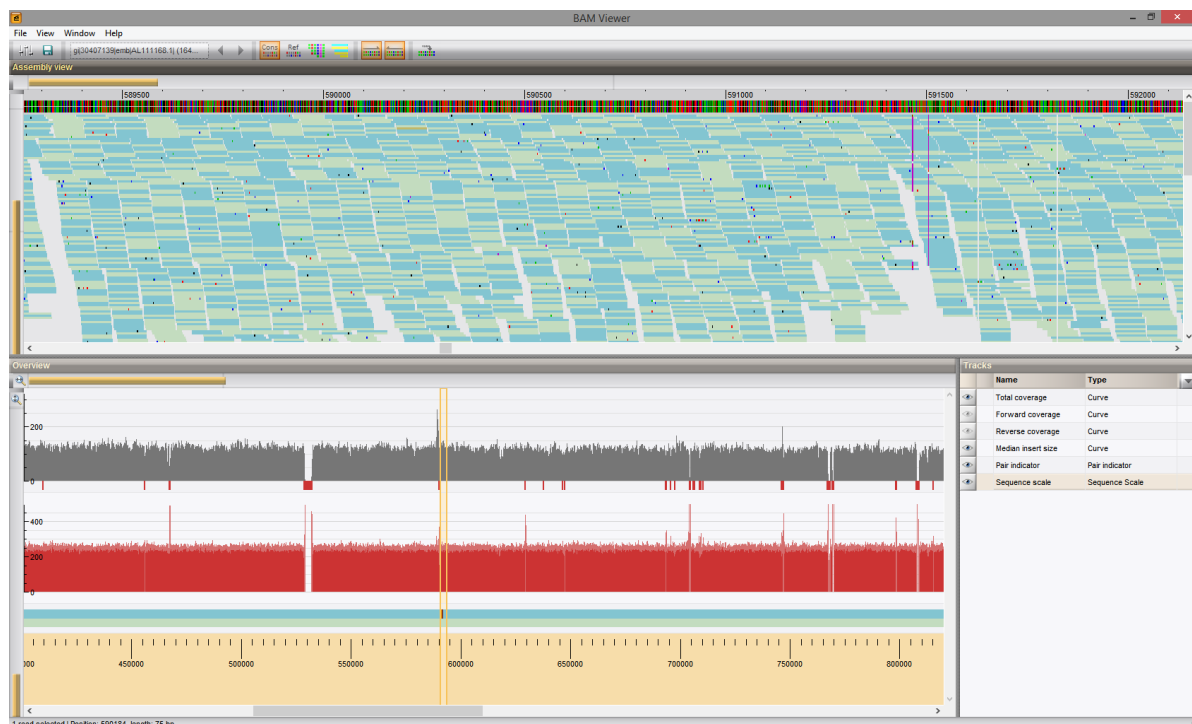




Figure 8.1.84: The *BAM viewer* window.

By default, the *Assembly view* panel displays the sequence scale based on assembly position, i.e. consensus positions including gaps, the reference sequence, the consensus sequence and the alignment data, forward



mapped reads indicated in blue and reverse mapped reads indicated in green. Read bases that are identical to the consensus sequence are displayed in blue or green, depending on the mapping direction, whereas read bases that are different from consensus base call have their specific base color, i.e. green for A, blue for C, black for G and red for T.



The sequence against which the deviant positions are shown, can be chosen from the consensus sequence and the reference sequence, as defined in the sequence experiment type. In this way, e.g. SNPs can be very easy graphically highlighted. To toggle the views between consensus- and reference-based comparison of the reads, select **View > Compare to consensus** () and **View > Compare to reference** (), respectively (see also Figure 8.1.85).






The reference sequence is only displayed in the *Assembly view* panel if it was defined in the corresponding sequence experiment type and if only one contig was imported from the BAM/SAM file.

The reads can be displayed at different levels of resolution. Zooming is done by using the yellow zoom sliders or scrolling the mouse wheel while holding the **Ctrl**-button. This way, it is possible to zoom into individual read base level.

Instead of displaying the mapping directions as colors with the deviant base calls, one can also display only the read outlines by selecting **View > Show only outlines** () or display all bases by using **View > Color all bases** () (see Figure 8.1.86).

By default, both forward and reverse mapped reads are shown. However, they can be selectively displayed using **View > Show forward** () or **View > Show reverse** (). This allows to search for e.g. strand-specific sequencing artifacts leading to strand-specific SNPs (see Figure 8.1.87).

To quickly access a specific position in the alignment, use **View > Go to...** (). This opens the *Go to assembly position* dialog box where the specific assembly position can be entered. After confirmation, the requested position is displayed in the center of the updated assembly view. Another way to visualize a specific part of the view is by holding the **Ctrl**-key and using the left mouse button to select a position in the assembly view and move it around within the *Assembly view* panel. By holding the **Shift**-button, one can drag a selection box around a specific region of interest in the assembly view and the selected region is automatically transferred to the overview panel. Similarly, **Shift**-clicking one of the reads selects a row in the assembly view, i.e. all reads that cover one of the base positions of the selected read.

If multiple contigs were defined in the BAM or SAM file, all contigs will be saved as a concatenated sequence to the same sequence experiment type. One can jump between the different imported contigs by selecting any of the contigs from the drop-down list in the toolbar or via **View > Show contig**. One can navigate between the loaded contigs using **View > Show next contig** () and **View > Show previous contig** ()

In the *Tracks* panel, the different curves displayed in the *Overview* panel are listed. Clicking the icon in front of each track, displays/hides the tracks from the overview. By default, the total coverage, the median insert size of the paired-end reads, a pair indicator and the sequence scale are displayed (see Figure 8.1.88).

Following tracks are available:

- The *Total coverage* contains the sum of forward and reverse read coverage information, as calculated over the consensus sequence. The red bars below the curve are visual indicators of the regions that do not satisfy the coverage criteria as defined in the consensus settings.
- The *Forward coverage* and *Reverse coverage* tracks contains similar information as the total coverage, but limited to forward and reverse mapped reads, respectively.
- The *Median insert size* visualizes the median absolute deviation of the insert sizes that cover the selected position. The median insert size is a robust measure of the variability in the insert size of all reads that cover a specific position. In contrast to the standard deviation, this measure is more resilient to outliers in insert sizes.

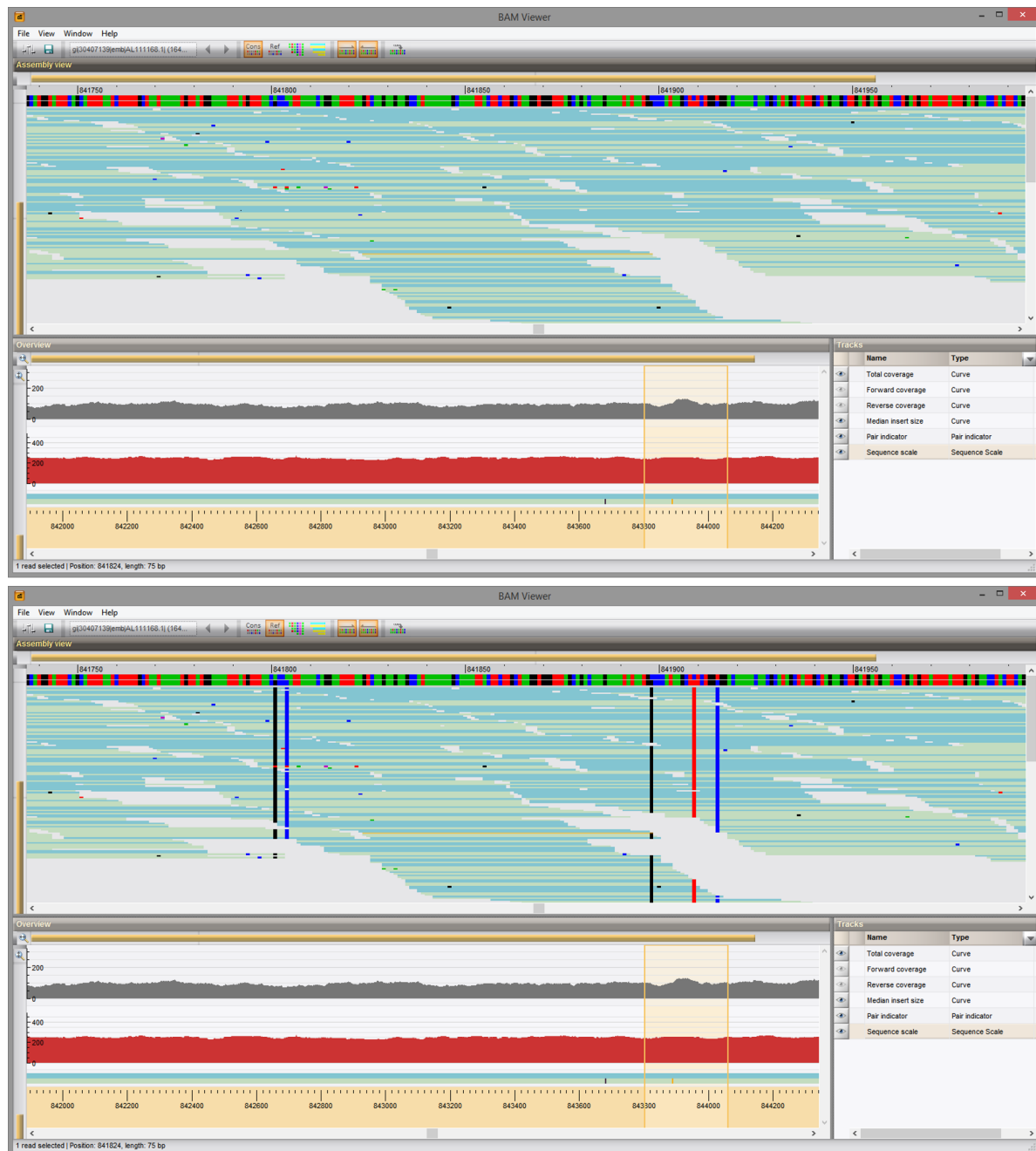


Figure 8.1.85: Looking for deviant positions in the *Assembly view* panel: consensus-based comparison of the reads at the top and reference-based comparison of the reads at the bottom.

- The *Pair indicator* marks the mapping position of the selected reads and that of its corresponding paired read. For the selected forward mapped reads, the mapping positions are displayed in the blue channel, whereas for the selected reverse mapped reads, the corresponding mapping positions of the pairs are displayed in the green channel. Each selected read is indicated in orange, whereas its paired read is indicated in red (forward) and blue (reverse).
- The *Sequence scale* contains the base pair indication as imported from the BAM/SAM file.

Zooming in the *Overview* panel is managed by the yellow sliders or hovering over the channels and scrolling with the mouse wheel.

A typical phenomenon at the beginning and end of each mapping region is the lack of coverage in the



Figure 8.1.86: Detail of *BAM viewer* window with all bases color-coded.

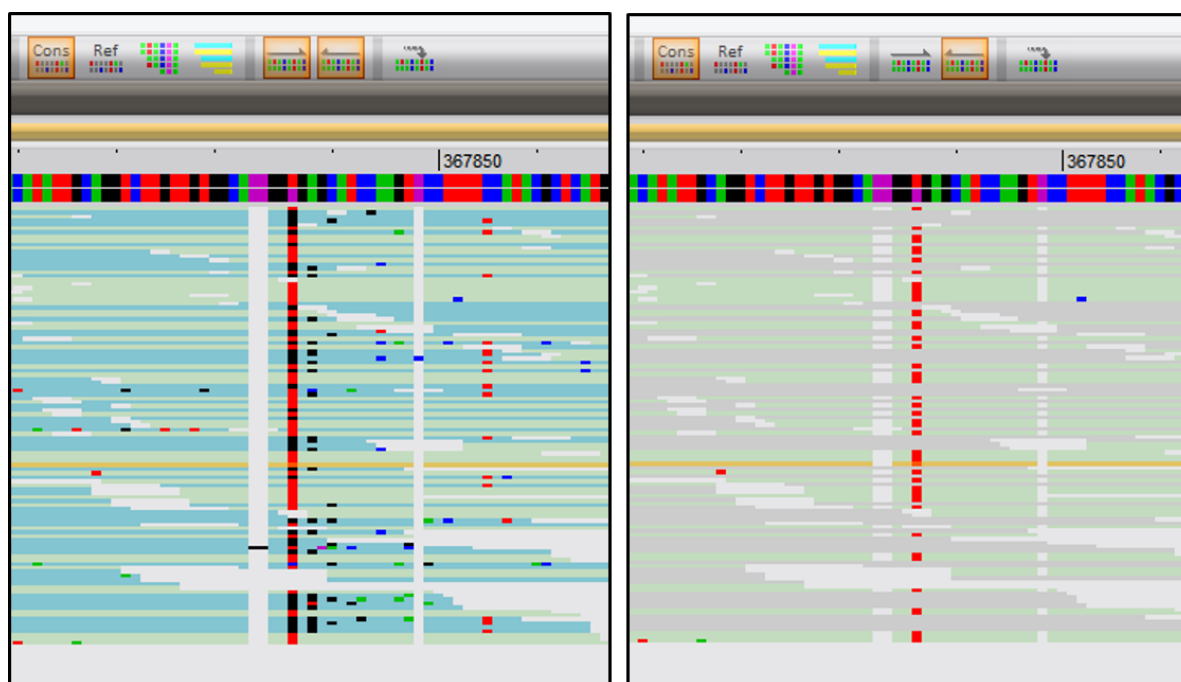


Figure 8.1.87: Looking at strand-specific sequencing artifacts by showing forward and/or reverse mapped reads.

read alignments. This typically results in read alignments as displayed in Figure 8.1.89. For this example, a minimal total coverage 10x, forward coverage 5x and reverse coverage 5x is required, resulting in a consensus sequence which only starts from position 9 compared to the reference sequence. The sequence that is saved to the database will start with the nucleotides AAGCCAAA.

When a mapping algorithm inserted base positions in individual reads, these inserts are indicated as + signs on the read (Figure 8.1.90).

Figure 8.1.91 illustrates a situation in which one of the reads has one additional base position, and as such,



Figure 8.1.88: The *BAM viewer* window: linking the *Assembly view* panel with the *Overview* panel.

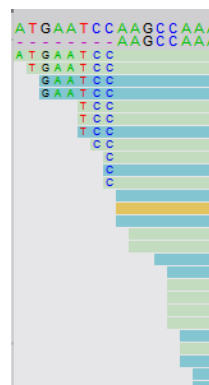


Figure 8.1.89: Deviant coverage at the beginning and end of a mapping region.

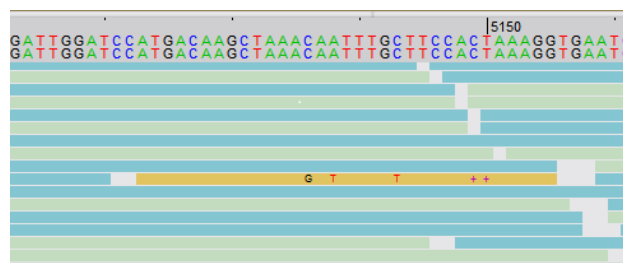


Figure 8.1.90: Illustration of inserted base positions in the reads.

a position is inserted in the reference (indicated by the + -sign in the reference sequence). However, as the gap thresholds are not met, this additional base position is not retained in the consensus sequence (indicated by the - -sign in the consensus sequence).

Figure 8.1.92 illustrates a similar situation, but in this case, the reads have one additional base position, and

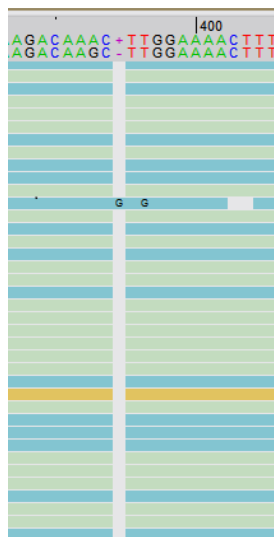


Figure 8.1.91: Illustration of an inserted base position which is not retained in the consensus sequence.

as such, a position is inserted in the reference (indicated by the +-sign in the reference sequence). In this case, all the reads indicate the same base and the single base threshold is met, resulting in an additional base position versus the reference that is retained in the consensus sequence and called a G. The second position of interest illustrates a position in the reference which is not covered by half of the reads, and as such, is not retained in the consensus sequence.

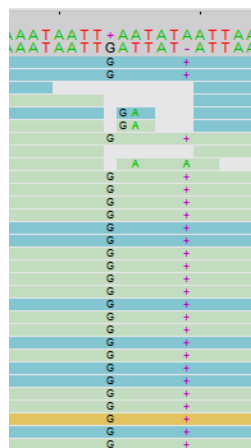


Figure 8.1.92: Illustration of an inserted base position which is retained in the consensus sequence and a deleted base position with respect to the reference.

8.1.3.3.4 Creating a consensus from a BAM sequence assembly

The final goal of the import of BAM and/or SAM files is to import a consensus sequence that was created based on an external sequence read assembly.

The consensus settings can be called by selecting **File > Consensus settings** (🔧). This opens the *Consensus settings* dialog box (see Figure 8.1.93).

In the *Consensus settings* dialog box, the minimum coverage settings and the base calling settings for consensus calling can be managed.

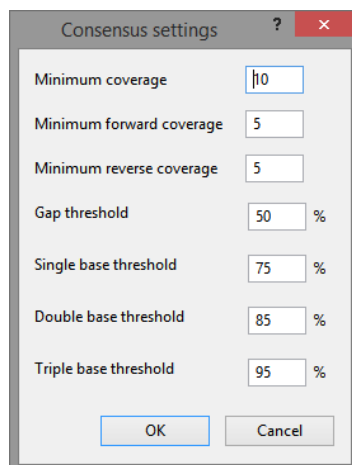


Figure 8.1.93: The *Consensus settings* dialog box.

- **Minimum coverage:** Minimum total coverage of a base to be considered for consensus base calling. If the coverage is too low, the base position is omitted from the consensus sequence.
- **Minimum forward coverage:** Minimum forward coverage of a base to be considered for consensus base calling. If the coverage is too low, the base position is omitted from the consensus sequence.
- **Minimum reverse coverage:** Minimum reverse coverage of a base to be considered for consensus base calling. If the coverage is too low, the base position is omitted from the consensus sequence.
- **Gap threshold:** Minimum frequency of a base position before that position is considered in the consensus sequence.
- **Single base threshold:** Minimum frequency of the most frequent base before this base is considered the unique base at a certain position in the consensus sequence.
- **Double base threshold:** Minimum summed frequency of the two most frequent bases before these bases are considered the two possible bases at a certain position in the consensus sequence and are denoted with IUPAC code for 2-fold degenerated positions (R: A/G; M: C/A; S: C/G, Y: C/T; W: A/T; K: G/T). Only applicable for positions that do not fulfill the criterion for single base calling.
- **Triple base threshold:** Minimum frequency of the three most frequent bases before these bases are considered the three possible bases at a certain position in the consensus sequence and are denoted with IUPAC code for 3-fold degenerated positions (V: A/C/G; H: A/C/T; D: A/G/T; B: C/G/T). Only applicable for positions that do not fulfill the criteria for single or double base calling. Any position that does not reach the required consensus for triple degeneracy is denoted as N.

When changing the consensus settings from the *Consensus settings* dialog box, the consensus present in the assembly view of the *BAM viewer* window is automatically updated.

To save the consensus sequence determined by the current consensus settings to the underlying sequence experiment, select **File > Export consensus** (📄). After creating the full consensus, the consensus sequence and its coverage information are exported to the database and a confirmation message is displayed.

The *BAM viewer* window is closed by selecting **File > Close**.

8.1.3.4 Importing FASTA sequences from text files

8.1.3.4.1 Introduction

When sequences are stored in FASTA format, each sequence begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than (“>”) symbol. The description line contains the *FASTA tags*, separated by a pipe (“|”) symbol.



Figure 8.1.94: FASTA formatted text files.

Using the **Import FASTA sequences from text files** option, listed under the topic *Sequence type data* in the *Import* dialog box (see Figure 8.1.95), sequences in FASTA format can be imported from text formatted files and linked to new or existing database entries. Optionally, FASTA tags can be stored in entry information fields.

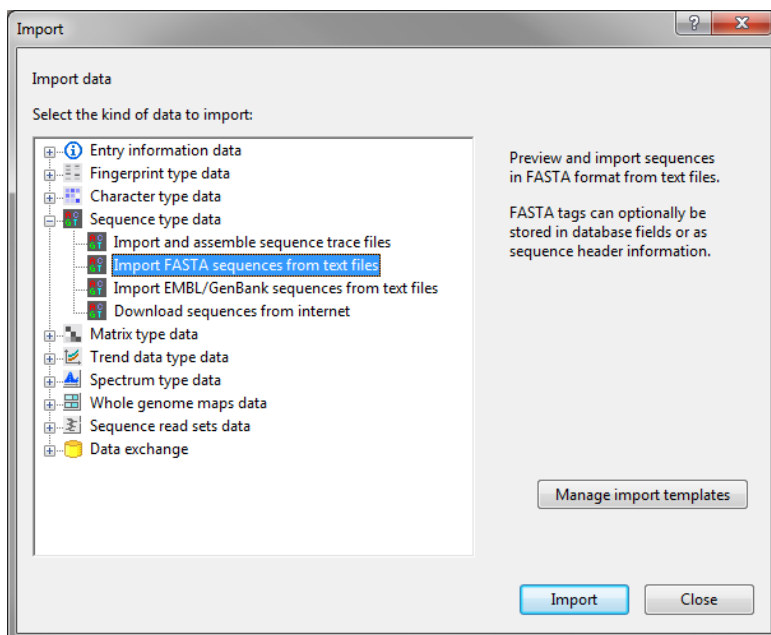


Figure 8.1.95: Import FASTA sequences from text files.

8.1.3.4.2 The Import wizard

Selecting **Import FASTA sequences from text files** under **Sequence type data** in the *Import* dialog box and pressing <**Import**> opens the *Input* wizard page (see Figure 8.1.96).

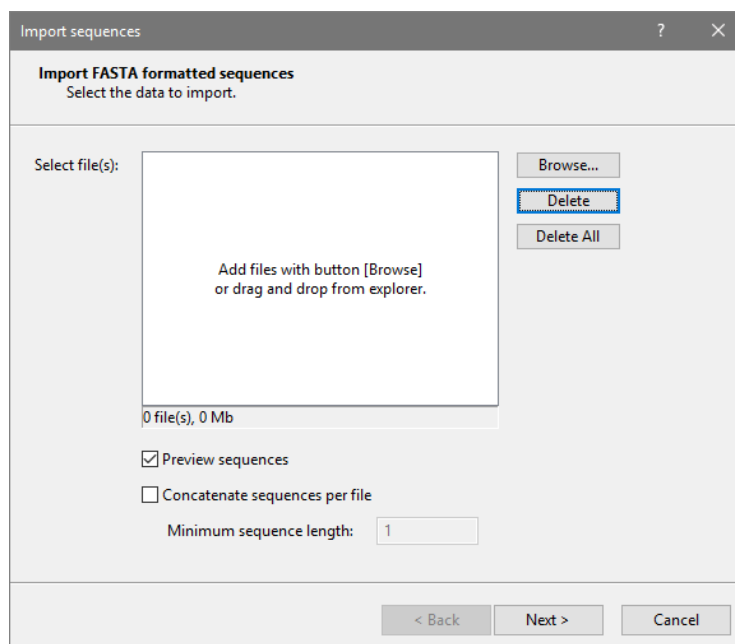


Figure 8.1.96: The *Input* wizard page.

Pressing the <**Browse**> button allows you to select the file(s) that you want to import, located on your computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. The number of files and total size is displayed below the list.

With the <**Delete**> button all selected files are removed from the import list. All files are deleted at once from the import list when pressing <**Delete All**>.

Checking the option **Preview sequences** displays all sequences in the next step of the wizard.

When checking the option **Concatenate sequences per file** all sequences found in a selected file are concatenated to a single sequence if their length exceeds the **Minimum sequence length**. The individual sequences are separated by a pipe (|) and will be concatenated in the same order as they appear in the file. The **Minimum sequence length** parameter can be used e.g. to exclude smaller contigs from a draft genome.

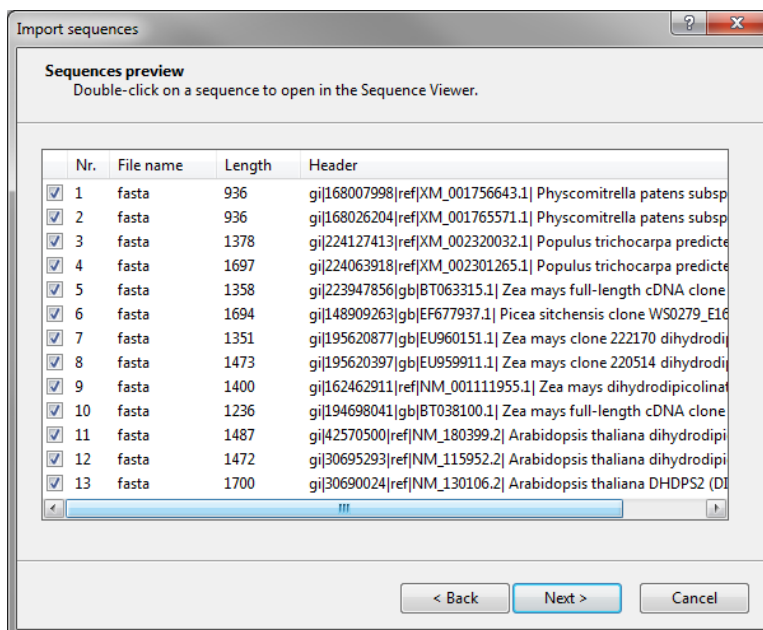
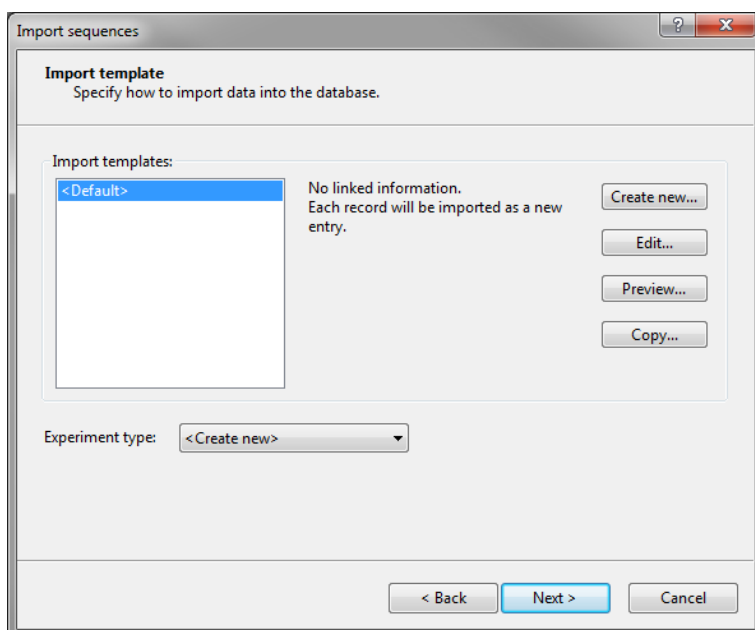
The **File name** column holds the name of the selected file, the **Length** column displays the size of the sequence, and the **Header** column holds the information that is present in the description line. All sequences are default checked and will be imported into the database. Unchecked sequences will not be imported in the database.

Double-clicking on a sequence in the preview list opens the *Sequence editor* window for the selected sequence. In the *Sequence editor* window the sequence can be saved to the database with **File > Save** (📁, **Ctrl+S**).

Pressing <**Next**> displays the *Import template* wizard page.

The way the sequence information should be imported in the database can be specified with an import template. The *Import templates panel* lists all FASTA templates that have been created and stored in the database.

The **Default** template will import the sequences in the database and link the sequences to new entries in the database (if the option **Create x entries** is checked in the final step). The keys are automatically created by

Figure 8.1.97: The *Preview* wizard page.Figure 8.1.98: The *Import template* wizard page.

the import routine.

Pressing the **<Create new>** button brings up the *Import rules* dialog box allowing you to define a new import template.

When sequences are stored in FASTA format, each sequence begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than ("**>**") symbol. The description line contains the *FASTA tags*, separated by a pipe ("**|**") symbol. Each *FASTA tag* corresponds to a row in the grid (maximum 20 FASTA tags can be parsed from the description line). The text **FASTA field** is specified in the **Source type** column and the position of the tags in the description line is displayed in the **Source** column (see Figure 8.1.99).

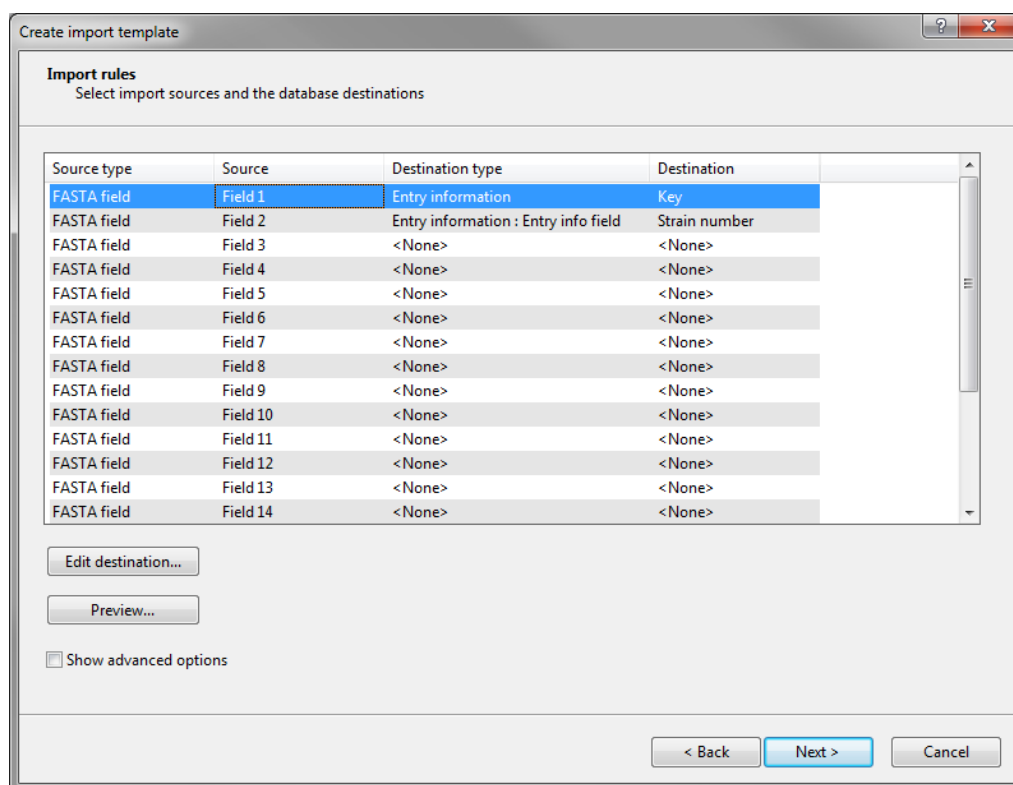


Figure 8.1.99: The *Import rules* dialog box.

Using the last row in the grid, the (parsed) file name of the selected file can be stored in the database. You might need to scroll down the list to view the last row entry. The text **File** is specified in the **Source type** column and the text **Name** is displayed in the **Source** column.

The rows in the grid can be associated with new or existing entry information fields, sequence information fields, sequence types, or header tags in the *Sequence editor* window. Initially the rows are not linked to any information (the **Destination type** and **Destination** for all rows is set to <None>). Specifying a *destination* for one or more selected rows can be done by pressing the <Edit destination> button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

When only one row is selected in the grid, the information of this row can be linked to (see Figure 8.1.100):

- The default information field **Key**.
- A **Sequence type** name. The (parsed) information will hold the sequence type name.
- A new or existing non-default entry information field (select the <Create new> option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select <Create new> or select an existing field under the topic **Sequence info field**, respectively).
- A **Sequence header** tag. The mapped information will be shown after import in the *Header tab* of the *Sequence viewer* next to the mapped header tag.

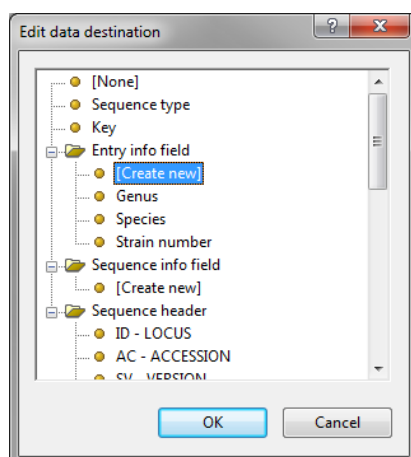


Figure 8.1.100: Edit data destination for a single selected row entry.

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button. This new dialog box prompts for the name.

When multiple rows are selected in the grid, the information of these rows can be linked to (see [Figure 8.1.101](#)):

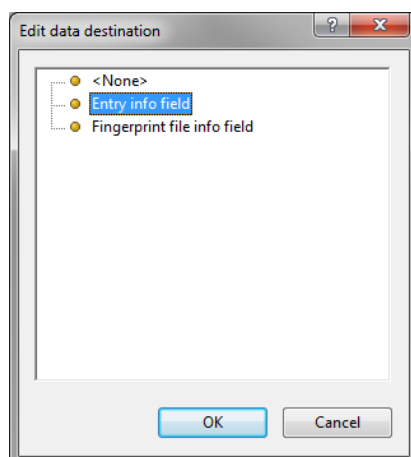


Figure 8.1.101: Edit data destination for multiple selected row entries.

- Non-default entry information fields (select the **Entry info field** option).
- Sequence information fields (select the **Sequence info field** option).

When pressing the **<OK>** button, the import routine checks if the selected rows can automatically be mapped to existing entry information fields or sequence information fields in the database. If no entry information fields or sequence information fields exist with the same name, a new dialog box pops up prompting for the names.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. When a row is linked to a sequence type name, the **Destination type** and **Destination** columns display the text **Sequence type**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to sequence information fields, the text **Sequence info field** is displayed in the **Destination type** column; the name of the field is listed in the **Destination** column. When one or more rows

are linked to a header tag, the name of the tag is shown in the *Destination* column and the *Destination type* column displays the text *Sequence header*.

Pressing <**Preview**> opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the <**Close**> button.

When the <**Show advanced options**> check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the <**Cancel**> button cancels the operation and the template settings are not saved to the database.

Pressing the <**Next**> button calls a new dialog where the entry link field needs to be defined.

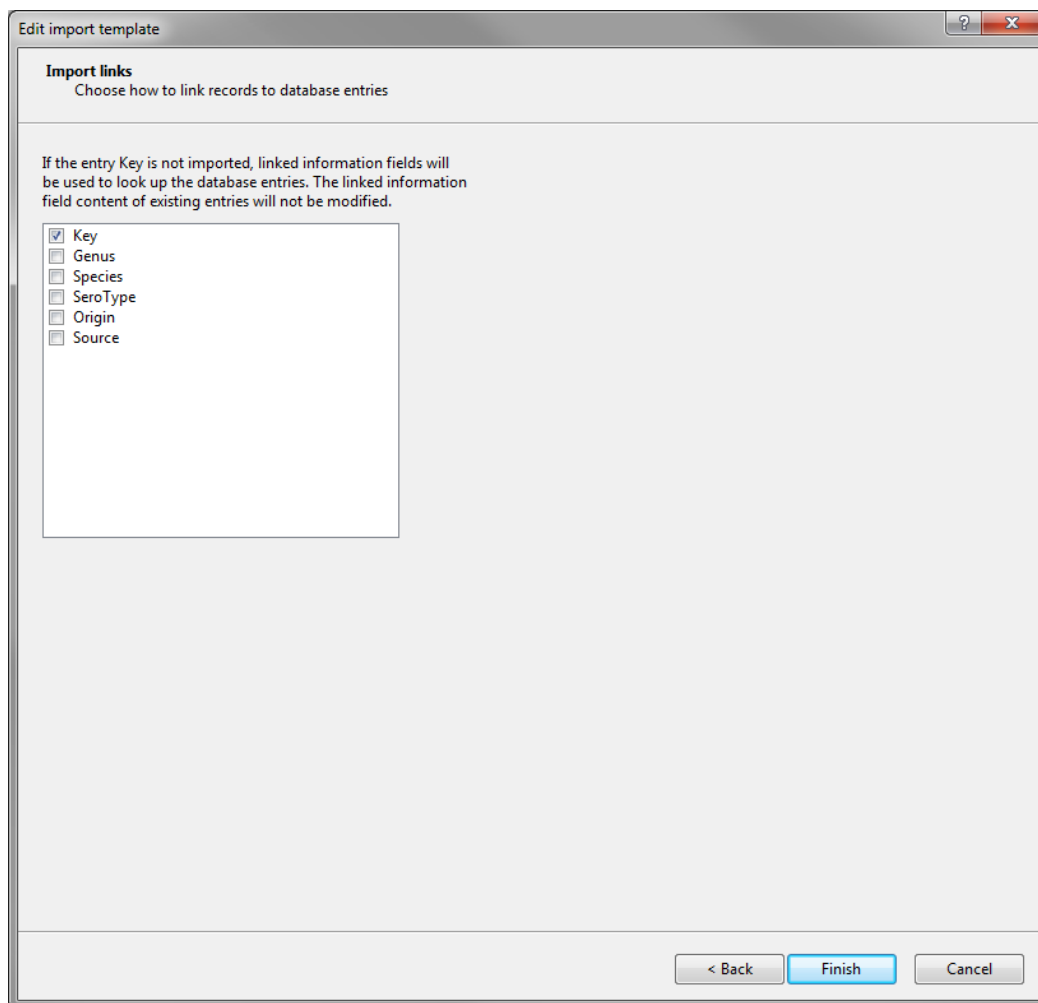


Figure 8.1.102: Specify the entry link field.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.

- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create *x* entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

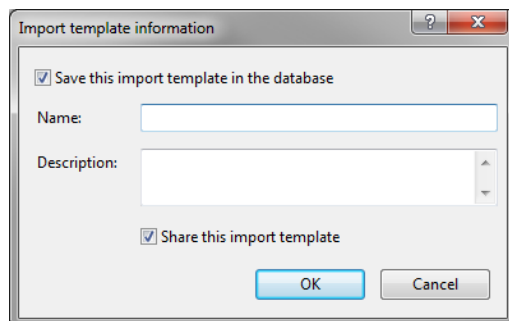


Figure 8.1.103: The *Import template information* dialog box.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel (see Figure 8.1.104).

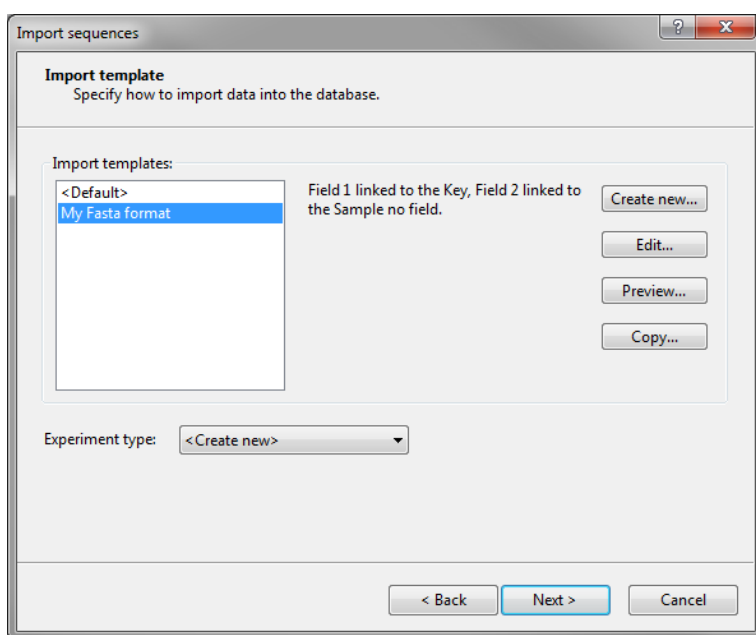


Figure 8.1.104: Import template added to the list.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

If no row entry in the grid is linked to the **Sequence type name** destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (**Create New**). When sequences are linked to a new sequence type experiment, a dialog box pops up when pressing the **<Next>** button, prompting for the sequence type name.

If a row in the grid is linked to the **Sequence type name** destination, the text **From import template** is automatically selected in the **Experiment type** text box. The import tool will link the sequences to the corresponding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the **<Next>** button, prompting for the names.

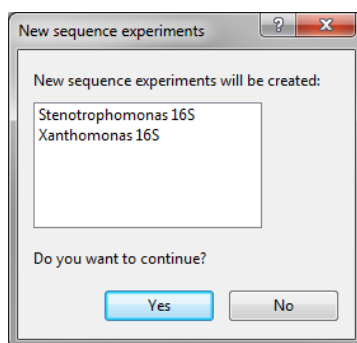


Figure 8.1.105: The *New experiment types* dialog box.

This dialog asks you to confirm the creation of the sequence type(s).

The *Database links* wizard page prompts for some final settings.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

8.1.3.5 Importing sequences from EMBL/GenBank files

8.1.3.5.1 Introduction

Using the **Import EMBL/GenBank sequences from text files** option, listed under the topic **Sequence type data** in the *Import* dialog box (see Figure 8.1.107), sequences in EMBL or GenBank format can be imported

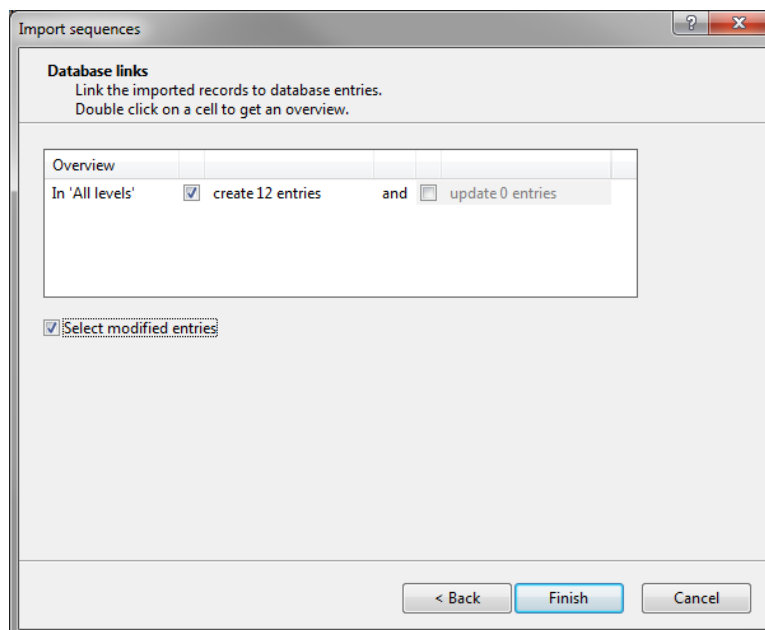


Figure 8.1.106: The *Database links* wizard page.

from text formatted files and linked to new or existing database entries. Header and feature descriptions are automatically stored with the sequences. Optionally, EMBL and GenBank tags can be stored in entry information fields.

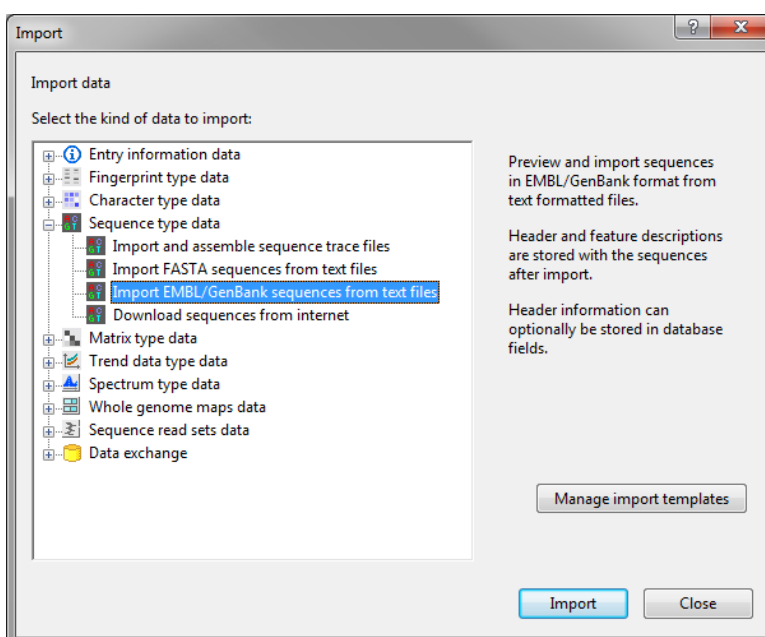


Figure 8.1.107: Import EMBL/GenBank sequences from text files.

8.1.3.5.2 The Import wizard

Selecting *Import EMBL/GenBank sequences from text files* under *Sequence type data* in the *Import* dialog box and pressing <Import> opens the *Input* wizard page (see Figure 8.1.108).

Pressing the <Browse> button allows you to select the file(s) that you want to import, located on your

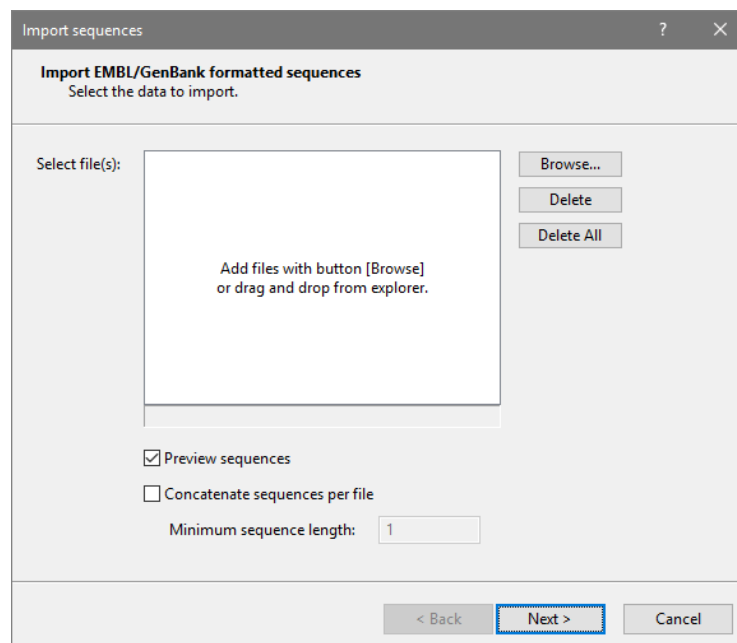


Figure 8.1.108: The *Input* wizard page.

computer, external drive or on a network location. Alternatively, files can be added to the import list through drag and drop. The number of files and total size is displayed below the list.

With the **<Delete>** button all selected files are removed from the import list. All files are deleted at once from the import list when pressing **<Delete All>**.

Checking the option **Preview sequences** displays all sequences in the next step of the wizard.

When checking the option **Concatenate sequences per file** all sequences found in a selected file are concatenated to a single sequence if their length exceeds the **Minimum sequence length**. The individual sequences are separated by a pipe (|) and will be concatenated in the same order as they appear in the file. The **Minimum sequence length** parameter can be used e.g. to exclude smaller contigs from a draft genome.

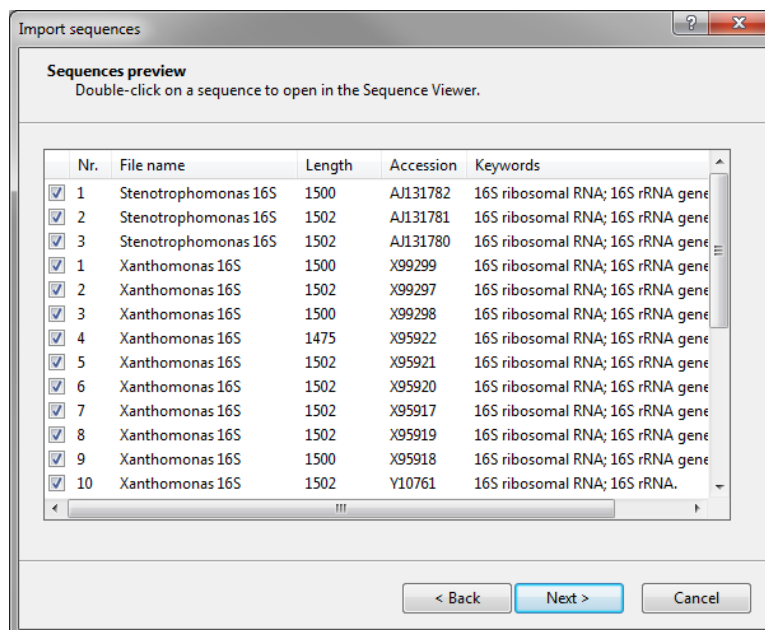
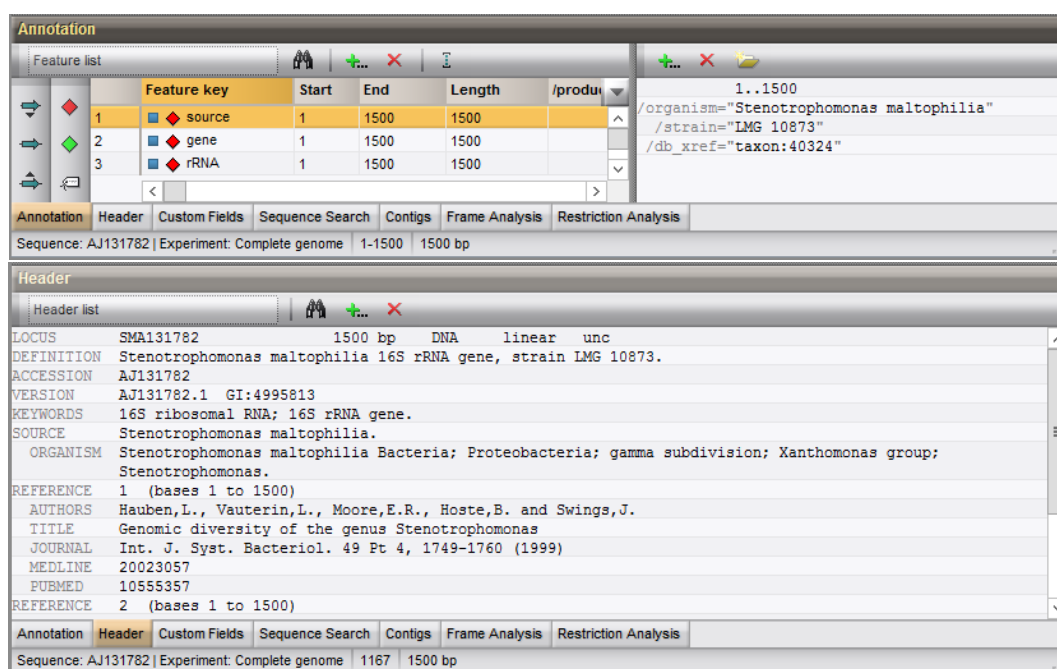
The **Length** column displays the size of the sequence, the **Accession** column displays the accession code, and the last column holds the information that is contained in the **Keywords** (NCBI) or **KW** (EBI) tag of the sequence. All sequences are default checked and will be imported into the database. Unchecked sequences will not be imported in the database.

Double-clicking on a sequence in the preview list opens the *Sequence editor* window for the selected sequence. The *features* (e.g. source, exon, CDS, ...) and *header tags* (e.g. ID, AC, KW, ...) of EMBL/GenBank formatted sequences are automatically recognized by BioNumerics and are displayed in the *Annotation* panel and *Header* panel of the *Sequence editor* window, respectively. The header information can also be stored in entry information fields. This needs to be specified in the import template (see further). In the *Sequence editor* window the sequence can be saved to the database with **File > Save** (📁, **Ctrl+S**).

Pressing **<Next>** displays the next step of the wizard.

The way the sequence information should be imported in the database can be specified with an import template. The *Import templates panel* lists all EMBL/GenBank templates that have been created and stored in the database.

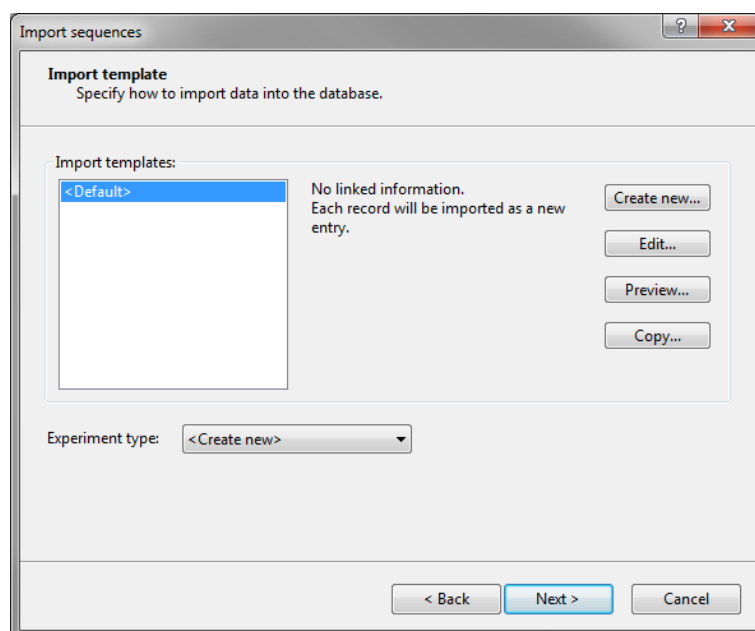
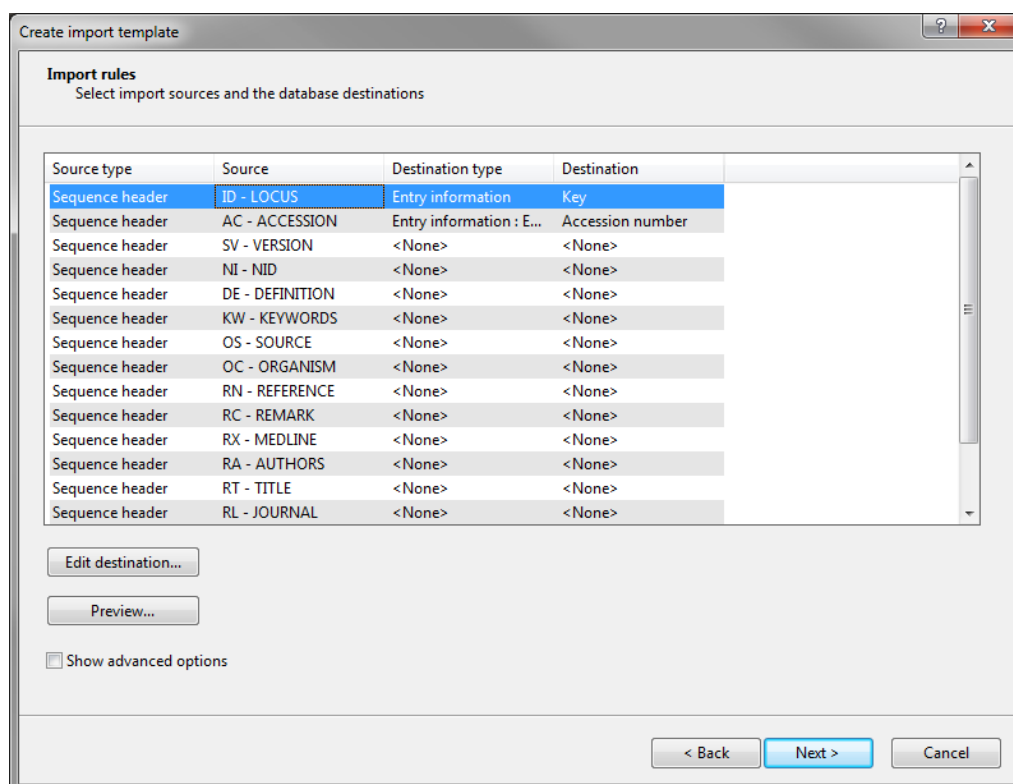
The **Default** template will import the sequences in the database and link the sequences to new entries in the database (if the option **Create x entries** is checked in the final step). The keys are automatically created by the import routine. If header and feature information is available for the sequences in the selected file(s), this information is stored in the *Sequence editor* window.

Figure 8.1.109: The *Preview* wizard page.Figure 8.1.110: The *Sequence editor* window: *Annotation* panel and *Header* panel.

Pressing the **<Create new>** button brings up the next step of the wizard allowing you to define a new import template.

Each EMBL/GenBank header tag corresponds to a row in the grid. The text *Sequence header* is specified in the *Source type* column and the name of the tag is displayed in the *Source* column. Using these rows, the header information can be stored in new or existing entry information fields.

Using the last row in the grid, the (parsed) file name of the selected file can be stored in the database. You might need to scroll down the list to view the last row entry. The text *File* is specified in the *Source type* column and the text *Name* is displayed in the *Source* column.

Figure 8.1.111: The *Import template* wizard page.Figure 8.1.112: The *Import rules* dialog box.

The rows in the grid can be associated with new or existing entry information fields, sequence information fields, or sequence type names. Initially the rows are not linked to any information (the **Destination type** and **Destination** for all rows is set to **<None>**). Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>** button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

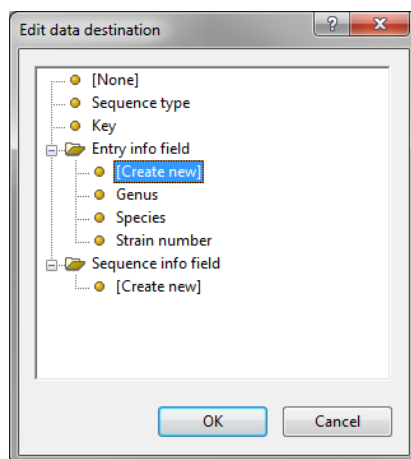


Figure 8.1.113: Edit data destination for a single selected row entry.

When only one row is selected in the grid, the information of this row can be linked to:

- The default information field **Key**.
- A **Sequence type** name. The (parsed) information will hold the sequence type name.
- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select **<Create new>** or select an existing field under the topic **Sequence info field**, respectively).

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button (see below for an example).

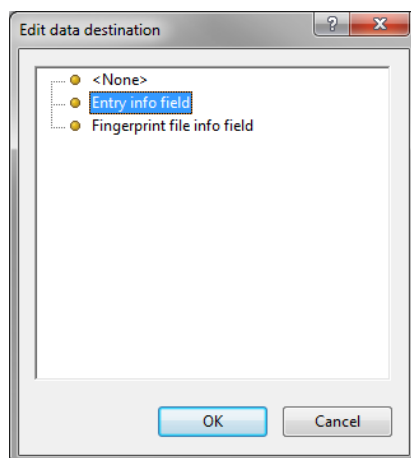


Figure 8.1.114: Edit data destination for multiple selected row entries.

When multiple rows are selected in the grid, the information of these rows can be linked to:

- Non-default entry information fields (select the *Entry info field* option).
- Sequence information fields (select the *Sequence info field* option).

When pressing the **<OK>** button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields or sequence information fields in the database. If no entry information fields or sequence information fields exist with the same name, a new dialog box pops up prompting for the names.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. When a row is linked to a sequence type name, the **Destination type** and **Destination** columns display the text **Sequence type**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to sequence information fields, the text **Sequence info field** is displayed in the **Destination type** column; the name of the field is listed in the **Destination** column.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in [3.3.5.5](#).

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.
- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template **Name** is shown in the *Import templates panel* and is automatically selected. The template **Description** is shown in the right panel.

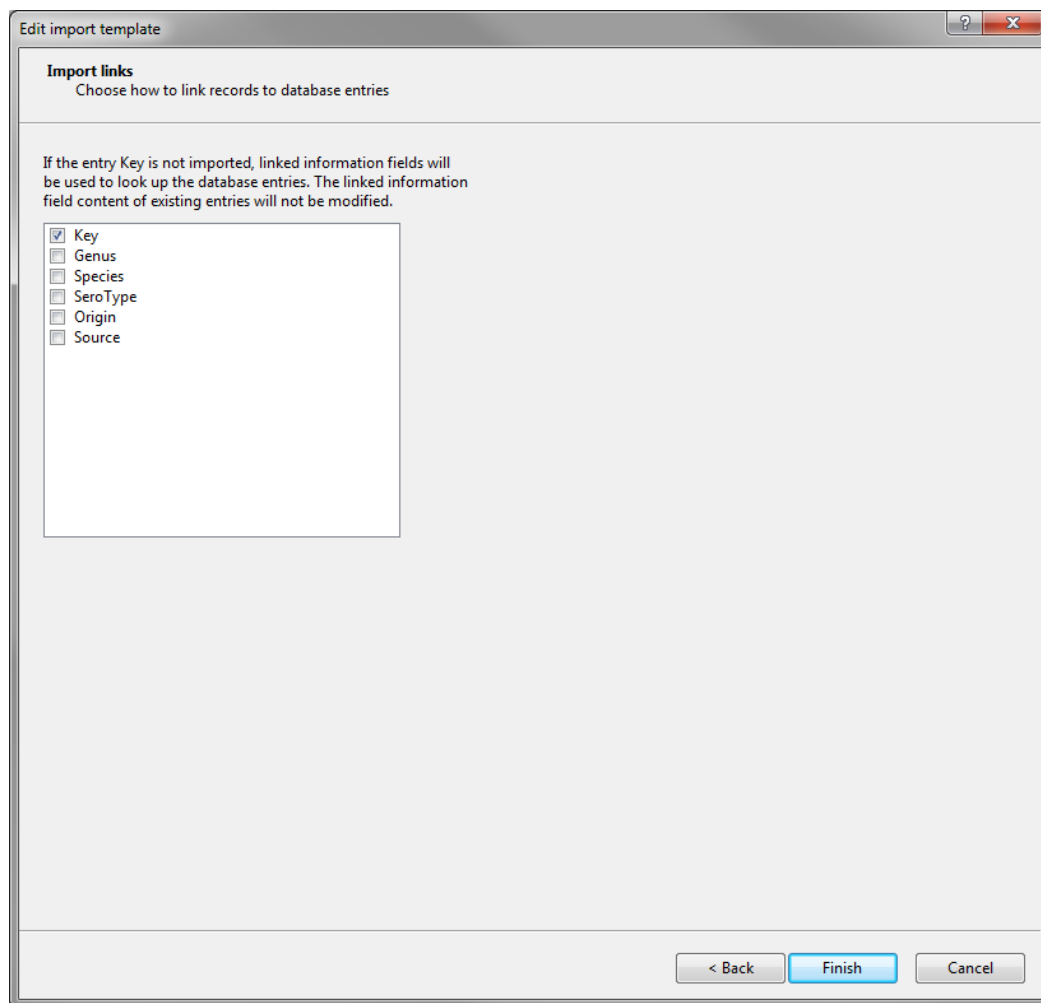


Figure 8.1.115: Specify the entry link field.

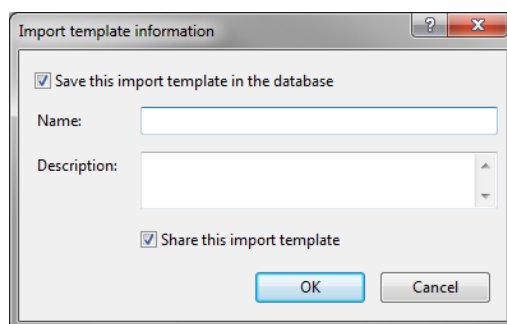


Figure 8.1.116: The *Import template information* dialog box.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the **Source** column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

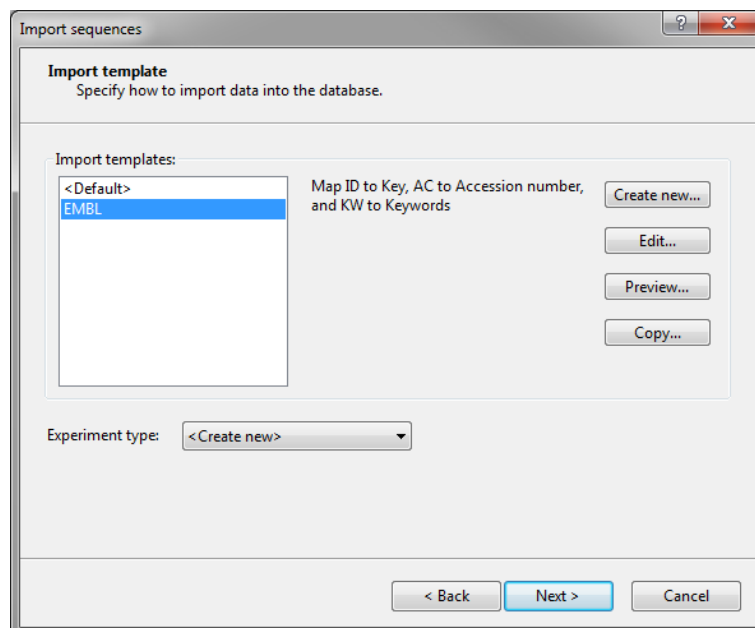


Figure 8.1.117: Import template added to the list.

If no row entry in the grid is linked to the *Sequence type name* destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (*Create New*). When sequences are linked to a new sequence type experiment, a dialog box pops up when pressing the *<Next>* button, prompting for the sequence type name.

If a row in the grid is linked to the *Sequence type name* destination, the text *From import template* is automatically selected in the *Experiment type* text box. The import tool will link the sequences to the corresponding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the *<Next>* button, prompting for the names.

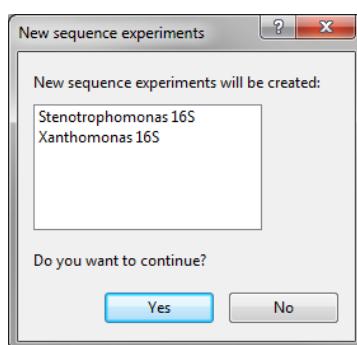


Figure 8.1.118: The *New experiment types* dialog box.

This dialog asks you to confirm the creation of the sequence type(s).

The *Database links* wizard page prompts for some final settings.

- When *Create x entries* is checked, the import tool is allowed to create the new entries in the database.
- Check the option *Update x entries* if you want the software to be able to update the information for existing entries.
- If the option *Select modified entries* is checked, entries in the database that were modified during the import routine will be selected after import.

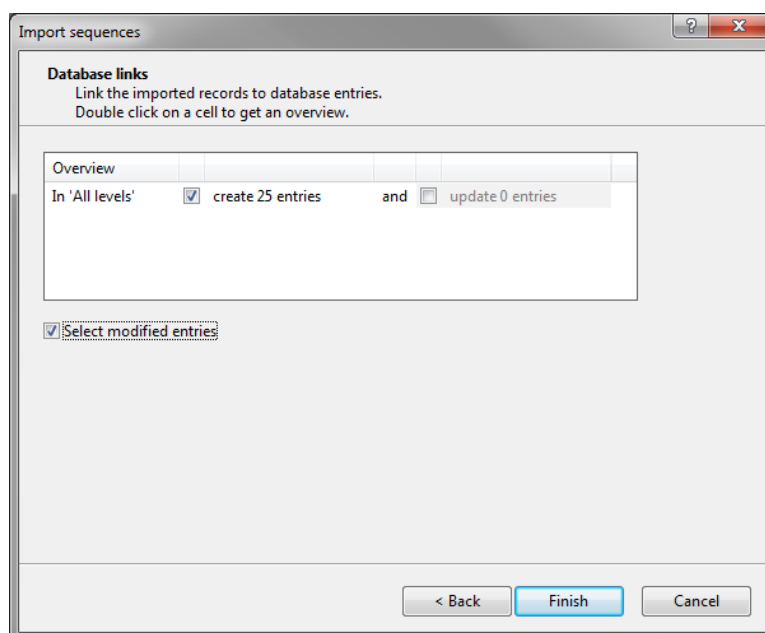


Figure 8.1.119: The *Database links* wizard page.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing <Next> will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.



Mapped sequence field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

8.1.3.6 Downloading sequences from internet

8.1.3.6.1 Introduction

With the *Download sequences from internet* option, listed under the topic *Sequence type data* in the *Import* dialog box (see Figure 8.1.120), it is possible to access the EBI, NCBI and NIG/DBJ databases directly from the BioNumerics software, which allows one to import single nucleotide sequences, or batches of nucleotide sequences in the BioNumerics database. During the import routine it is even possible to view the sequences in the *Sequence editor* window, without importing the data in the BioNumerics database.

8.1.3.6.2 The Import wizard

Selecting *Download sequences from internet* under *Sequence type data* in the *Import* dialog box and pressing <Import> opens the *Import sequences* wizard page (see Figure 8.1.121).

This dialog box prompts for the *Accession code(s)* and the *Preferred download site* which can be *EBI (EMBL-bank)*, *NCBI (GenBank - RefSeq)* or *NIG (DBJ)*.

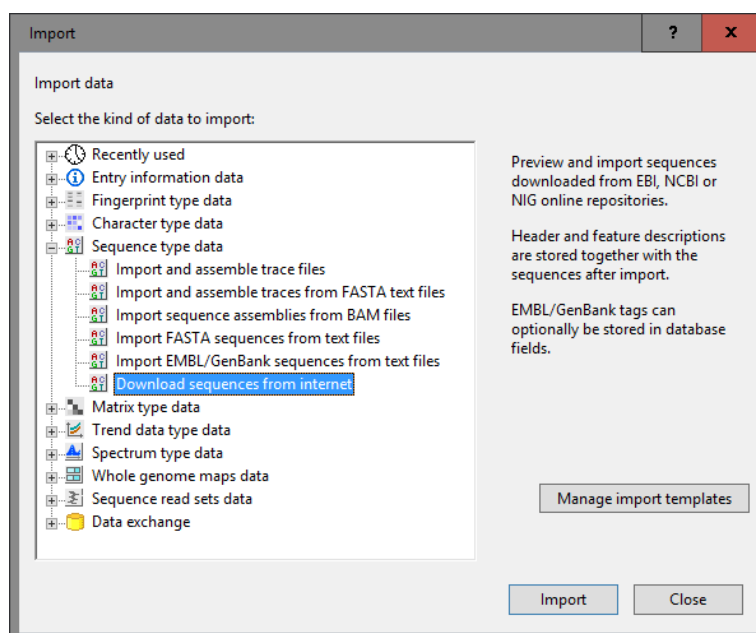


Figure 8.1.120: Download sequences from online repositories.

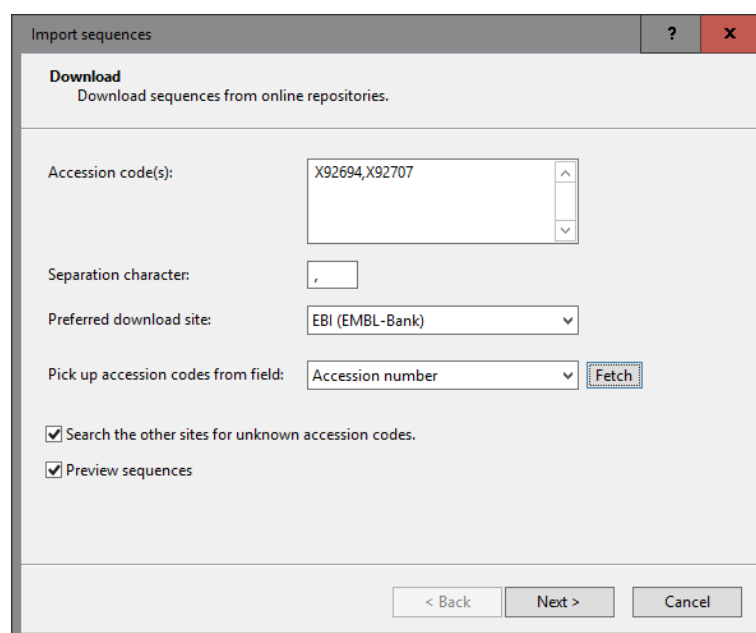


Figure 8.1.121: The *Import sequences* wizard page.

When fetching multiple sequences in the same import routine, the different accession codes need to be separated by the same separation character in the **Accession code(s)** input box. The character that separates the different codes in the upper input box needs to be specified in the **Separation character** input field.

With the **Pick up accession codes from field** option, accession codes stored in an entry information field in the database can be added to the **Accession code(s)** panel by selecting the entry field from the list and pressing the **<Fetch>** button. When no information is detected for the selected entries an error message is generated.

If **Search the other sites for unknown accession codes** is checked, the import routine will check the other online repositories for accession numbers that were not found at the preferred download site.

When pressing the **<Next>** button, the import routine tries to fetch the information from the online database(s). If the import routine is unable to find an accession code, a warning message is generated.

When the option **Preview sequences** was enabled in the previous step, information about the fetched sequences is displayed in the next step.

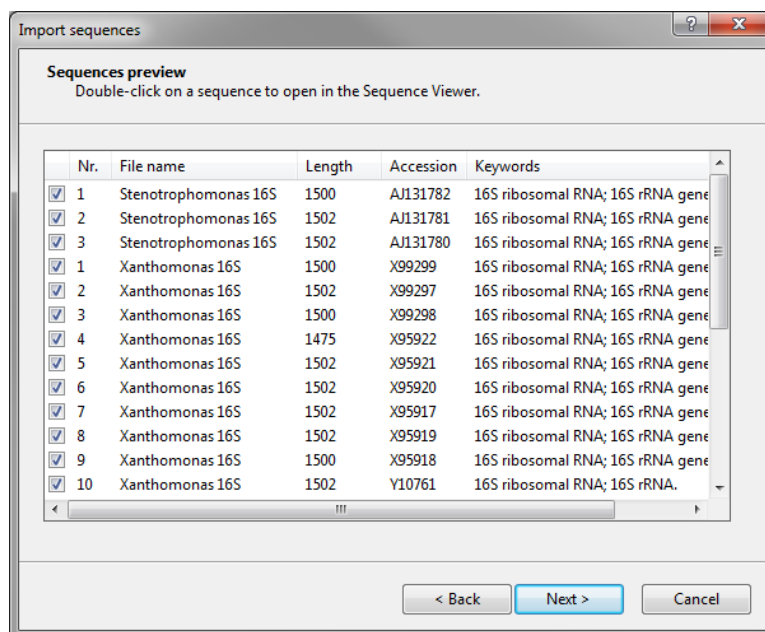


Figure 8.1.122: The *Preview* wizard page.

The **Length** column displays the size of the sequence, the **Accession** column displays the accession code, and the last column holds the information that is contained in the **Keywords** (NCBI) or **KW** (EBI) tag of the sequence. All sequences are default checked and will be imported into the database. Unchecked sequences will not be imported in the database.

Double-clicking on a sequence in the preview list opens the *Sequence editor* window for the selected sequence. The *features* (e.g. source, exon, CDS, ...) and *header tags* (e.g. ID, AC, KW, ...) of EMBL/GenBank formatted sequences are automatically recognized by BioNumerics and are displayed in the *Annotation* panel and *Header* panel of the *Sequence editor* window, respectively. The header information can also be stored in entry information fields. This needs to be specified in the import template (see further). In the *Sequence editor* window the sequence can be saved to the database with **File > Save** (📁, **Ctrl+S**).

Pressing **<Next>** displays the next step of the wizard.

The way the sequence information should be imported in the database can be specified with an import template. The *Import templates panel* lists all EMBL/GenBank templates that have been created and stored in the database.

The **Default** template will import the sequences in the database and link the sequences to new entries in the database (if the option **Create x entries** is checked in the final step). The keys are automatically created by the import routine. If header and feature information is available for the sequences in the selected file(s), this information is stored in the *Sequence editor* window.

Pressing the **<Create new>** button brings up the next step of the wizard allowing you to define a new import template.

Each EMBL/GenBank header tag corresponds to a row in the grid. The text **Sequence header** is specified in the **Source type** column and the name of the tag is displayed in the **Source** column. Using these rows, the header information can be stored in new or existing entry information fields.

Using the last row in the grid, the (parsed) file name of the selected file can be stored in the database. You

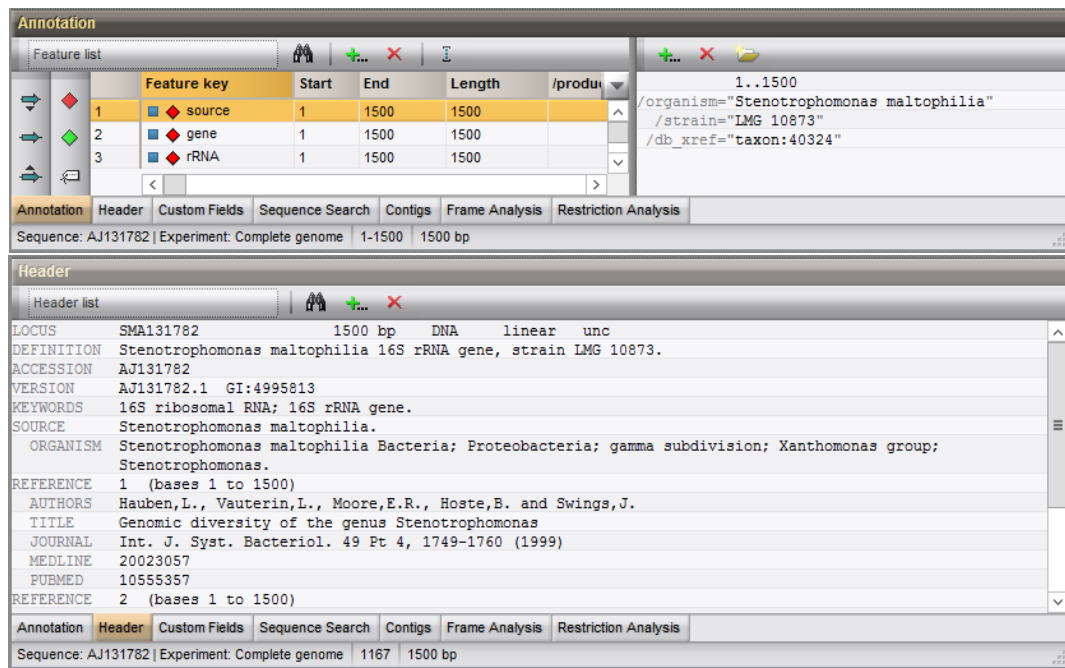


Figure 8.1.123: The *Sequence editor* window: *Annotation* panel and *Header* panel.

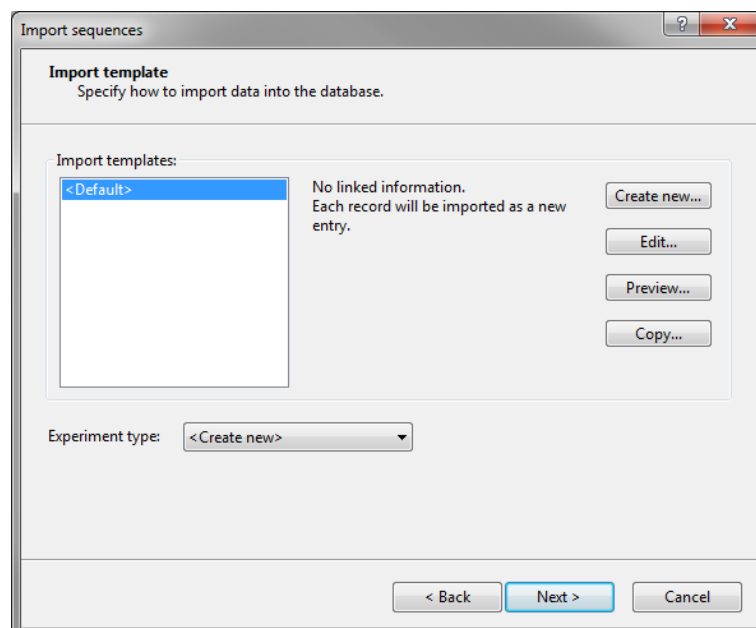


Figure 8.1.124: The *Import template* wizard page.

might need to scroll down the list to view the last row entry. The text *File* is specified in the *Source type* column and the text *Name* is displayed in the *Source* column.

The rows in the grid can be associated with new or existing entry information fields, sequence information fields, or sequence type names. Initially the rows are not linked to any information (the *Destination type* and *Destination* for all rows is set to <None>). Specifying a *destination* for one or more selected rows can be done by pressing the <Edit destination> button or by double-clicking. This action pops up a new dialog box prompting for the new destination for the selected row(s).

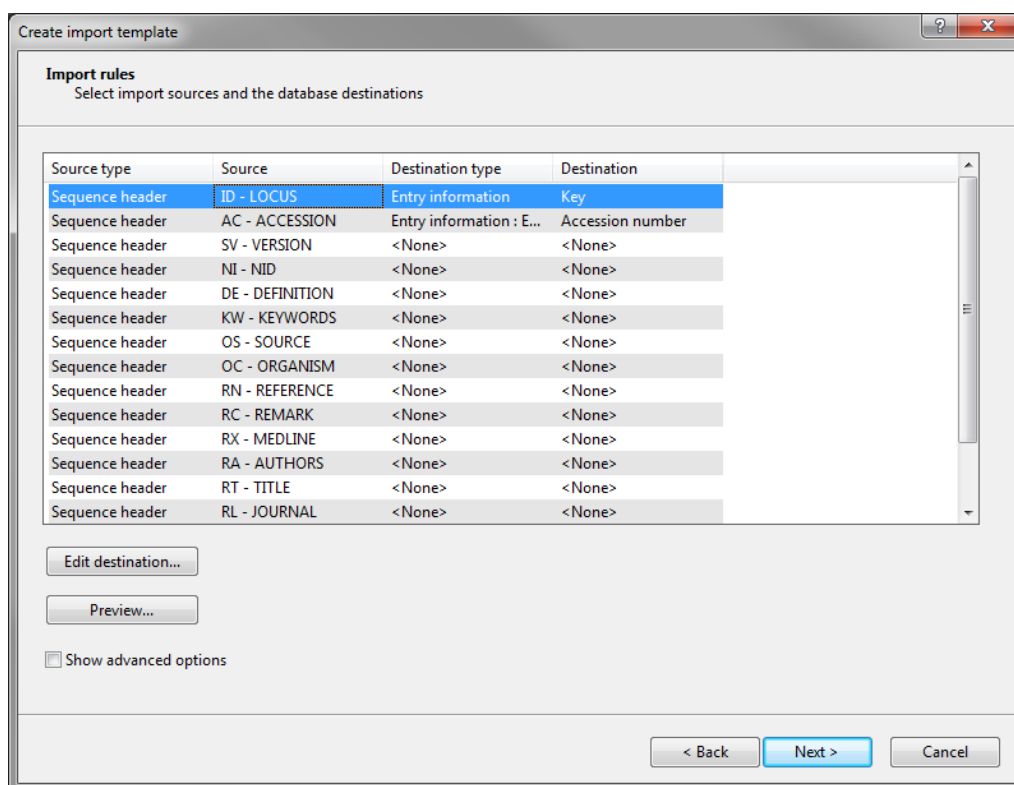


Figure 8.1.125: The *Import rules* dialog box.



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

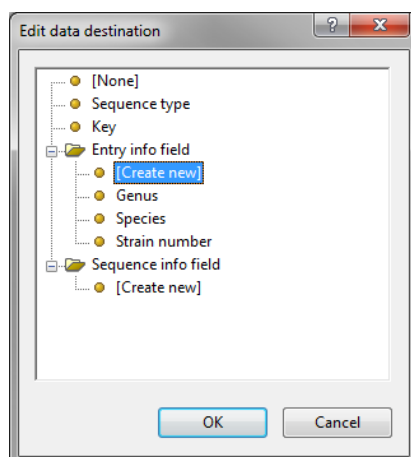


Figure 8.1.126: Edit data destination for a single selected row entry.

When only one row is selected in the grid, the information of this row can be linked to:

- The default information field **Key**.
- A **Sequence type** name. The (parsed) information will hold the sequence type name.

- A new or existing non-default entry information field (select the **<Create new>** option or an existing field under the topic **Entry info field**, respectively).
- A new or existing sequence information field (select **<Create new>** or select an existing field under the topic **Sequence info field**, respectively).

If a row is linked to a new entry information field or a new sequence information field, a new dialog box pops up when pressing the **<OK>** button (see below for an example).

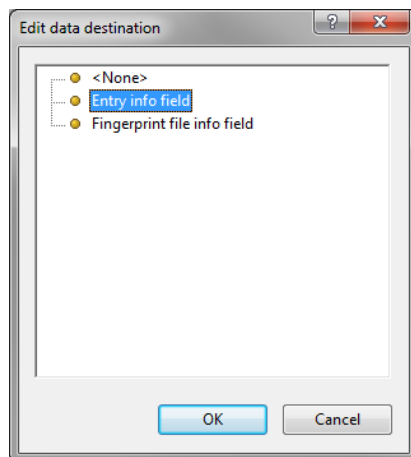


Figure 8.1.127: Edit data destination for multiple selected row entries.

When multiple rows are selected in the grid, the information of these rows can be linked to:

- Non-default entry information fields (select the **Entry info field** option).
- Sequence information fields (select the **Sequence info field** option).

When pressing the **<OK>** button, the plugin checks if the selected rows can automatically be mapped to existing entry information fields or sequence information fields in the database. If no entry information fields or sequence information fields exist with the same name, a new dialog box pops up prompting for the names.

When a row in the grid is linked to the **Key** field, the **Destination** column in the grid displays the name **Key**. When a row is linked to a sequence type name, the **Destination type** and **Destination** columns display the text **Sequence type**. Rows that are linked to entry information fields in the database display the text **Entry info field** in the **Destination type** column; the **Destination** column holds the name of the entry information field. When rows are linked to sequence information fields, the text **Sequence info field** is displayed in the **Destination type** column; the name of the field is listed in the **Destination** column.

Pressing **<Preview>** opens the *Preview* dialog box displaying the parsed information using the template settings. The preview can be closed with the **<Close>** button.

When the **<Show advanced options>** check box is enabled, three more columns appear inside the grid and eight extra buttons appear below the grid. These advanced options are explained in 3.3.5.5.

Pressing the **<Cancel>** button cancels the operation and the template settings are not saved to the database.

Pressing the **<Next>** button calls a new dialog where the entry link field needs to be defined.

- If a row in the grid is linked to the **Key** field in the database, **Key** is automatically selected as the entry link field. If entries are already present in the database with the same (parsed) key information, the import tool will link the data to these entries. If the **Key** field was linked for several levels, the import tool will link to each of these levels and also create cross links between the levels.

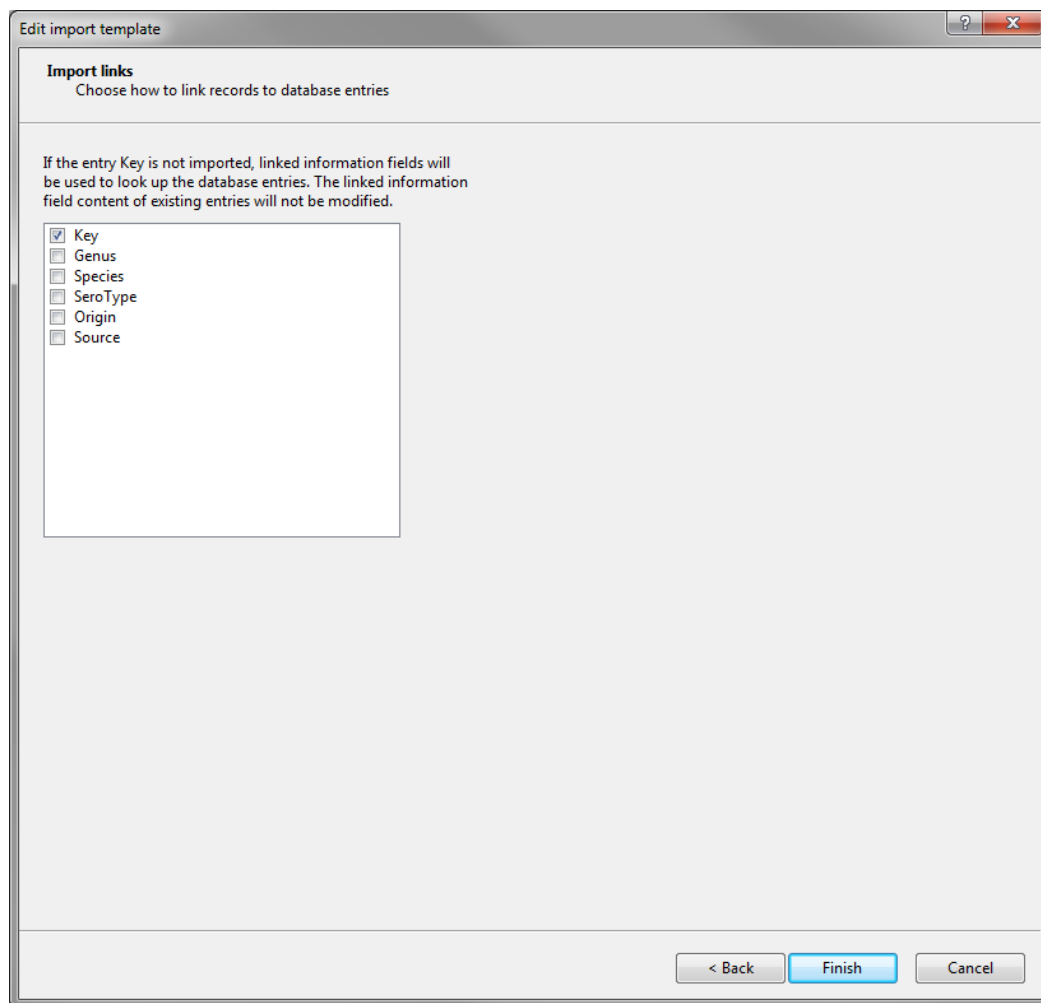


Figure 8.1.128: Specify the entry link field.

- If no row entry in the grid is linked to the **Key** field, but one or more rows are linked to an entry information field in the database, these fields can be selected from the list. If entries are already present in the database with this linked information, the import tool will link the data to these entries (Note: If the linked field content is not unique in the database, the data is linked to only one entry in the database, i.e. the first entry in the database holding the linked field content). If the entries are not yet present in the database, the data will be linked to new entries in the database.
- If no fields are selected from the list, no check for existing entries will be performed, and all data will be linked to new entries in the database (if the option **Create x entries** is checked in the last step of the wizard). New keys are automatically generated during import.

Pressing **<Finish>** brings up the last step of the wizard.

Each import template has its own unique **Name**.

Optionally, a descriptive text string can be entered in the **Description** input field. This descriptive text will be displayed when the template is selected from the template list.

The import template is saved to the database when the option **Save this import template in the database** is checked.

Check or uncheck the option **Share this import template** when the import template should be shared with other database users or when the template is intended exclusively for the current database user, respectively. Shared templates can be used and modified by all other database users.

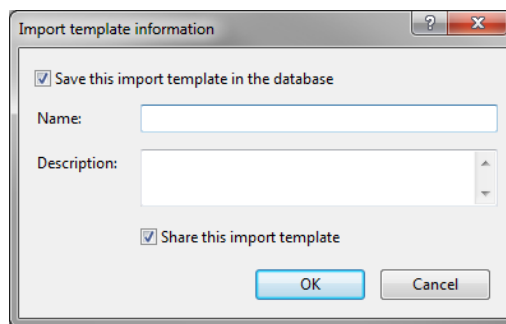


Figure 8.1.129: The *Import template information* dialog box.

When pressing the **<OK>** button, all template settings are saved to the database.

When a the template has been created, the template *Name* is shown in the *Import templates panel* and is automatically selected. The template *Description* is shown in the right panel.

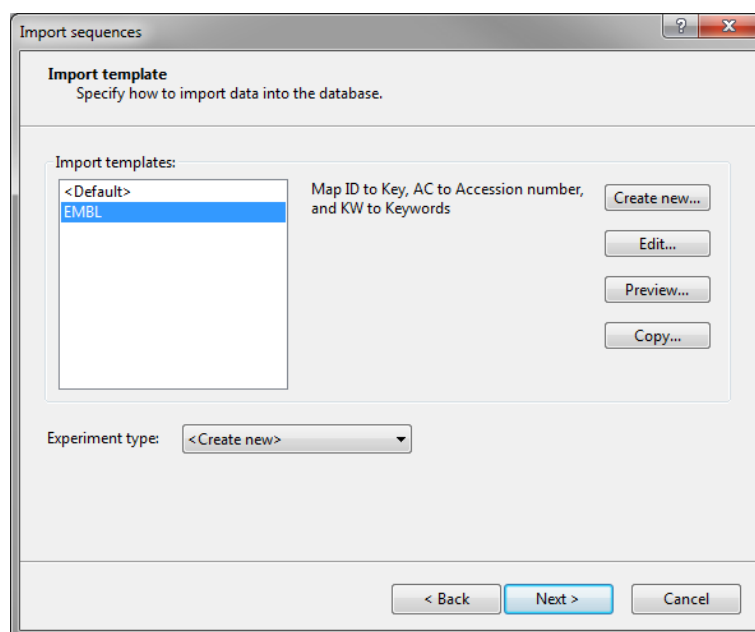


Figure 8.1.130: Import template added to the list.

A highlighted import template can be copied and saved under a different name by pressing **<Copy>**.

Pressing the **<Edit>** button brings up the *Import rules* dialog box again. If a conversion rule could not be applied to a selected file (e.g. a template column name could not be found in the selected file), the tag **<Absent>** is shown in the *Source* column next to the source name. Only if all conversion rules of a template can be applied to the selected file(s), the template can be used to import data in the database.

The *Preview* dialog box is displayed when pressing the **<Preview>** button. The *Preview* dialog box displays the parsed information using the template settings. The preview can be closed with the **<Close>** button.

If no row entry in the grid is linked to the *Sequence type name* destination, the sequences can be linked to an existing sequence type experiment or to a new sequence type experiment (**Create New**). When sequences are linked to a new sequence type experiment, a dialog box pops up when pressing the **<Next>** button, prompting for the sequence type name.

If a row in the grid is linked to the *Sequence type name* destination, the text **From import template** is automatically selected in the *Experiment type* text box. The import tool will link the sequences to the cor-

responding (parsed) sequence type names. If the sequence type experiments are not present in the database, a dialog box pops up when pressing the **<Next>** button, prompting for the names.

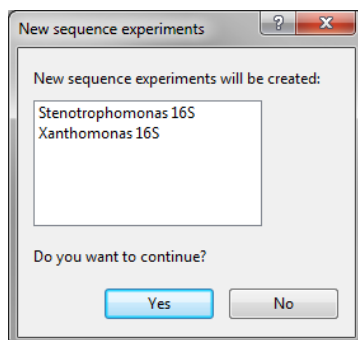


Figure 8.1.131: The *New experiment types* dialog box.

This dialog asks you to confirm the creation of the sequence type(s).

The *Database links* wizard page prompts for some final settings.

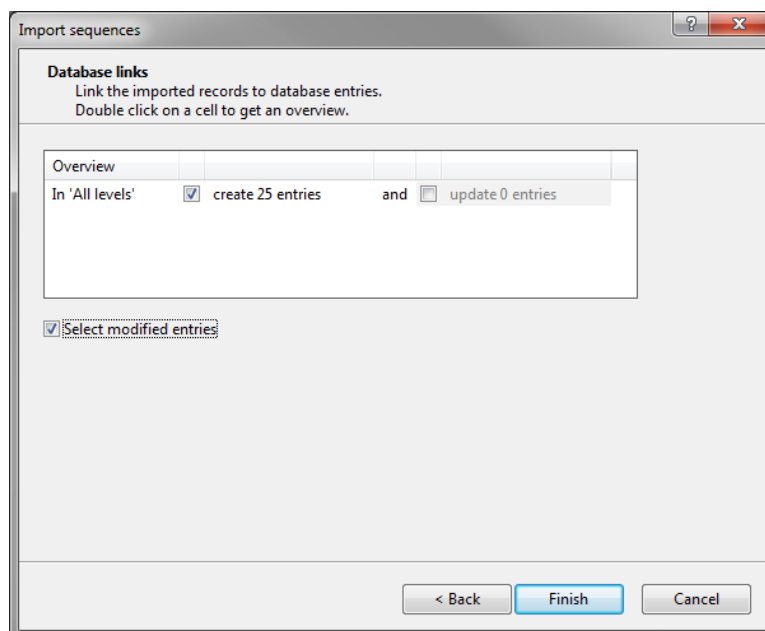


Figure 8.1.132: The *Database links* wizard page.

- When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database.
- Check the option **Update *x* entries** if you want the software to be able to update the information for existing entries.
- If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), the **Create *x* entries** option will have a red background. The entries with a **Key** exceeding the maximum number of allowed characters will not be created in the database. Double-clicking on the red menu item opens the *Entry key import* dialog box.

Pressing **<Next>** will start the import.



Mapped entry field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.



Mapped sequence field information, exceeding the maximum number of allowed characters (i.e. 79 characters) will be truncated to 79 characters during import. A message pops up asking the user to confirm the import action.

8.1.3.7 Assembling sequences from trace files

8.1.3.7.1 Introduction


The BioNumerics Assembler is a program to assemble contig sequences from partial sequences which result from sequencing experiments. The program accepts flat text files as well as binary chromatogram files from Applied Biosystems, Beckman, and Amersham automated sequencers, including the SCF sequence trace format. In the latter cases, Assembler allows the user to verify base assignments by inspecting the chromatograms along with the partial sequences and the consensus sequence. Assembler also investigates the quality and ambiguity of the curve profiles to assign a quality label to the partial sequences and trim off bad parts where necessary.


Contig sequences are saved into contig projects, which contain all the information about the partial sequences, the editing made by the user, the multiple alignment, and the editing done on the contig. A contig project and its full information can be opened at any time from the BioNumerics sequence entry to which it is associated. Assembler can handle thousands of sequences in one single contig project and is optimized for speed and editability in large projects. The program can be launched from BioNumerics but not as a separate program.




The BioNumerics batch sequence assembly import routine allows the Assembler program to run in batch mode, thereby assembling a large number of trace files into multiple contigs. See [8.1.3.2](#) for more information.

8.1.3.7.2 Opening the Assembler window

The BioNumerics Assembler program can be launched from the *Entry* window by clicking on the empty flask button next to the name of the sequence type in the *Experiments* panel. The program will ask "The experiment 'sequence type name' is not defined for this entry. Do you want to create a new one?". By answering <Yes>, the *Sequence editor* window opens (see Figure [8.1.150](#)). Pressing the  button in the toolbar will launch the Assembler program to assemble a contig sequence from a series of partial sequencing experiments for this entry.

The Assembler Program can also be launched by clicking the  button in the *Experiment card* window (see [8.1.5](#)).

From a sequence in BioNumerics assembled using the Assembler or Power Assembler program, the project can be opened in the (Power) Assembler again by pressing the  button in the sequence *Experiment card* window or in toolbar of the *Sequence editor* window.

Double-clicking on a base of a sequence in the *Comparison* window pops up the *Sequence editor* window.

Selecting **File > Open assembler** () in the *Sequence editor* window toolbar launches Assembler.

8.1.3.7.3 Importing sequence files in Assembler

Choose **File > Import sequence files...** (📁) to import sequence files into the *Contig assembly* window. Under **Files of type**, different formats can be imported: **AB**, **SCF**, **MegaBACE**, and **Flat text** files.

Pressing the **<Open>** button imports the selected files into the *Contig assembly* window. Sequences can also be imported directly from the clipboard with **File > Import sequence from clipboard**.

The *Contig assembly* window consists of two main panels: the *Trimming* panel and the *Assembly* panel. The sequences are shown in the *Trimming* panel of the *Contig assembly* window (see Figure 8.1.133).

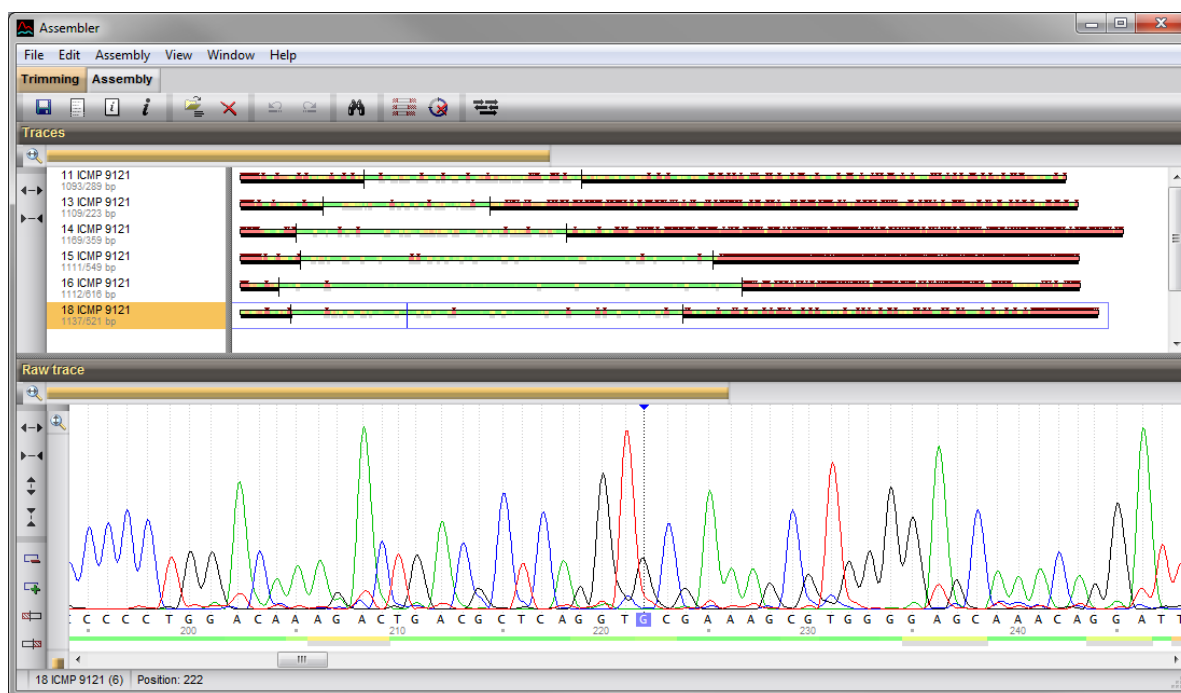


Figure 8.1.133: The *Contig assembly* window with the *Trimming* panel selected.

The *Trimming* panel displays the original sequences and gives an indication of the quality.

The colors of text, background, bases, and all other symbols may be changed by the user with **View > Display settings...**

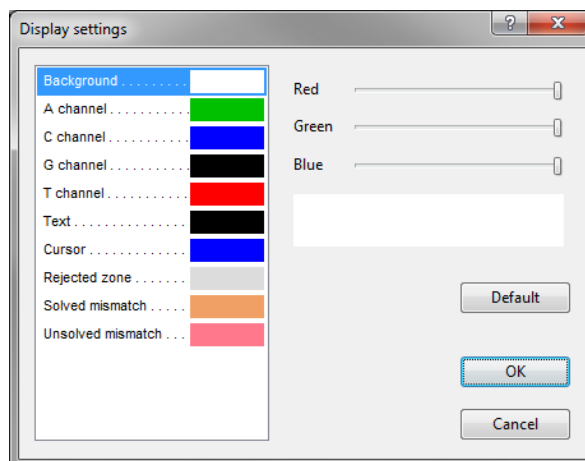




Figure 8.1.134: The *Display settings* dialog box.

The color of the selected item can be changed using the **Red**, **Green** and **Blue** sliders in the right panel. To restore the default configuration, press the **<Default>** button.


The *Traces overview* panel (top right) shows the sequences in a graphical representation. For each sequence, there is a quality assignment, based on the quality of the densitometric curves and the base assignment. Based on the quality, the program will automatically trim the bad parts from the sequences, which are underlined with a black bar. Unknown bases (ambiguous positions) are indicated with a dark red flag on top of the sequence.


The *Traces list* panel (top left) shows the corresponding file names in the upper line. In the bottom line, the original size in base pairs and the size after trimming are shown for the sequence.


Zooming in on the *Traces overview* panel is done with **View > Zoom in (overview)**, by pressing  in the *Traces* panel toolbar or by using the zoom slider.


To zoom out on the *Traces overview* panel, use **View > Zoom out (overview)**, press  in the *Traces* panel toolbar or use the zoom slider.

The *Raw trace* panel (bottom) displays the densitometric curve in four colors and the corresponding bases for the selected sequence.

You can horizontally zoom in on the curve with **View > Zoom in (trace)**, with  in the toolbar of the *Raw trace* panel or by using the zoom slider in the *Raw trace* panel.

To zoom out on the curve in the horizontal direction, use **View > Zoom out (trace)**, or  in the toolbar of the *Raw trace* panel or use the zoom slider in the *Raw trace* panel.

To enlarge the curve vertically, select **View > Enlarge trace**, click  in the toolbar of the *Raw trace* panel or use the zoom slider in the *Raw trace* panel.

To shrink the curve vertically, select **View > Shrink trace**, click  in the toolbar of the *Raw trace* panel or drag the zoom slider in the *Raw trace* panel.

A sequence can be selected from the *Traces overview* panel, or from the *Traces list* panel. The selected sequence is highlighted and its graphical overview is bordered by a colored rectangle.

A position can be selected on any sequence in the *Traces overview* panel by clicking it with the mouse. The selected position is indicated with a blue vertical line. The corresponding sequence chromatogram is shown in the *Raw trace* panel, with the selected position centralized and highlighted in blue. Likewise, a base position can be selected on the curve in the *Raw trace* panel, which causes the selection to be updated in the upper panels as well.

The logical work flow for a contig assembly is:

1. Cleaning trace files and quality assignment
2. Manual inspection of cleaning result
3. Removal of vector sequences (optional)
4. Assembling the contig (multiple alignment)
5. Manual inspection and correction of mismatches and unresolved positions
6. Trimming the consensus sequence according to known start and end signatures (e.g. primers) (optional)


8.1.3.7.4 Cleaning chromatogram readouts

Before aligning the sequences, the bad parts need to be cut out, i.e. the outermost left and/or right parts from the curves with unreliable signal or no signal at all. This process, called cleaning of sequences, consists of

two levels:

1. *Trimming* of the sequences, i.e. physically removing the unusable ends. This level of cleaning is based upon the percentage of unresolved positions at both ends of the sequence. Trimmed ends are neither used, nor shown in the *Assembly* panel of the *Contig assembly* window.
2. *Inactivating* doubtful parts of the sequence. This level of cleaning is based both on the quality of the densitometric curves and the proportion of unresolved positions. Inactivated parts are still shown, but do not actively contribute to obtain the consensus. However, they are aligned to the consensus. In case there is no consensus base at a position, the inactivated regions will not be considered by the program. The user can still compare the consensus position with the base in an inactivated sequence region. Inactive regions can still be set as active at anytime, whereas active regions can be set as inactive as well. In case an inactivated region is the only information available in a part of the consensus sequence, it will be used to fill in the consensus sequence. In case a position on an inactivated region conflicts with other sequences, it will be ignored.

Cleaning of the sequences happens automatically and is based on the *quality assignment* settings. The quality of the sequence is shown on the *Traces overview* panel in the *Trimming* panel (see Figure 8.1.133). A color scale ranges from green (acceptable quality) over yellow and orange to red (unacceptable quality). The trimmed ends are indicated by a black bar underlining the sequence. Inactivated zones are indicated by a gray bar. Unresolved positions (N) are indicated with a small flag on top of the sequence.

To modify the quality assignment settings, select **File > Quality assignment...** (). This pops up the *Quality assignment* dialog box (see Figure 8.1.135).

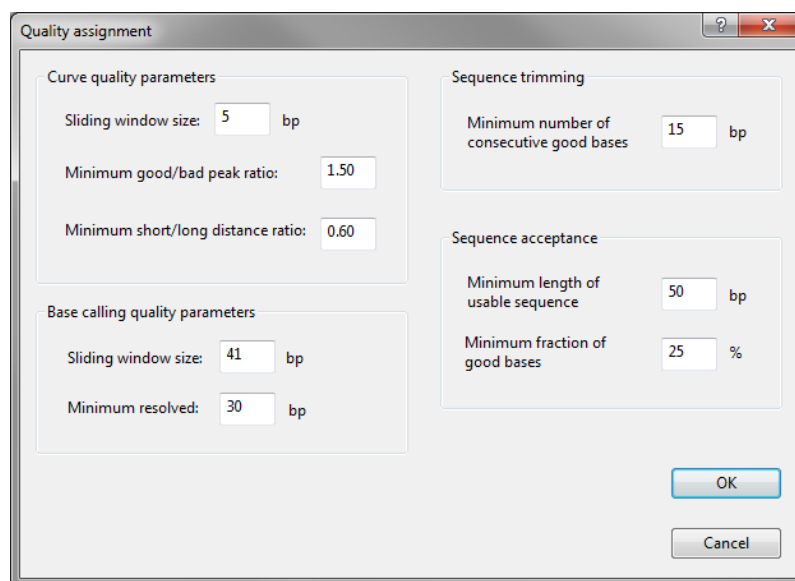


Figure 8.1.135: The *Quality assignment* dialog box.

The *Curve quality parameters* determine how the program will investigate the quality of signal derived from the curves. They include two ratios that are considered in a certain window, determined by the *Sliding window size*. The latter should be an odd number, including the position itself and a number of positions at either side. A typical value for the *Sliding window size* is 5 positions; increasing this value will result in a more stringent quality assignment. The *Minimum good/bad peak ratio* is the ratio between the signal strength of the weakest peak resulting in a base and the strongest peak not resulting in a base within the sliding window. The higher this ratio is set, the more stringent the quality assignment becomes. A suitable starting value for most systems is 1.50. The *Minimum short/long distance ratio* is the ratio of the shortest distance between two positions and the longest distance within the sliding window. A suitable starting value is 0.60; the larger it is set, the more stringent the quality assignment will be.


Under **Base calling quality parameters**, you can specify a **Sliding window size** and the number of resolved positions that should be found within the sliding window (**Minimum resolved**). Similar as under **Base quality assignment**, the **Sliding window size** should be an odd number. Suggested starting values are a **Sliding window size** of 41 of which minimum 30 resolved positions.


Sequence trimming is based upon the **Minimum number of consecutive good bases**, as defined by the **Curve quality parameters** and the **Base calling quality parameters**. A suggested value is 15; the larger the number, the heavier the sequence will be trimmed.


The **Sequence acceptance parameters** determine whether a sequence as a whole will be accepted to contribute to the consensus or not. The **Minimum length of usable sequence** determines the length of the non-trimmed part of the sequence. The **Minimum fraction of good bases** determines the ratio of good bases over the total number of bases in the usable part of the sequence. Suggested values are 50 bases of which minimum 25% good bases.


Automatic cleanup (trimming and assignment of inactive zones) happens automatically after pressing the **<OK>** button in the **Quality assignment** dialog box. Any manual trimming and (in)activation done will be lost at this point.

After automatic quality assignment and trimming, the user can still manually correct the trimmed ends and inactive zones:

To mark the start of a sequence, click on the position to start (this can be done both on the overview and on the curve) and select **Edit > Mark start of sequence** (, **Ctrl+Home**).



To mark the end of a sequence, click on the position to end (this can be done both on the overview and on the curve) and select **Edit > Mark end of sequence** (, **Ctrl+End**).

To mark a zone as inactive, click on the start position of the zone, then hold down the **Shift**-key while clicking on the end position of the zone (this can be done both in the overview and on the curve) and choose **Edit > Inactivate selected region** ()

To mark a selected zone as active, choose **Edit > Activate selected region** ()


A sequence can be inactivated as a whole with **Edit > Inactivate selected sequence**. When inactivated, a sequence is marked with a red cross in the **Traces list** panel (upper left).

A sequence that was inactivated by the **Sequence acceptance** parameters in the **Quality assignment** dialog box can be activated manually with **Edit > Activate selected sequence**.

A sequence can be removed from the project with **File > Remove selected sequence** ()**.** Conversely, sequences can be added to a project at any time with **File > Import sequence files...** ()

8.1.3.7.5 Removing vectors

If the sequences contain residues from vector sequences, these need to be removed before the sequences are assembled.

Vectors can be removed from the unaligned sequences with **File > Remove vectors...** ()**.** This pops up the **Remove vectors** dialog box (see Figure 8.1.136).

Vector sequences to be removed can be added from the clipboard (by copying from another application). They can be pasted in the list by pressing **<Add from clipboard>**. This opens the **Import vector from clipboard** dialog box (see Figure 8.1.137).

The sequence on the clipboard is automatically pasted into the editor, which the user can still edit. An input field **Name** allows a name to be entered for the vector. Vectors can be deleted from the list using the **<Delete selected>** button.

Vectors entered are automatically saved along with the project.

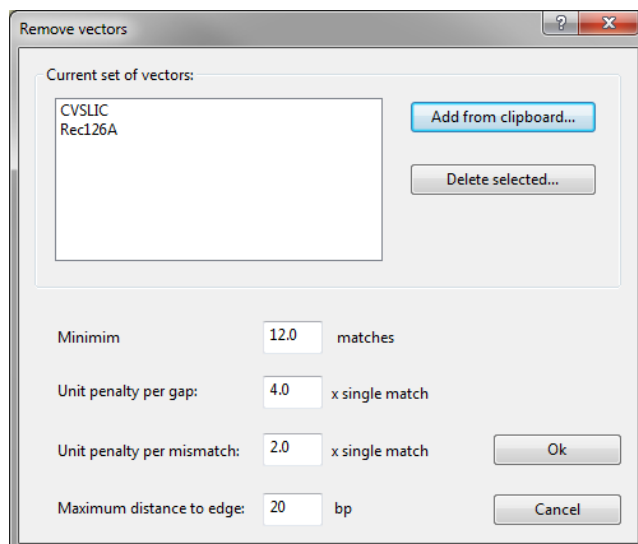


Figure 8.1.136: The *Remove vectors* dialog box.

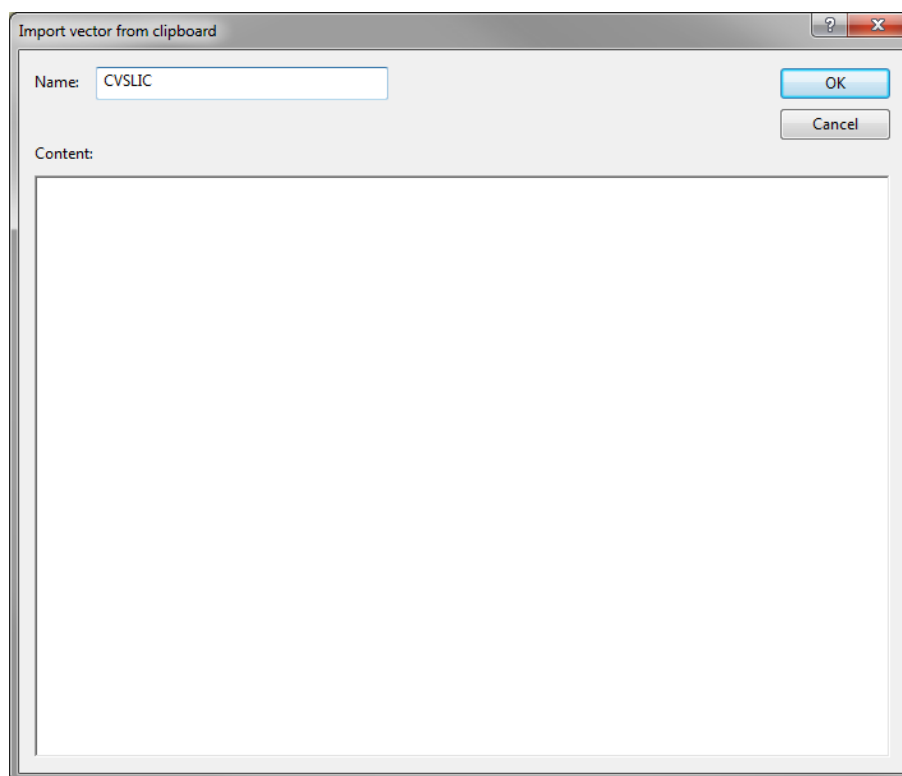


Figure 8.1.137: The *Import vector from clipboard* dialog box.

The *Remove vectors* dialog box contains a number of alignment parameters:

- **Minimum score:** the minimum number of matching bases the sequence and the vector should have in order for the vector sequence to be removed. This number is the result of the total number of matching bases minus the total penalty resulting from mismatches and gaps.
- **Unit penalty per gap:** the penalty, as a factor of the match score, assigned to a gap in either the sequence or the vector after the alignment.
- **Unit penalty per mismatch:** the penalty, as a factor of the match score, for a single mismatch between

the vector and the sequence after the alignment

- **Maximum distance to edge:** the maximum number of unmatched bases at the end of the sequence. Normally a vector sequence will extend over the end of the trace sequence, so one will not expect unmatched bases at the end of the sequence. Therefore, this number should be set very low (e.g. 5 or less).

By pressing <OK>, the vector sequences are automatically searched for and removed from the unaligned sequences. Removed vector sequences are indicated in blue in the overview panel.

To undo vector removal, delete all vectors defined and press <OK>.



Vector removal as well as undoing vector removal can only be executed on unaligned sequences. If sequences are already aligned, you will first have to remove the alignment.

8.1.3.7.6 Alignment to consensus

After trimming and (optional) vector removal, individual sequences should be assembled into a consensus sequence. To do so, select **Assembly > Assemble sequences...** (). This action will display the *Calculate assembly* dialog box (see Figure 8.1.138).

Figure 8.1.138: The *Calculate assembly* dialog box with alignment parameters.

In this dialog box, the various alignment parameters can be specified:

- The **Minimum match word size** determines the number of bases that are taken together into one *word*. The algorithm creates a lookup table of groups of bases to accelerate the alignment, which increases the speed of the algorithm. In an alignment to a consensus sequence, no mismatches are expected, except due to bad base calling. In that case, it is justified to choose a high *word size* number. In the default setting of 7, only stretches of 7 identical bases or more will be considered as matches.
- **Minimum score:** the minimum number of matching bases the two sequences should have before they will be aligned. This number is the result of the total number of matching bases minus the total penalty resulting from mismatches and gaps.
- **Unit penalty per gap:** the penalty, as a factor of the match score, assigned to a gap introduced in one of the sequences after the alignment.
- **Unit penalty per mismatch:** the penalty, as a factor of the match score, for a single mismatch between the two sequences after the alignment.

- **Maximum number of gaps** relates to the alignment technique that is used, i.e. a fast algorithm based upon Needleman and Wunsch (1970) [30]. The number of gaps the algorithm can create is proportional to the number of diagonals specified. The larger the number, the more accurate but the slower the calculations. The suggested default setting is 25 diagonals.

The check box **Ignore current assemblies** allows the algorithm to recalculate the consensus sequence(s) from individual trace sequences without taking into account any already calculated contigs.

Pressing <OK> will calculate the assembly using the specified assembly settings.

8.1.3.7.7 Editing a consensus sequence

When the alignment is finished, the *Assembly* panel is shown by default (Figure 8.1.139). As compared to the *Trimming* panel (see Figure 8.1.133), a central *Alignment* panel now shows the consensus sequence (upper line) and the individual trace sequences that contribute to the displayed consensus.

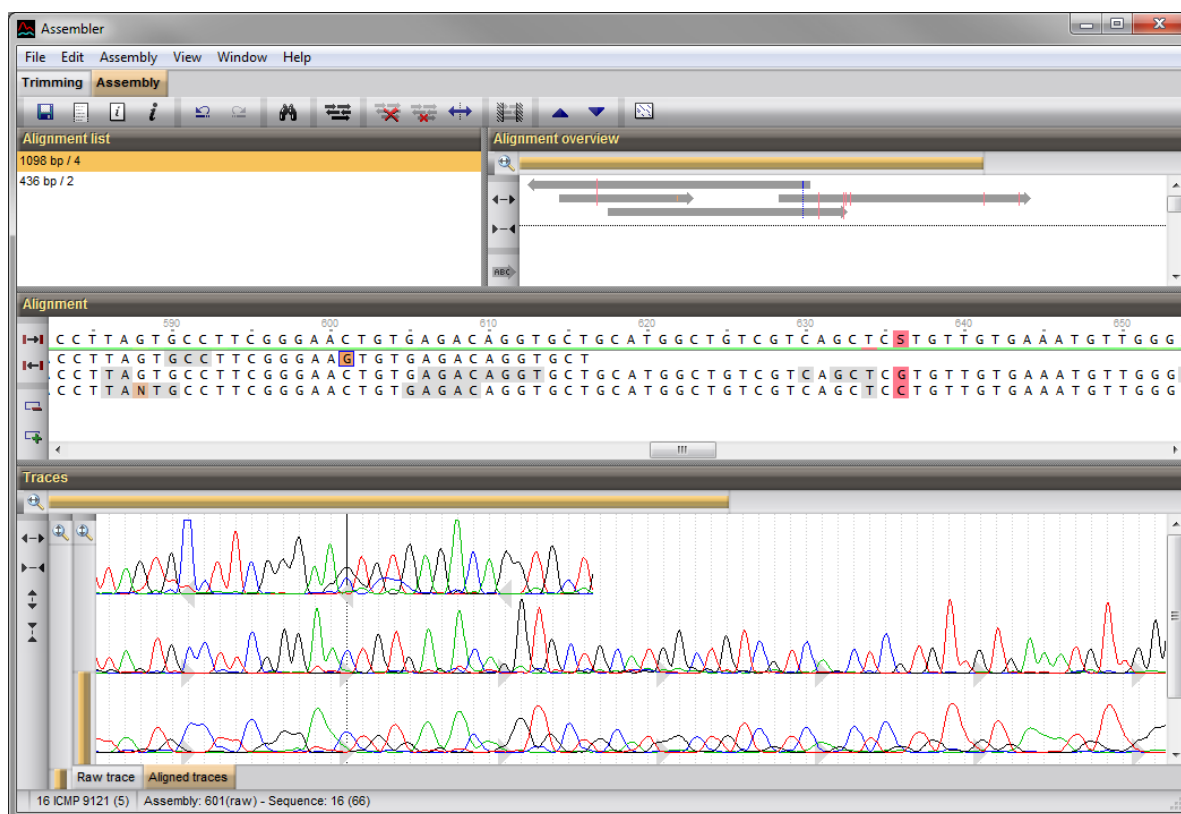


Figure 8.1.139: The *Contig assembly* window, with the *Assembly* panel displayed.

The *Alignment overview* panel (top right) displays the aligned trace sequences. If the arrow points to the left, the program has invert-complemented the sequence to obtain the correct alignment.

You can have the names of the trace files displayed on top of the bars in the *Alignment overview* panel by selecting **View > Show trace names** (REC).

The *Alignment list* panel (top left) now displays the selected consensus with its length and the number of sequences that are part of it. If the program could not align all trace sequences to a single consensus, the panel lists the different consensus sequences with their lengths and number of trace sequences. One should click on a particular consensus sequence to select it for viewing and editing.



The *Traces* panel (bottom) has two tabs, corresponding to the *Raw trace* panel and the *Aligned traces* panel. Depending on which view was last displayed when *Contig assembly* window was closed, the *Raw trace*



panel or *Aligned traces* panel is shown.



The *Raw trace* panel displays the chromatogram file for the selected trace sequence. Regardless of whether the sequence is invert-complemented in the alignment, the chromatogram is always shown in original mode. This means that, when the sequence has been invert-complemented, a G on the original sequence, for example, will appear as a C on the consensus. Due to the fact that the direction of the curve can be opposite from the sequence and that the bases are not aligned, it is not possible to select bases on the raw curves directly.



The *Aligned traces* panel (Figure 8.1.139, bottom) has the following features:


- Curves have been stretched or shrunk to obtain equidistant spacing between the base positions.
- Trace sequences are always shown as transformed and oriented in the consensus. If a sequence is invert-complemented, the complement of the bases is shown, and the colors of the curves are adjusted likewise.
- Multiple trace chromatograms can be shown together and are aligned to each other and to the consensus.
- Arrows on the curves indicate the direction of the sequence: if the sequence has been inverted, the arrow points to the left.
- Bases can be selected on the curves.



You can zoom in on the curves with **View > Zoom in (trace)** () or by dragging the  zoom slider in the *Traces* panel. See 2.3.7 for a description of zoom slider functions.

To zoom out on the curves, use **View > Zoom out (trace)** () or drag the  zoom slider in the *Traces* panel.

To zoom in on the curves vertically, select **View > Enlarge trace** () or drag the  zoom slider with the mouse.

To zoom out on the curves vertically, select **View > Shrink trace** () or drag the  zoom slider with the mouse.

With a second vertical  zoom slider in the *Traces* panel, the vertical space reserved for the curves can be determined.

A sequence can be moved up or down by selecting it and choosing **Edit > Move sequence up** (, **Page Up**) or **Edit > Move sequence down** (, **Page Down**), respectively.

Bases on the consensus sequence are assigned according to *consensus determination parameters*, which can be set with **Assembly > Consensus determination...** The *Consensus determination* dialog box (see Figure 8.1.140) allows four parameters to be set:

- **Required bases to include position:** The percentage of sequences that need to have a base at a certain position in order for the position to be inserted in the consensus. For example using the default value 50, if the consensus is determined by three sequences at a certain position, it will not be accepted as a base if there is a gap in two of the three sequences (33.3%).
- **Required consensus for unique base calling:** The percentage of sequences that need to have the same base at a position in order for the base to be accepted as resolved.
- **Required consensus for 2-fold degeneracy:** The summed percentage of sequences having two different bases at a position in order to be denoted with IUPAC code for 2-fold degenerated positions (R: A/G; M: C/A; S: C/G, Y: C/T; W: A/T; K: G/T). Only applicable for positions that do not fulfill the criterion for unique base calling.

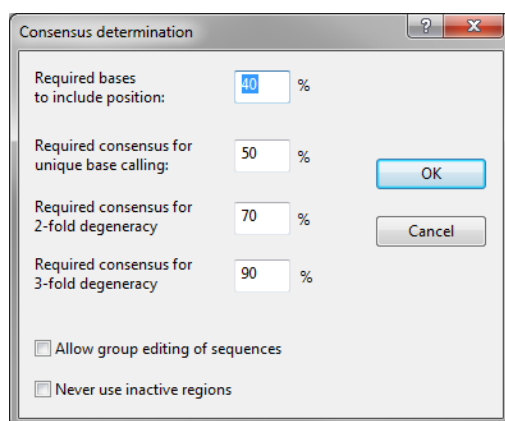


Figure 8.1.140: The *Consensus determination* dialog box.

- **Required consensus for 3-fold degeneracy:** The summed percentage of sequences having three different bases at a position in order to be denoted with IUPAC code for 3-fold degenerated positions (V: A/C/G; H: A/C/T; D: A/G/T; B: C/G/T). Only applicable for positions that do not fulfill the criteria for unique base calling and 2-fold degeneracy. Any position that does not reach the required consensus for 3-fold degeneracy is denoted as N.

Allow group editing of sequences is a feature that allows bases to be changed directly on the consensus sequence. If this feature is enabled, corresponding positions on the trace files will be changed accordingly. If disabled (default), bases can only be changed on the individual trace sequences.

If **Never use inactive regions** is checked, inactivated regions on the individual trace sequences will not be contribute to the consensus sequence.



If you do not wish to use the parameters for 2-fold or 3-fold degeneracy, you can leave the corresponding text boxes empty.

Unresolved positions on the consensus are indicated in pink and extend over all sequences shown (*Alignment* panel, see Figure 8.1.139).

Problem positions on individual trace sequences, which have been solved under the current consensus determination parameters are indicated in orange. Such problem positions include mismatches as well as unresolved positions.

To change a base in a trace sequence, place the cursor on the base or on the position on the chromatogram and type the base, which can be A, G, C, or T, or any IUPAC code for denoting ambiguous positions.

To delete a base, select **Edit > Delete base (Del)**.

To insert a position, select **Edit > Insert column (Ins)**.

If group editing is enabled in the *Consensus determination* dialog box (see Figure 8.1.140), it is also possible to place the cursor on the consensus sequence and type a base, which causes the base to be changed on all sequences that have signal at the selected position.

The Assembler program contains a multi-step undo and redo function. To undo a command, select **Edit > Undo** (🗑️, **Ctrl+Z**). To redo a command, select **Edit > Redo** (📄, **Ctrl+Y**).

In addition, the program also stores a history of editing actions done on each individual sequence. This information can be popped up by selecting the sequence (clicking on any position on the sequence, in the chromatogram or on the overview) and calling **File > Sequence information...** (📄, **Ctrl+I**).

The *Sequence information* dialog box lists all base corrections that are made to the sequence. The corrections recorded include base changes, deletions and insertions.

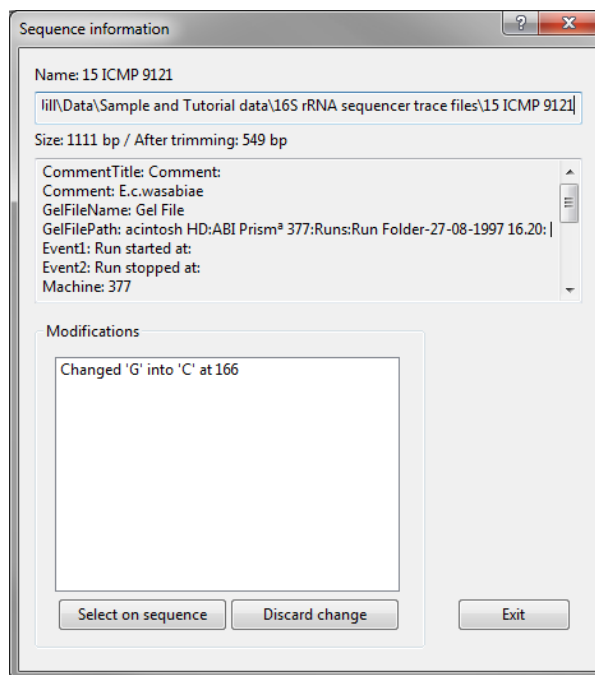


Figure 8.1.141: The *Sequence information* dialog box.

From this dialog box, you can select a particular editing action in the list, and press **<Select on sequence>**. The position will be selected on the sequence. A correction made can be undone by pressing the **<Discard change>** button.

A range of bases can be selected on the curves or on the sequences in the central panel by clicking the first position of the range, then holding down the **Shift**-key while clicking on the last position. A selected range is highlighted by a blue rectangle in the sequence view. Range selection by dragging the mouse is also possible in the sequence view.

If a selection of bases is flanked by a gap at one side, it is possible to shift the selection towards that gap, to correct misalignments. Shifting towards the left can be done with **Edit > Shift block left (Alt+Left)** and shifting towards the right with **Edit > Shift block right (Alt+Right)**.

To check the consensus sequence for correctness, you can let the program jump to each next unresolved problem position using **View > Next unresolved problem (F5, Ctrl+Right)**. To jump to the previous unresolved problem, use **View > Previous unresolved problem (F4, Ctrl+Left)**.

In case the program has incorrectly aligned a sequence to one or more other sequences, you can place the cursor on the misaligned sequence and select **Assembly > Break selected sequence apart**.

New sequences can be added at any time to the existing alignment project by switching to the first view and selecting **File > Import sequence files...** (F6), and subsequently selecting **Assembly > Assemble sequences...** (F7). In the *Calculate assembly* dialog box (see 8.1.3.7.6), **Ignore current assemblies** should normally be unchecked, to preserve the assembly or assemblies already present.

Although Assembler automatically inverts and complements subsequences wherever necessary to obtain the consensus sequence, the program cannot know the correct orientation of the consensus sequence. Hence, it may be necessary to invert and complement the consensus sequence before entering it into the database. Invert-complement the consensus sequence by selecting the consensus to invert and **Assembly > Invert direction** (F8).



In case the program could not find one single consensus for all subsequences, two or more assemblies will exist. Therefore you will need to select the assembly to invert from the list in the *Alignment list* panel (upper left) before executing the invert-complement function.

Manual editing such as assigning the start/end of a sequence and activating/inactivating parts (see 8.1.3.7.4) for the *Trimming* panel remains available to further clean up sequences.

It is also possible to extend a sequence that has been trimmed off too far. To do so, select the outermost base on the sequence and **Edit > Extend sequence (Ctrl+X)**. An input box will ask you to enter the number of bases to extend.

A region on an individual sequence or on the consensus can be selected as explained above and can be copied to the clipboard using **Edit > Copy (Ctrl+C)**.

The entire sequence on which the cursor stands, or the entire consensus, can be selected with **Edit > Select all (Ctrl+A)**.

A selected sequence can be removed from a contig with **File > Remove selected sequence** (✖).

A consensus sequence and its associated alignment can be removed by selecting it in the *Alignment list* panel and choosing **Assembly > Delete selected contig...** (✖).

All alignments and consensus sequences can be removed with **Assembly > Delete all contigs....**

The latter two options can be useful if you want to load stored templates (see 8.1.3.7.13), remove vectors (8.1.3.7.5) or change the quality assignment parameters (8.1.3.7.4). These actions cannot be performed if an alignment is present.

The overview panel of a contig project can be printed with **File > Print overview....**

8.1.3.7.8 Advanced alignment editing using dot plots

Using a dot plot, regions of homology between two sequences are displayed graphically. To allow the dot plot to display the homology between very long sequences in an efficient way, three reduction factors will be applied: (1) bases are grouped together into *words* of a specific length, (2) a minimum number of bases should match before the match is displayed on the dot plot, and (3) the entire plot is reduced in size.

The *Contig dot plot* window is called with **Assembly > View dot plot** (🖨). First, the *Dot plot* dialog box appears (Figure 8.1.142).

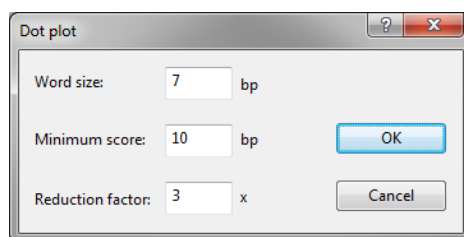


Figure 8.1.142: The *Dot plot* dialog box.

This dialog box prompts to enter the parameters for **Word size**, **Minimum score**, and **Reduction factor**. The values to enter depend strongly on the size of the project.

When pressing <OK>, the *Contig dot plot* window appears (see Figure 8.1.143).

In this window, each consensus sequence is represented as one gray square. Repeats found within a consensus are shown within the gray squares; whereas repeats found between the consensus sequences are shown in the rectangles that form the intersections between the consensus sequences. The upper left part of the window displays the *direct repeats* (in green), whereas the lower left part of the window displays the *inverted repeats* (in blue).

You can zoom in or out on the plot using **Edit > Zoom in** and **Edit > Zoom out**, or by using the zoom buttons 📐 and 📐 in the toolbar.

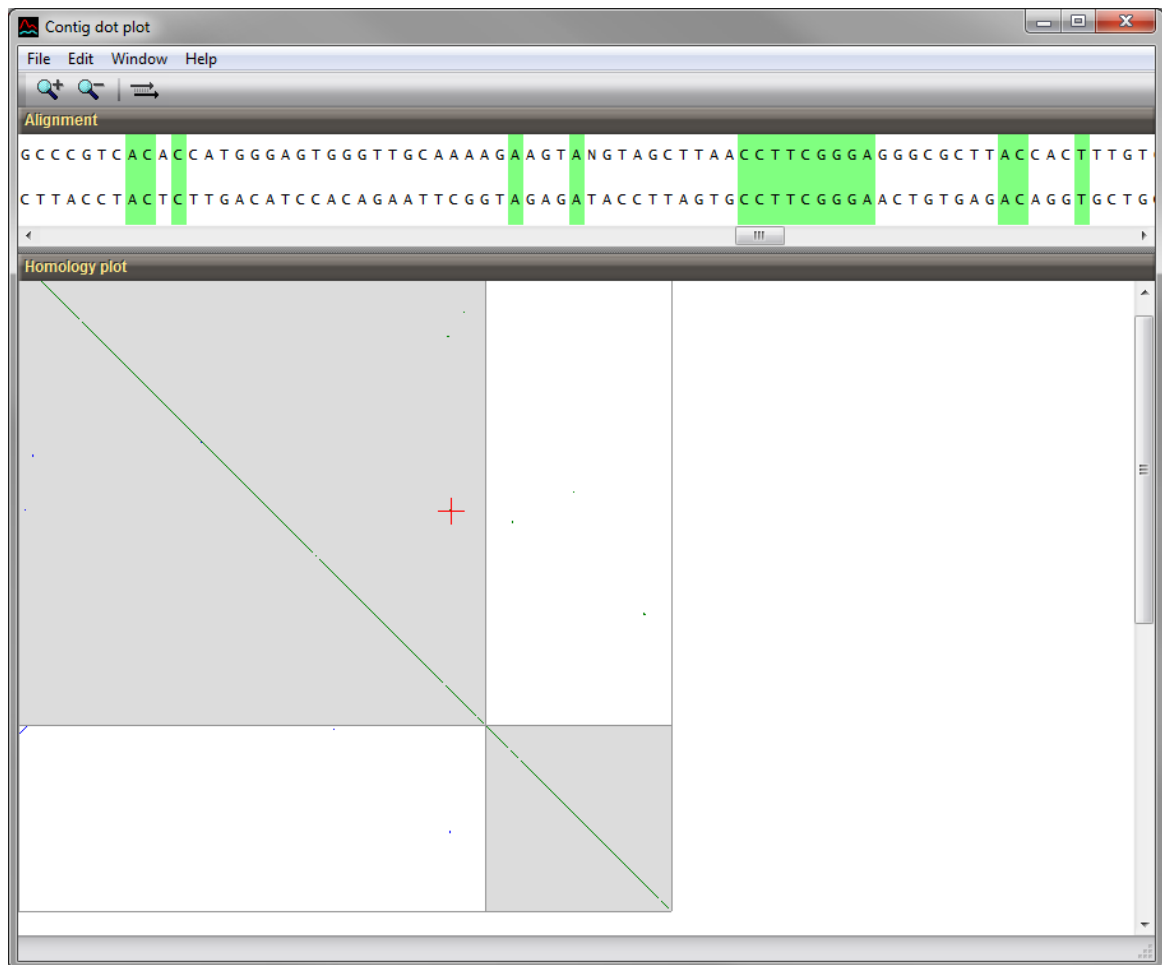



Figure 8.1.143: The *Contig dot plot* window.

A repeat (direct or inverted) can only be considered interesting in a contig project if it extends from a vertical side of a rectangle to a horizontal size: only then there is a complete overlapping end between consensus sequences, which can thus be merged.

Inside the *Contig dot plot* window, you can click on a particular dot or stretch of dots. A red cursor appears, and the upper panel displays the matching region between the two sequences, matching bases on a green background.

To merge two sequences that have a terminal match, select **Edit > Merge contigs** or press . The consensus sequences are now merged in the *Contig assembly* window and the *Contig dot plot* window is updated accordingly.

8.1.3.7.9 Approving a contig project

A contig project can be marked as being approved or not approved in the *Experiment presence* panel. For approved sequences, the colored dot indicating experiment presence is surrounded by a square of contrasting color, whereas for non-approved sequences the dot appears in a transparent square. The same squares are also indicated on the sequence *Experiment card* window (8.1.5). Sequences can be marked approved or non-approved with **File > Approved**.

8.1.3.7.10 Storing a contig project

When the aligned sequences are ready for importing in the sequence database, select **File > Save** (📁, **Ctrl+S**).

In case the program could not align the trace sequences to one single consensus, the different contig sequences will be saved into one sequence, separated by a pipe (|). They will be saved in the same order as they appear on the screen.

The order of the contigs can be changed by selecting a contig in the *Alignment list* panel and using **Assembly > Move contig up** or **Assembly > Move contig down**.

Select **File > Show report** (📄) to generate a report of all settings defined for the contig project.

The report can be saved as HTML or text with **File > Save as html** (📄) or **File > Save as text** (📄), respectively.

8.1.3.7.11 Finding subsequences

With **Edit > Find...** (🔍, **Ctrl+F**), you can pop up the *Find sequence* dialog box to find subsequences (Figure 8.1.144).

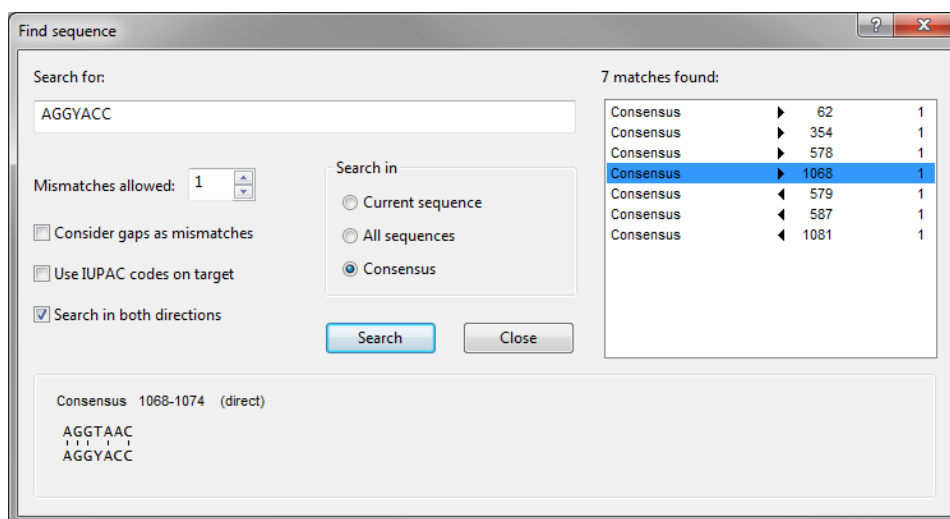


Figure 8.1.144: The *Find sequence* dialog box.

In the **Search for** text box, you can fill in a subsequence including unresolved positions according to the IUPAC code.

Under **Search in**, you can choose between **Current sequence** (the selected one), **All sequences**, and **Consensus**.

Using **Mismatches allowed**, it is possible to find subsequences that differ in a defined number of bases from the entered string.

The check box **Consider gaps as mismatches**, allows the search algorithm to introduce gaps in either the search sequence or the target sequence to match them. Gaps are considered in the same way as mismatches, and thus depend on the **Mismatches allowed** setting.

Use IUPAC codes on target allows the search sequence to be matched with uncertain positions denoted as IUPAC unresolved positions (e.g. R, Y, etc., including N).

With **Search in both directions** enabled, the invert-complemented sequence will be searched through as well.

Press **<Search>** to execute the search command. The result set displays all the instances that were found (Figure 8.1.144), indicating with arrows if they have been found on the sequence as is, or after invert-complementing. The positions are also indicated.

If you click on an item in the result set, the matching subsequence is selected in the *Alignment* panel. The bottom part of the *Find sequence* dialog box displays the alignment of the search sequence and the target sequence, indicating mismatches and gaps introduced (if allowed).

8.1.3.7.12 Trimming the consensus sequence

The purpose of this tool is to locate two fixed subsequences on the consensus to define the start and end position, respectively. One can choose to include or exclude the locator sequences in the final consensus. In many cases, but not always, these subsequences will correspond to primers used. For generality, the subsequences are called *trimming targets* in the program and in the description that follows.

Select **Assembly > Consensus trimming...** (🔍) to open the *Consensus trimming* dialog box (see Figure 8.1.145).

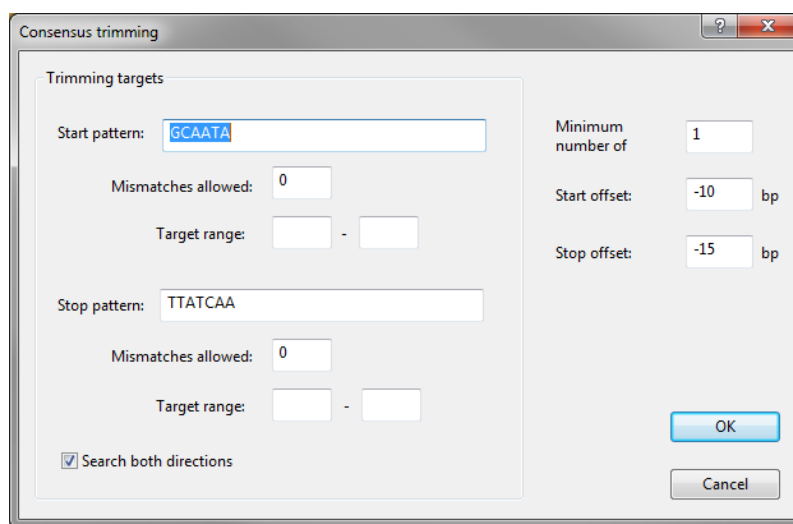


Figure 8.1.145: The *Consensus trimming* dialog box.

Under *Trimming targets*, you can fill in a *Start pattern* and an *End pattern*. For both the start and end patterns, you can specify *Mismatches allowed*, and fill in a *Target range* on the consensus. The latter is to restrict the search to certain regions on the consensus, e.g. to prevent incidental matches inside the targeted consensus sequence.

With *Search both directions*, the entered trimming targets will be searched for on the consensus sequence as it appears as well as on its complementary strand. In case the trimming targets match the complementary strand of the consensus, it will be automatically invert-complemented.

Minimum number of specifies a minimum number of trace sequences that should be contributing to the subsequence on the consensus that matches the trimming targets. For example, if “2” is entered, a trimming target will only be set if the matching region on the consensus is fully defined by at least 2 sequences.

With *Start offset* and *End offset*, one can specify that the consensus is trimmed at a certain offset from the start and end trimming target positions, respectively. If no offset is specified (zero), the trimming targets are **included** in the trimmed consensus.

When the trimming targets have been set by pressing the **<OK>** button in the *Consensus trimming* dialog box, the *Alignment overview* panel shows black hatched lines at the positions of the trimming targets. Likewise, the consensus sequence in the *Alignment* panel is grayed where it is trimmed off.

8.1.3.7.13 Storing and using assembly templates

The Assembler program automatically stores all user defined settings from the last saved project in a template called "DefaultSettings" (see Figure 8.1.146). These settings include the display settings, the quality assignment parameters, the vectors to remove and their parameters, the alignment parameters, the consensus determination parameters, the consensus trimming targets and their parameters. When opening a new project, these settings are automatically applied to the new project. In addition to the "DefaultSettings" template, other templates can be stored.

Select **File > Templates...** () to open the *Templates* dialog box (see Figure 8.1.146).

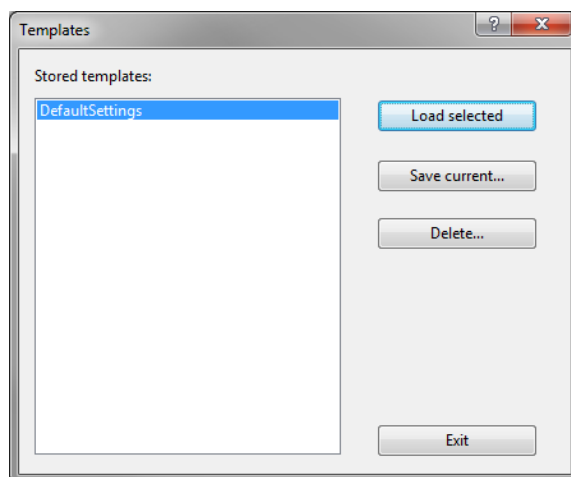


Figure 8.1.146: The *Templates* dialog box.

This dialog box lists all **Stored templates** and allows the current template to be saved with **<Save current>** or a selected template from the list to be loaded with **<Load template>**. A selected template can be deleted with **<Delete selected>**.




A template can only be loaded if no alignment is present. To load a template, you will need to remove the assemblies first, which can be done with **Assembly > Delete all contigs...**

The sequence *Experiment card* window and *Sequence editor* window is filled with the assembled sequence as soon as the project is saved.

8.1.4 Exporting sequence data

With the **Export sequences** option, listed under the topic **Sequence type data** in the *Export* dialog box (see Figure 8.1.147), sequences can be exported in FASTA, EMBL or GenBank format.

In the *Database entries* panel of the *Main* window, select the entries to export. A single entry can be selected by holding the **Ctrl**-key and left-clicking (**CTRL+click**). Check boxes for selected entries are indicated as . In order to select a group of entries, hold the **Shift**-key and click on another entry. All the entries in the database can be selected using the keyboard shortcut **Ctrl+A** or with **Edit > Select all**.

Selecting **Export sequences** under **Sequence type data** in the *Export* dialog box and pressing **<Export>** brings up the *Export sequences* dialog box (see Figure 8.1.148).

In the *Export sequences* dialog box, all sequence types defined in the database are displayed in the **Sequence types** list and all database fields are shown in the **Header fields** list.

Browse for a new or existing export file with the **<Browse>** button on top.

Select the **Sequence types** and **Header fields** to export. To select multiple rows, hold the **Ctrl**-key on the

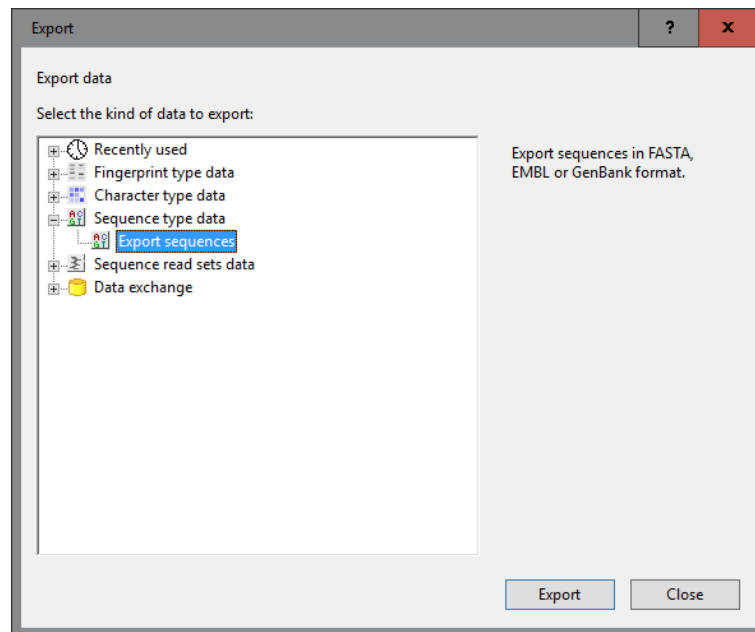
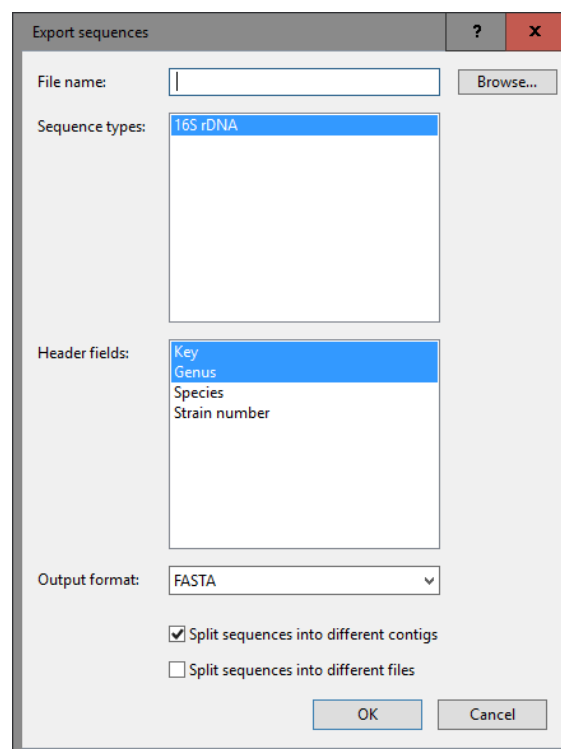


Figure 8.1.147: Export tree.

Figure 8.1.148: The *Export sequences* dialog box.

keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.

The *Output format* can be:

- **FASTA**: This option exports the sequences in FASTA format. Each sequence begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than ("**>**") symbol. The description line contains name of the *Sequence*

type and the selected **Header fields**, separated by a pipe (") symbol.

- **EMBL**: This option exports the sequences in EMBL format. The *features* (e.g. source, exon, CDS, ...) and *header tags* present in the *Annotation* panel and *Header* panel of the *Sequence editor* window are exported along. The name of the *Sequence type* and selected **Header fields** are stored as KW tags.
- **GenBank**: This option exports the sequences in GenBank format. The *features* (e.g. source, exon, CDS, ...) and *header tags* present in the *Annotation* panel and *Header* panel of the *Sequence editor* window are exported along. The name of the *Sequence type* and selected **Header fields** are stored as KEYWORDS.

When the option *Split sequences into different contigs* is checked, different contigs (separated by a pipe (") symbol in a sequence) are split into individual sequences. The contig number (contig1, contig2, etc.) is added to the KW/KEYWORDS tags with an EMBL/GenBank export and appears as first tag in the header line in case of a FASTA export.

When the option *Split sequences into different files* is checked, each sequence is saved into a separate file. The suffix "_1", "_2", "_3", etc. is added to the file name.

8.1.5 The Sequence experiment card

In case of a sequence type, holding the **Shift**-key while clicking on the colored dot in the *Experiment presence* panel opens the *Experiment card* window. Clicking on the colored dot of a linked sequence type without holding the **Shift**-key opens the *Sequence editor* window.

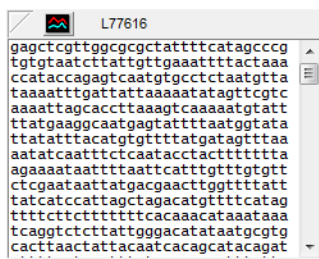




Figure 8.1.149: The Sequence *Experiment card* window.

When the sequence has been generated using the Assembler or Power Assembler program, pressing the  button will launch the (Power) Assembler to open the contig project associated with this sequence.

When no contig project is available for this entry, pressing the  button will launch Assembler with a new project associated. See 8.1.3.7 for instructions how to work with Assembler.

Clicking the right mouse button inside the card calls different edit functions. Sequence data can be pasted from the clipboard using *Paste from clipboard*. It is also possible to type bases directly from the keyboard. Right-clicking offers further editing tools: *Undo*, *Select all (Ctrl+A)* and *Copy to clipboard* to copy the current selection. With the option *Remove this experiment* the character set is deleted from the database. This is an irreversible operation.



When pasting a nucleotide or an amino acid sequence in the *Experiment card* window, please use the correct IUPAC codes and amino acid abbreviations respectively. Avoid the use of improper letters, symbols and spaces.

8.1.6 The Sequence Editor

8.1.6.1 Introduction

The *Sequence editor* window is a tool to edit, and analyze nucleotide and amino acid sequences. When sequences in GenBank or EMBL formats are imported in the BioNumerics database, the feature and header description fields available in these formats are recognized and displayed in the *Sequence editor* window.

Frame analysis, restriction enzyme analysis, and primer analysis can be executed from the *Sequence editor* window on an imported nucleotide sequence and require the presence of the Sequence types modules (SQ) in the BioNumerics configuration.

Clicking on the colored dot of a linked sequence type in the *Database entries* panel opens the *Sequence editor* window displaying the imported nucleotide or amino acid sequence in the upper panel for that entry (see Figure 8.1.150).

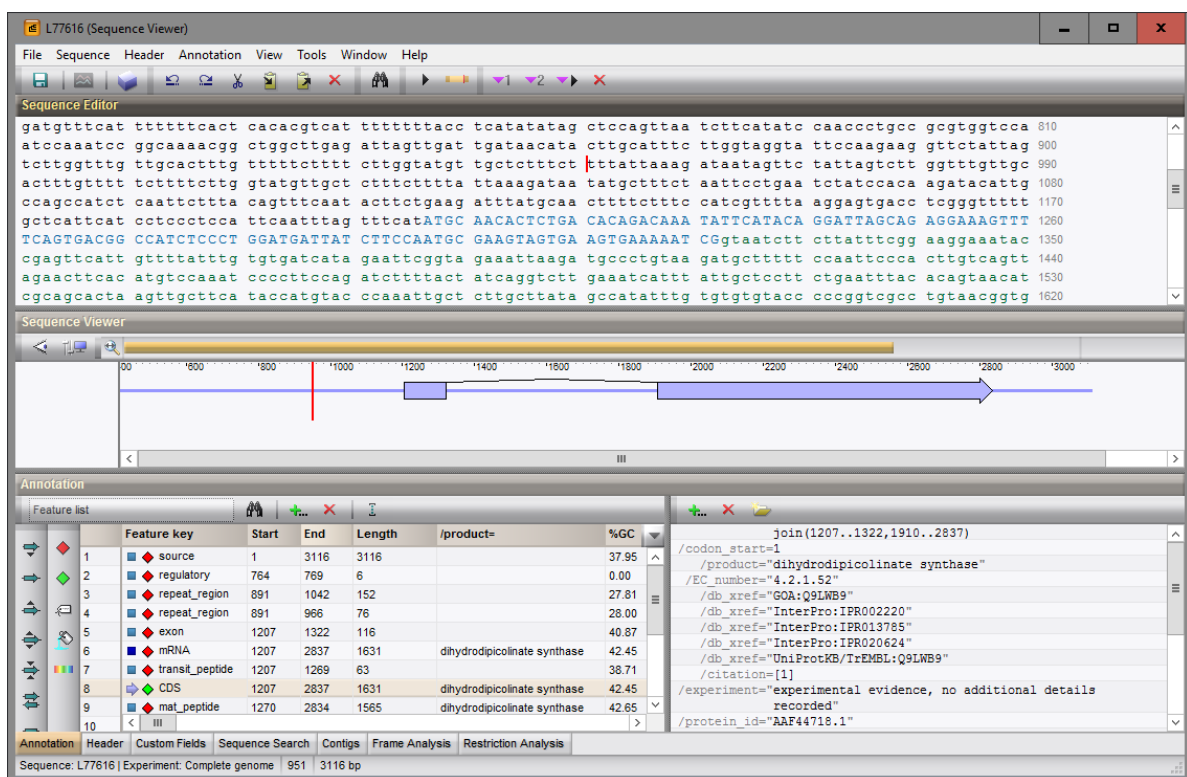


Figure 8.1.150: The *Sequence editor* window.

The *Sequence editor* window is divided in three panels:

The *Sequence Editor* panel shows the numbered sequence, with bases (or amino acids) grouped in blocks of 10. Non-translated sequence parts are displayed in black lowercase, exons in blue uppercase, and introns in green lowercase.

The *Sequence Viewer* panel shows a graphical representation of the sequence. The zoom slider allows one to continuously zoom in or out on the graphical sequence view. Zooming can be done up to base (amino acid) level. The red vertical line in the upper and middle panel indicates the cursor position on the sequence.

The first two tabs in the lower panel of the *Sequence editor* window describe the sequence by means of its EMBL/GenBank features (*Annotation* panel) and header information (*Header* panel). A sequence search can be executed (*Sequence Search* panel) and contig information is summarized in the *Contigs* panel. Each analysis tool that can be applied to a nucleotide sequence (frame analysis, restriction enzyme analysis, and

primer design) has its own tab in the lower panel, except for the primer design tool.

8.1.6.2 General functionality

8.1.6.2.1 Importing and exporting sequences

A sequence can be imported from the clipboard into an empty *Sequence editor* window with **Sequence** > **Paste** (📄, **Ctrl+V**).

To import a sequence in FASTA format from the clipboard into the *Sequence editor* window use **File** > **Import from clipboard** > **Import FASTA format from clipboard...** (📄). This calls the *Field conversions* dialog box (see Figure 8.1.151).

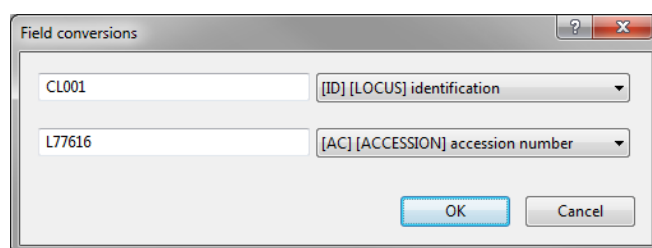


Figure 8.1.151: The *Field conversions* dialog box: two FASTA tags.

A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than (">") symbol. The description line contains the *FASTA tags*, separated by a pipe (|) symbol.

The *FASTA tags* are parsed from the description line and are displayed as separated rows in the *Field conversions* dialog box. The FASTA tags can be linked to any of the header tags using the drop-down list. FASTA tags linked to header tags are saved with the sequence and are displayed in the *Header* panel of the *Sequence editor* window.

To import a sequence in EMBL/GenBank format from the clipboard into the *Sequence editor* window use **File** > **Import from clipboard** > **Import Embl/GenBank format from clipboard...** (📄).

When pasting a nucleotide or an amino acid sequence in the *Sequence editor* window, please use the correct IUPAC codes and amino acid abbreviations respectively. Avoid the use of improper letters, symbols and spaces.

To launch the (Power) Assembler from the *Sequence editor* window, select **File** > **Open assembler** (🔧). Note that the Assembler or Power Assembler cannot be launched when the sequence displayed in the *Sequence editor* window is not linked to an Assembler or Power Assembler contig sequence project, respectively.

The sequence contained in *Sequence editor* window can be exported to the clipboard from where it can be pasted as text, e.g. in Notepad.

To export the sequence in EMBL/GenBank format to the clipboard select **File** > **Export to clipboard** > **Embl/GenBank format**.

The sequence can be copied in FASTA format to the clipboard with **File** > **Export to clipboard** > **FASTA format**.

The command **File** > **Export to file...** saves the sequence in EMBL/GenBank or FASTA format in a text file.

8.1.6.2.2 Selecting and editing sequences

Selecting **Sequence > Go to...** () calls the *Go to* dialog box.

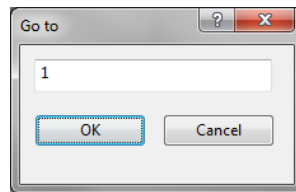



Figure 8.1.152: The *Go to* dialog box: update cursor position to position entered in the text box.

The cursor position in the upper two panels of the *Sequence editor* window will be updated based on the positions entered in the text box.

The red vertical line in the upper and middle panel indicates the cursor position on the sequence. Clicking on a new position in the upper or middle panel, or using the arrow keys changes the cursor position.

A selection in the upper two panels can be made using the mouse pointer: place the mouse pointer left or right from the region to be selected, click and hold down the left mouse button, while dragging the mouse pointer to the left or right and release the mouse button. The corresponding region is highlighted in the upper two panels. The complete sequence is selected with **Sequence > Select all (Ctrl+A)**.

Choosing **Sequence > Select subsequence...** ( , **Ctrl+G**) opens the *Select* dialog box.

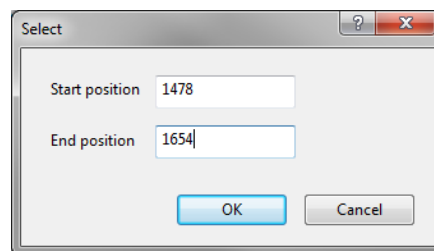



Figure 8.1.153: Select a region on the sequence.

The dialog prompts for the **Start position** and **Stop position**. The corresponding subsequence region will be selected in the upper two panels. A selection in the *Sequence editor* window can be copied to a new *Sequence editor* window with **File > Save selection...** ().

The dialog box prompts for the entry key and sequence type. The suggested entry key can be changed by the user.

The sequence will be stored in the database and a new *Sequence editor* window will open, displaying the selected region in the upper panels. Information on the sequence where the partial sequence was derived from, will be displayed in the comment field in the *Header* panel. The features located within the boundaries of the selected region will be transferred to the new *Sequence editor* window and will be displayed in the *Annotation* panel.

The sequence displayed in the *Sequence editor* window can be edited by the user in the upper two panels. Editing a sequence that is derived from an Assembler contig sequence project will force the contig project to adapt these changes. The software warns for this.

With **Allow only overwrite modus** checked, the base at the cursor position can be overwritten with another base or a gap, but bases can not be added or deleted. In this modus, editing a highlighted block is not possible.

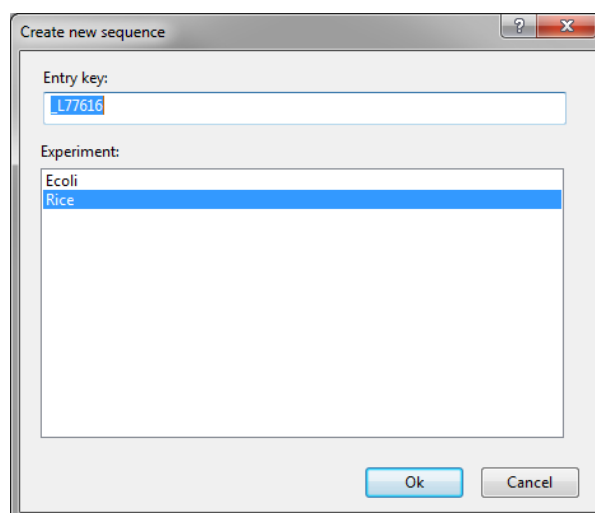


Figure 8.1.154: The *Create new sequence* dialog box: store a new sequence in the database.

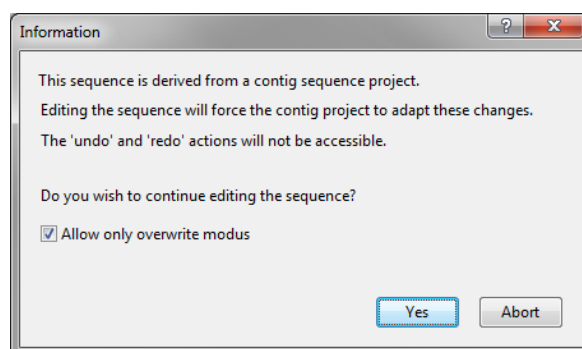


Figure 8.1.155: Editing a sequence derived from a contig project.

Sequence editing in the upper two panels is possible with following actions:

Move the cursor (red vertical line) using the mouse or by using the arrow buttons on the keyboard to a position on the sequence.

Type a character to insert a base (or amino acid).

Use the **Backspace** to delete the base (or amino acid) that is standing before the red vertical line.

Select **Sequence > Delete** (✂, Del) to delete selected nucleotide bases (or amino acids). If no selection is present, the base (or amino acid) that is standing after the red vertical line (cursor position) is deleted.

Replace a base (amino acid) by first deleting it and typing a new base (amino acid) instead.

To copy a selected region to the clipboard, use **Sequence > Copy** (📋, Ctrl+C).

With **Sequence > Cut** (✂, Ctrl+X), the selected region is removed from the sequence and copied to the clipboard.

With **Sequence > Paste** (📋, Ctrl+V), the sequence on the clipboard is pasted into the sequence at the position of the cursor. Start and stop *anchors* of a region of interest can be defined on the sequence.

Place the mouse pointer before the start position of the region of interest and select **Sequence > Anchors > Set start anchor** (📌) to specify a start anchor on the sequence. A pink triangle appears at the position of the cursor.

Place the cursor position after the end position of the region of interest and select **Sequence > Anchors >**

Set end anchor (📌) to specify a stop anchor on the sequence. A pink triangle appears at the position of the cursor.

When start and stop anchors are defined on the sequence, these positions automatically appear in the **Start position** and **Stop position** boxes when calling the *Select* dialog box.

To jump from one anchor to the other, use **Sequence > Anchors > Jump to next anchor** (📌➡). The cursor position in the upper two panels is updated.

Removing anchors from the sequence is done with **Sequence > Anchors > Remove anchors** (🗑).

For any changes made to the sequence - not derived from an Assembler contig sequence project - there is a multi-step undo function: select **Sequence > Undo** (↶, **Ctrl+Z**) to undo the previous action. To redo one or more actions use **Sequence > Redo** (↷, **Ctrl+Y**).

8.1.6.2.3 Viewing options

When the sequence, displayed in the *Sequence editor* window, is derived from an Assembler contig sequence project, the curves (chromatograms) of the individual trace files can be displayed in the *Sequence Viewer* panel with **View > Load curves**.

The curves (chromatograms) of the sequences can be displayed superimposed or on different lines. The display of the curves can be changed with **View > Figure display settings...**. This calls the *Display settings* dialog box.

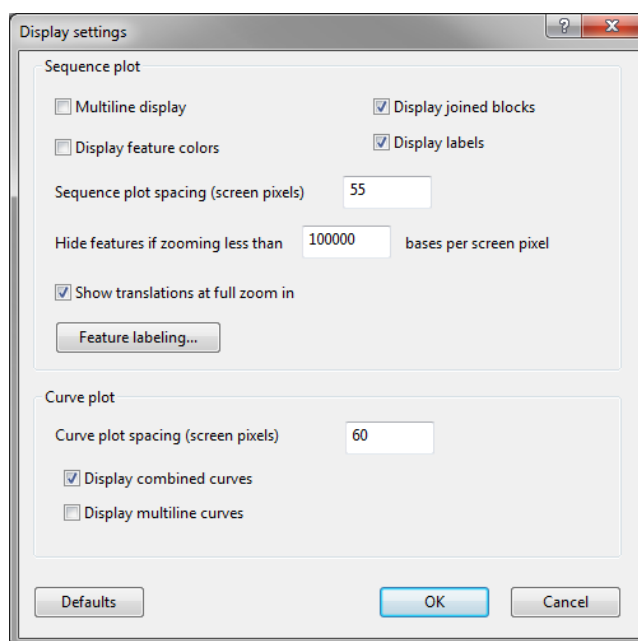


Figure 8.1.156: The *Display settings* dialog box.

If the **Multi-line display** option is checked in the *Sequence plot panel*, the alignment is wrapped into the width of the panel and displayed on more than one line. Features built up of joined sequence stretches (for example eukaryotic CDS features with exon-intron structures, disrupted genes, ...) can be mapped as joined blocks when enabling **Display joined blocks**. Unchecking this check box displays features built up of joined sequence stretches as one continuous block from start to end position.

Check **Display feature colors** to show different colors assigned to the features. Standard feature colors are shown if no feature colors have been defined on the sequence, or if the check box is unchecked. To display the labels for the visualized features on the plot check **Display labels**. The **Sequence plot spacing** (default 55) determines the distance between the lines of the sequences in the multi-line graphical overview

(in screen pixel units). The drawing of features on the plot by higher zoom out levels can be eliminated by decreasing the cut-off value entered in the field **Hide features if zooming less than**. A high cut-off value, for example 10.000.000 will result in feature drawing even at full zooming out, whereas a cut-off value of 0 will cause the features never to be drawn. When **Show translations at full zoom in** is checked, the amino acid translation of the CDS features is displayed in the *Sequence Viewer* panel below the nucleotide sequence if zoomed in up to base level.

Pressing the **<Feature labeling>** button brings up the *Feature labeling* dialog box. This dialog box can also be called with **Annotation > Feature labeling...**

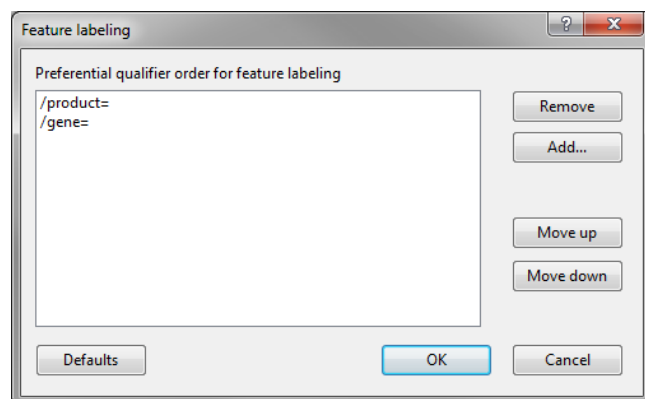


Figure 8.1.157: The *Feature labeling* dialog box.

The label that is shown below the displayed features in the *Sequence Viewer* panel is standard based on the list defined in this window. If for example, the qualifiers **"/product="**, **"/gene="** and **"/note="** are defined as preferential qualifiers for labeling (in this order), then the label of choice will be the text defined under the qualifier **"/product="** of the feature annotation. If a feature is not annotated with a **"/product="** qualifier label, the next qualifier of preference is taken, namely **"/gene="**, and so on. The list of preferred qualifiers for labeling can be edited as follows: the **<Remove>** button removes the currently selected qualifier from the list.

The **<Add>** button calls the *Qualifier types* dialog box.

A qualifier can be selected out of the EMBL-GenBank feature annotation table. Pressing the **<OK>** button adds the qualifier to the list of qualifiers.

With the **<Move up>** and **<Move down>** buttons, the order of the qualifiers within the list can be changed. The **<Defaults>** button resets the default labels and label order.

The **Curve plot spacing** (default 60) determines the distance between the lines of the curves of the trace files in the multi-line graphical overview. **Display combined curves** and **Display multi-line curves** display the curves (chromatograms) of the sequences respectively superimposed or on different lines. Pressing the **<Defaults>** button resets all parameters to their default value.

The options **Tools > Curves > Plot GC-content...** and **Tools > Curves > Plot GC-Skewing...** call the *Sequence Plot Settings* dialog box.

The *Sequence Plot Settings* dialog box prompts for following plot screening settings:

- **Window screening size:** the window size, expressed in number of base pairs, for which the average %GC is calculated.
- **Step screening progress:** the number of bases the screening window shifts before calculating the next windowed average %GC value on the sequence.

Default settings can be restored by pressing **<Reset to Default>**.

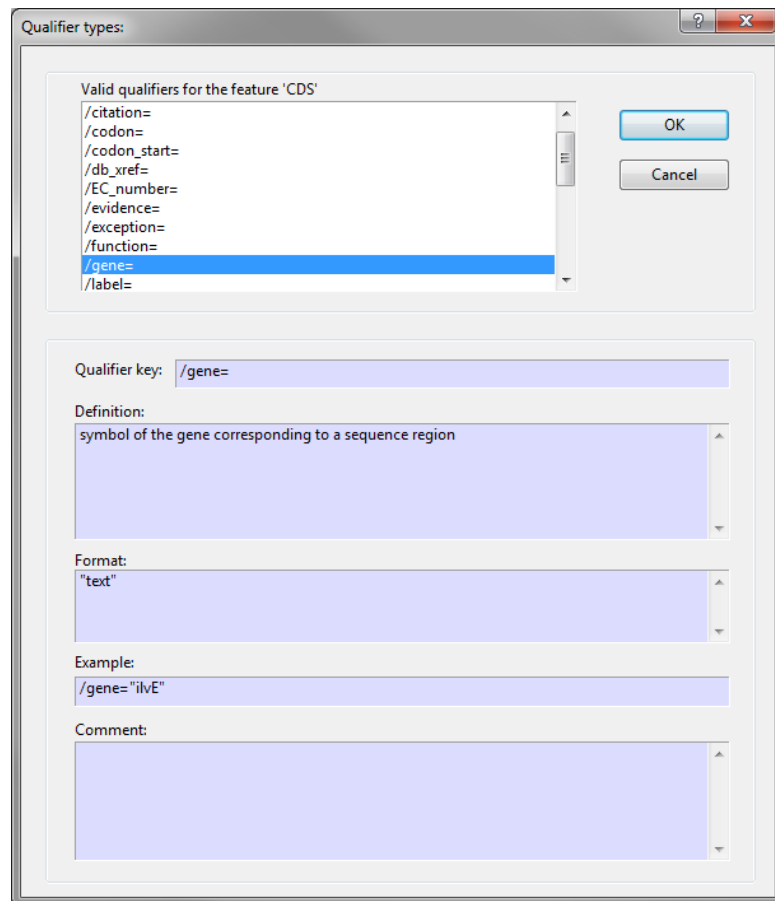


Figure 8.1.158: Add a new qualifier to the list.

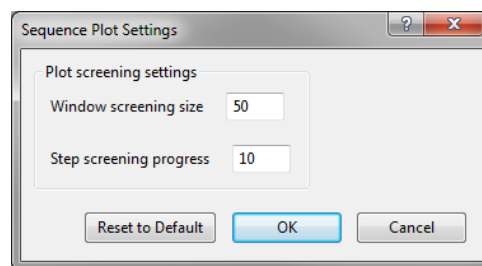


Figure 8.1.159: The *Sequence Plot Settings* dialog box.

The commands **Tools** > **Curves** > **Sequence base-coverage...** and **Tools** > **Curves** > **Sequence base-call quality...** plot the sequence coverage and sequence quality over the length of the sequence. The latter parameters are only available if the sequence was assembled from sequence reads in BioNumerics.

With the command **Tools** > **Curves** > **Plot non-uniqueness...**, a non-uniqueness curve is plotted. This curve has for each base a value that corresponds to the minimum length to have a unique sub-sequence around that base. The non-uniqueness curve can be used to detect problematic repeat regions that hinder resequencing and for defining position masks in the context of whole genome SNP analysis (see 8.10.2.4.3).

With **Tools** > **Curves** > **Create mask curve...**, a masking curve for wgSNP analysis can be created as explained in 8.10.2.4.3.

8.1.6.2.4 Saving changes to the database

Any changes made to a sequence in the *Sequence editor* window can be saved in the database with **File** > **Save** (📁, **Ctrl+S**). This action will save the sequence to the same sequence experiment (i.e. entry - experiment type combination). Alternatively, selecting **File** > **Save as...** (**Ctrl+Shift+S**) allows you to save the sequence and coverage information (if available) to another sequence experiment.

Select **File** > **Exit** to close the *Sequence editor* window. If unsaved changes are detected in the sequence, a dialog box pops up, prompting to save the changes in the database.

8.1.6.3 Annotation

8.1.6.3.1 Annotation panel

When sequences in GenBank or EMBL formats are imported in the BioNumerics database, or directly pasted from the clipboard into the *Sequence editor* window, the *feature* description fields available in these formats are recognized and interpreted by BioNumerics. All these recognized features are displayed in the *Annotation* panel of the *Sequence editor* window.

A selected feature is highlighted with an orange background in the upper and middle panels and is highlighted in the feature list in the lower panel. The *qualifiers* associated with the selected feature are given in the right panel.

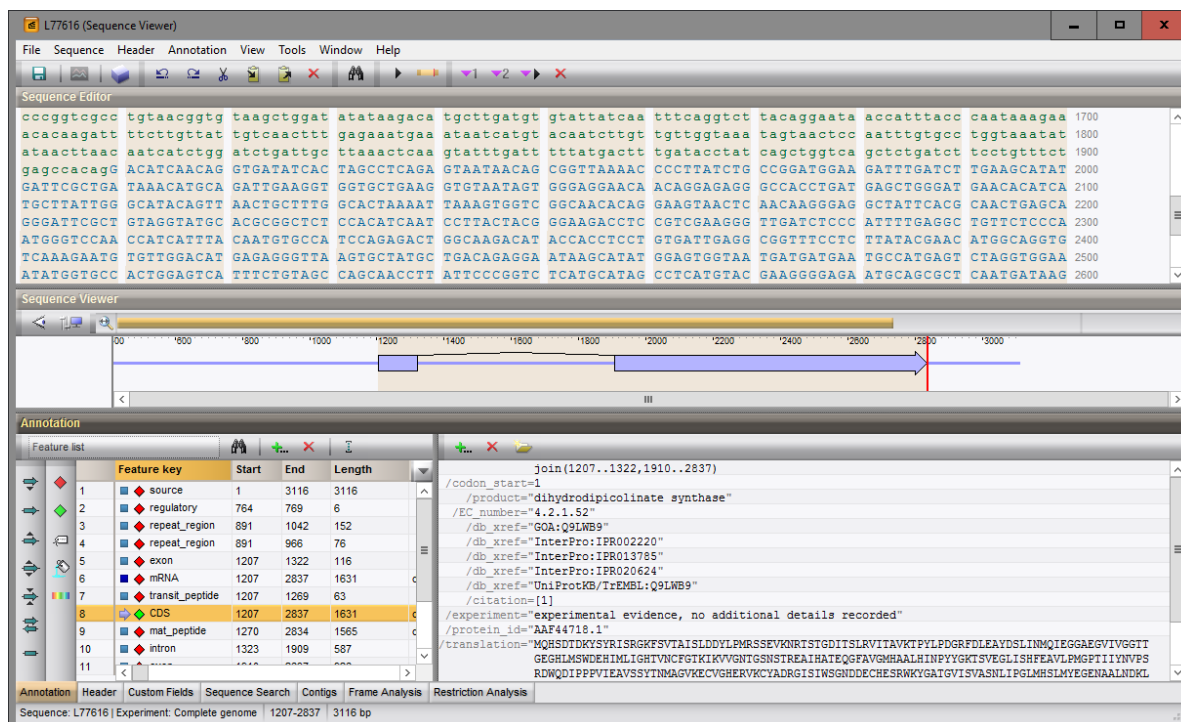




Figure 8.1.160: The *Annotation* panel with the features (left) and qualifiers (right).

8.1.6.3.2 Annotation features

When editing the sequences in one of the upper two panels, the feature (left panel) and qualifier information (right panel) is instantly updated by these actions.

Selecting more than one feature in the list is done by holding down the **Ctrl**-key and click on the feature. To select a range of features use the **Shift**-key.

In the feature list (left panel) four columns are shown by default. These include the name of the feature (**Feature key**), the **Start** and **End** position of the feature, and its **Length**. It is possible to display and hide columns by clicking on the  button in the header of the left lower panel. A menu appears, containing all available *qualifiers* which can either be displayed or hidden (the check marks respectively present or absent). The column separators can be dragged to the right or to the left in the header to reduce or increase space for each column.

Pressing the  button calls the *Find features* dialog box.

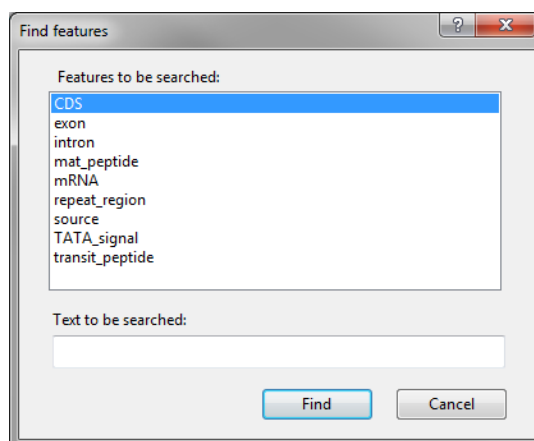



Figure 8.1.161: The *Find features* dialog box, to select features to be searched for.

This search function allows to display a specific set of features (for example only CDS features) or a set of features containing a specified selection text.

Standard all features are displayed (**Feature list**). Another listing can be displayed by selecting the listing from the drop-down list in the toolbar of the *Annotation* panel.




The command **Annotation** > **Add feature...** () calls the *Edit qualifier* dialog box.

Select a feature type from the list and choose the **Feature orientation**. Pressing <Next> brings you to the next step. The same dialog is called with **Annotation** > **Edit locations...**

The region selected in the *Sequence viewer* is shown as the fragment that corresponds to the new feature. Use the <Delete fragment>, <Add fragment> or <Edit fragment> buttons to remove, add, or edit a sequence fragment. To remove a feature, select the feature in the left panel and select **Annotation** > **Remove selected feature** (.

The nucleotide sequence of a selected feature can be copied to a new *Sequence editor* window with **Annotation** > **Create nucleotide sequence entry from selected feature...**. The *Create new sequence* dialog box prompts for the entry key and sequence type. The suggested entry key can be changed by the user.

Features that are marked with a green diamond in the **Feature key** column next are displayed on the graphical plot, whereas features that are marked with a red diamond are not displayed on the plot.

To display a selected feature on the sequence plot, select **Annotation** > **Feature layout tools** > **Show feature** (). Hiding features is done with **Annotation** > **Feature layout tools** > **Hide feature** (). A qualifier is added to a selected feature with **Annotation** > **Add qualifier...** (). The *Qualifier types* dialog box appears.

A qualifier can be selected out of the EMBL-GenBank feature annotation table. Pressing the <OK> button adds the qualifier to the list of qualifiers.

Qualifiers can be edited by double-clicking on the qualifier in the right panel. The information will appear highlighted and can be edited.

To remove a qualifier for a selected feature, select the qualifier and select **Annotation** > **Remove selected**

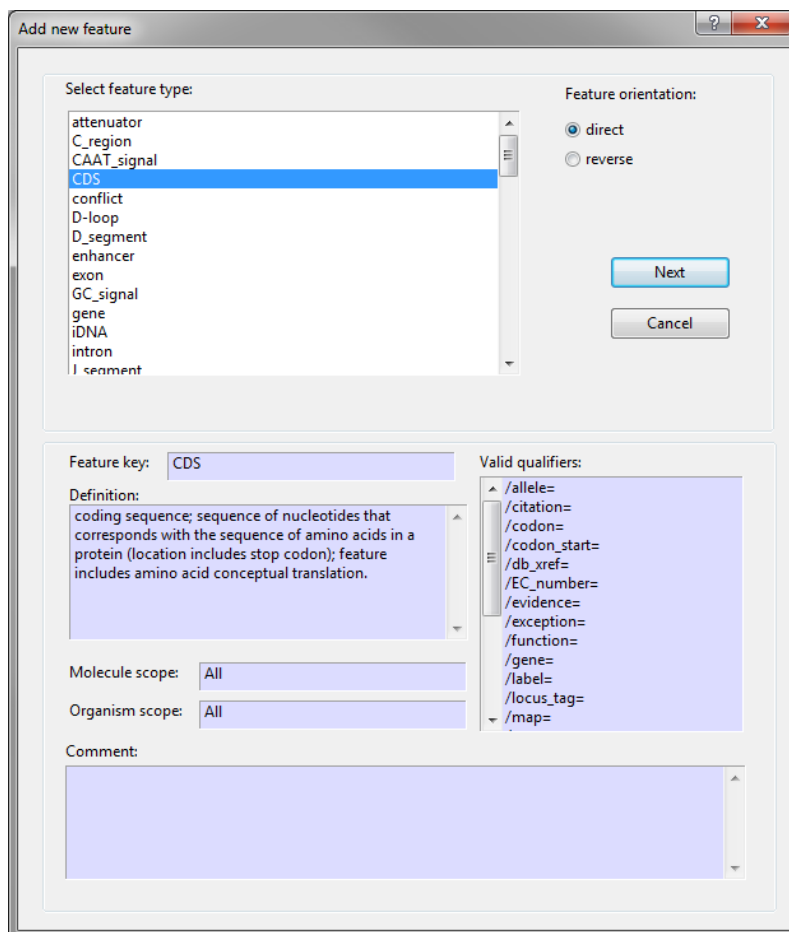


Figure 8.1.162: Add a new feature to the list.

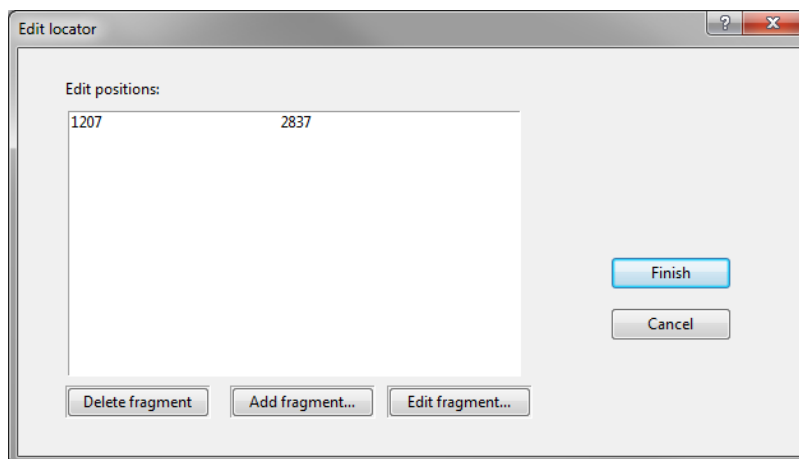


Figure 8.1.163: Choose the positions of the new feature.

qualifier (✖).

The qualifier **/translation=** holds the amino acid translation of a CDS feature. When **Show translations at full zoom in** is checked in the *Display settings* dialog box, the amino acid translation of the CDS features is displayed in the *Sequence Viewer* panel below the nucleotide sequence if zoomed in up to base level. The translation is shown below the nucleotide sequence of the CDS feature.

The translation of a selected CDS feature can be copied to a new *Sequence editor* window with **Annotation**

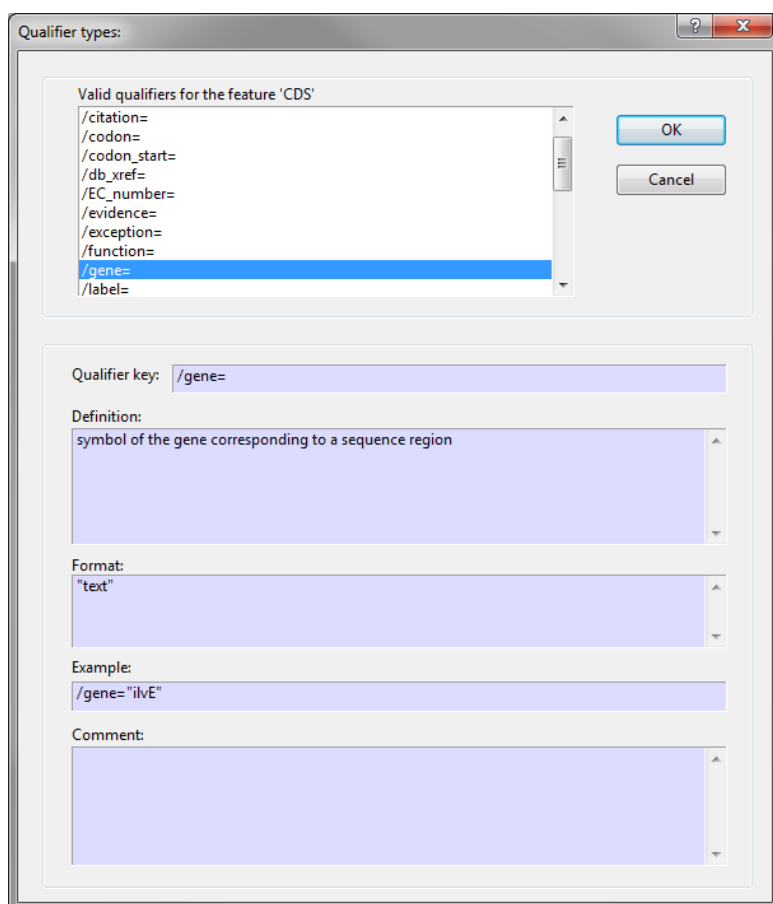


Figure 8.1.164: Add a new qualifier to the list.

> **Create protein entry from selected CDS...** (📁). The *Create new sequence* dialog box pops up listing all amino acid sequence types that are present in the database. The suggested entry key can be changed by the user.



A message is displayed if no amino acid sequence types are found in the database. In that event, create an amino acid sequence type in the database and repeat the action.

Features that are marked with a green diamond in the **Feature key** column next are displayed on the graphical plot, whereas features that are marked with a red diamond are not displayed on the plot.

To display a selected feature on the sequence plot, select **Annotation** > **Feature layout tools** > **Show feature** (🔍). Hiding features is done with **Annotation** > **Feature layout tools** > **Hide feature** (🔍).




The label that is shown below the displayed features in the middle panel is standard based on the list defined in the *Feature labeling* dialog box.



Changing the label layout of a selected feature is done with **Annotation** > **Feature layout tools** > **Show/hide feature label** (🏷️). A check sign marks which option is selected as current label layout for the selected feature. Default the **Preferential qualifier listing** is used. The pop-up window displays all qualifiers found in the selected feature. When a qualifier is selected from the list, the qualifier name is shown below the feature in the middle panel, if the feature is displayed on the plot. To display the feature without any label on the plot select the **No label** option from the list.





If several features are selected at the time (hold the **Ctrl**-key to do so), the pop-up window will display all types of qualifiers found in the selected features. Selecting a specific qualifier as new label will cause all selected features to take this qualifier as label, if this qualifier is present.

Selected features can be moved upwards or downwards with **Annotation** > **Feature layout tools** > **Move**


feature up () and **Annotation > Feature layout tools > Move feature down** () respectively. To center the selected features, use **Annotation > Feature layout tools > Set feature central** ()


The thickness of the selected features can be changed with **Annotation > Feature layout tools > Make feature thicker** () , making the feature thicker and **Annotation > Feature layout tools > Make feature smaller** () - making the feature smaller.

A direction arrow can be shown with **Annotation > Feature layout tools > Show feature direction** () . The command **Annotation > Feature layout tools > Hide feature direction** () shows the selected features without direction arrow.



Double-stranded features like repeats, replication origins, usually have no direction and cannot be marked with direction arrows.

To select a different color for the selected features, choose **Annotation > Feature layout tools > Set feature color...** () . From the pick list of basic colors that appears, select a specific color. Custom colors can be created by pressing **<Define Custom Colors>** and defining the color in RGB components.

Another tool, which can be of particular interest for large sequences with many features, is the color randomizer. When selecting the option **Annotation > Feature layout tools > Randomize feature color** () each selected feature gets a random color.

The qualifier **/translation=** holds the amino acid translation of a CDS feature. When **Show translations at full zoom in** is checked in the *Display settings* dialog box, the amino acid translation of the CDS features is displayed in the *Sequence Viewer* panel below the nucleotide sequence if zoomed in up to base level. The translation is shown below the nucleotide sequence of the CDS feature.

The translation of a selected CDS feature can be copied to a new *Sequence editor* window with the command **Annotation > Create protein entry from CDS**. The *Create new sequence* dialog box pops up listing all amino acid sequence types that are present in the database. The suggested entry key can be changed by the user.



A message is displayed if no amino acid sequence types are found in the database. In that event, create an amino acid sequence type in the database and repeat the action.


8.1.6.4 Header

8.1.6.4.1 Header panel

When sequences in GenBank or EMBL formats are imported in the BioNumerics database, or directly pasted from the clipboard into the *Sequence editor* window, the **header** description fields available in these formats are recognized and interpreted by BioNumerics. These features are displayed in the *Header* panel of the *Sequence editor* window.

Each header line is characterized by a tag (light gray) and the content line(s). The header lines **ID** and **AC** in the EMBL format, and **LOCUS** and **ACCESSION** in the GenBank format, are protected lines which cannot be changed. When double-clicking on such a protected line, a box with the message "Protected line!" appears. When double-clicking on a non-protected header line, the field will appear highlighted and can be edited. Pressing **ENTER** saves the information.

8.1.6.4.2 Header features

To add new header lines to a sequence, first select the line after which you want to insert the new line, and choose **Header > Add line...** () . This calls the *Add line* dialog box.

The name tags for the header lines are displayed according to the stored format (GenBank or EMBL). Pressing **<OK>** adds the new header line to the sequence.

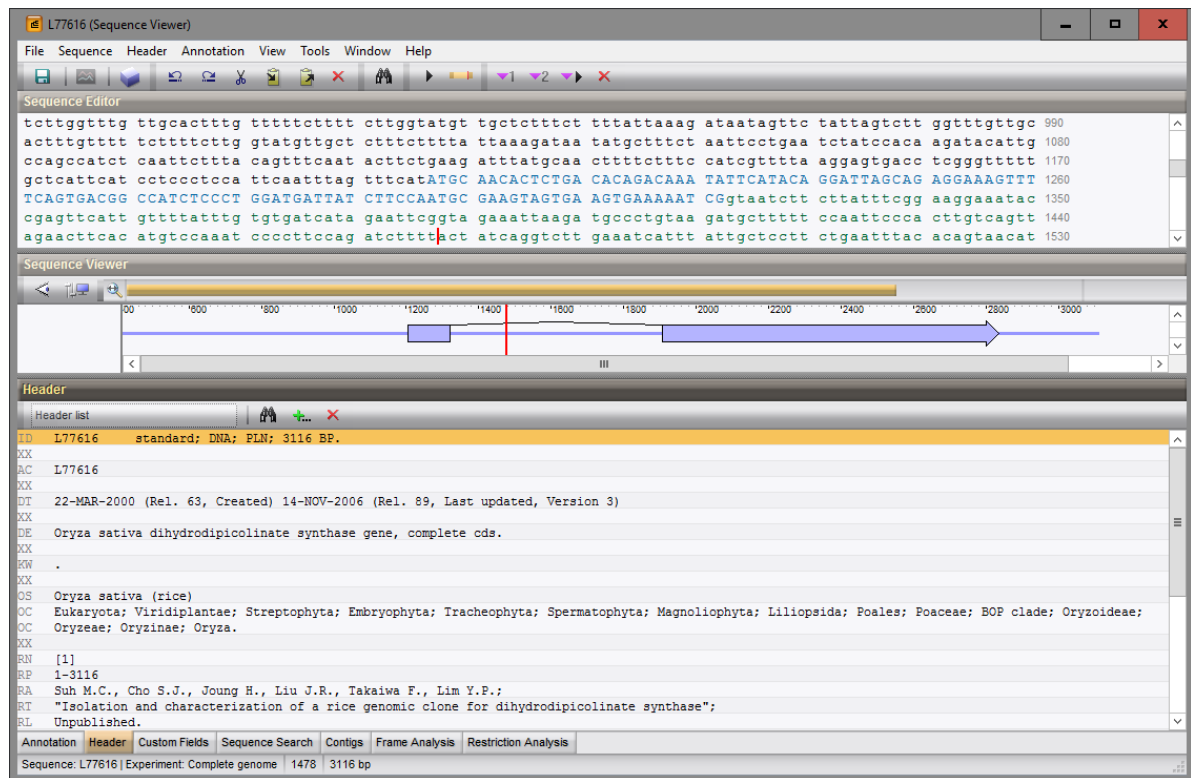
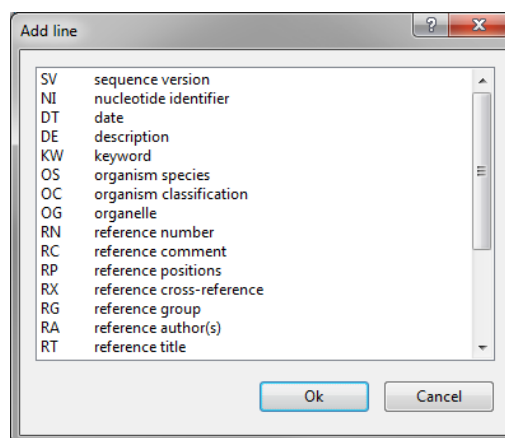
Figure 8.1.165: The *Header* panel with the header lines.

Figure 8.1.166: Add a new header line.

Non-protected header lines can be removed from the sequence with **Header > Remove selected line** (✖).

Selecting **Header > Find...** (🔍) calls the *Find features* dialog box.

A search text can be specified in the input field. If **Match case** is checked, the search will be case-sensitive.

Standard all header lines are displayed (**Header list**). Another listing can be displayed by selecting the listing from the drop-down list in the toolbar of the *Header* panel.

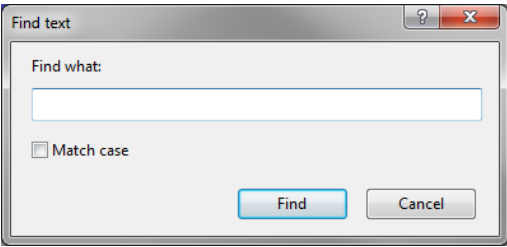


Figure 8.1.167: Specify a search text.

8.1.6.5 Sequence Search

8.1.6.5.1 Sequence Search panel

In the *Sequence Search* panel one can search for subsequences. The positions of the matching subsequences with the query sequence are shown in the *Position* column, and the number of mismatches are indicated in the *Mismatch* column. Subsequences found in the sequence are indicated by a blue arrow in the *Direction* field, subsequences found in the complementary strand are marked with a red arrow.

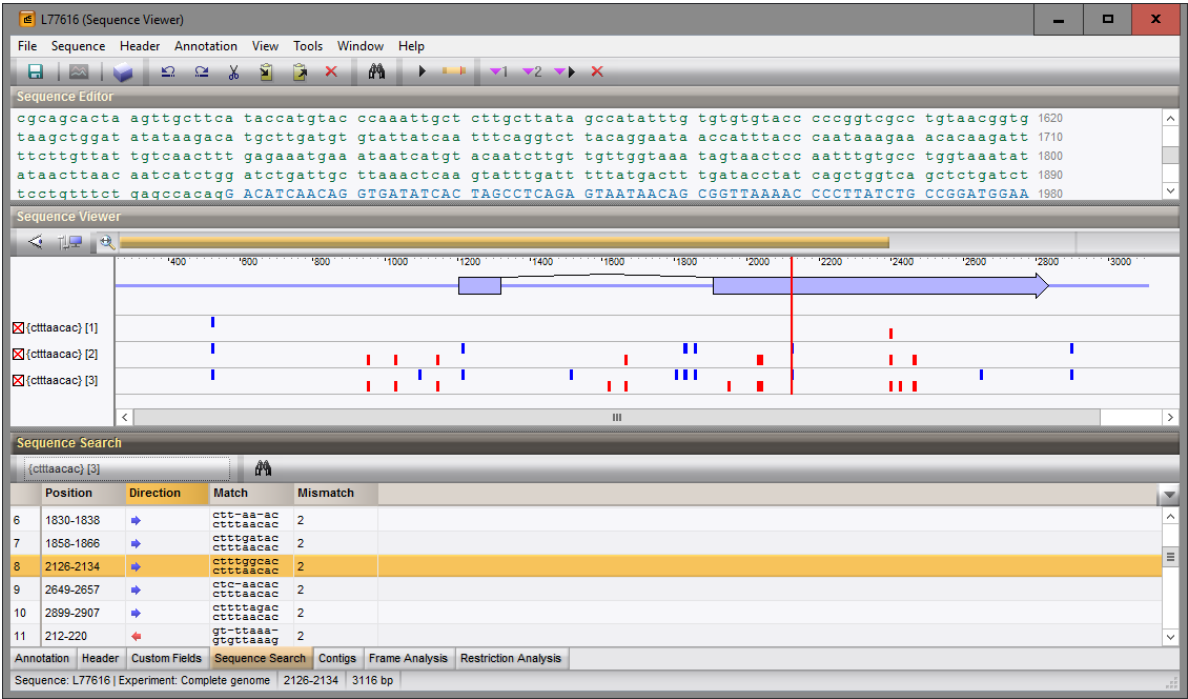


Figure 8.1.168: The *Sequence Search* panel.

8.1.6.5.2 Sequence Search features

Selecting *Sequence > Find sequence...* () calls the *Find features* dialog box.

In the *Subsequence* text box the query sequence needs to be provided. The *Number of mismatches allowed* can be specified using the spin control, or by entering the value in the input field. A query sequence can also be matched against the target sequence with the introduction of one or more gaps (*Allow gaps*). When checking the option *Allow gaps*, the mismatch tolerance is automatically set to "1", since a gap is considered equal to a mismatch. With a mismatch tolerance set to 2, and introduction of gaps enabled, matches will thus be found with either 2 mismatches, one mismatch and a gap, or two gaps (if all possibilities occur). The

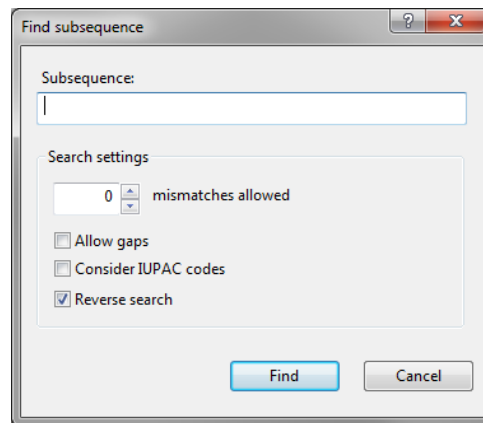



Figure 8.1.169: Search for subsequences.

IUPAC code for ambiguous bases is supported when *Consider IUPAC codes* is checked. When the option *Reverse search* is checked, the invert-complemented sequence will be searched as well. If more than one sequence search is present, the results of a previous search can be displayed again by selecting the search from the drop-down list in the toolbar of the *Sequence Search* panel.

When a match is selected, the corresponding region is highlighted in the upper two panels and the cursor position is updated.

Standard all matches of the sequence search(es) are displayed in the *Sequence Viewer* panel. Matches found in the sequence are indicated with a blue bar below the plot, matches found in the complementary strand are marked with a red bar.

The matches of a search can be removed from the plot by clicking on the red cross next to the search name in the *Sequence Viewer* panel. All matches are removed from the plot (or added to the plot) with the  button.

8.1.6.6 Contig information

Different contig sequences can be imported and saved into one single sequence in BioNumerics. The individual sequences are separated by a pipe (|), visible in the sequence experiment card and in the *Sequence Editor* panel of the *Sequence editor* window (see Figure 8.1.170).

Information about the individual sequences is present in the *Contigs* panel (see Figure 8.1.170). This includes the *Start* and *End* position of each sequence and the sequence *Length*.

8.1.7 The Genome viewer

8.1.7.1 Introduction

The *Genome Viewer* window integrated in BioNumerics is a visualization tool for interactive exploration of (large) sequences. Selecting **View > Genome viewer...** in the *Sequence editor* window opens the *Genome Viewer* window (see Figure 8.1.171).

The *Genome Viewer* window is divided in three panels:

The *Genome* panel shows the graphical representation of the sequence. The circular representation of the sequence is the default view.

The *Tracks* panel gives an overview of the information available in the *Sequence editor* window that can be

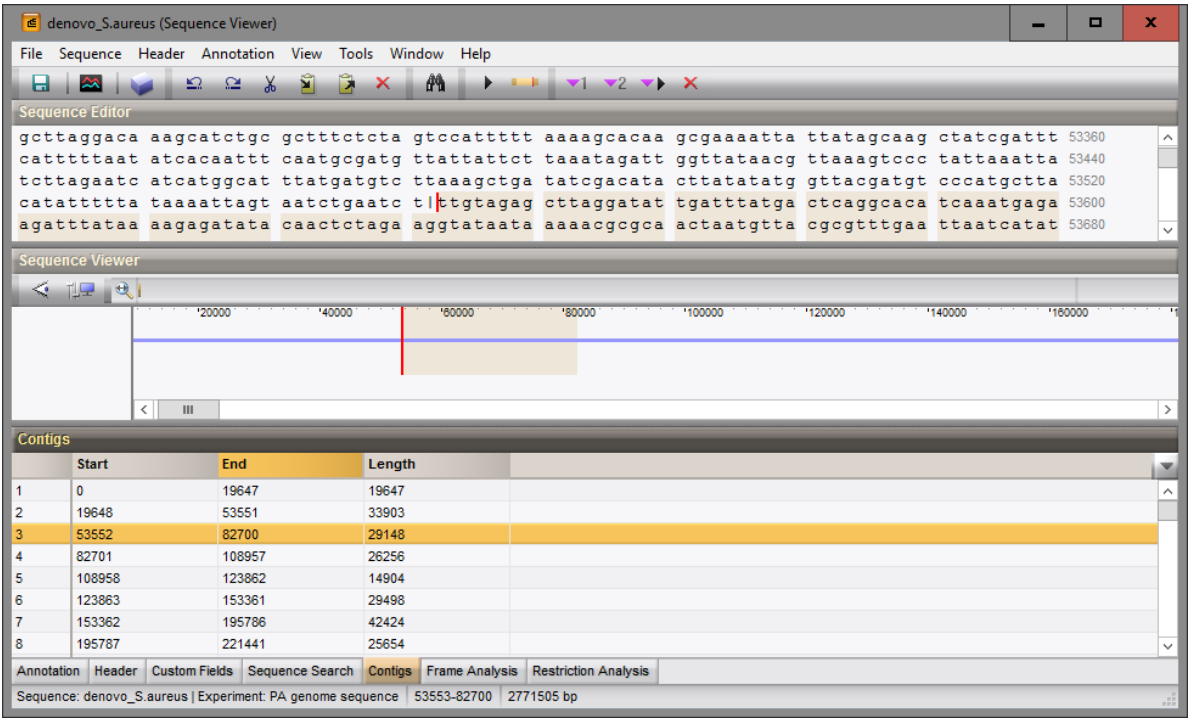


Figure 8.1.170: The *Contigs* panel.

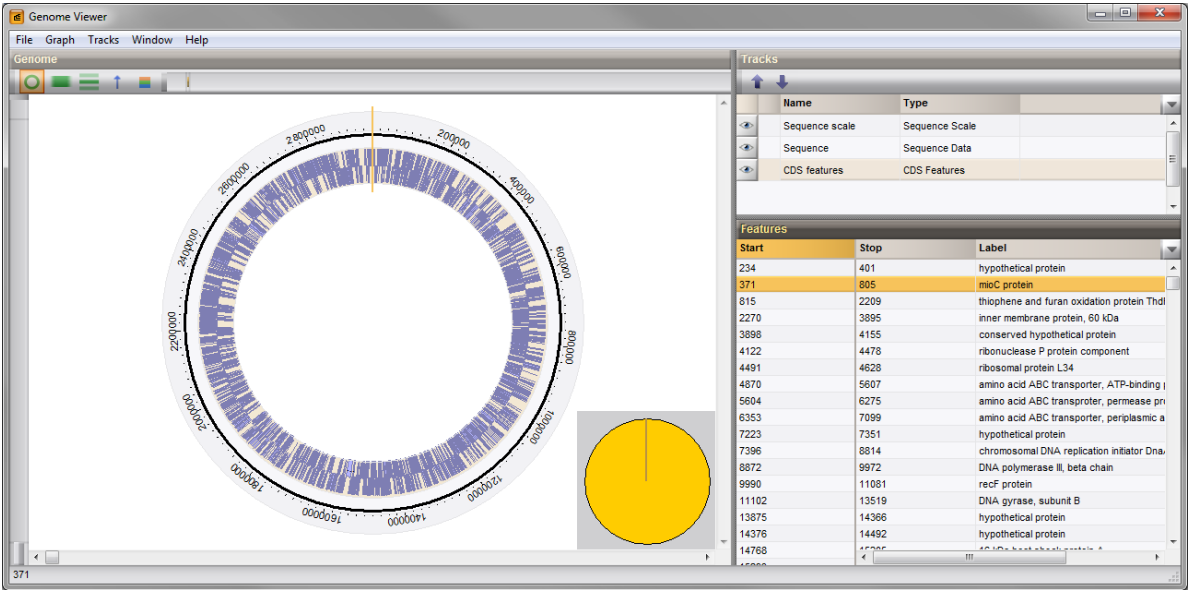


Figure 8.1.171: The *Genome Viewer* window.

plotted on the graph.

Depending on the *track* that is highlighted in the *Tracks* panel, the features (if any) of the selected track are displayed in the *Features* panel.

8.1.7.2 The *Genome* panel

The *Genome* panel shows the graphical representation of the sequence. The circular representation of the sequence is the default view.

With the zoom slider – located next to the toolbar – one can zoom in or out on the sequence. Alternatively one can use the mouse wheel or the + and - keys on the keyboard. When zooming in on the circular sequence, zooming is done on the upper area of the circular sequence.

Zooming can be done up to base level. The bases are colored based on following color scheme: green - A, blue - C, red - T, black - G, and gray for any IUPAC code denoting ambiguous positions (see Figure 8.1.172). The base numbers shown on top of the sequence correspond to the base numbering as used in the *Sequence editor* window.

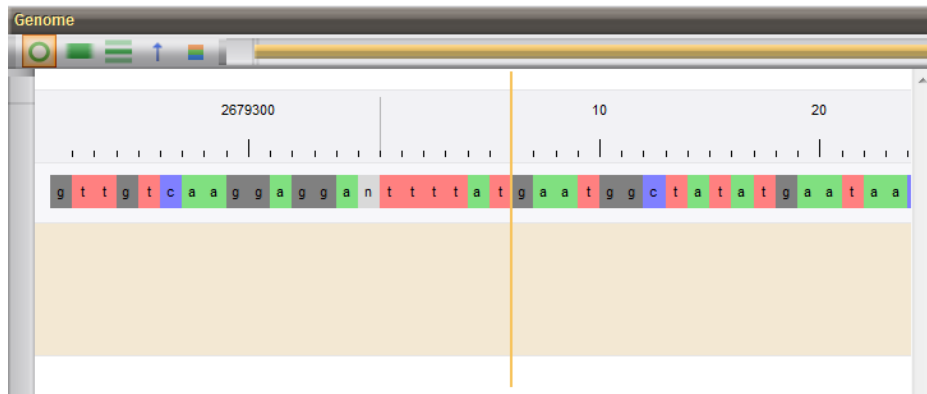


Figure 8.1.172: Zooming up to base level.

A zooming area can be specified with **Graph > Set view range....** This action calls the *Set view range* dialog box (see Figure 8.1.173).

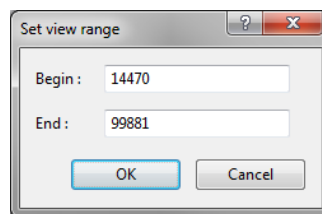


Figure 8.1.173: The *Set view range* dialog box.




The start (**Begin**) and stop (**End**) positions of the zoom area are prompted for. Pressing the **<OK>** button, updates the visible sequence part in the *Genome* panel based on the entered positions.

The gray vertical line on the circular map corresponds to the start position of the sequence (see Figure 8.1.172). The circular sequence can be rotated by holding down the left mouse button while dragging the mouse. With **Graph > Reset cursor** (↑) the circular sequence is rotated back to its original representation, i.e. with the start position located at the top of the map.

The cursor position is visible as an orange vertical line on the sequence. Double-clicking on a position on the circular map, rotates the map by placing the selected position at the top of the map. The cursor can be extended to cover a range of bases by holding down the **Shift**-key while selecting a position with the mouse.

The cursor position can be moved using the left and right arrow keys on the keyboard. In combination with the **Ctrl**-key this results in larger jumps. Using the **Home** button the cursor is placed at the start of the sequence. The end of the sequence is selected when the **End** button is pressed.

A *miniature map* is displayed below the circular sequence, representing the entire circular sequence present in the *Genome* panel. The portion of the sequence currently visible in the *Genome* panel is highlighted with a white color on the mini map, showing the relative position of the visible sequence to the entire sequence. To hide the mini map, click on the arrow in the left upper corner of the mini map. Un-hiding the map is done by clicking on the arrow again.



The portion of the sequence currently visible in the *Genome* panel can be displayed as a linear sequence using the option **Graph > Linear** (). With **Graph > Multi-line** () the complete sequence is wrapped into the width of the *Genome* panel and is displayed on more than one line. To go back to the circular representation, use **Graph > Circular** (.



8.1.7.3 The Tracks and Features panels


The *Tracks* panel gives an overview of the information that can be displayed on the sequence in the *Genome* panel. Selecting a track is done by clicking on the track in the *Tracks* panel. Alternatively one can use the **Up** and **Down** arrows, to change the selection. The selected track is highlighted with an orange background and the associated information (if any) is updated in the *Features* panel.

The *Sequence Data* and *Sequence Scale* tracks are available for every sequence. Other tracks might be present, depending on the information available in the underlying *Sequence editor* window:

- The *CDS Features* track is listed if at least one CDS feature is present in the *Annotation* panel. Selecting this track in the *Tracks* panel will display all CDS features in the *Features* panel with their **Start** and **Stop** position on the sequence and their qualifier label name as defined in the *Feature labeling* dialog box (**Label**). Clicking on a CDS in the *Features* panel will update the cursor selection on the map. The CDS features are plotted as blue arrows on the map.
- A *Sequence Search* track is listed for each sequence search performed in the *Sequence Search* panel. Selecting a sequence search track in the *Tracks* panel will display the sequence search results in the *Features* panel, each with their **Start** and **Stop** position on the sequence and sequence search pattern (**Label**). Clicking on a sequence match in the *Features* panel will update the cursor selection on the map. Matches found on the sequence are indicated with a blue rectangle on the map, while matches found in the complementary strand are marked with a red rectangle.
- For each restriction analysis performed in the *Restriction Analysis* panel a *Restriction Enzyme Search* track is listed. Selecting a restriction enzyme search track in the *Tracks* panel will display the cut positions in the *Features* panel together with the name of the enzyme (**Label**). Clicking on a feature in the *Features* panel will update the cursor selection on the map. The recognition sites, cut positions and name of the enzyme are plotted on the map.
- A *Curve* track is listed for each plot calculated in the *Sequence editor* window (**Tools > Curves**). Selecting a plot track in the *Tracks* panel will highlight the track in the *Genome* panel.

The order of tracks in the *Tracks* panel reflects the way this information is displayed in the *Genome* panel. The order of the tracks can be changed using the **Tracks > Move up** () and **Tracks > Move down** () options.

Default, the information of all *tracks* is shown on the sequence (). Clicking on the  icon next to a track will hide the track from the map.

With **Graph > Toggle channel color display** () , all tracks are assigned a different color (see Figure 8.1.174). This makes it easy to detect the different tracks on the graphical representation at a glance.

8.1.7.4 Export options

The image can be exported with **File > Export...**

This dialog box allows you to export graphical information to a file or to the Windows clipboard in one of several available formats. In case a file is exported, a file **Name** should always be entered or browsed for via the **<Browse>** button. Exported files will open in their default editor. Information on the Windows clipboard can be pasted into other applications. Following export options are available:

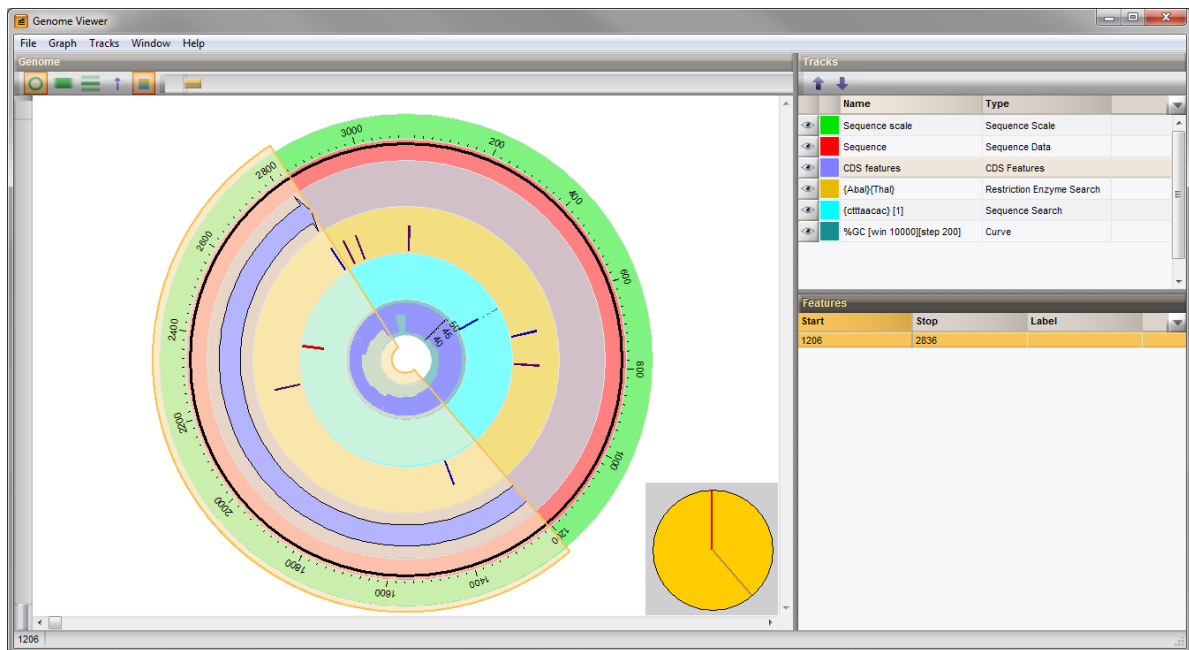
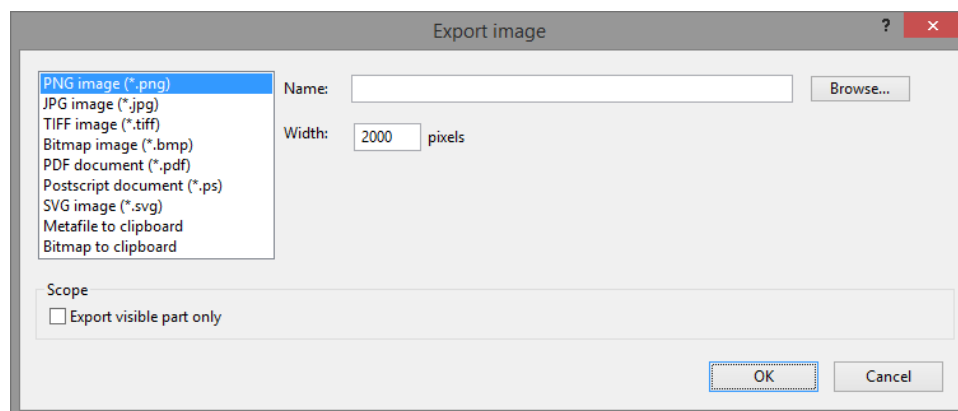


Figure 8.1.174: Tracks displayed in color.

Figure 8.1.175: The *Export image* dialog box.

- **PNG image (*.png)**: exports to a Portable Network Graphics (PNG) file. PNG is a raster graphics file format that supports lossless data compression. A **Name** and **Width** (in pixels) should be specified; the height will be determined automatically.
- **JPG image (*.jpg)**: exports to a Joint Photographic Experts Group (JPEG) file. JPEG or JPG is a raster graphics file format that uses a lossy data compression. A **Name** and **Width** (in pixels) should be specified, as well as a **Quality** parameter. With the latter, a tradeoff can be obtained between storage size and image quality.
- **TIFF image (*.tiff)**: exports to a Tagged Image File Format (TIFF) file. TIFF is a raster graphics file format with optional lossless data compression. A **Name** and **Width** (in pixels) should be specified.
- **Bitmap image (*.bmp)**: exports to a BMP bitmap image or device independent bitmap (DIB) file. BMP is a raster graphics image file format used to store bitmap digital images, independently of the display device. A **Name** and **Width** (in pixels) should be specified.
- **PDF document (*.pdf)**: exports to a Portable Document Format (PDF) file. PDF is a file format used to present documents in a manner independent of application software, hardware, and operating


systems. A *Name* and the *Orientation* (either Landscape or Portrait) should be specified.

- **Postscript document (*.ps)**: exports to a PostScript (PS) file. PostScript is a computer language for creating vector graphics. A *Name* and the *Orientation* (either Landscape or Portrait) should be specified.
- **SVG image (*.svg)**: exports to a Scalable Vector Graphics (SVG) file. SVG is an XML-based vector image format for two-dimensional graphics. A *Name* should be specified.
- **Metafile to clipboard**: copies the graphics as Windows enhanced metafile to the clipboard. Enhanced metafile is the standard clipboard exchange format between native Windows applications.
- **Bitmap to clipboard**: copies the graphics as a bitmap to the Windows clipboard. The *Width* (in pixels) should be specified.

Chapter 8.2

Analysis tools for individual sequences

8.2.1 Introduction

Frame analysis, restriction enzyme analysis, and primer analysis can be executed from the *Sequence editor* window on a nucleotide sequence and require the presence of the Sequence data module () in the BioNumerics configuration.


In the *Sequence editor* window of each nucleotide sequence a separate tab is present in the lower panel for each analysis tool, except for the primer design tool.

8.2.2 Frame analysis

8.2.2.1 Frame Analysis panel

The frame analysis function in the *Sequence editor* window analyzes a nucleotide sequence in function of its six possible translation frames.

Upon selecting the *Frame Analysis* panel, BioNumerics analyzes the nucleotide sequence that is present in the *Sequence editor panel* in function of its six translation frames.

Mapping the reading frames on the sequence plot is done with **Tools > Frame analysis > Show frame analysis in sequence viewer** () . Repeating this action will remove the frames again from the plot.

The middle panel shows a graphical view of the sequence along with its 6 possible reading frames. The 3 reading frames of the forward strand are mapped above the sequence plot, the 3 reading frames of the reverse strand are mapped below the sequence plot.

The zoom slider in the *Sequence Viewer* panel allows zooming from full-length sequence view up to base level view.

8.2.2.2 Frame Analysis features

Two items are of interest when considering reading frames of a sequence:

(1) An open reading frame (ORF), which is a sequence stretch that produces a continuous translation free of stop codons when translated into amino acid code,

and

(2) A protein coding sequence (PCS), which is a sequence stretch that produces a continuous translation free of stop codons and starts with a codon that serves as initiation of translation (for example methionine).

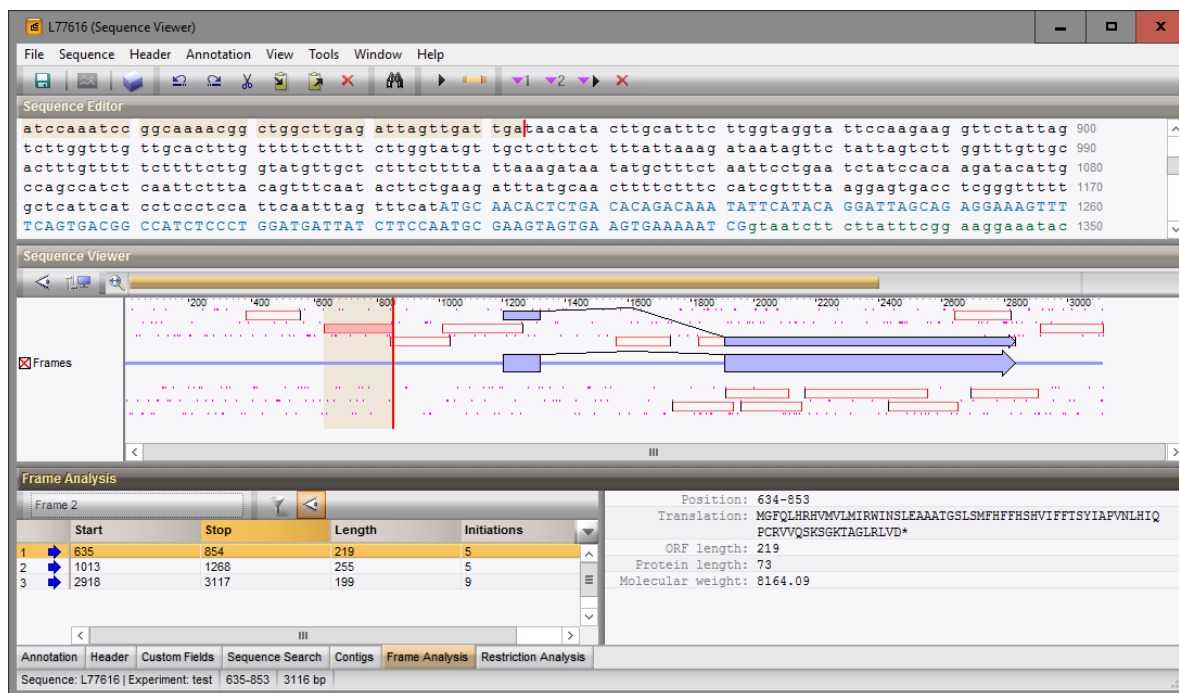


Figure 8.2.1: The *Frame Analysis* panel.



An ORF is a PCS, if the ORF contains an initiation of translation. A PCS is always an ORF, but will often be smaller in length.

The right panel in the *Frame analysis* tab shows details about the currently selected ORF or PCS in the left listing: **Position** on the sequence, **Translation**, size of the nucleotide sequence stretch (**ORF length**), size of the corresponding translation product (**Protein length**), and protein **Molecular weight** data.

When editing the sequences in one of the upper two panels, the settings in both panels in the *Frame analysis* tab are instantly updated.

When mapping the reading frames on the sequence plot, the ORF stretches are default plotted in the *Sequence Viewer* panel on the reading frames. The ORF stretches are drawn on the reading frames as blue boxes with black lines marking the upstream and downstream stop codons of the ORF. In the left panel of the *Frame Analysis* panel, all open reading frame fragments are listed for the currently selected reading frame (default **Frame 1**).

The **Start** and the **Stop** position, the fragment **Length** and the number of **Initiations** are shown next to each fragment. A blue arrow is displayed next to the fragments of the forward reading frames if the fragments corresponds to a PCS. A red arrow is displayed next to the fragments of the reverse reading frames, if the fragments correspond to a PCS.

The ORF fragments of another reading frame can be displayed in the fragment panel by selecting the frame from the drop-down list in the toolbar of the *Frame Analysis* panel.

In the fragment list, the reading frame and ORF selection focus are automatically updated when another ORF stretch is selected in the *Sequence Viewer* panel.

Double-clicking with the mouse pointer on an ORF in the *Sequence Viewer* panel results in the selection of the ORF fragment (stop-to-stop codon).

To plot the PCS stretches in the *Sequence Viewer* panel, select **Tools** > **Frame analysis** > **Filter settings...** (🔍) and check **Show only protein coding sequences**.

The **PCS stretches** are drawn on the reading frames as red boxes with black lines marking the stop codons.

In the left panel of the *Frame analysis tab*, all protein coding sequences are listed for the currently selected reading frame (default **Frame 1**). The **Start** and the **Stop** position, the fragment **Length** and the number of **Initiations** are shown next to each fragment. Blue arrows are displayed next to the PCS fragments of the forward reading frames, red arrows are displayed next to the PCS fragments of the reverse reading frames.

The PCS fragments of another reading frame can be displayed in the fragment panel by selecting the frame from the drop-down list in the toolbar of the *Frame Analysis* panel.

In the fragment list, the reading frame and PCS selection focus are automatically updated when another PCS stretch is selected in the *Sequence Viewer* panel.

Double-clicking with the mouse pointer on a PCS in the *Sequence Viewer* panel results in the selection of the PCS fragment (start-to-stop codon).

The right panel in the *Frame analysis tab* shows details about the currently selected ORF or PCS in the left listing: **Position** on the sequence, **Translation**, size of the nucleotide sequence stretch (**ORF length**), size of the corresponding translation product (**Protein length**), and protein **Molecular weight** data.

When editing the sequences in one of the upper two panels, the settings in both panels in the *Frame Analysis* panel are instantly updated.

Selecting **Tools** > **Frame analysis** > **Filter settings...** (🔧) opens the *Frame analysis settings* dialog box.

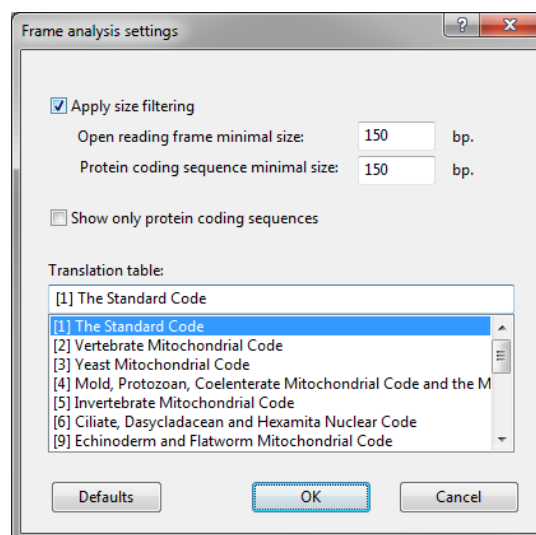


Figure 8.2.2: The *Frame analysis settings* dialog box.

As one will mostly be interested in ORF or PCS stretches with a considerable length, cut-off sizes for open reading frames and protein coding sequences can be specified when enabling the check box **Apply size filtering** (default enabled). The minimal sizes for open reading frames and protein coding sequences can be specified with **Open reading frame minimal size** and **Protein coding sequence minimal size**, respectively. Stretches smaller than these cut-off values are not considered and not mapped on the figure. Standard, ORF and PCS fragments with a minimal size of 50 bp are shown on the plot.

Upon selecting the *Frame analysis tab*, BioNumerics analyzes the nucleotide sequence in function of its six translation frames using the default translation table (**The Standard Code** translation table). The translation table to be used for analysis can be changed in the lower panel of the dialog. The analysis is automatically updated if a new table is selected from the list.

To help finding true protein encoding regions (CDS features), organism-specific codon usage can be displayed along the 6 possible frames. Codon usage tables are tables indicating the preferred use of specific codons within protein encoding sequences. In *Escherichia coli*, for example, glycine is encoded by four codons, GGG, GGA, GGT, and GGC. GGT and GGC will be for 78% the preferred codons to encrypt a glycine,

whereas GGG will only be used in 13% of the cases, and GGA in 9% of the cases. This means that the combinations GGT and GGC will be more represented in coding regions than are the combinations GGG and GGA, whereas in non-coding regions GGG and GGA will be represented as frequently as GGT and GGC are. By plotting the appearance of those codons that are rarely used in coding sequences (like GGA for glycine in *Escherichia coli*) through spots along the reading frames, one can visually locate coding sequences as regions where spots are absent or rarely represented.

The dialog with the available codon usage tables is called with **Tools > Frame analysis > Select codon usage...**

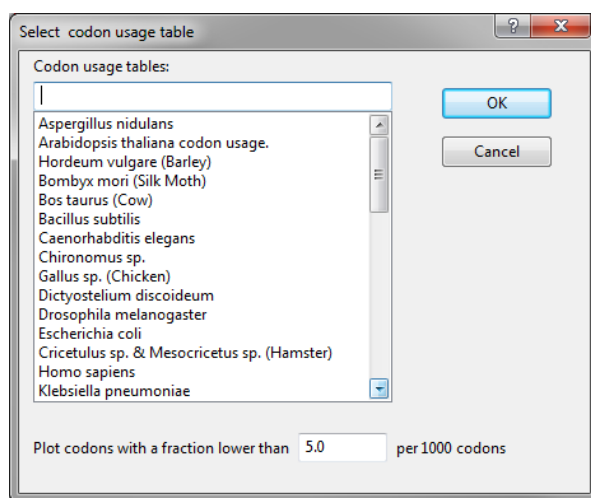


Figure 8.2.3: The *Select codon usage table* dialog box: select a codon usage table from the list.

Codons having a fraction lower than the cut-off value specified in the field **Plot codons with a fraction lower than**, will be plotted as a dot on the sequence plot. These fractions are expressed as "number of appearance per 1000 codons".

With **Tools > Frame analysis > Show codon usage in sequence viewer** the rare codons of the selected organism are plotted on the six reading frames in the *Sequence Viewer* panel. The pink spots are less abundant at each position along the frame, where a protein is encoded. It is clear that this option can help a lot in predicting protein encoding sequences, for example in deciding which coding sequence in case of overlapping open reading frames is the most probable true coding region. The stretches can be removed again from the plot with the same action.

8.2.3 Restriction analysis

8.2.3.1 Restriction Analysis panel

The restriction analysis tool that is present in the *Sequence editor* window has been designed to map restriction enzyme sites on nucleotide sequences. By default, both panels in the *Restriction analysis tab* are empty. The enzyme list that will be used to search for enzymes, is displayed in toolbar of the *Restriction analysis tab*. The **full-list** - holding the entire REBASE list of enzymes - is automatically selected. If other lists are defined in the database, these lists can be selected from the drop-down list in the toolbar.

8.2.3.2 Restriction Analysis features

A restriction enzyme analysis in the *Sequence editor* window consists of two steps:

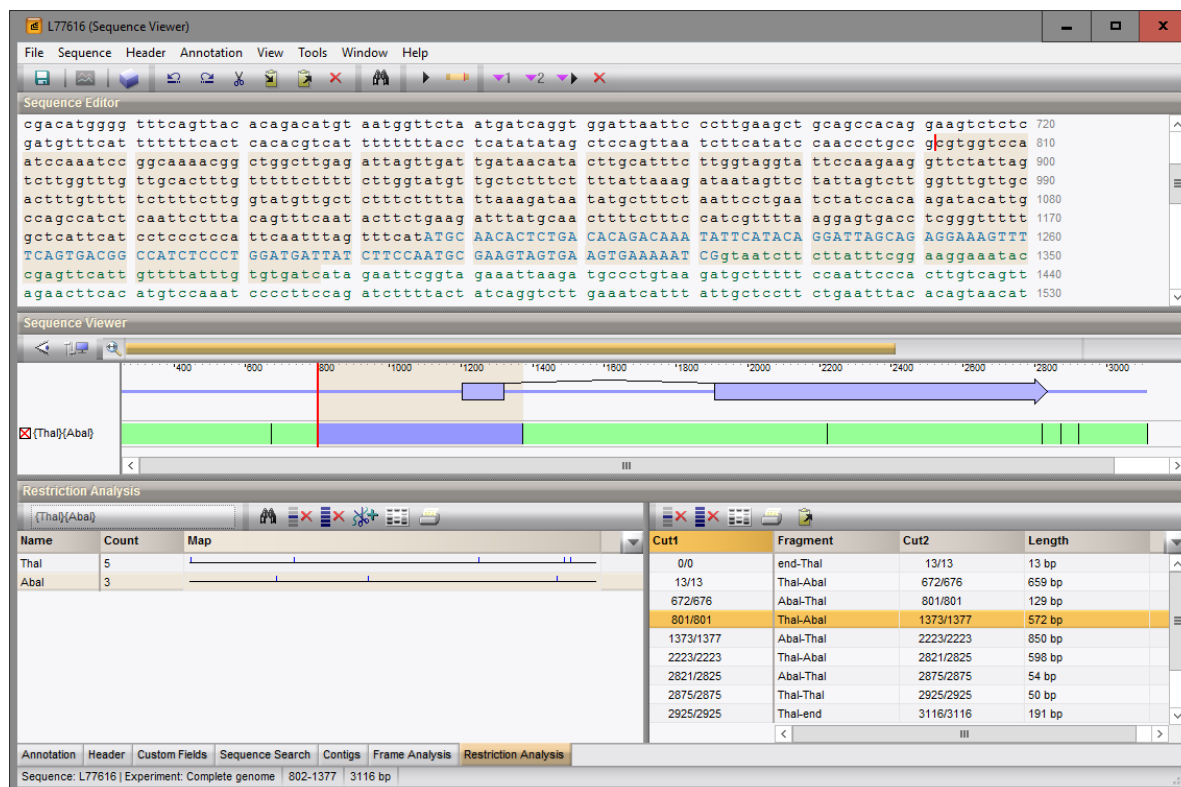


Figure 8.2.4: The *Restriction Analysis* panel.

- Step 1: Selection of enzymes that fulfill certain criteria into an *enzyme map* (left panel in the *Restriction Analysis* panel).
- Step 2: Mapping of the most suitable enzymes from the enzyme map onto the sequence (*fragment list*, right panel in the *Restriction Analysis* panel).

Select **Tools** > **Restriction analysis** > **Add enzyme map...** (🧬) to open the *Restriction enzyme filter* dialog box.

A selection of enzymes into the *enzyme map* can be made by selecting an enzyme from the enzyme list (check **Select single enzyme**).

Besides selecting restriction enzymes from a list of enzymes, restriction enzymes can also be selected by using more general criteria, like the number of bases in the recognition site, blunt or 5'/3' sticky ends, etc. These settings can be specified in the *Restriction enzyme filter* dialog box when checking **use filter settings** and subsequently pressing <**Change filter settings**>.

The *Restriction enzyme filter* dialog box allows a number of parameters to be specified:

- The *Cleavage type* can be **Blunt end**, **5' protruding end**, and/or **3' protruding end**. The *Recognition pattern* can be **palindromic**, **Non-palindromic single cleavage**, and/or **Non-palindromic double cleavage**.
- The *Length recognition site* can be defined as **n=4**, **n=5**, **n=6**, **n=7**, **n=8**, and **Other**, or any combination of these.
- **Restriction enzymes with unknown sites** and **unknown cleavage patterns** can be included or excluded from the search criteria. If the option **Include restriction enzymes not commercially available** is unchecked, only enzymes which are specified in the restriction enzyme database as being available from at least one commercial institution will be searched for.

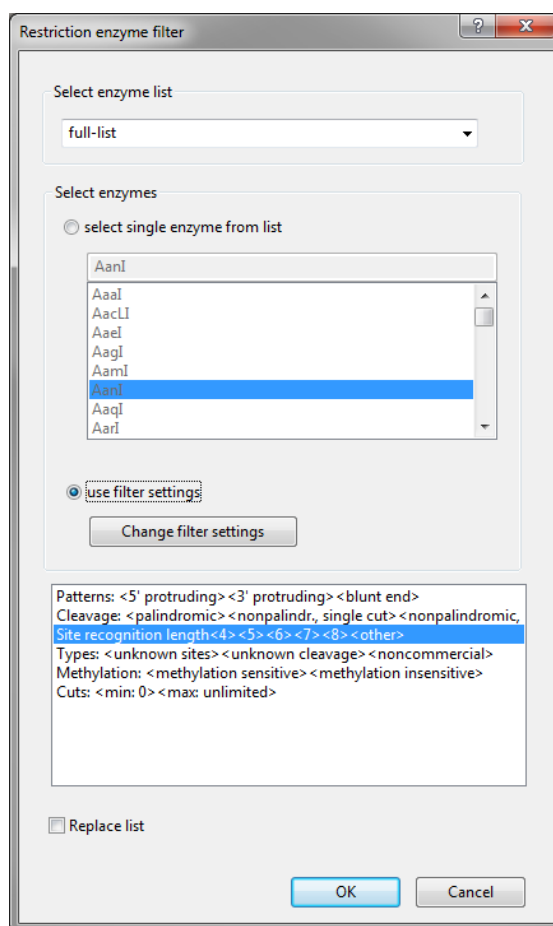


Figure 8.2.5: The *Restriction enzyme filter* dialog box.

- **Methylation-sensitive** and **-insensitive restriction enzymes** can be included or excluded from the search criteria.
- The number of cleavage sites can be specified with a **Minimal** and a **Maximal cut** limit.
- Finally, in respect of enzymes present in the panel of *Restriction enzyme filter* dialog box, one can specify to replace the enzymes (check **Replace list**), or to add the enzymes to the existing list (uncheck **Replace list**).
- To restore the default settings press the **<Default>** button.

Pressing **<OK>** shows the enzymes that fulfill the specified criteria in the *Restriction enzyme filter* dialog box.

In respect of enzymes present in the enzyme map, one can specify to replace the enzymes (check **Replace list**), or to add the enzymes to the existing list (uncheck **Replace list**).

When pressing **<OK>** the selected enzyme(s) are added to the enzyme map, displaying the number of cleavage sites on the nucleotide sequence in the **Count** column, and the position of the cleavage sites on the sequence in the **Map** column.

Double-clicking on an enzyme in the left panel maps the restriction enzyme onto the sequence. Alternatively, choose **Tools > Restriction analysis > Add selected enzyme to fragment list** (🔍➕).

The fragment list (right panel) lists the restriction fragments obtained using the mapped enzyme (**Fragment**), the restriction enzyme cut positions (**Cut1** and **Cut2**) and number of base pairs (**Length**).

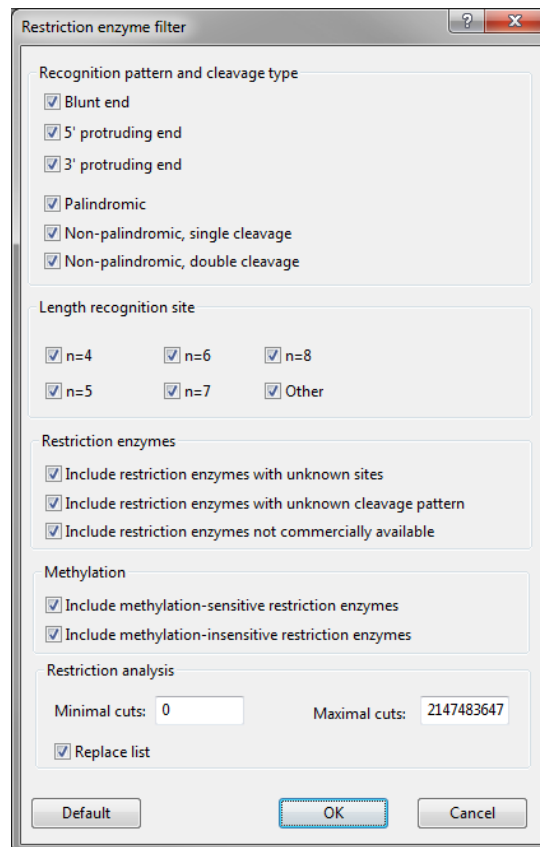


Figure 8.2.6: The *Restriction enzyme filter* dialog box.



Fragments of some enzymes cannot be mapped on the sequence because their information provided is incomplete (e.g. unknown cleavage positions). A message will pop up when trying to map the fragments of these enzymes.

The fragment list (right panel) can be exported to a tab-delimited text file with **Tools** > **Restriction analysis** > **Export fragment list...** (📄).

Select **Tools** > **Restriction analysis** > **Print fragment list...** (📄) to print the information present in the fragment list. Use **Tools** > **Restriction analysis** > **Print enzyme map...** (📄) to print the enzyme map (left panel).

To plot the fragments in the middle panel, press the 📊 button. The center panel shows a schematic presentation of the fragments (black vertical lines in green block). A selected fragment is highlighted in purple on the map image. Non-selected fragments are shown in green.

Use the zoom slider in the *Sequence Viewer* panel to zoom in on the sequence plot. Zooming can be done up to base level and shows the cleavage patterns at the respective cleavage sites.

Remove the fragments again from the *Sequence Viewer* panel with the 🗑️ button.

To remove all fragments from the fragment list for one or more enzymes, press **Tools** > **Restriction analysis** > **Remove enzyme from fragment list** (🗑️).

Select the enzyme(s) you want to remove the fragments from, and press <**Delete**>.

A virtual gel can be created with gel patterns of the restriction enzymes that are present in the enzyme map (left panel). Select **Tools** > **Restriction analysis** > **Add selected enzyme to gel window** (📄) to call the *Fingerprint fragment lists* window.

Similar as for the enzyme list, a virtual gel pattern can be generated for the fragments obtained from a

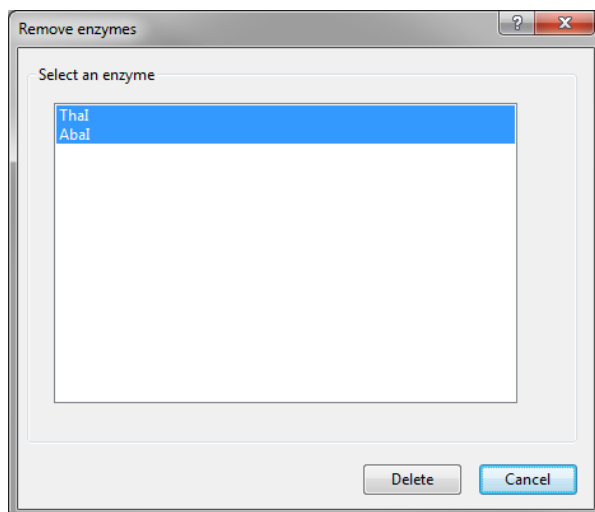


Figure 8.2.7: The *Remove enzymes* dialog box: remove fragments for selected enzyme(s).

combination of enzymes (right panel). Selecting **Tools** > **Restriction analysis** > **Add fragments to gel window** (📊) add the lane with the fragments.

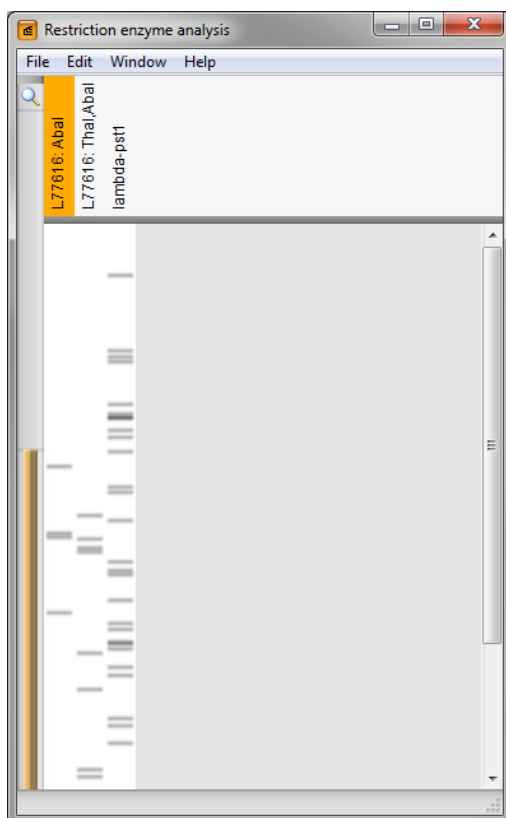


Figure 8.2.8: Virtual gel.

When moving with the mouse over a band in one of the lanes, the molecular size of the band is displayed in a small tool tip window.

To change the position of a gel pattern, click on the gel pattern and select **Edit** > **Move to the left** (Ctrl+LEFT) or **Edit** > **Move to the right** (Ctrl+RIGHT).

A selected pattern is deleted from the virtual gel with **Edit** > **Delete**.

A molecular weight marker pattern can be added to the virtual gel with **Edit > Add marker**. This calls a new dialog displaying all markers that are present in the database. By default, only one marker is present in the database, the *Lambda-PstI* marker. Other markers can be added to the database (see 8.5.3 for more information). Pressing <OK> add the pattern of the selected marker to the virtual gel.

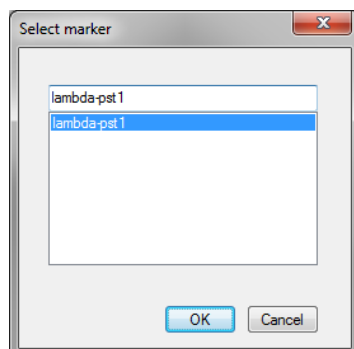


Figure 8.2.9: Add a marker to the virtual gel.

To remove all fragments from the fragment list choose **Tools > Restriction analysis > Clear fragment list** (🗑️).

Select **Tools > Restriction analysis > Remove enzyme from map** (🗑️) to remove a selected enzyme from the enzyme map. If the enzyme was mapped onto the sequence (right panel), the fragments obtained with the selected restriction enzyme are removed from the fragment list.

To remove all enzymes from the enzyme map, select **Tools > Restriction analysis > Clear map list** (🗑️). Both the enzyme map and the fragment list are cleared.

8.2.4 Primer analysis

8.2.4.1 Introduction

The primer analysis application in BioNumerics has been designed to calculate optimal primer and primer combinations for the amplification of a target region in function of various experimental parameters. The primer analysis tool can be launched from the *Sequence editor* window or from the *Sequence alignment* window, both for nucleotide sequences or amino acid sequences. In the latter case, nucleotide sequences are generated through back-translation. When launching the primer analysis tool from the *Sequence editor* window, the target region can be adapted from the complete sequence, or from a (feature) selection.

Choose **Tools > Primer design...** to launch the primer analysis tool.

The *Primer design* window is divided into an upper panel displaying the nucleotide sequence (the *Sequence viewer panel*), and a lower panel, giving information concerning the primers and primer combinations found.

Zooming can be done up to base level with the zoom slider. The translation of the CDS features – if present – is shown underneath the nucleotide sequence. The red vertical line indicates the cursor position on the sequence. If a selection was present in the *Sequence editor* window before launching the primer analysis tool, the selected region is highlighted with an orange background.

8.2.4.2 Creating a locus

Primers and primer combinations are searched in a target region for PCR amplification, also referred to as a *locus*. Depending on the situation, following target regions are used by default:

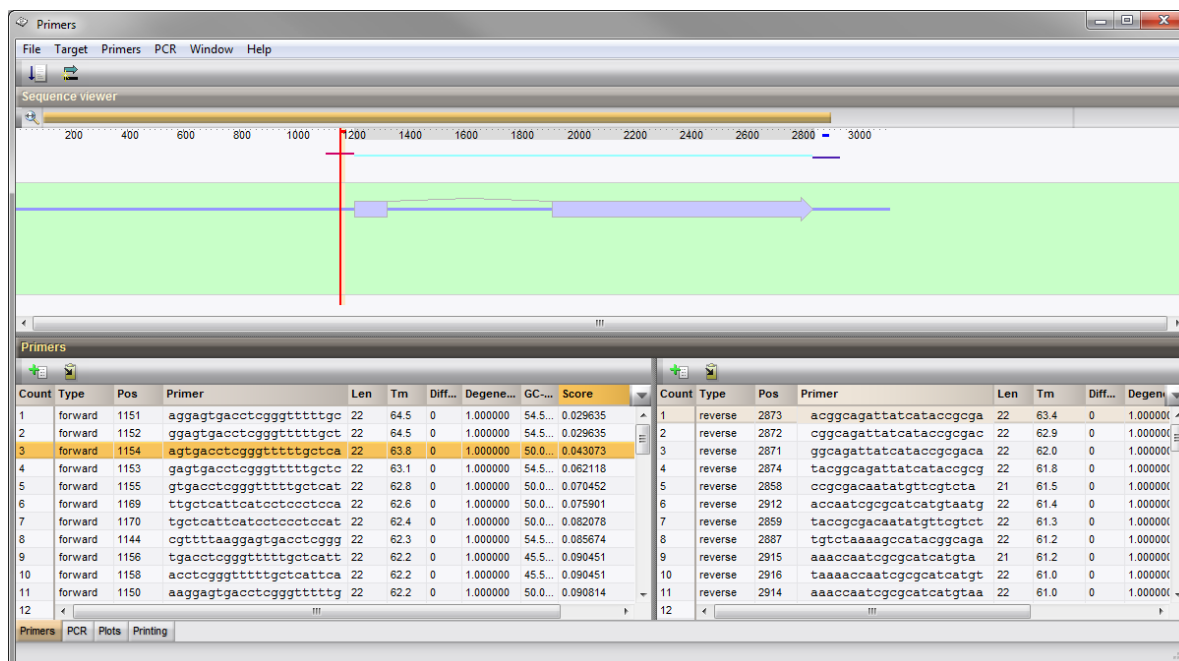


Figure 8.2.10: The *Primer* design window.

- If no selection is defined, the full sequence is taken as target.
- If a feature is selected in the *Sequence Viewer*, this feature will be taken as target region.
- If a sequence selection is defined in the *Sequence Viewer*, this selection will be taken as target region.

The default target region can be modified with **Target > Edit...**

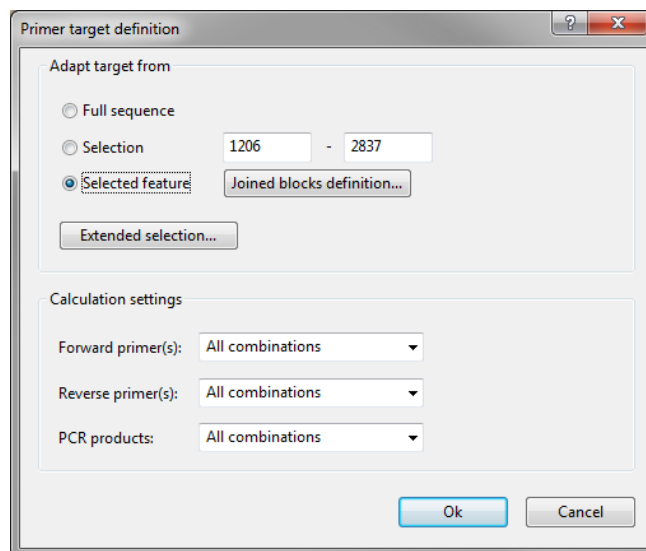


Figure 8.2.11: The *Primer target definition* dialog box: specify the target settings.

The *Primer target definition* dialog box prompts for some specifications about the target region (upper panel):

- With the **Full sequence** option checked, the whole sequence will be specified as forward and reverse primer regions and will be used when searching for primers and primer combinations.

- Check the **Selection** option if a user-defined selection has been made in the *Sequence editor* window (independent from feature). The subsequence is specified by the start and stop positions in the two input boxes. If a selection is present in the *Primer design* window, the start and stop positions of the selection are automatically displayed in the start and stop input boxes. Forward and reverse primers will be searched within this selection.
- Select the **Selected feature** option if a feature has been selected in the *Sequence editor* window. In this case, forward and reverse primers can be searched spanning the feature exons only, or spanning feature introns and exons. Note that the **Selected feature** option will only be accessible if the sequence of a feature was selected in the *Sequence editor* window before launching the primer analysis tool.

When the **Selection** or **Selected Feature** option is checked, the target design around the region can further be specified by pressing the button **<Extended selection>** in the *Primer target definition* dialog box.

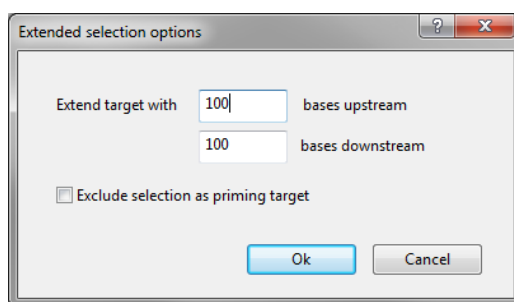


Figure 8.2.12: Extended selection options.

Three possible target designs can be constructed:

1. The forward and reverse primer target regions correspond to the selected region.
2. The primer regions include the selected region and some defined distance upstream and/or downstream from this region.
3. The primer regions are located up- and downstream only in the extended regions.

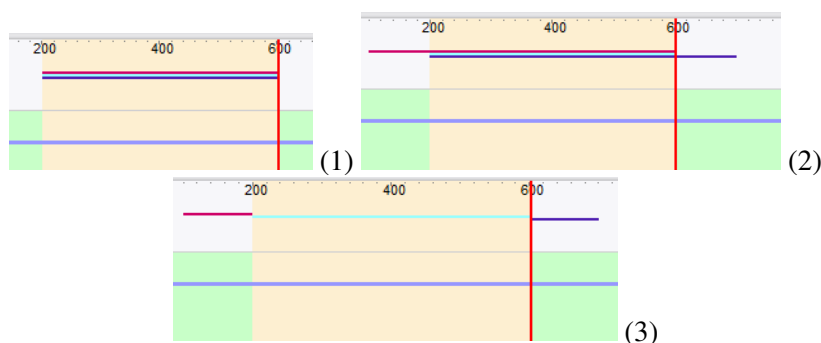


Figure 8.2.13: Three possible target design strategies (see text for explanation).

Strategy 1 is achieved when the selected region is not extended with bases up- and downstream, and the option **Exclude selection as priming site** remains unchecked. The resulting PCR products will be smaller than the selected region.

Strategy 2 is achieved by specifying a number of bases for primer binding upstream and/or downstream of the selected region and leaving the option **Exclude selection as priming site** unchecked.

Procedure 3 is the same as number 2 except that the check box **Exclude selection as priming site** should now be checked in order to exclude the selection as priming site. The resulting PCR products will fully cover the selection and will thus be greater than the selected region.

When the **Selected feature** option is checked, the target design around the selected feature region can further be specified in the *Joined blocks definition* dialog box. This window is called by pressing the **<Joined blocks definition>** button in the *Primer target definition* dialog box.

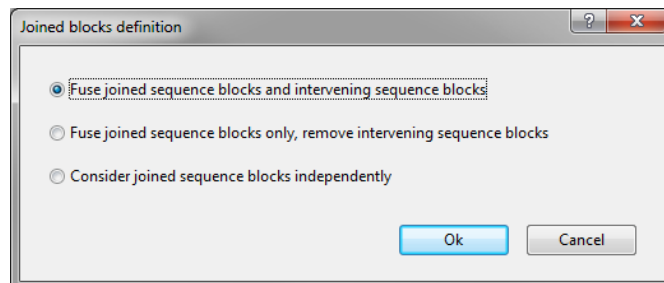


Figure 8.2.14: Joined blocks definition.

The settings in this window are only of interest when looking for priming sites around features containing intervening sequences.

When the **Fuse joined sequence blocks and intervening sequence blocks** option is checked, exons and introns will be considered as one continuous sequence. Choose this item when performing a PCR on genomic DNA with the intron sequences also of interest, or at least not interfering with the experiment.

When checking the option **Fuse joined sequences only, removing intervening sequence blocks**, the introns will be excised from the sequence, as if splicing would have been occurred. Choose this option when the experiment will be carried out on RNA-derived templates (e.g. copy-DNA).

When the option **Consider joined sequences independently** is checked, each exon will be considered as an independent target. During primer and PCR product calculation (see further), the most compatible exon target is chosen.

Press **<OK>** in the *Primer target definition* dialog box plots the target design on the sequence.

The target design is plotted in the *Sequence viewer panel* below the sequence position numbering. The region used to construct the design is shown in green. The forward primer region is shown in red, and the reverse primer region is shown in blue. Some example designs:

8.2.4.3 Searching for primers and PCR products

After having specified the forward and reverse primer regions on the sequence, these regions can be screened for optimal primers and primer combinations. Using the default search settings, the software calculates optimal primers and primer combinations in function of various settings that can be changed by the user: selecting **File > Run calculation** (📄) calls the *Specifications for primer calculation* dialog box.

In this dialog, the following primer calculation parameters can be set:

- The **Preferential primer size** is the preferential length of the primers that will be searched for (standard 20 base pairs). The **Maximal primer size** defines the maximal length of the primers and **Minimal primer size** defines the minimal length. When specifying a primer with a preferential length 20 and a minimal and maximal primer size of 18 and 22 respectively, all primers ranging from length 18 till 22 will be considered. Increased primer sizes result in increased melting temperatures.
- The **Preferential melting temperature** is by default set to 67.5 degrees Celsius. The **Minimal melting temperature** (standard 55.0 degrees Celsius) specifies the lower temperature cut off value. Primers

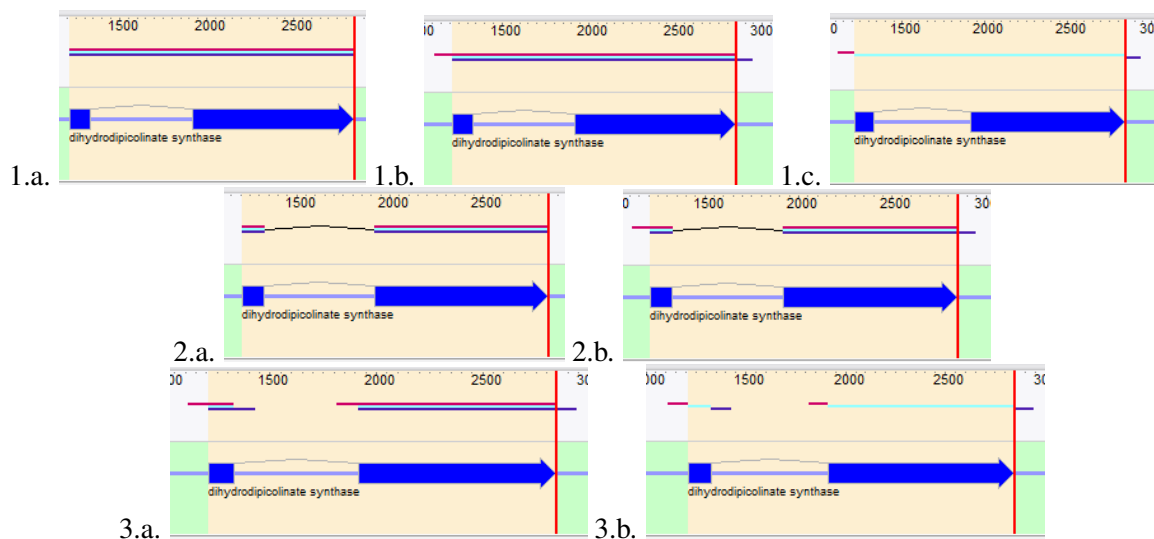


Figure 8.2.15: Target designs for a selected feature with an intervening sequence: 1. Exons and intron are considered as one continuous region: a. Primer regions include the exon and intron sequences, b. Primer regions include the exon and intron sequences, and some defined distance up- and downstream from this region, c. Primer regions are located up- and downstream from the selected feature region, excluding the feature region itself. 2. Intron sequence is excised from the sequence, resulting in one joined exon sequence: a. Primer regions correspond to the joined exon sequence, b. Primer regions include the joined exon sequence, and some defined distance up- and downstream from the feature region. 3. Each exon is considered as an independent target: a. Primer regions include the exon sequences and some defined distance up- and downstream from each exon region, b. Primer regions are located some defined distance up- and downstream from each exon region, excluding the exon region itself.

with a calculated melting temperature under this value will be discarded. The **Maximal melting temperature** is standard set to 80 degrees Celsius. This setting specifies the higher temperature cut off value. Primers with a calculated melting temperature above this value will be discarded.

- The **Minimal** and **Maximal percentage GC-content** parameters specify the minimal and maximal percentage G and C bases. The values are default set to 20 and 80 respectively. Note that a high GC-content mostly indicates a stable binding (GC has 3 hydrogen bonds whereas AT has only 2) and thus a higher melting temperature. If a high GC-content is demanded, one cannot expect low melting temperatures.
- The **[Monovalent cation]** concentration in the solution has a great influence on the charge of the DNA molecules. Melting temperature of the DNA molecules will be in function of concentration in the solution. Higher monovalent salt concentrations lower the DNA melting temperature. The monovalent cation concentration is by default set to 50 mM.
- The **[Mg⁺⁺]** concentration in the solution also influences the charge of the DNA molecules and the melting temperature of the DNA molecules. Higher [Mg⁺⁺] lower the DNA melting temperature. The default value is 2 mM.
- The **[DNA top strand]** reflects the concentration of the primer; the **[DNA bottom strand]** corresponds to the template DNA concentration. Although the primer and template concentration in the solution does not have a great impact on hybridization of the DNA molecules, higher DNA concentrations lower primer melting temperatures weakly.
- The **Maximal degeneracy** is standard set to 1024. This setting specifies how degenerated a primer can be. A maximal degeneracy of 1024 means that maximally 5 positions in the primer are uncertain

Primer and PCR settings 'standard'

Primers

Preferential primer size: 20
Minimal primer size: 18
Maximal primer size: 22

Preferential melting temperature: 67.5
Minimal melting temperature: 55.0
Maximal melting temperature: 80.0

Minimal percentage GC-content: 20
Maximal percentage GC-content: 80

[Monovalent cation] 50 mmol/l [DNA top strand] 50 nmol/l
[Mg++] 2 mmol/l [DNA bottom strand] 50 nmol/l

Maximal degeneracy: 1024
Self-complementary check size: 8
End-complementary check size: 5
Minimal: 1

Weight factor balance:
Degeneracy:
Melting temperature:
Discrimination:

PCR products

Minimal length: 86
Maximal length: 10000

Weight factor balance:
Degeneracy of primers:
Length of PCR products:

Calculation

☐ Scan for false priming on template
Exclude if false priming site contains 4 mismatches or less (including gaps)

☐ Eliminate overlapping primer candidates

Top ranking primers to be shown: 100
Top ranking PCR products to be shown: 100

Output options

☒ Write reverse primers in conformance with the minus strand

Defaults OK Cancel

Figure 8.2.16: The *Specifications for primer calculation* dialog box.

bases. Of course, degeneracy of the primer has to be seen in function of the specified primer length, and whether or not inosine is incorporated into the primer.

- The **Self-complementary check size** value indicates the size of the window with which primers are screened upon self-complementary. The default value is set to 8. A **Self-complementary check size** of 8 means that each primer showing a stem-loop of 8 identical bases will be eliminated.
- The **End-complementary check size** is by default set to 5. This value indicates the size of the window with which primers are screened for complementary with each other at the 3' end. Primers are selected that show a base pairing at the 3' end reduced to the minimum.
- The **Minimal discrimination** parameter is only considered when performing a *discriminative* primer analysis, launched from the *Sequence alignment* window. A minimal discrimination of 1 means that at least one mismatch should occur within a possible binding site of the primer on any sequence of the negative selection. The user has to set this value in function of the primer length. For short primer lengths, one mismatch will mostly prevent binding of the primer, however one should take into account that, with longer primer lengths, a single mismatch occurring at the far 5' phosphate end of the primer may allow binding of the primer at lower temperatures. Therefore it is advised to select primers with more mismatches to negative sequences (increasing **Minimal discrimination**), or to select primers with a single mismatch at the middle or 3'-OH end of the primers.

- **Weight factor balance:** Ranking the primer that do fulfill the individual criteria can be done by specifying weights of importance. The criteria which are specific for each individual primer are **Degeneracy** and **Melting temperature** (and **Discrimination** when performing a discriminative primer analysis). The sliders represent the weights of these criteria. The more the weight is shifted to the left, the less the criterion is taken into account.

The settings concerning calculation of **PCR products** are the following:

- The **Minimal length** is standard set to 86. This setting specifies the lower cut off value of the PCR product sizes. PCR products with a length under this value will be discarded.
- The **Maximal length** is standard set to 10000. This setting specifies the higher cut off value of the PCR product sizes. PCR products with a length above this value will be discarded.
- **Weight factor balance:** Ranking the PCR products that do fulfill the individual criteria can be done through specifying weights of importance upon two selection criteria. These important criteria, which are specific for each individual PCR product, are **Degeneracy of the primers**, and **Length of the PCR products** (longer PCR products are difficult to synthesize). The two sliders represent the weights of these two criteria. The more the weight is shifted to the left, the less the criterion is taken into account.

A number of **Calculation** settings are listed in the lower panel:

- When enabling the check box **Scan for false priming on template** the primers are checked for false priming on the template, eliminating the formation of false reaction products. In the input box, one can define the cutoff value, for which primers showing a possible priming site with a number of mismatches less or equal to the cutoff value, should be treated as false priming cases.
- Enabling the check box **Eliminate overlapping primer candidates** eliminates all primers candidates that show overlap with other primers candidates.
- The **Top ranking primers to be shown** is standard set to 100. After ranking the primers upon their criteria, only the first 100 will be listed. Primer combinations are based on these selected primers.
- The **Top ranking PCR products to be shown** is standard set to 100. After ranking the PCR products upon their criteria, only the first 100 will be listed.
- Pressing the **<Defaults>** button restores the default settings.

When the **Output** setting **Write reverse primers in conformance with the minus strand** is checked, all reverse primers will be displayed, printed and exported in the reverse-complement direction. If this option is unchecked, reverse primers are written in conformance with the leader (plus) DNA strand.

After having executed the search for primers and PCR products, the left and right panels in the **Primers tab** list the most suitable forward and reverse primers respectively. The **PCR tab** contains the PCR products.

The primer sequence is displayed in the **Primer** column. The length of the primer is shown in the **Len** column and the start position in the **Pos** column. The melting temperature (calculated according to the nearest-neighbor thermodynamics published by SantaLucia [34] using the default thermodynamic settings), the number of different bases, the degeneracies, and GC-content are displayed in the **Tm**, **Difference**, **Degeneracy**, and **GC-content** columns respectively.

The primers are ordered according to a score, which is calculated from the primer calculation parameters (**Score** column). The primers can also be sorted according to the other features, such as position, length, degeneracy, melting temperature, and number of different bases (in case discriminating primers were searched for). Arranging the primer list according to another feature (for example base difference) can help select

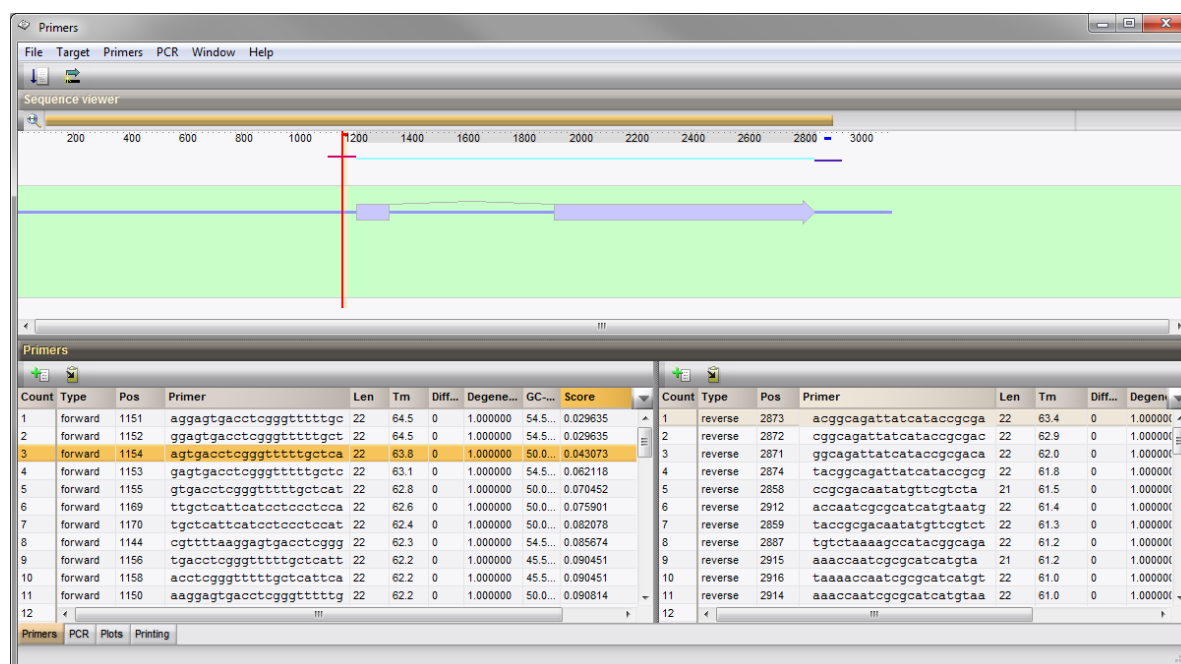


Figure 8.2.17: The *Primer* design window with the *Primers* tab selected.

suitable primers out of the full list. This is done by right-clicking with the mouse pointer on a header field and selecting **Sort**.

The sequence of the selected primer is highlighted with an orange background in the *Sequence viewer* panel. A red line is shown in the sequence position bar when a forward primer is selected, a reverse primer is recognized by a blue line. Changing the primer selection in the primer lists updates the selection in the *Sequence viewer* panel.

To export the primer information of the selected forward primer to the clipboard choose **Primers > Forward > Export to clipboard** (📋); to export the primer information of the selected reverse primer to the clipboard select **Primers > Reverse > Export to clipboard** (📋).

To add a selected forward or reverse primer to the printing list, select **Primers > Forward > Add to printing list** (🖨️) or **Primers > Reverse > Add to printing list** (🖨️) respectively.

The *PCR* panel lists the PCR amplification products - created using the primers listed in the *Primers* tab - that fulfill the criteria defined in *PCR products* panel in the *Specifications for primer calculation* dialog box.

The forward and reverse primer sequences are displayed in the **Forward** and **Reverse** columns, respectively. The start position of the forward primer is shown in the **Start** column, the stop position of the reverse primer in the **Stop** column. The length of the product is shown in the **Length** column. The melting temperature (calculated according to the nearest-neighbor thermodynamics published by SantaLucia [34] using the default thermodynamic settings) of the forward and reverse primers are shown in the **Tm1** and **Tm2** columns respectively.

The PCR products are ordered according to a score, which is calculated from the PCR products calculation parameters (**Score** column). The PCR products can also be sorted according to the other features, such as start or stop position, length, and melting temperatures. Arranging the PCR products list according to another feature can help select suitable products out of the full list. This is done by right-clicking with the mouse-pointer on the header of a feature and selecting **Sort**.

The sequence of a selected PCR product is highlighted with an orange background in the *Sequence viewer* panel, and a gray bar is shown in the sequence position bar. Changing the PCR product selection in the PCR list updates the selection in the *Sequence viewer* panel.

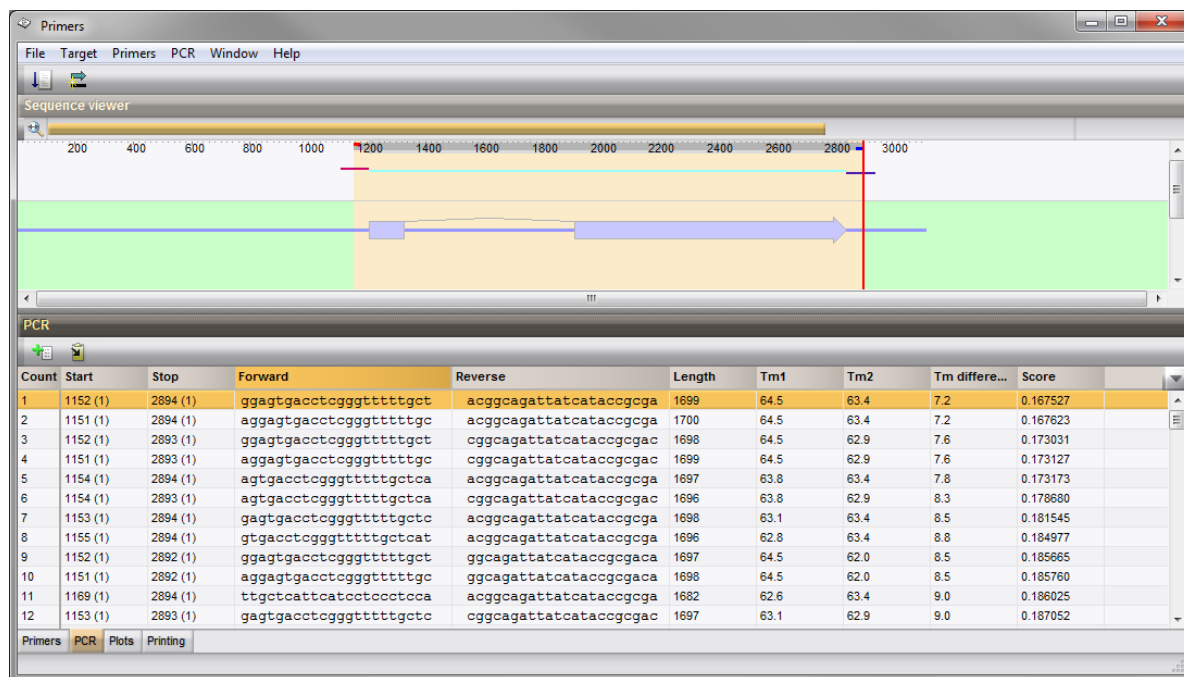


Figure 8.2.18: The *Primer design* window with the *PCR* tab selected.

To export the PCR product information of the selected PCR product to the clipboard select **PCR > Export to clipboard** (📄).

To add the forward and reverse primers of the selected PCR product to the printing list choose **PCR > Add to printing list** (🖨).

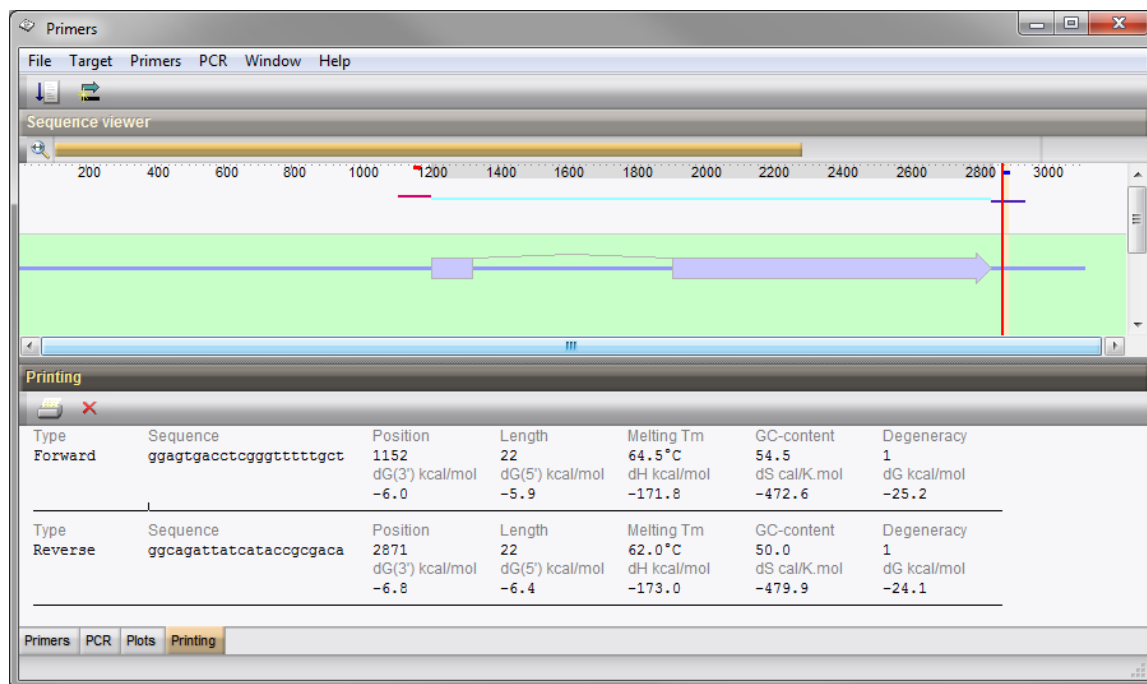


Figure 8.2.19: The *Primer design* window with the *Printing* tab selected.

The *Printing* tab displays all primers added with the commands **Primers > Forward > Add to printing list**, **Primers > Reverse > Add to printing list**, and **PCR > Add to printing list**.

The primer sequence is displayed together with its start position, length, melting temperature (calculated according to the nearest-neighbor thermodynamics published by SantaLucia [34] using the default thermodynamic settings), the GC-content, and degeneracies.

All primer information present in the *Printing tab* is printed with **File > Printing list > Print...** (🖨).

To clear all primer information shown in the *Printing tab* choose **File > Printing list > Clear** (✖).

Forward primers, reverse primers and PCR products are default included in the primer search. To exclude primer combination from the search select the option "None" from the drop-down list next to **PCR products**. Likewise, select the option "None" from the drop-down list next to **Reverse primer(s)** or **Forward primer(s)**, to remove reverse or forward primers from the search.

Forward and reverse primer regions can also be screened for *fixed* forward and reverse primer sequences: select the option "User-defined primer" from the drop-down list next to **Forward primer(s)** or **Reverse primer(s)** and press the <Edit> button to define a fixed forward or reverse primer. This calls the *Enter primer sequence* dialog box.

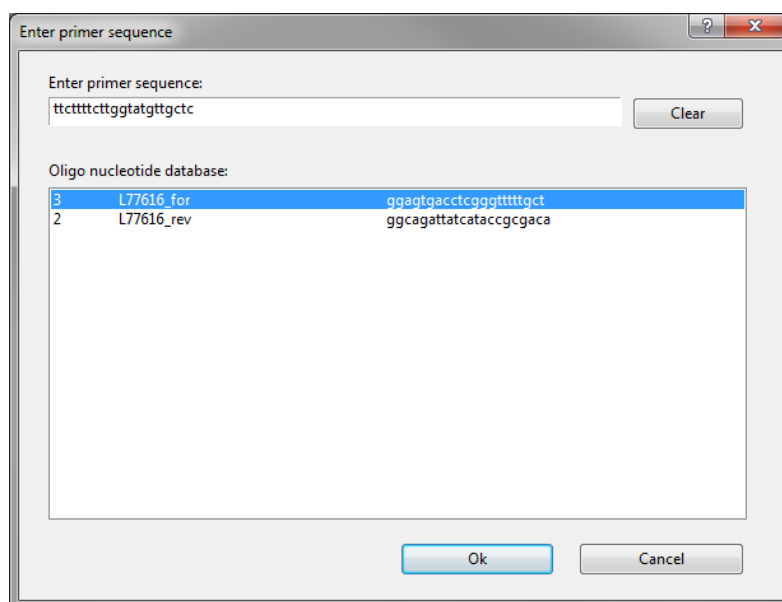


Figure 8.2.20: The *Enter primer sequence* dialog box.

A sequence can be entered in the **Primer sequence** input field or an oligo can be selected from the oligo nucleotide database listed in the lower panel.

BioNumerics will check if the primer regions contain binding sites for the user-defined primers, and displays a warning message if multiple sites or no sites are found.

If a fixed forward and reverse primer was specified, both fixed primer sequences will be searched for within the primer regions.

If a fixed forward primer was specified, and the **All combinations** option was specified for the reverse primer option, the fixed forward primer sequence will be searched for within the forward primer region. The reverse primer combinations will be searched for in the corresponding reverse primer region using the primer filter settings.

If a fixed reverse primer was specified, and the **All combinations** option was specified for the forward primer, the fixed reverse primer sequence will be searched for within the reverse primer region. The forward primer combinations will be searched for in the corresponding forward primer region using the primer filter settings.

8.2.4.4 Degeneracy and melting temperature plots

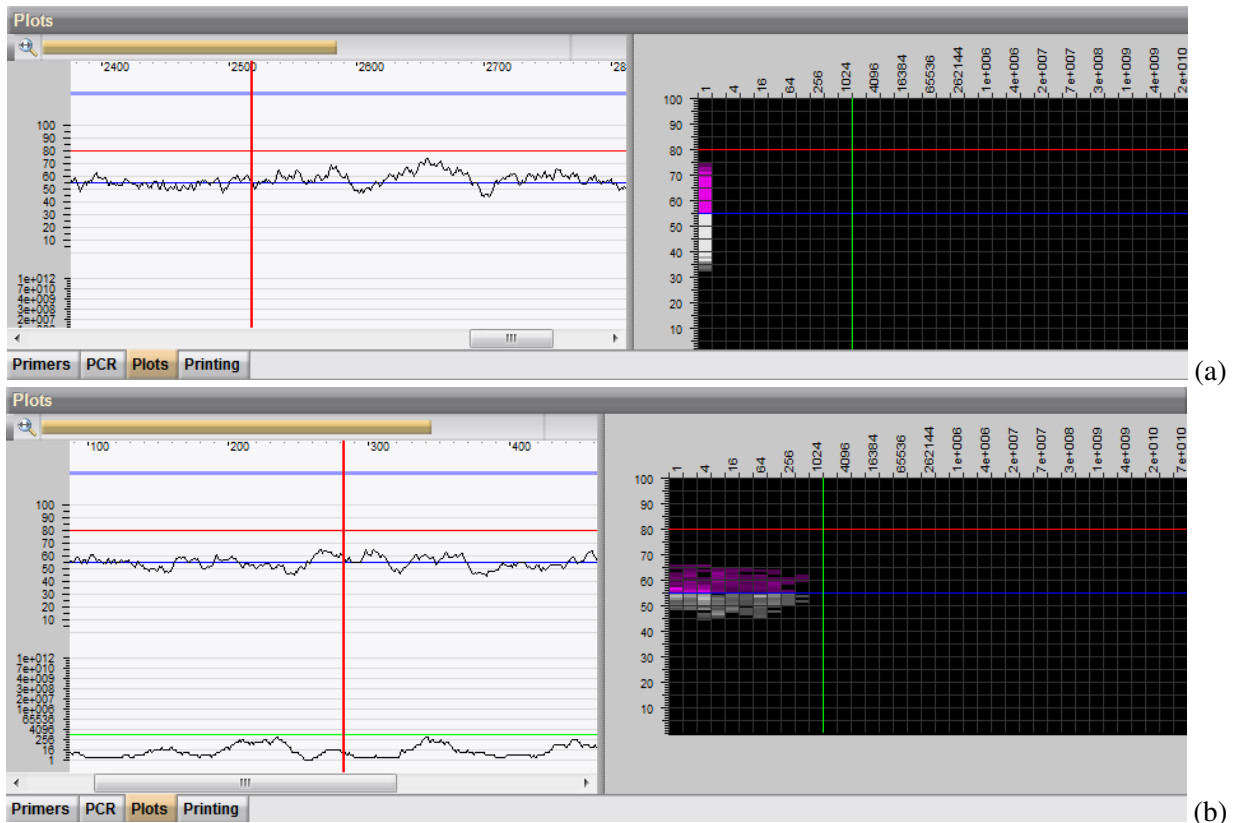


Figure 8.2.21: The *Plots* tab: (a) Sequence containing no degeneracies; (b) Sequence containing degeneracies.

In the *Plots* tab the degeneracy and melting temperature, the two most important criteria for selecting primers, are graphically plotted in function of the sequence position.

- The left upper graph shows the melting temperature of the partial sequence with a window size equal to the primer length, in function of the position along the total sequence. The zone between the upper red line (**Maximal melting temperature**) and lower blue line (**Minimal melting temperature**) indicates the temperature interval in which primers will be accepted.
- The left lower graph plots the degeneracy of the partial sequence with a window size equal to the primer length, in function of the position along the total sequence. The green line indicates the cut-off degeneracy (**Maximal degeneracy**).
- The right plot shows the degeneracy of each possible primer (X-axis) plotted against its melting temperature (Y-axis). If a primer falls within the settings specified, it is plotted purple, if not it is plotted gray. The more primers are plotted on the same x-y coordinate, the lighter the purple or gray shade will be. The shade thus indicates how many primers are plotted on this spot.

8.2.4.5 Adding primers to the oligo nucleotide database

Primer sequences generated within the primer design functionality of BioNumerics can be added to the oligo nucleotide database that comes with each BioNumerics database (see 8.5.2).

To add a forward primer to the oligo nucleotide database, select the forward primer from the left list in the *Primers* panel and select **Primers > Forward > Add to oligo nucleotide database**.

To add a reverse primer to the oligo nucleotide database, select the reverse primer from the right list in the *Primers panel* and choose **Primers > Reverse > Add to oligo nucleotide database**.

This calls the *Edit oligo nucleotide sequence* dialog box.

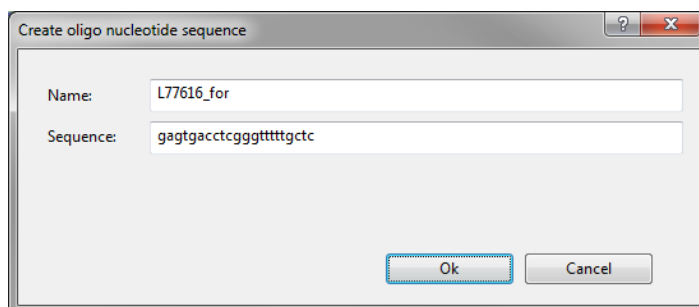


Figure 8.2.22: Add primer sequence to the nucleotide database

The *Edit oligo nucleotide sequence* dialog box displays the sequence of the selected primer in the **Sequence** input field. The suggested **Name** consists of the key of the entry the primer is derived from and the orientation of the primer (for or rev), separated by a "_" sign. The name can be changed if desired. Pressing <OK> adds the primer sequence to the oligo nucleotide database. The *Oligo Database* window opens and the imported sequence is highlighted (see 8.5.2).

To add a primer combination to the oligo nucleotide database select the PCR product from the list in the *PCR tab* in the *Primer design* window and choose **PCR > Add primers to oligo nucleotide database**.

This calls the *Edit oligo nucleotide sequence* dialog box.

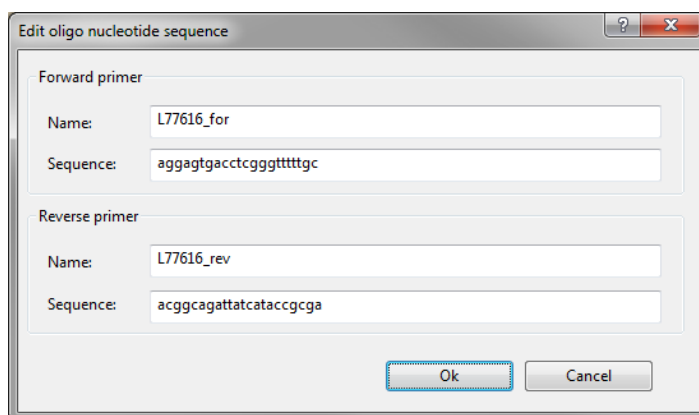


Figure 8.2.23: The *Edit oligo nucleotide sequence* dialog box for adding primer sequences to the nucleotide database.

This dialog displays the sequences of the forward and reverse primer of the selected PCR product in the **Sequence** input fields. The suggested **Name** consists of the key of the entry the primer is derived from and the orientation of the primer (for or rev), separated by a "_" sign. The name can be changed if desired.

Pressing <OK> adds the primer sequences to the oligo nucleotide database. The *Oligo Database* window opens and the imported sequences are automatically selected.

Chapter 8.3

Multiple alignment and cluster analysis of sequences

8.3.1 An introduction to sequence analysis

Among all types of experimental data, cluster analysis of sequence data is by far the most complex in steps and possibilities. The fact that sequences need to be *aligned* before one can estimate similarity requires a number of additional steps before a dendrogram can be obtained. Furthermore, sequence data are a suitable substrate for a number of phylogenetic clustering algorithms which can rarely be applied to other types of data. Please note that the Sequence data module (**sq**) and the Tree and network inference module (**tn**) need to be present in your BioNumerics configuration in order to calculate a cluster analysis based on sequence data.

There are two ways to obtain a dendrogram from sequence data: by aligning the sequences *pairwise* (steps 1-2 in Figure 8.3.1), or by obtaining a *multiple alignment* of all sequences (steps 1-6 in Figure 8.3.1).

The best multiple alignments that can be achieved, particularly for large numbers of sequences, involve the following steps.

1. Pairwise alignment and calculation of similarity of all possible pairs of sequences, resulting in the *Pairwise alignment similarity matrix*.
2. Construction of a UPGMA *dendrogram* based on the similarity matrix obtained.
3. Determination of *consensus sequences* at each linkage node of the dendrogram, down to the root.
4. Alignment of all sequences based on the local and the root consensus sequences.
5. Calculation of a similarity matrix based on the aligned sequences, the *Multiple alignment similarity matrix*.
6. Construction of a Neighbor Joining *dendrogram* based on the multiple alignment similarity matrix.

In **step 1**, each individual sequence is aligned with each other sequence, and for each pair of aligned sequences, the similarity value is calculated and registered in a similarity matrix. The obtained matrix (*pairwise alignment similarity matrix*) will serve as the basis for cluster analysis (**step 2**). Neighbor Joining or other algorithms resulting in unrooted dendrograms would not be suitable here, as in such dendrograms, the closest linked sequences are not necessarily the most related ones. This is a requirement for step 3, discussed below.

Steps 3 and 4 are very important for obtaining a meaningful global alignment. Each linkage node on the dendrogram represents a *local alignment* of the sequences linked at the node, resulting in a *local consensus*.

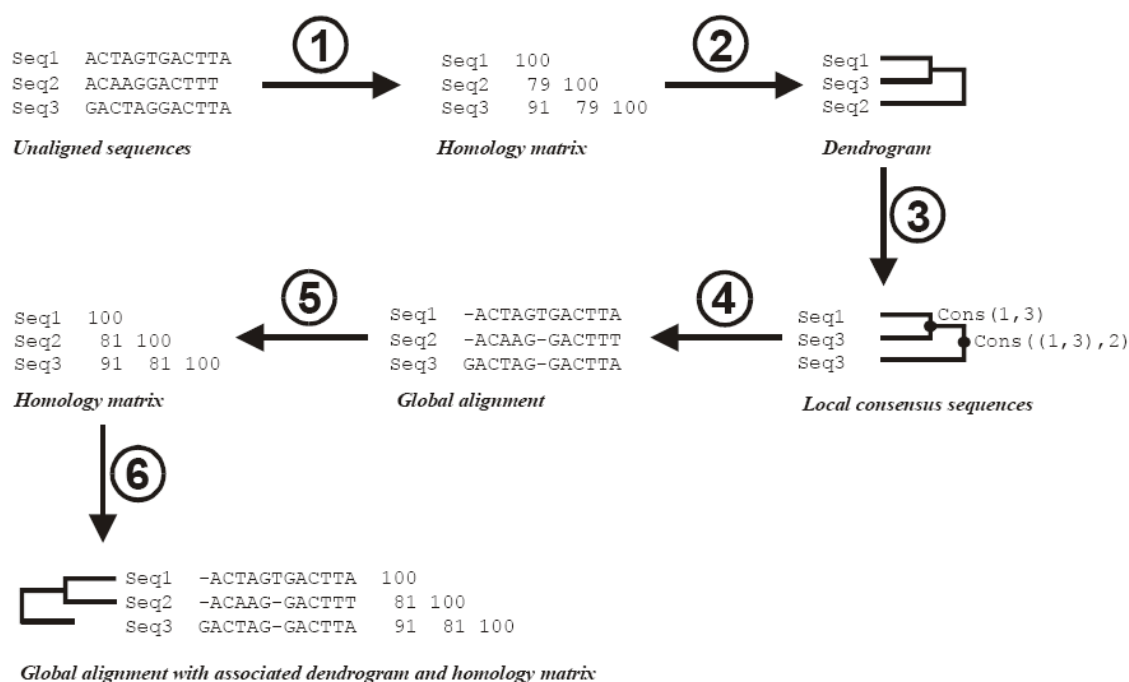


Figure 8.3.1: Steps in a cluster analysis of sequences: dendrogram based on pairwise alignment (steps 1 to 2), and dendrogram based on multiple alignment (steps 1 to 6).

These local consensus sequences are calculated downwards, i.e. starting from the highest related sequences down to the dendrogram root (**step 3**). In the above example, the highest linkage observed is between sequences 1 and 3, leading to consensus (1,3). The next linkage level is the branch that links sequences 1 and 3 with sequence 2. At this node, the consensus (1,3) is aligned with sequence 2. This results in a consensus ((1,3),2), which will, in turn, be aligned with the consensus of another group linked to this one. For each sequence or local consensus, the program keeps track of the positions of the gaps that are introduced to align it with the branch it is linked to. Finally, a *global consensus* for the whole dendrogram is inferred.

The program now introduces to each individual sequence all the gaps that were introduced on the subsequent consensus sequences following the path from the sequence itself down to the global consensus (**step 4**). This results in a *global* or *multiple alignment*.

The multiple alignment in turn can be used as the basis for the calculation of a similarity matrix. Now, instead of aligning each sequence with each other sequence to determine their mutual similarity, the multiple alignment is used to calculate the *multiple alignment-based similarity* between each pair of sequences (**step 5**).

Once the multiple alignment is present, this step is extremely fast. The *multiple alignment-based similarity matrix* can be used for Neighbor Joining or UPGMA clustering, or other clustering algorithms (**step 6**).

8.3.2 Calculating a cluster analysis based on a pairwise alignment

A comparison is created from a selection of database entries and displayed in a new *Comparison* window by clicking in the *Comparisons* panel in the *Main* window and selecting **Edit > Create new object...** (🔍).

To display the data for a sequence type, press the eye button (👁) next to the experiment type name in the *Experiments* panel. When the image is displayed, the button displays a green V-sign.

Initially, the sequences are not aligned and no similarity matrix exists.

To calculate a cluster analysis based on pairwise alignments of sequences select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**.... The *Comparison settings* wizard appears (see Figure 8.3.2). In the *Similarity coefficient* wizard page, the settings are shown in the right panel of the dialog box and depend on the algorithm selected in the left panel.

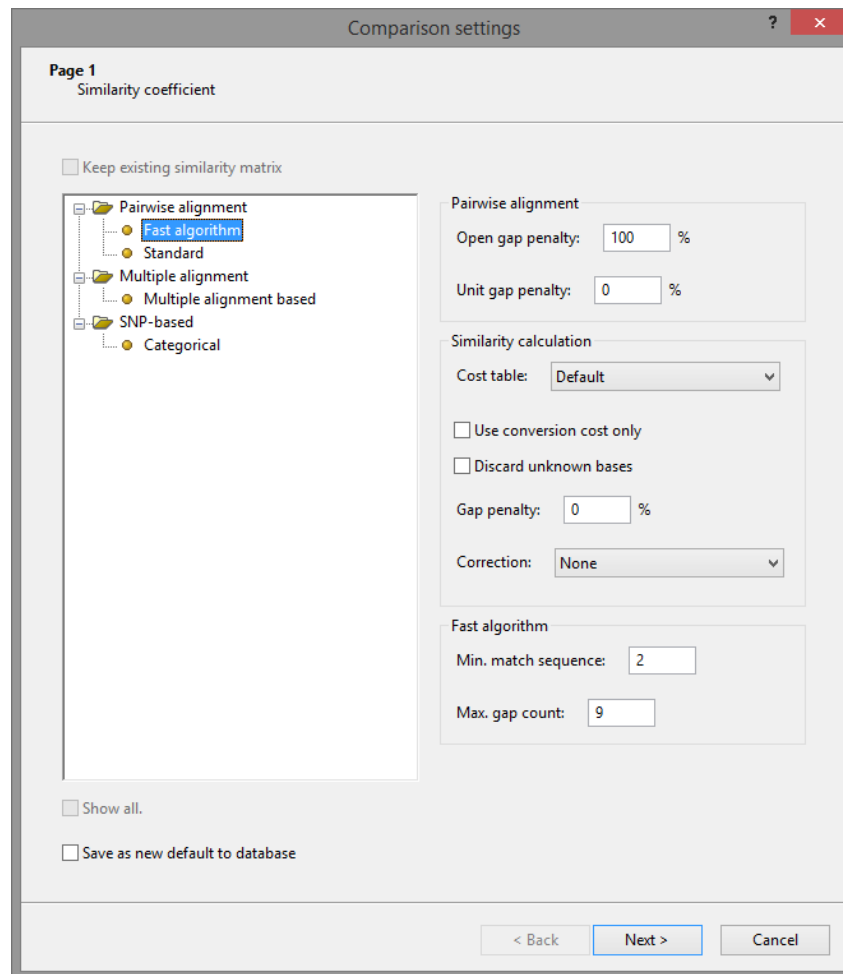


Figure 8.3.2: The *Similarity coefficient* wizard page: Fast algorithm for pairwise alignment.

The *Pairwise alignment* settings involve an *Open gap penalty* and a *Unit gap penalty*. The *Open gap penalty* is the percentage cost of the mismatch cost for introducing one single gap in one of the sequences. The *Unit gap penalty* is the percentage cost of that score to increase the gap by one base position. The default setting is 100% open gap penalty and 0% unit gap penalty, which means that introducing a gap in one of both sequences has the same cost as a mismatch, whereas there is no extra cost for gaps of multiple positions. It should be emphasized that the pairwise alignment settings will only determine the way the alignment is done: if a large unit gap cost is set (e.g. 350%), the program would not easily introduce gaps between sequences; for example, the program would rather allow three successive mismatches than one single gap. If no gap cost is chosen (0%) the program would introduce gaps to match every single base. The pairwise alignment settings have no direct influence on the similarity values, but of course, if the obtained alignments differ, the similarity values may differ too.

Contrary to the pairwise alignment settings, the *Similarity calculation* parameters will not influence the alignments, but determine the way the similarity is calculated. The *Cost table* corrects for differences between the nucleic acids and amino acids when clustering nucleic acid and amino acid sequences, respectively. The *Default* nucleic acid cost table is embedded in the software and assigns the same cost to all possible nucleotide mismatches. The *Default* amino acid table is stored in the text file `DefaultAminoAcids.txt`, which can be found in the `SequenceData \CostTables` folder of the BioNumerics installation directory.

Custom nucleic and amino acid cost matrices stored in this path will also appear in the **Cost table** list. **Use conversion cost only** is a parameter which makes calculation of the pairwise similarity matrix faster. Both alignment methods (standard and fast algorithm) work in two steps: first they determine the total maximal conversion score to convert one sequence into the other (given the current alignment settings) and then they realize the alignment using the minimal gap cost and maximal matching score. If **Use conversion cost** is enabled, the calculated conversion cost is transformed into a similarity value. This method is two times faster than the usual similarity calculation, but the obtained values cannot be described as real "similarities". When **Discard unknown bases** is disabled, the program will use a predefined cost table for scoring uncertain or unknown bases. For example, N with A will have 75% penalty, as there is only 25% chance that N is A. Y and C will be counted 50% penalty because Y can be C or T with 50% chance each. If this setting is enabled, all uncertain and unknown bases will not be considered in calculating the final similarity. The **Gap penalty** is a parameter which allows to specify the cost the program uses when one single gap is introduced. This cost is determined as a percentage of the mismatch cost. The program uses 0% as default. Under **Correction**, one can select the one parameter correction for evolutionary distance, as calculated from the number of nucleotide substitutions as described by Jukes and Cantor [21]. The resulting dendrogram displays a distance scale which is proportional to an evolutionary time, rather than a similarity scale.

The **Fast algorithm** option offers an interesting accelerated algorithm. In addition to the parameters discussed above, two adjustable parameters are displayed in the right panel if the **Fast algorithm** is selected: the **Minimum match sequence** and the **Maximum gap count**. The program creates a lookup table of groups of bases for both sequences. The **Minimum match sequence** is the size of such a group. The smaller the groups are, the more precise the alignment will be, but the longer the alignment will take. The parameter can be varied between 1 and 5, with 2 as default. The **Maximum gap count** is the maximum number of possible gaps that you allow the algorithm to introduce in one of both sequences. The values can be varied between 0 and 99 with 10 as default. The larger the number, the more gaps the program can create to align every two sequences, but the longer the alignment will take. If zero is selected, no gaps at all would be introduced. Thus, you can custom-define its accuracy between very fast and fairly rough, to slow and very accurate.

If you wish to use the similarity matrix that is present in the *Comparison* window for the selected experiment type, then you should enable **Keep existing similarity matrix**.

Checking **Save as new default to database** will save the settings specified in the right panel as the new defaults in the database for the selected experiment type.



When the **Multiple alignment based** option is checked, no alignment is calculated. The calculation of the similarity matrix will be based on the alignment that is present in the *Experiment data* panel. More information can be found in 8.3.13.

Pressing <Next> brings you to the *Cluster analysis* wizard page. This step deals with the calculation of a dendrogram from the similarity matrix and is discussed in 13.2.6.



Only sequences with a maximum length of 200,000 bases can be analyzed in the *Comparison* window if an alignment is needed.

Pressing <Next> in the *Cluster analysis* wizard page starts the cluster analysis. When the calculations are finished, the dendrogram and the matrix are shown in the *Comparison* window. The sequences are still unaligned since no multiple alignment is calculated yet.

If desired, a *Report* window can be displayed with all settings that were used to calculate the matrix and dendrogram, by selecting **Clustering > Similarity matrix > Show information** (📄) (see 13.3.1).

8.3.3 Calculating a multiple alignment

Selecting **Sequence > Multiple alignment...** (📄) calls the *Sequence display settings* dialog box (Figure 8.3.3).

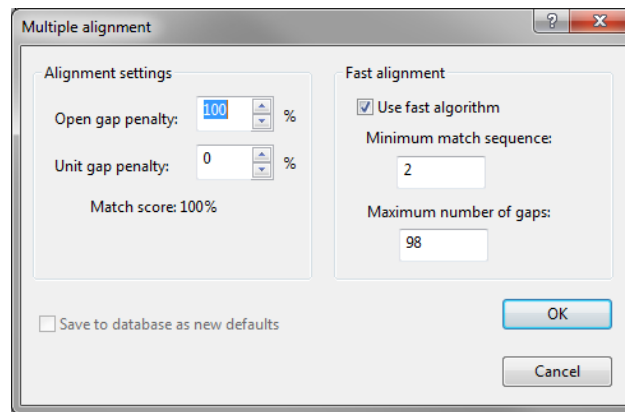


Figure 8.3.3: The *Sequence display settings* dialog box.

The significance of the **Open** and **Unit gap penalties** is the same as explained for pairwise alignment (see 8.3.2): they are the percentage of the mismatch cost to create a gap, and to increase the gap by one base position, respectively. The default setting is 100% for the **Open gap penalty** and 0% for the **Unit gap penalty**, which means that introducing a gap in one of both sequences has the same cost as a mismatch, whereas there is no extra cost for gaps of multiple positions. These alignment settings will only determine the way the alignment of the local consensus sequences is done: if a large unit gap cost is set (e.g. more than 100%), the program would not easily introduce gaps between sequences. If no gap cost is chosen (0%) the program would introduce gaps in order to match single bases. The alignment settings have no direct influence on the similarity values obtained from a global alignment, but if the eventual multiple alignment differs, the derived similarity values may differ too.

Use fast algorithm is an algorithm with two adjustable parameters: the **Minimum match sequence** and the **Maximum number of gaps** (see also under pairwise alignment). The **Minimum match sequence** can be varied between 1 and 5, with 2 as default. The **Maximum number of gaps** can be varied between 0 and 198 with 98 as default. The smaller the first number and the larger the second number, the more accurate the multiple alignment should be. If the default values are not satisfactory, e.g. for very diverse sequences, some experimenting is recommended.



The *Sequence display settings* dialog box does not contain settings for similarity calculation, unlike the *Similarity coefficient* wizard page (see Figure 8.3.2). The similarity matrix based upon the global alignment is not calculated automatically by the program, but requires a further command by the user (see Instruction 8.3.13).

Pressing <OK> starts the calculation of the multiple alignment.

If no pairwise alignment based dendrogram is present in the *Comparison* window for the selected sequence type, BioNumerics will automatically create the pairwise alignment, its associated dendrogram and the multiple alignment by means of this single command. BioNumerics will use default pairwise alignment and pairwise based dendrogram settings. After calculation of the multiple alignment, select **Clustering > Similarity matrix > Show information** () to view these settings.

If a pairwise alignment based dendrogram is present in the *Comparison* window for the selected sequence type, the multiple alignment will be calculated based on this dendrogram.

When the calculations are done, all sequences are aligned in the *Experiment data* panel (see Figure 8.3.4).

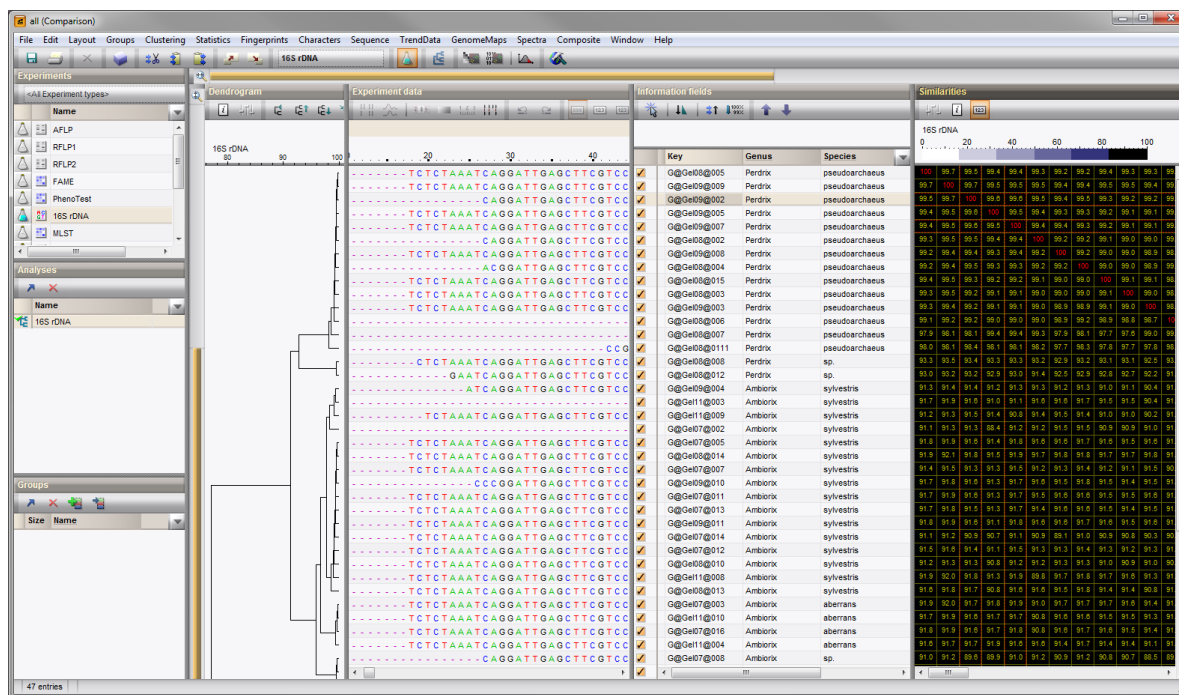


Figure 8.3.4: A Multiple alignment in the *Experiment data* panel.

8.3.4 Multiple alignment display options

With *Sequence > Display settings...*, the general display options such as colors and symbols, can be changed. These display settings are specific to the sequence type and can therefore also be accessed from the *Sequence type* window (see 8.1.2.1).

In order to facilitate visual interpretation of multiple alignments, there are three methods to highlight homologous regions.

Select *Sequence > Block type > Neighbor blocks* to show the *Neighbor match* representation. The neighbor match representation shows bases as blocks (highlighted) if at least one of the neighboring sequences has the same base at the corresponding position. Between two different groups of consensus, a small white line is drawn (Figure 8.3.5).

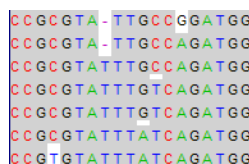


Figure 8.3.5: Neighbor match representation.

The *Consensus match* requires a consensus sequence to be present. A consensus sequence is defined from one or more sequences, and in case a user-defined percentage of the sequences have the same base at a given position, this base will be written in the consensus. With *Sequence > Create consensus of branch* the consensus is based on all entries present in the selected branch. When *Sequence > Create consensus of current selection* is selected, the consensus is based on the sequences of all selected entries in the *Comparison* window.

The dialog box prompts to *Enter minimum consensus percentage*. If a minimum percentage of “50” is specified, a base at a given position will only be shown in the consensus sequence if at least 50% of the

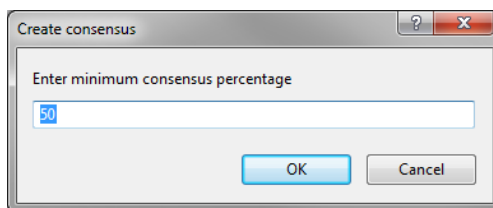


Figure 8.3.6: The *Create consensus* dialog box.

sequences have that base at the given position. A consensus sequence can be copied to the clipboard with **Sequence > Copy consensus to clipboard**.

A consensus sequence is shown on the header of the *Experiment data* panel. Bases for which there is a consensus in at least 50% of the sequences are named, the other bases are unnamed (N).

Select **Sequence > Block type > Consensus blocks** to show the *Consensus match* representation (Figure 8.3.7). The consensus match representation highlights bases (shown as blocks) on the aligned sequences if they are the same as on the consensus sequence.

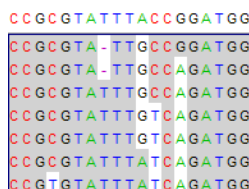


Figure 8.3.7: Consensus match representation.

The *Consensus difference* displays the consensus sequence in the editor caption, and only shows bases that differ from the consensus while bases that are the same as the consensus are shown as |. Use **Sequence > Block type > Consensus difference** to select the consensus difference representation as in Figure 8.3.8.

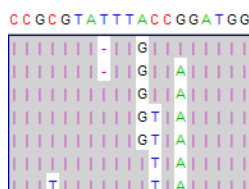


Figure 8.3.8: Consensus difference representation.

8.3.5 Editing a multiple alignment

A multiple alignment can be edited manually and is saved along with the comparison. In order to rearrange the multiple alignment as desired, any sequence can be moved up or down with **Edit > Move entry up** (↑, **Shift+Up**) or **Edit > Move entry down** (↓, **Shift+Down**) respectively.

To move a sequence to the top or the bottom of the alignment, hold the **Shift**-key and press the up or down button, respectively.

As soon as an entry is moved up or down, the dendrogram disappears: a dendrogram imposes a certain order to the entries, which is not compatible with freely moving sequences up or down. You can display the dendrogram again using **Layout > Show dendrogram**, however, this will reorder the entries again so that all manual changes you made to the sequence order are lost.

A number of manual alignment editing tools are described below. For these editing tools, the multiple alignment editor contains a multilevel undo and redo function. The undo function can be accessed with **Sequence > Edit alignment > Undo** (🖱, **Ctrl+Z**), the redo function is accessible through **Sequence > Edit alignment > Redo** (🖱, **Ctrl+Y**). The undo/redo function works for the following sequence editing functions: drag-and-drop realignments (8.3.6), inserting and deleting gaps (8.3.8), removing common gaps (8.3.8), and changing sequence bases (8.3.9). The undo/redo function also works for all automatic alignment functions, including full multiple alignment (Figure 8.3.3) and partial alignments obtained with one of the following commands: **Align internal branch**, **Align external branch**, and **Align selected sequences** (8.3.11).

8.3.6 Drag-and-drop manual alignment

A cursor, visible as a black rectangle can be placed on any base of any sequence, and can be moved up, down, left, and right using the arrow keys.

The cursor can also be extended to cover a range of bases both in the vertical and the horizontal direction. This can be achieved by holding down the **Shift**-key while pressing the arrow keys. The result is that blocks of bases can be selected as shown in Figure 8.3.9. By dragging the mouse towards the left or the right, the block of bases can be realigned within the alignment.

Figure 8.3.9: Selecting blocks of bases for drag-and-drop manual alignment.

While moving the block it remains displayed so that the user can see the resulting alignment at each position. The realignment is executed as soon as the mouse button is released. If necessary, the block can be moved over other bases at the left or right side. This will then force a gap to be introduced in the sequences up and down from the block, in order to both preserve the original alignments left and right from the block, and align the block the way the user has forced it to.

A useful tool to select a group of identical bases at once is to click on one of the bases and choose **Sequence > Edit alignment > Highlight identical positions** (**Ctrl+Shift+E**).

8.3.7 Inserting and deleting gaps

Besides the drag-and-drop realignment tool (see 8.3.6), a number of menu-items and corresponding keyboard shortcuts are available to manually edit a multiple alignment (see Figure 8.3.10). Using the editing tools listed below, all changes made to a sequence, i.e. inserting gaps or deleting gaps, cause shifts no further than the next gap. You can consider an aligned sequence as a series of blocks with some space in between (the gaps), just like carriages on a railway: if one block is shifted to the right, it will move alone until it touches the next block, which will then move together, until they touch the next block etc.

Following manual alignment editing tools are available:

Sequence > Edit alignment > Insert gap to the right (Ins): Inserts a gap at the position of the cursor, by shifting the block right from the cursor position to the right. This function can be used on a gap as well as

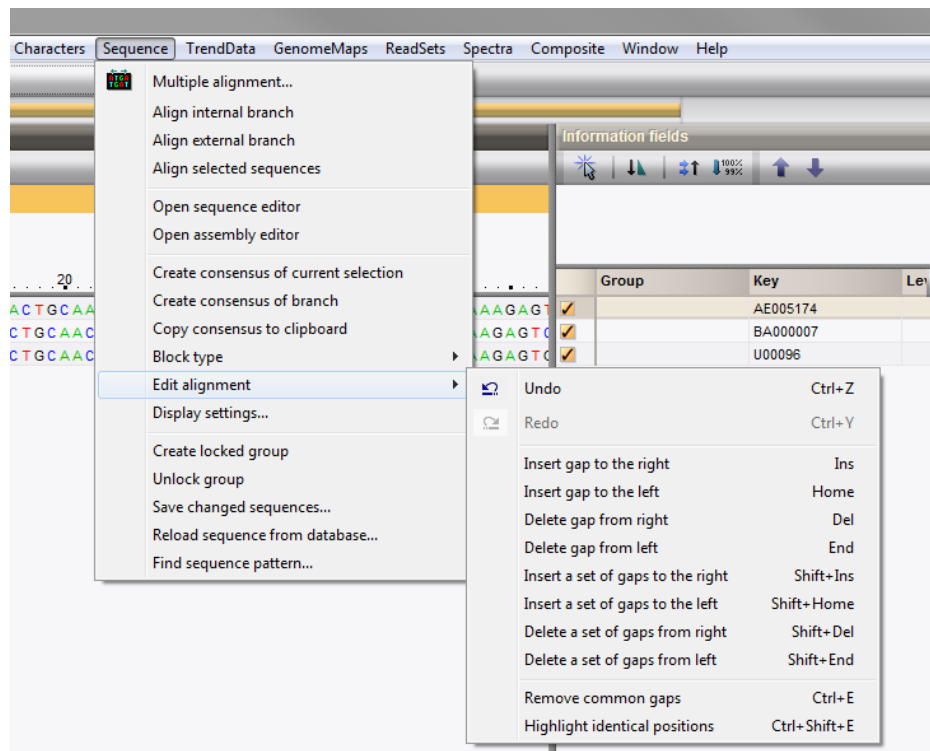


Figure 8.3.10: Menu-items to insert and delete gaps.

on a base. In the latter case, the base at the cursor position will also shift to the right, i.e. the gap will be inserted left from it.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCG-GT--ACC-TCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCG-GT--ACC-TCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Insert gap to the left (Home): Inserts a gap at the position of the cursor, by shifting the block left from the cursor position to the left. This function is similar to the previous function.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCG-GT--ACC-TCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCG-GT--ACC-TCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Insert a set of gaps to the right (Shift+Ins): Inserts gaps at the position of the cursor, by shifting the block right from the cursor position to the right, until it closes up with the next block.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCGG-ACC---TCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATGTCCTTA
TCTTACCGG---ACC-TCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Insert a set of gaps to the left (Shift+Home): Inserts gaps at the position

of the cursor, by shifting the block left from the cursor position to the left, until it closes up with the next block.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGGTACC---TTCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Delete gap from right (Del): Deletes a gap by shifting the block right from the gap to the left.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Delete gap from left (End): Deletes a gap by shifting the block left from the gap to the left.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Delete a set of gaps from right (Shift+Del): Deletes all gaps right from, and including the cursor, by shifting the block right from the gap to the left.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Sequence > Edit alignment > Delete a set of gaps from left (Shift+End): Deletes all gaps left from, and including the cursor, by shifting the block left from the gap to the right.

Example:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TCTTACCGG---TACC-TTCTGGCTG-TGGTCCTTA
```

Result:

```
TTGGACCGGGAAAAATTTCTTTCTGATAGTCCTTA
TATCTTACCG---TACC-TTCTGGCTG-TGGTCCTTA
```

To insert and delete gaps or move blocks of a group of sequences as a whole, it is possible to *lock* sequences in the *Comparison* window. Groups of locked sequences are created as follows: select a consecutive group of entries and select **Sequence > Create locked group**. Locked sequences are connected by a red brace in the *Dendrogram* panel. All sequences that are part of the locked group are moved/deleted as a whole.



When the dendrogram is shown, the locked groups will not be seen anymore. When the dendrogram is removed again, the locked groups become visible and active again.



Locked groups are not taken into account when using the drag-and-drop alignment tool described in 8.3.6. Only when using the editing tool buttons or keyboard shortcuts, locked sequences are considered as one block.

To unlock locked groups of sequences, click on any of the entries within the group, and select **Sequence > Unlock group**.

8.3.8 Removing common gaps in a multiple alignment

After a series of manual realignments, it may be possible that the multiple alignment contains one or more common gaps, i.e. gaps that occur over all sequences. Instead of having to remove those gaps for all sequences separately, the user can let the software find and remove all common gaps automatically.

To remove common gaps automatically, select **Sequence > Edit alignment > Remove common gaps (Ctrl+E)**.

8.3.9 Editing sequences in a multiple alignment


In some cases, it is possible that ambiguous positions in certain sequences can be filled in when a multiple alignment of highly homologous sequences is present. BioNumerics offers the possibility to change bases in sequences within a multiple alignment.

To change a base in the multiple alignment, place the cursor on the base in the alignment, hold the **Ctrl**-key and type a base letter, a space (gap) or any letter corresponding to the IUPAC nucleotide naming code. The sequence is now changed in the multiple alignment, but not yet in the BioNumerics database.

In order to reload the original sequence, select **Sequence > Reload sequences from database...**. The existing alignment will be preserved. To save the changed sequence(s) to the database, select **Sequence > Save changed sequences...**

As an alternative, a base in a sequence can also be changed by selecting **Sequence > Open sequence editor**, which will pop up the *Sequence editor* window of the sequence. Simply double-clicking that base will do the same thing. If the **Shift**-key is pressed while executing this command, the *Experiment card* window will pop up instead of the *Sequence editor* window. To change the base, simply type in another base from the keyboard. Upon exiting the *Experiment card* window or *Sequence editor* window, the software will ask to save the changes. These changes are immediately updated in the multiple alignment.



If the sequence was assembled from trace files using the Assembler or Power Assembler program, it is *NOT* recommended to modify the sequence in the *Comparison* window, *Experiment card* window or *Sequence editor* window. Instead, open the (Power) Assembler by selecting **Sequence > Open assembly editor** in the *Comparison* window or pressing the  button in the caption of the *Experiment card* window or in the toolbar of the *Sequence editor* window, and change the base there. When the assembly is saved, the *Comparison* window is automatically updated.

8.3.10 Finding a subsequence

In order to find certain subsequences in a sequence from a multiple alignment, e.g. restriction sites, primer sequences, repeat patterns etc., you can perform a subsequence search.

Select **Sequence > Find sequence pattern...** to pop up the *Find pattern* dialog box (Figure 8.3.11).

As **Sequence to search for**, you can enter any sequence including unknown positions, which are entered as a question mark. You also can allow a number of mismatches to occur in matching subsequences, by specifying a number under **Mismatches allowed**.

For rare subsequences which you do not expect to occur more than once, select **Complete sequence**. For

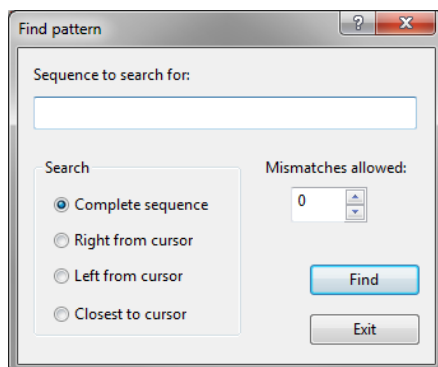


Figure 8.3.11: The *Find pattern* dialog box.

frequently occurring subsequences, you can place the cursor at the start of the sequence, and check ***Right from cursor***. By successively pressing <***Find***>, all subsequent matching patterns will be shown. Similarly, ***Left from cursor*** shows the first matching pattern left from the cursor, whereas ***Closest to cursor*** only shows the matching pattern closest to the cursor, in any direction.

8.3.11 Adding entries to and deleting entries from a multiple alignment

The feature of BioNumerics that makes it possible to add entries to (or delete entries from) an existing cluster analysis also applies to sequence clustering: it is not necessary to recalculate the complete similarity matrix because the program can calculate the similarity of the new sequence(s) with each of the other sequences and will add these new similarity values to the existing matrix. Particularly in case of sequence clustering, this feature is extremely time-saving and causes no degeneration of the clustering.

In case a multiple alignment exists, the problem is slightly more complex. As soon as sequences are added, the program will have to recalculate the multiple alignment (steps 3 and 4 of scheme in Figure 8.3.1) to find the optimal alignment again for the new set of sequences. This could cause corrections in the alignment made by the user to be lost each time sequences are added. Therefore, the program offers some additional features to add sequences to existing multiple alignments without affecting the existing alignment, including manual corrections.

When cutting entries from the analysis with ***Edit > Cut selection*** (✂, **Ctrl+X**), the dendrogram that is present in the *Dendrogram* panel is recalculated immediately, and the multiple alignment is preserved. Gaps that occur in all sequences in the alignment can be removed with ***Sequence > Edit alignment > Remove common gaps*** (**Ctrl+E**).



A dendrogram based on a multiple alignment is not displayed any more when entries are removed from or added to an existing multiple alignment.



If the dendrogram that is displayed in the *Dendrogram* panel is not the last dendrogram that was calculated for the experiment type, the dendrogram will be removed from the comparison if entries are added to (or deleted from) the existing cluster analysis. The software will warn you for this.

When pasting the selection (which is still on the clipboard) again with ***Edit > Paste selection*** (📋, **Ctrl+V**) into the analysis, the matrix based upon pairwise alignments and the corresponding dendrogram are being updated, and when finished, the pasted sequences are shown in the multiple alignment. However, the program has **NOT** aligned them.

If the pasted sequences constitute one single branch, select that branch on the dendrogram, and choose ***Sequence > Align internal branch***. The sequences within the branch are now being aligned internally, and

once this is finished, select **Sequence > Align external branch** to align the sequences from the rest of the dendrogram with the selected branch.

The advantage of this approach is that by using **Sequence > Align internal branch**, only the sequences within the selected branch are aligned. This is useful to update a part of a multiple alignment without affecting the non-selected branches. With **Sequence > Align external branch**, the selected branch is aligned to the rest of the dendrogram as a whole: all sequences within the branch are treated as one block, and all the other sequences are treated as another block. The two blocks are aligned to each other. These features give the user full control over how new sequences are added to a multiple alignment without affecting any editing.

A similar result can be obtained with the **Sequence > Align selected sequences** function (see 8.3.12).

8.3.12 Automatically realigning selected sequences

With the function **Sequence > Align selected sequences**, any set of selected sequences can be realigned within an existing multiple alignment. The function preserves any automatic and manual alignment that exists between all the non-selected sequences, which are treated as one block. The selected sequences are aligned one by one to the non-selected sequences. The difference with the method described in 8.3.11 is that the new sequences are not first aligned among each other, which may produce a slightly different result.

8.3.13 Calculating a clustering based on a multiple alignment

Based on a multiple alignment, calculated as described in 8.3.2, a global clustering can be calculated via **Clustering > Calculate > Cluster analysis (similarity matrix)...**

In the *Similarity coefficient* wizard page that appears, select the **Multiple alignment based** option under **Multiple alignment** (see Figure 8.3.12).

The similarity calculation settings for the **Multiple alignment based** option are shown in the right panel of the dialog box. The calculation of the similarity matrix will be based on the alignment that is already present in the *Experiment data* panel, so no alignment settings are present.

The **Similarity calculation** settings determine the way the similarity is calculated between the pairs of sequences. The **Cost table** corrects for differences between the nucleic acids and amino acids when clustering nucleic acid and amino acid sequences, respectively. The **Default** nucleic acid cost table is embedded in the software and assigns the same cost to all possible nucleotide mismatches. The **Default** amino acid table is stored in the text file `DefaultAminoAcids.txt`, which can be found in the `SequenceData \CostTables` folder of the BioNumerics installation directory. Custom nucleic and amino acid cost matrices stored in this path will also appear in the **Cost table** list. When **Discard unknown bases** is disabled, the program will use a predefined cost table for scoring uncertain or unknown bases. For example, N with A will have 75% penalty, as there is only 25% chance that N is A. Y and C will be counted 50% penalty because Y can be C or T with 50% chance each. If this setting is enabled, all uncertain and unknown bases will not be considered in calculating the final similarity. The **Gap penalty** is a parameter which allows you to specify the cost the program uses when one single gap is introduced. This cost is relative to the score the program uses for a base matching, which is equal to 100%. The program uses 0% as default. Under **Correction**, one can select the **Jukes and Cantor** correction [21], a *one parameter* correction for the evolutionary distance as calculated from the number of nucleotide substitutions. Alternatively, the **Kimura 2 parameter** correction [22] can be selected. In either case, the resulting dendrogram displays a distance scale which is proportional to an evolutionary time, rather than a similarity scale. The check box **Use active zones only** is only applicable when a reference sequence is defined, and when certain zones on this reference sequence are excluded for analysis (see 8.1.2.3 for more information on excluding regions for comparison).

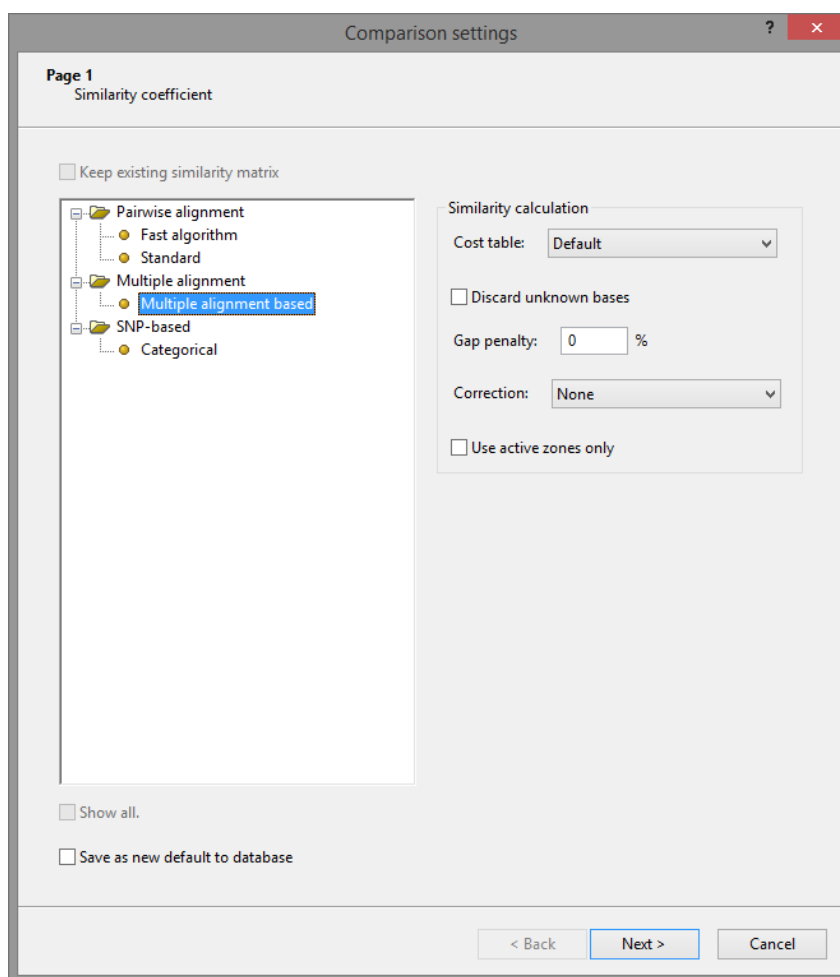


Figure 8.3.12: The *Similarity coefficient* wizard page: Multiple alignment based.

Checking **Save as new defaults to database** will save the settings specified in the right panel of the wizard as the new defaults in the database.

Pressing **<Next>** in the *Similarity coefficient* wizard page will display the second page, which groups the options related to the clustering algorithms. This step is discussed in 13.2.6.

Select a clustering method (e.g. **Neighbor Joining**) and press **<OK>** to start calculating the multiple alignment-based dendrogram. This calculation is usually fast. The clustering based on the multiple alignment will be displayed in the *Dendrogram* panel.

Selecting **Clustering > Similarity matrix > Show information** (i) will pop up a report, listing all settings that were used to calculate the dendrogram (see 13.3.1).

Selecting **File > Save** (S, **Ctrl+S**) will save the clustering based on the multiple alignment along with the comparison.

8.3.14 Exporting a multiple alignment

Aligned or unaligned sequences in a comparison can be exported as text file with the command **File > Export > Export sequences (tabular)...** The program now asks "Do you want to export the database fields?". Answer **<Yes>** to export tab-delimited database fields along with the sequences. Next, the program asks "Do you want to include regions with gaps?". Answer **<Yes>** if you want to preserve the gaps introduced in

the multiple alignment. This allows aligned sequences to be exported from BioNumerics to other software applications. Gaps are represented as spaces.

A more advanced way to export alignments is provided by **File > Export > Export sequences (formatted)...**, which displays the *Formatted sequence export* dialog box (Figure 8.3.13).

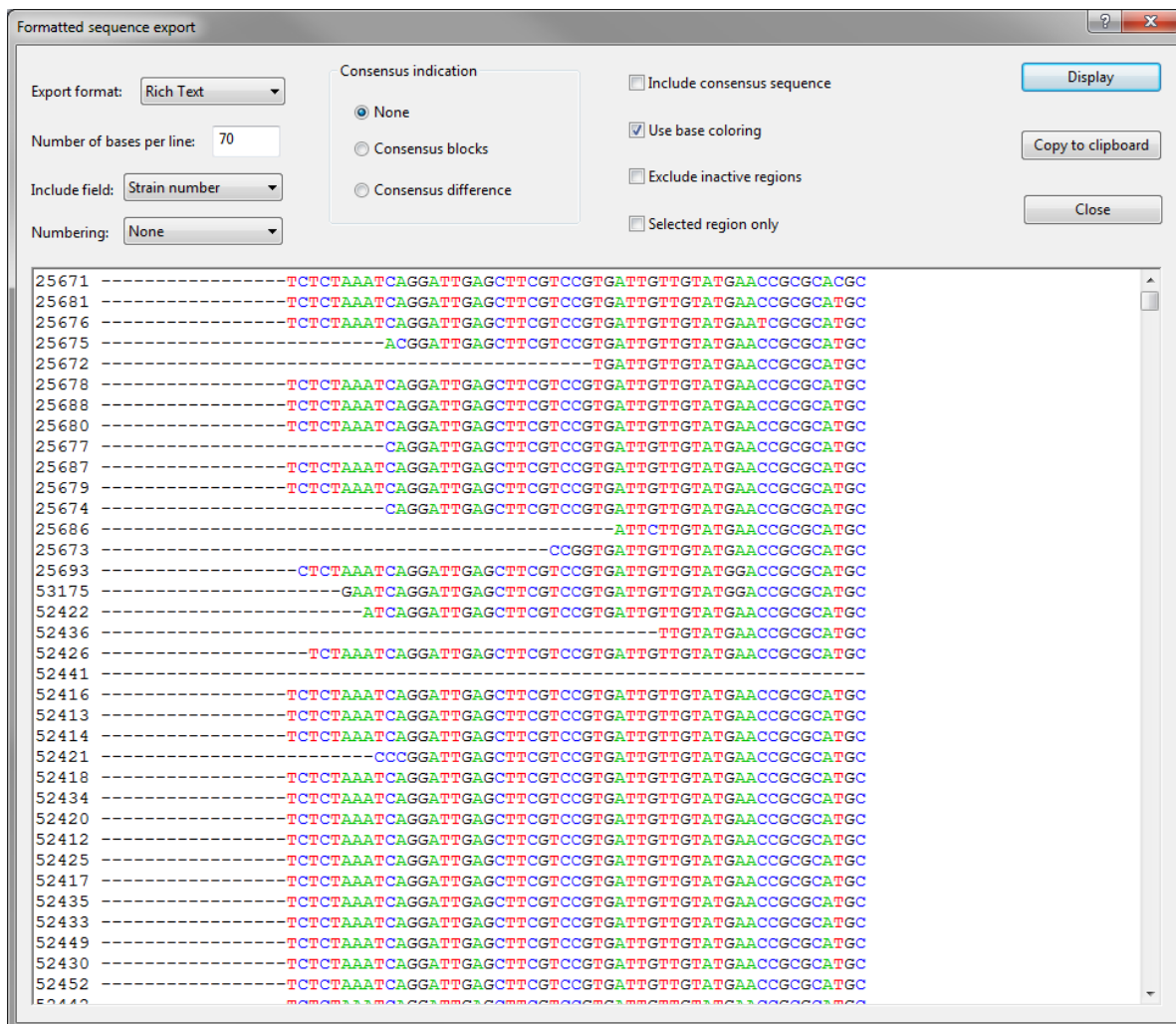


Figure 8.3.13: The *Formatted sequence export* dialog box.

This tool allows alignments to be exported in blocks of a defined number of bases, so that multiple blocks can be presented underneath each other on the same page. Initially, the dialog box is empty; the view is updated by pressing the **<Display>** button.

The *Formatted sequence export* dialog box has the following options:

- **Export format** can be *Raw text* or *Rich text*. The latter will export the alignment as RTF, which allows text to be formatted. When *Raw text* is chosen, some options such as consensus blocks and base coloring do not apply.
- **Number of bases per line** determines the number of base positions to be displayed in one line block. If the total alignment is longer than the number specified, subsequent blocks of the same length will be displayed beneath each other.
- **Include field** allows an information field to be displayed left from the alignment.
- **Numbering** displays a position numbering above the alignment blocks. With *Decimals*, only multi-plets of 10 are indicated, whereas with *All numbers*, multi-plets of 10 are indicated as an upper line,

and unit numbering is indicated in a lower line.

- **Consensus indication** allows the **Consensus blocks** or the **Consensus difference** (if the consensus is calculated in the *Comparison* window) to be indicated (rich text only).
- With **Include consensus sequence**, the consensus sequence is shown on top of the alignment blocks.
- **Use base coloring** will display the bases in color (only rich text mode).
- **Exclude inactive regions** is to exclude regions on the reference sequence that are marked as inactive (see 8.1.2.3).
- **Selected region only** will only display and export the region in the alignment that is selected using the block selection tool (see 8.3.6). Only the horizontal selection is taken into account (base positions); all sequences are exported regardless of the vertical selection (entries).

The view is only updated after pressing the **<Display>** button.

With **<Copy to clipboard>**, the current view is copied to the clipboard of your operating system and can be pasted in other programs. Depending on the choice, the alignment will be exported as RTF or flat text.

8.3.15 Converting sequence data to categorical character sets

Nucleic acid or amino acid sequence data can be converted into categorical character data, whereby each nucleic acid or amino acid is represented by an integer number (e.g. A = 1, C = 2, G = 3, and T = 4 for DNA sequences). The converted categorical data can be visualized and analyzed as a composite data set (see 11). The possibility to convert sequences into categorical character sets requires that a multiple alignment is calculated from the sequences, and also, that a composite data set exists which includes the sequence type (exclusively or in combination with other sequence types).

In addition to the states displayed in the composite data set, an extra state (zero) can optionally be assigned to a gap position. As another option, it is possible to consider only the mutating positions, i.e. the positions that differ in at least one sequence from the others.



If nucleic acid sequences containing IUPAC code are converted into categorical character sets, each of the ambiguous bases (e.g. M, R, Y, etc.) becomes an additional state.

The settings for converting sequences into character data can be changed in the *Sequence type* window: Choose **Settings > Character conversion settings...** (🔧) to open the *Sequence display settings* dialog box (see Figure 8.3.14).

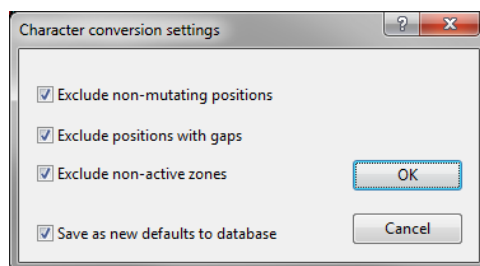


Figure 8.3.14: The *Character conversion settings* dialog box.

With the option **Exclude non-mutating positions**, only those positions in a multiple alignment that do not contain the same character for all sequences will be included in the character set.

With the option **Exclude positions with gaps**, those positions where one or more sequences have a gap in the multiple alignment will be excluded from the character set. If gaps are not excluded, an extra state is assigned to gaps (zero).

With the option **Exclude non-active zones**, the non-active defined zones will be excluded from the character set. This option is only applicable when a reference sequence is defined, and when certain zones on this reference sequence are excluded for analysis (see 8.1.2.3 for more information on excluding regions for comparison).

Converting sequences into character data can have several useful applications:

- Minimum spanning trees (see 16.4.4) can be calculated from the composite data set, thus allowing sequence data to be analyzed using MSTs.
- Different genes, each represented in a separate sequence type, can be combined in one composite data set, so that the information from the different genes can be condensed in one single dendrogram. The option **Exclude non-mutating positions** (Figure 8.3.14) thereby offers the possibility to reduce the amount of information to only those positions that are polymorphic in the entries analyzed.
- In addition to clustering the entries, it is also possible in the composite data set to cluster the nucleic acid or amino acid positions, using the transversal clustering method (see 11.2.5). The result looks like in Figure 8.3.15, where groups of characters are clustered together according to their discriminatory behavior between groups of entries.

In this view (Figure 8.3.15), the characters can be shown as letters (default, to be obtained with **Composite > Show presence/absence** (📄)), as colors (using **Composite > Show quantification (colors)** (🎨)), or as numbers (with **Composite > Show quantification (values)** (📊)).

In the numbered view, A is 1, C is 2, G is 3, and T is 4; a gap is zero. In the default color view, A is shown in magenta, C in green, G in orange and T in red; a gap is blue. However, the colors can be modified in the *Preferences* window (see 2.3.3).

8.3.16 Writing comments in an alignment

In order to mark special regions on the reference sequence or on the multiple alignment, a simple comment editor allows you to add any comment to the comparison. The comments can only be added when a reference sequence is present, when the sequences are aligned, and when a consensus sequence is shown. The comment line is saved along with the sequence type, and new comments can be added at any time.

When one of the aligned sequences in the image is clicked, you can start writing comments at the cursor position. The comments appear in the image header, above the consensus sequence (Figure 8.3.16). Any character input is supported. A's, C's, G's and T's are written in the colors of the bases.

To delete a comment, place the cursor on any sequence at the position of the first character of the comment and enter spaces.

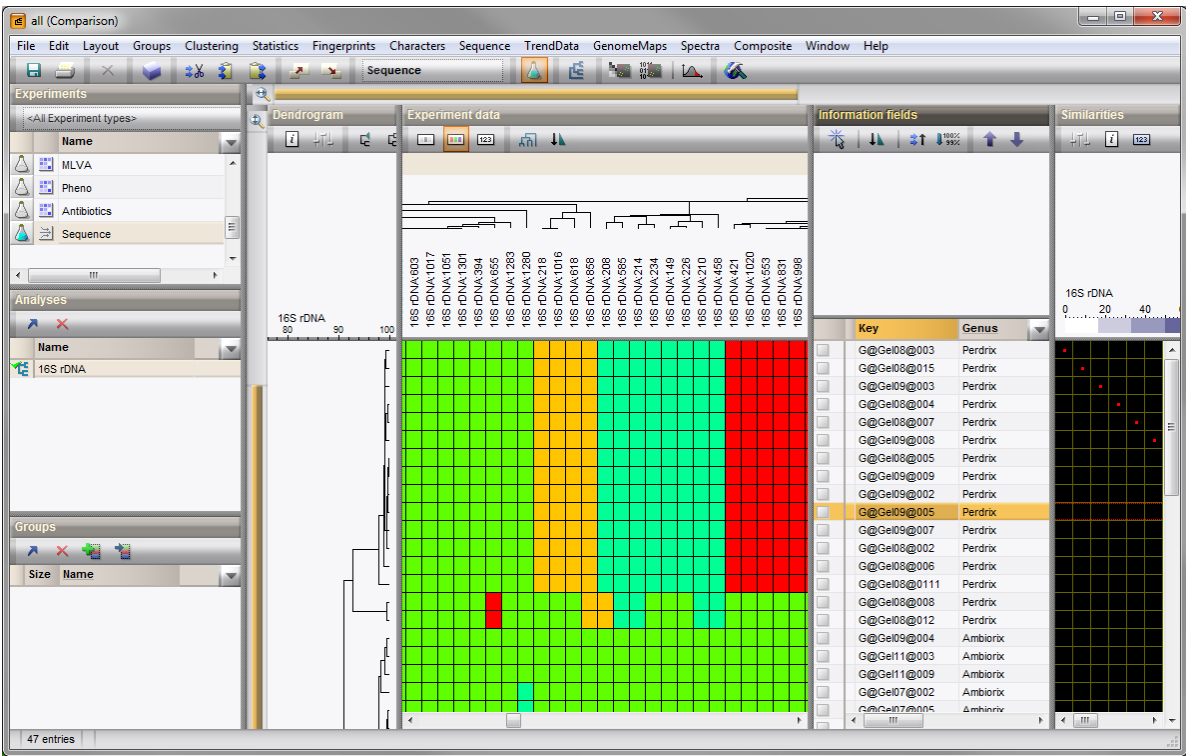


Figure 8.3.15: Comparison window showing a composite data set generated from a sequence type. Bases were converted into categorical characters and clustered in both directions.

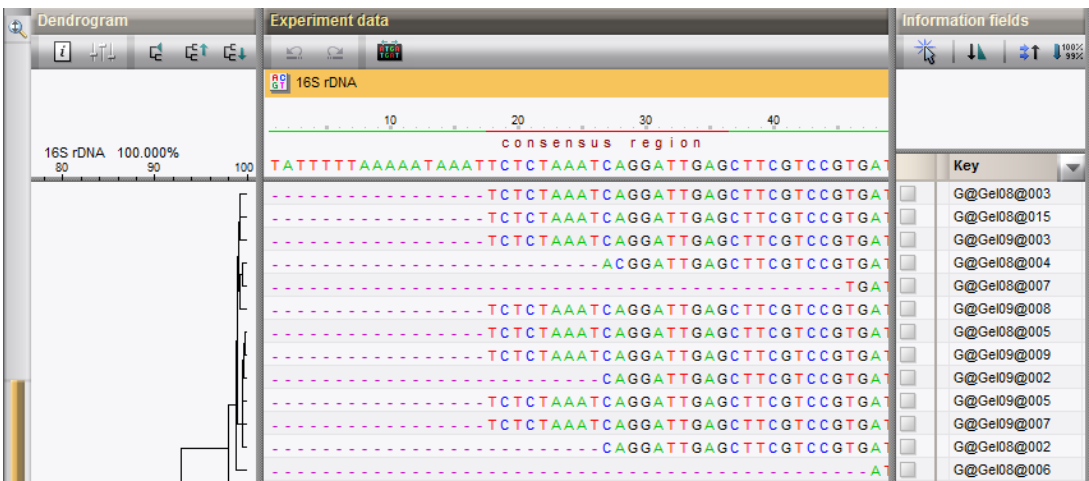




Figure 8.3.16: Comparison window showing excluded/included regions and comments.

Chapter 8.4

Sequence alignment and mutation analysis


8.4.1 Introduction

The *Sequence alignment* window is a convenient tool for the calculation of multiple sequence alignments, subsequence searches and mutation analysis. In this respect, it forms a more powerful alternative to the sequence analysis features available in the *Comparison* window. The *Sequence alignment* window allows different views of the alignment to be displayed in different panels, for example a panel with the chromatograms (curves) and a panel with the bases. The cursor position is synchronized between the different panels. This feature, together with the fact that curves can be displayed for the trace files in the contigs, allows for a quick and reliable evaluation of the correctness of positions of interest, including mutations and single nucleotide polymorphisms (SNPs).


To be able to work with the *Sequence alignment* window, the Sequence data module () and the Tree and network inference module () need to be present in your BioNumerics configuration.



8.4.2 Creating a new alignment project

In the *Main* window, the *Alignments* panel is displayed in default configuration as tabbed view with the *Comparisons* panel, *Decision networks* panel, *Chromosome comparisons* panel, *Annotations* panel and *BLAST projects* panel in the bottom right part of the window. If desired, the configuration of the *Main* window can be customized as described in 2.3.4. In the manual, however, the default configuration will be used.

To create a new alignment project, select the *Alignments tab* in the *Main* window and select **Edit > Create new object...** () . A name for the new alignment project is prompted for.

The new alignment project is added to the *Alignments* panel in the *Main* window. The date on which the annotation project was created and last modified is displayed in the default information fields 'Creation date' and 'Modified date' respectively. When more than one alignment project is present, projects can be sorted and searched using the information present in the default or user-defined information fields. For a detailed explanation of the display options of the *Alignments* panel and other grid panels, see 3.2.7.

To delete one or more alignment projects from the list, select the project and select **Edit > Delete selected objects...** () .

Choose **Edit > Open highlighted object...** (, **Enter**) to open a selected alignment project. The first time an alignment project is opened, it will open with the currently selected entries in the *Database entries* panel. As soon as an alignment project has been saved, selecting **Edit > Open highlighted object...** (, **Enter**)

will open the alignment project with the entries that were present when the alignment project was last saved.

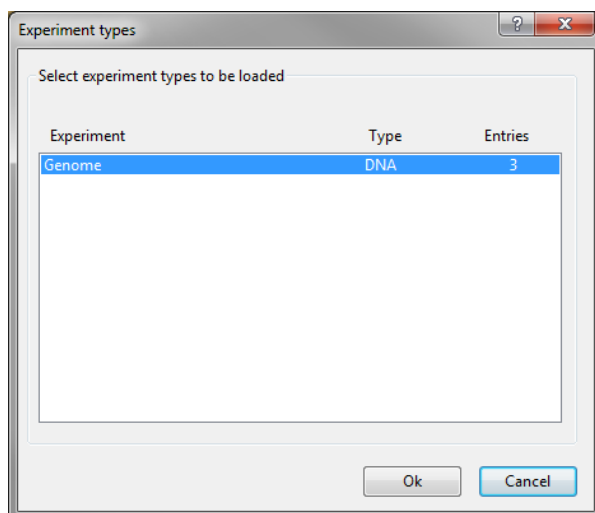


Figure 8.4.1: Select experiment(s) to be included in the alignment project.

The *Experiment types* dialog box displays a list of available sequence types and the number of associated entries. From this list, the user can select the experiment type(s) that should be included in the alignment project. Pressing <OK> opens the alignment project in the *Sequence alignment* window.

8.4.3 The Alignment window

The *Sequence alignment* window (see Figure 8.4.2) consists of eight panels: the *Dendrogram* panel, *Sequence display 1* panel, *Information fields* panel, *Similarities* panel, *Sequence display 2* panel, *Sequence search results* panel, *Mutation listing* panel, and *Bookmarks* panel. All panels are dockable, which enables the user to customize the layout of the *Sequence alignment* window according to personal preference and/or the type of analysis to be performed. See 2.3.4 for detailed information on the display options of dockable panels.



The *Dendrogram* panel, *Sequence display 1* panel, *Information fields* panel and *Similarities* panel behave as a group, i.e. these panels cannot be docked outside this group and they cannot be displayed in a separate window (undocked).

- *Dendrogram* panel: Displays a dendrogram calculated on the sequences present in the alignment project and for the sequence type selected. See 8.4.12 for dendrogram-related tools.
- *Sequence display 1* panel: Displays the nucleic acid sequences present in the alignment. Optionally, the corresponding curves and the translated (amino acid) sequences can be shown as well. See 8.4.8 for display options of the *Sequence display 1* panel.
- *Information fields* panel: This object grid panel displays the entry information fields in tabular format, similar to e.g. the *Database entries* panel in the *Main* window or the *Information fields* panel in the *Comparison* window (for display options of object grid panels, see 3.2.7).
- *Similarities* panel: Displays the matrix of similarity values, calculated on the sequences present in the alignment. This panel is similar to the *Similarities* panel of the *Comparison* window. See 13.3.2 for matrix display functions.
- *Sequence display 2* panel: Is nearly identical to the *Sequence display 1* panel except that it displays by default the curves (chromatograms) instead of the sequences. See 8.4.8 for display options of the

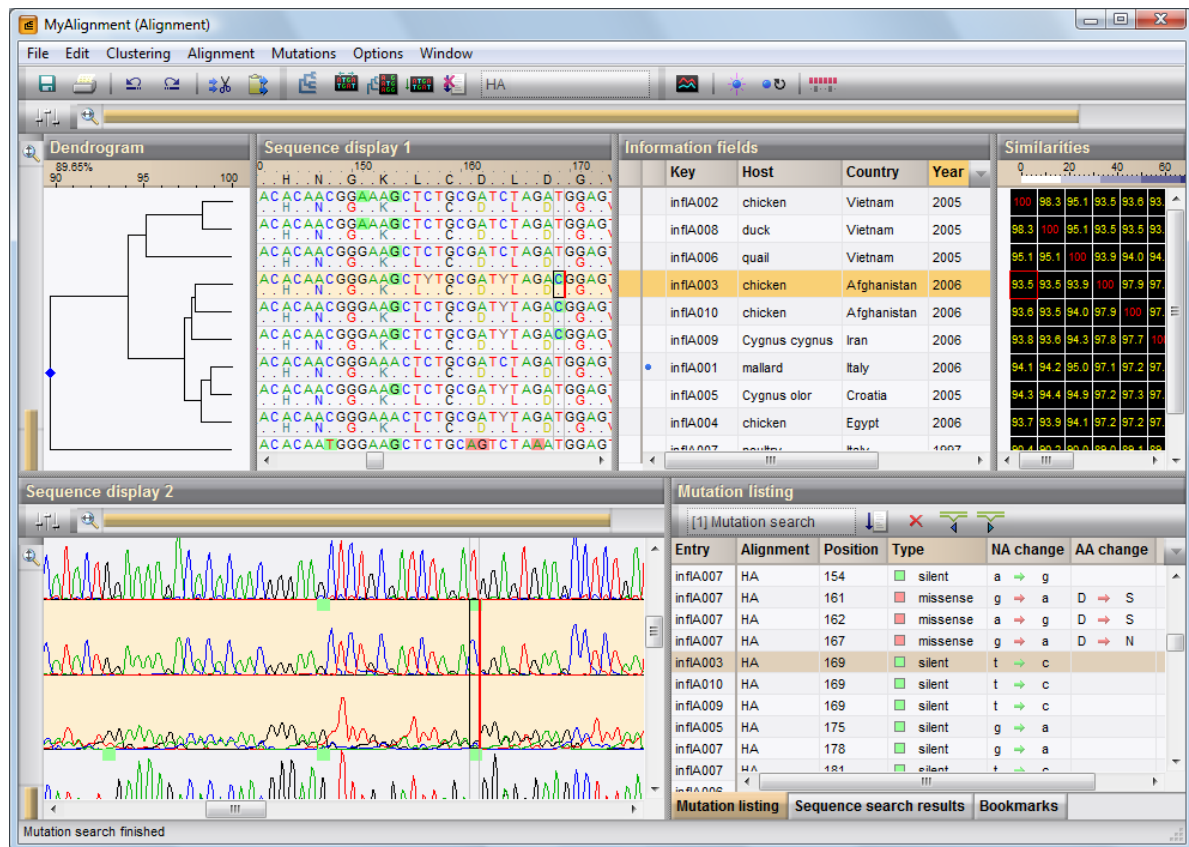


Figure 8.4.2: The *Sequence alignment* window with example data displayed.

Sequence display 2 panel. Since two sequence display panels are available, sequences and curves can be displayed simultaneously, each in their respective panel.

- *Mutation listing* panel: Allows a search to be launched of a selection of sequences from the alignment against a consensus sequence. When a search is performed, it lists all mutations for that search. See [8.4.19](#) for detailed information.
- *Sequence search results* panel: Allows a search for a subsequence to be launched for a selection of sequences from the alignment. When a search is performed, it lists all occurrences of this subsequence. See [8.4.18](#) for detailed information.
- *Bookmarks* panel: Lists all bookmarks that are added to sequences in the alignment. See [8.4.20](#) for detailed information.

The upper part of the *Sequence alignment* window contains the main menu and toolbars. The latter can be displayed or hidden according to your preferences. See [2.3.5](#) for display options of toolbars.

You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally.

In the *Information fields* panel, you can drag the separator lines between the information field columns to the left or to the right, in order to divide the space among the information fields optimally.

Clicking the column properties button (⌵) located on the right hand side in the information fields header in the *Information fields* panel gives access to functions to hide, freeze, or move information fields (see [3.2.7](#) for details).

The zoom sliders can be used to zoom selectively in the horizontal or vertical direction, respectively. See [2.3.7](#) for a detailed description of zoom slider functions.

8.4.4 General functions

An alignment project is saved with **File > Save project** (📁, **Ctrl+S**). When an alignment project is saved, all calculations done on the sequences it contains will be stored along. This includes similarity matrices and dendrograms for all sequence alignments. The alignment project will be saved by default within the connected database, so that it can be shared between users.

To reset the active sequence alignment, choose **Alignment > Reset**. The *Dendrogram* panel and *Similarities* panel will be emptied and sequences in the alignment project will be unaligned. In case position-based search results are mapped on the alignment, e.g. sequence searches (see 8.4.18), mutation listings (see 8.4.19) or bookmarks (see 8.4.20), the program will warn about this under default general settings. If you confirm the action, the position-based search results will be lost.

A copy of the active alignment can be created with **Alignment > Create copy**. This option allows the user to keep the current alignment unaltered, while at the same time, alternative settings can be evaluated on a copy of the alignment.

The active alignment can be selected from the drop-down list in the main toolbar of the *Sequence alignment* window.

Sometimes, the user is specifically interested in a certain region within an existing alignment. It is possible to create a new alignment, containing only this specific region. Using the mouse, a region can be selected from the alignment (sequence block) and with **Alignment > Create subset** a subset is created only containing the selected region. Similar as for a copy, the active alignment can be selected from the drop-down list in the main toolbar of the *Sequence alignment* window.

An active alignment is deleted with **Alignment > Delete**. If only one alignment is present (the active one), all entries will be removed from the alignment project upon executing this command.

A number of general settings for the alignment project can be accessed via **File > Settings....** This pops up the *General settings* dialog box (see Figure 8.4.3).

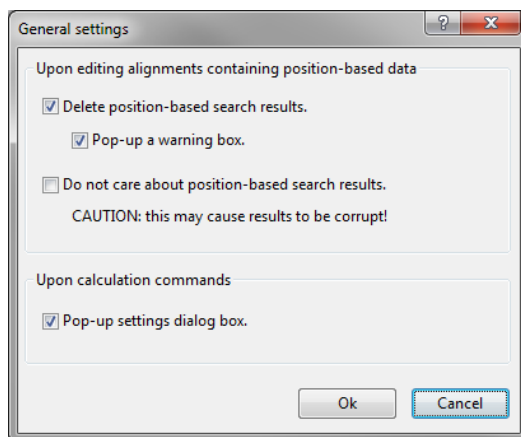


Figure 8.4.3: The *General settings* dialog box.

If **Delete position-based search results** is checked, sequence searches, mutation listings and bookmarks will be deleted when an alignment is changed, e.g. by recalculating a dendrogram, swapping branches of a dendrogram, or manual editing of an alignment. In case **Pop-up a warning box** is checked, the program will display a warning and will allow the user to cancel the operation prior to deleting the position-based search results. If **Do not care about position-based search results** is checked, the program will continue to display the sequence searches, mutation listings and bookmarks, even though the actual positions may not correspond anymore.

The setting **Pop-up settings dialog box (Upon calculation commands)** is checked by default. If it is

unchecked, commands to calculate alignments and/or clusterings will be executed using the last specified settings, without first displaying the corresponding settings dialog box.

In a multiple alignment with highly homologous sequences, it is possible that an ambiguous position in a certain sequence can be filled in.


To edit a sequence, place the cursor on the position to be edited (either in the *Sequence display 1* panel or *Sequence display 2* panel), right-click and select **Open sequence editor** from the floating menu.


The *Sequence editor* window opens. It is possible to edit the sequence directly in the *Sequence editor* window. However, this way of working is not advised when the sequence was imported as trace files via the Assembler or Power Assembler program, as this will break the link with the assembly. A better option is to edit the sequence in the (Power) Assembler program. The sequence can be edited and saved as described in 8.1.3.7. The edited sequence is automatically updated in the *Sequence alignment* window. Position-based search positions will be lost.

8.4.5 Adding and removing entries

Selections of entries made in the *Database entries* panel of the *Main* window are also shown in the *Information fields* panel of the *Sequence alignment* window and vice versa. The entries in a newly created alignment project are automatically selected (☑). Entries can be selected and unselected in the *Information fields* panel using the **Ctrl**- and **Shift**-keys as described in 3.3.8.



Selected entries can be added to or removed from an existing alignment.

With **Edit > Cut selected sequences** (, **Ctrl+X**) the selected entries are removed from the alignment project and copied to the clipboard.

With **Edit > Paste selected sequences** (, **Ctrl+V**) the same entries are placed back into the alignment project.

Entries can be added to an existing alignment project at any time. The entries first need to be copied to the clipboard from the *Main* window or e.g. from a comparison or another alignment project.

To copy entries to the clipboard, select the entries (e.g. in the *Main* window) first and use **Edit > Views > Copy selection** (**Ctrl+C**).


To cut entries from one alignment project into another, use **Edit > Cut selected sequences** (, **Ctrl+X**) in one alignment project and **Edit > Paste selected sequences** (, **Ctrl+V**) in the other alignment project.



When adding entries to or deleting entries from an alignment project, the dendrogram, similarity matrix and sequence alignment need to be calculated again.

8.4.6 Aligning sequences

In order to obtain a multiple sequence alignment, a pairwise alignment similarity matrix and a dendrogram (most often a UPGMA dendrogram) should be calculated first (see 8.3.1 for more background information on sequence analysis).

The *Sequence alignment* window of BioNumerics offers the possibility to calculate a multiple sequence alignment by means of a single command (**Alignment > Calculate > Multiple alignment...** ()). The settings for the successive pairwise and multiple alignment steps are grouped within a single dialog box.

The *Multiple alignment settings* dialog box allows a number of parameters to be set. The **Algorithm** for pairwise sequence comparisons can be set to be BioNumerics own proprietary algorithm (**BioNumerics**), **Needleman-Wunsch** [30] or **Wilbur-Lipman** [41] (ClustalW). Depending on which algorithm is selected, the settings which can be specified, differ.

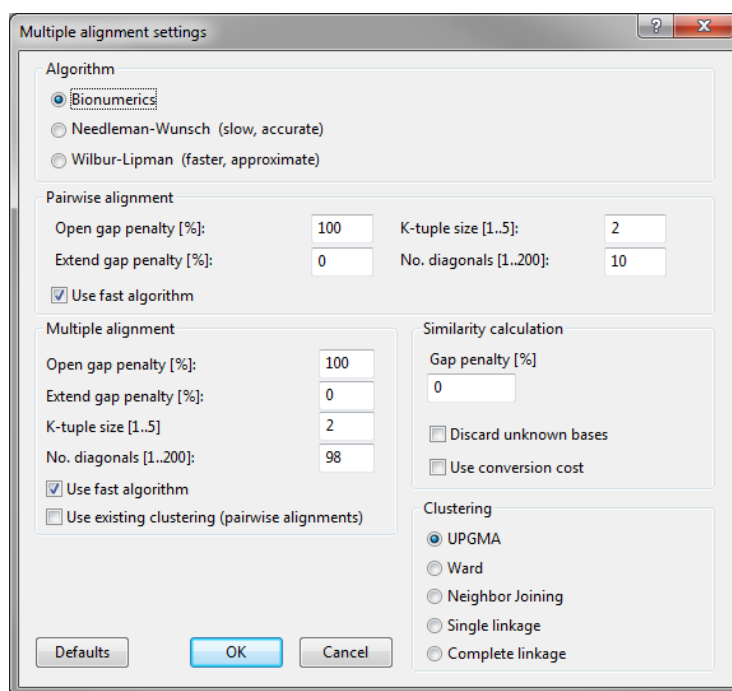


Figure 8.4.4: The *Multiple alignment settings* dialog box, BioNumerics algorithm checked.

If *BioNumerics* is selected as algorithm, the dialog box is displayed as in Figure 8.4.4 and the following settings can be specified.

The *Pairwise alignment* settings are grouped together: The *Open gap penalty* is the cost, expressed as a percentage, to introduce a gap in a sequence. The default value is 100%, which is the same penalty as a mismatch. The *Extend gap penalty* is the cost (in percent) to increase an existing gap with one position. The default value is 0. The parameters *K-tuple size* and *No. diagonals* are only available when the option *Use fast algorithm* is checked. This algorithm creates a lookup table of groups of bases for both sequences (*words*). The *K-tuple size* is the size of such a word. The smaller the words are, the more precise the alignment will be, but the more computing time it will take to calculate the alignment. The parameter can be varied between 1 and 5, with 2 as default. The number of diagonals (*No. diagonals*) is the maximum number of relative positions between both sequences the algorithm will consider. The values can be varied between 0 and 200 with 9 as default. The larger the number, the more gaps the algorithm can create to align every two sequences, but the longer the alignment will take. Therefore, the two parameters allow you to custom-define the alignment process from very fast and relatively rough, to slow and very accurate.

For the *Similarity calculation*, a *Gap penalty* is a cost that can be specified as a percentage of a match (default value is 0). *Discard unknown bases* lets you decide whether or not ambiguous bases are taken into account. If unchecked, the program uses a predefined cost table for scoring ambiguous bases. Checking *Use conversion cost* makes the similarity calculation faster, e.g. for draft alignments, by transforming the calculated conversion cost into a similarity value.

As *Clustering* algorithm can be selected between *UPGMA*, *Ward*, *Neighbor Joining*, *Single linkage* and *Complete linkage*.

The *Multiple alignment* settings include an *Open gap penalty* (default value: 100) and an *Extend gap penalty* (default value: 0). When the option *Use fast algorithm* is checked, the additional parameters *K-tuple size* (default value: 2) and *No. diagonals* (default value: 99) become available. Checking *Use existing pairwise clustering* allows you to calculate the multiple alignment based on an existing pairwise clustering. This option can be used e.g. to employ a ClustalW tree as input for the BioNumerics multiple alignment algorithm.

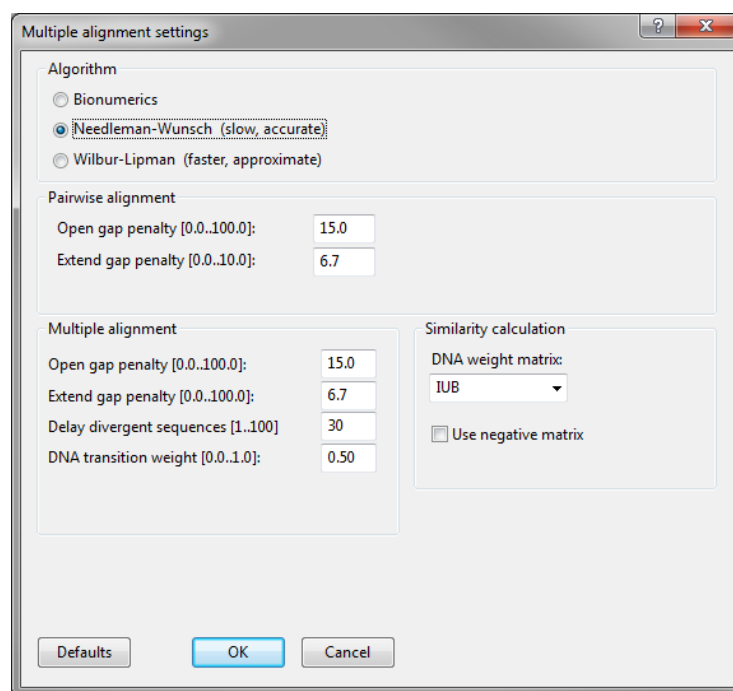


Figure 8.4.5: The *Multiple alignment settings* dialog box, Needleman-Wunsch algorithm checked.

If the *Needleman-Wunsch* algorithm is used, the *Multiple alignment settings* dialog box is displayed as in Figure 8.4.5 and following settings apply.

Under the *Pairwise alignment* settings, an *Open gap penalty* (default value: 15) and an *Extend gap penalty* (default value: 6.7) can be specified.

Under the *Similarity calculation* settings, the *DNA weight matrix* can be selected. The choice is offered between the default *IUB* (International Union of Biochemistry) and the *CLUSTALW* DNA weight matrix. Check *Use negative matrix* to allow the use of negative values in the DNA weight matrix.

The *Multiple alignment* settings include an *Open gap penalty* (default value: 15) and an *Extend gap penalty* (default value: 6.7). *Delay divergent sequences* allows you to delay the alignment of the most distantly related sequences until the most closely related sequences have been aligned. The value displayed is the percent identity level below which the addition of a sequence is delayed; sequences that are less identical than this level to any other sequences in the alignment will be aligned later. The default value is 30. The *DNA transition weight* gives transitions (A \leftrightarrow G or C \leftrightarrow T, i.e. purine-purine or pyrimidine-pyrimidine substitutions) a weight between 0 and 1; a weight of zero means that the transitions are scored as mismatches, while a weight of 1 gives the transitions the match score. The default value is 0.50. For distantly related DNA sequences, the weight should be set near to zero; for closely related sequences it can be useful to assign a higher score.

If the *Wilbur-Lipman* algorithm is used, the *Multiple alignment settings* dialog box is displayed as in Figure 8.4.6 and following settings apply.

Under the *Pairwise alignment* settings, a *Gap penalty* (default value: 5), a *K-tuple size* (default value: 2), and the *No. diagonals* (default value: 4) can be specified. These parameters are similar as explained for the *BioNumerics* algorithm. The *Window size* is a parameter specific for the Wilbur-Lipman algorithm and corresponds to a region within the similarity scores matrix where matches are considered. The higher this value is set, the more accurate the pairwise alignment will be, but the more calculation time required. The default value is 4.

The *Multiple alignment* options are identical to the options described above for the Needleman-Wunsch

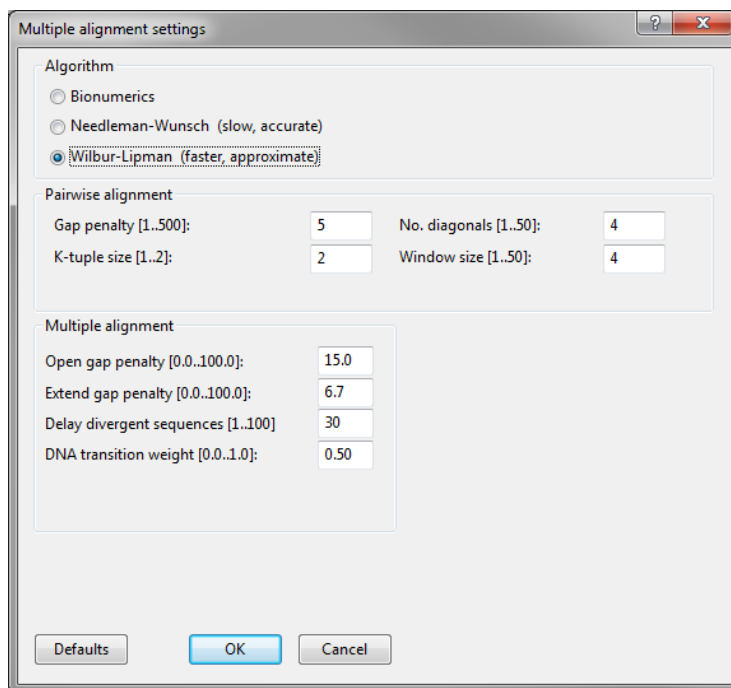


Figure 8.4.6: The *Multiple alignment settings* dialog box, Wilbur-Lipman algorithm checked.

algorithm.

Irrespective of the selected algorithm, pressing the **<Defaults>** button will restore the default settings for the *Multiple alignment settings* dialog box, i.e. the **BioNumerics** algorithm with default parameters.

Pressing **<OK>** will start the calculation of the multiple alignment using the selected algorithm.



The dendrogram and similarity matrix that are displayed after calculating a multiple alignment are still based on pairwise similarity values. See 8.4.10 on how to calculate a global cluster analysis.

In case the project consists of more than one alignment, e.g. if multiple sequence types were imported for the selected entries, the active (i.e. currently displayed) sequence alignment can be selected from the drop-down list in the main toolbar.

The *Sequence alignment* window also offers the option to calculate a multiple alignment as a two-step process, i.e. calculate a pairwise alignment and dendrogram first and then calculate a multiple alignment based on the previously obtained pairwise clustering. This option can be useful, e.g. when one wants to base a multiple alignment on a ClustalW dendrogram or when applying a Jukes and Cantor correction. A pairwise alignment and dendrogram can be calculated as follows:

First, reset the calculated alignment with **Alignment > Reset**. This step is strictly spoken not necessary, but it makes understanding of the work flow easier. Then, select **Clustering > Calculate > Clustering (pairwise alignments)...**

The *Clustering settings (pairwise alignments)* dialog box allows a number of parameters to be set (see Figure 8.4.7). The parameters that can be set depend on the algorithm selected.

For any of the three algorithms selected, the **Pairwise alignment**, **Clustering** and **Similarity calculation** settings are the same as explained for the *Multiple alignment settings* dialog box (see Figure 8.4.5).

In addition to the parameters discussed above, a parameter **Correction** is available for the BioNumerics algorithm in the *Clustering settings (pairwise alignments)* dialog box. The options are **None** to use no correction (the default choice) or to select the **Jukes and Cantor** correction [21], a *one parameter* correction

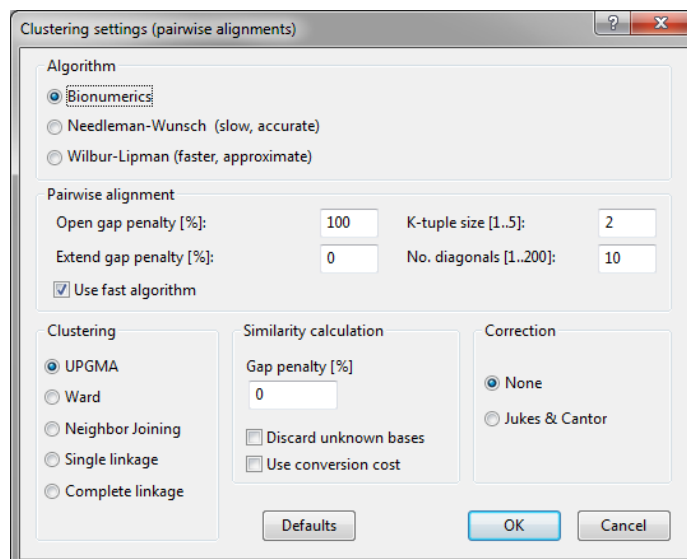


Figure 8.4.7: The *Clustering settings (pairwise alignments)* dialog box, BioNumerics algorithm checked.

for the evolutionary distance as calculated from the number of nucleotide substitutions.

Pressing <OK> will start the calculation of the pairwise clustering. The dendrogram and similarity matrix as calculated by the selected algorithm appear. The sequences in the *Sequence display 1* panel are still unaligned. The obtained pairwise clustering can now be used to calculate a multiple alignment (**Alignment > Calculate > Multiple alignment...** (🔍)). In the *Multiple alignment settings* dialog box, check **Use existing clustering (pairwise alignments)** and press <OK>. The multiple alignment that is displayed is calculated using the selected algorithm, based on a pairwise alignment and dendrogram.

8.4.7 Calculating a consensus sequence

A consensus sequence can be used to obtain a multiple alignment of all sequences against one single sequence. Depending on the type of analysis, the user may wish to assign a single sequence as consensus or may want to calculate the consensus sequence based on several sequences. A consensus sequence is also required for mutation searches (see 8.4.19) and allows additional identity display settings for the alignment (see 8.4.8) to be chosen. To allow maximum flexibility, a consensus sequence is always calculated for the currently selected entries in an alignment. With **Alignment > Consensus > Create from selected entries** (🔍) the *Consensus definition* dialog box is called.

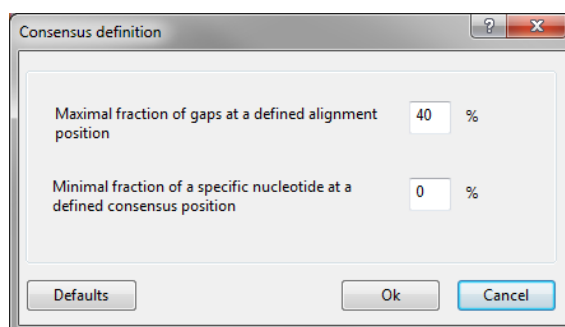


Figure 8.4.8: The *Consensus definition* dialog box.

The parameter *Maximal fraction of gaps at a defined alignment position* determines the maximum occurrence of a gap at a certain position in the sequences on which the consensus is calculated. If the actual occurrence is higher than the specified percentage, the position will be excluded from the consensus sequence. The default value is 40%.

The *Minimal fraction of a specific nucleotide at a defined consensus position* defines the "threshold" occurrence that a nucleotide should reach in order to contribute to the consensus. The default value of 0% will result in any mismatch leading to an ambiguous base in the consensus.

The consensus sequence is displayed in the header of the *Sequence display 1* panel. Entries that were used to calculate the consensus are preceded with a blue dot in the *Information fields* panel.

When changes are made to the alignment (e.g. when recalculated using other settings or after manual editing), the consensus can be recalculated using the same sequences and calculation settings (**Alignment** > **Consensus** > **Recalculate** (🔄)).

When a consensus sequence is present, it can be used to align other sequences against. This option obviously only makes sense in an alignment containing highly related sequences. Choosing **Alignment** > **Calculate** > **Template-based alignment...** (📄) calls the *Template-based alignment settings* dialog box (see Figure 8.4.9).

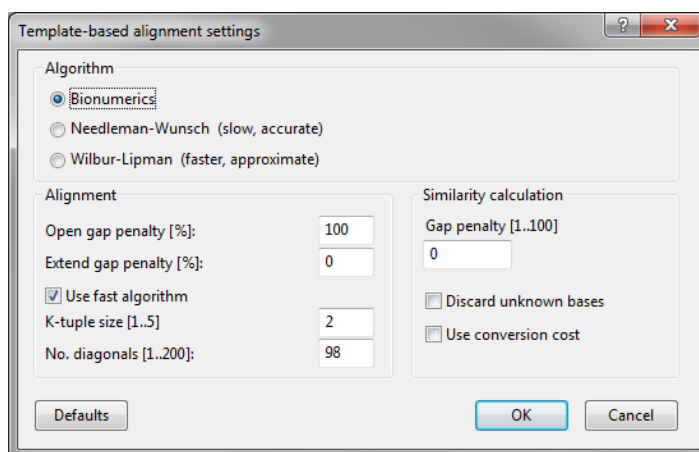


Figure 8.4.9: The *Template-based alignment settings* dialog box.

The parameters that can be set in the *Template-based alignment settings* dialog box are identical to those discussed for the *Clustering settings (pairwise alignments)* dialog box (see Figure 8.4.7). Pressing <OK> aligns the sequences against the consensus sequence.

To remove a calculated consensus sequence, press **F4** to unselect any entries in the *Sequence alignment* window and select **Alignment** > **Consensus** > **Create from selected entries** (🔍).

8.4.8 Display options for sequences and curves

The sequence alignment is displayed in the *Sequence display 1* panel by default. The *Sequence display 2* panel is configured to display the curves (chromatograms) of the individual trace files by default. Before the curves can be displayed, they need to be loaded first. To load the curves into the alignment project, choose **Alignment** > **Load curves** (📄).

The display options for the *Sequence display 1* panel and the *Sequence display 2* panel can be set via **Options** > **Sequence display 1...** and **Options** > **Sequence display 2...**, respectively.

For *Sequence display 1* panel, **Show sequence** is checked by default. **Multiple line display** means that the alignment will be wrapped into the width of the panel and displayed on more than one line. If a dendrogram

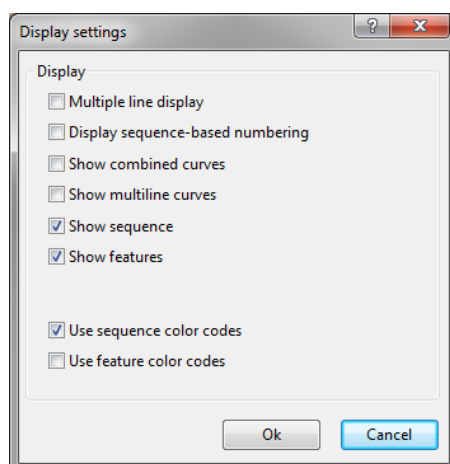


Figure 8.4.10: The *Display settings* dialog box.

is calculated, it will be repeated for each line. Consequently, the horizontal scroll option disappears in the sequence display panel. **Display sequence-based numbering** shows position numbers for each individual sequence. **Show combined curves** and **Show multi-line curves** display the curves (chromatograms) of the sequences respectively superimposed or on different lines. **Use sequence color codes** displays the bases and amino acids in the sequence in different colors.



Curves can only be displayed if they are available for the sequence (i.e. if the sequences were generated from chromatogram files and assembled using the *Assembler* program in BioNumerics, see 8.1.3.7) and if the curves are loaded into the alignment project.

Color codes for nucleic acids can be changed with **Options > Text color settings > Nucleic acids....** This calls the *Color code settings* dialog box for nucleic acids.

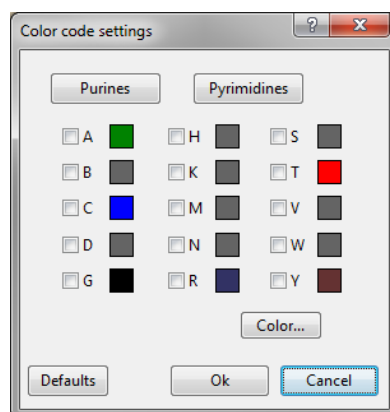


Figure 8.4.11: The *Color code settings* dialog box with nucleic acid color definitions.

To change the color for a specific nucleotide or a number of nucleotides, select the nucleotide(s) via the check box and press **<Color>**. This pops up a color picker from which standard colors can be picked and/or custom colors defined.

The buttons **<Purines>** and **<Pyrimidines>** provide a shortcut to select specifically the purine and pyrimidine nucleotides, respectively.

Pressing the **<Defaults>** button restores the default colors.

Color codes for amino acids can be changed by selecting **Options > Text color settings > Amino acids....** This calls the *Color code settings* dialog box for amino acids.

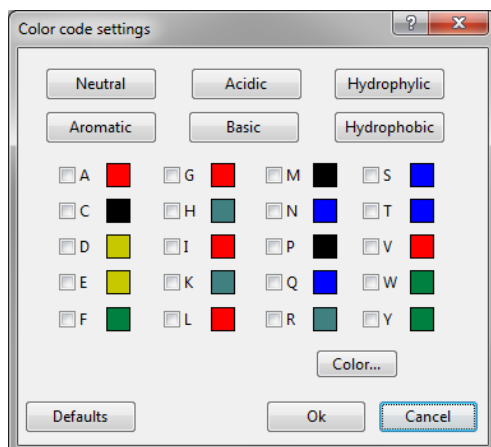


Figure 8.4.12: The *Color code settings* dialog box for amino acid color definitions.

Color settings for amino acids can be changed in a similar way as described for nucleic acids.

To change the color for a specific amino acid or a number of amino acids, select the amino acid(s) via the check box and press **<Color>**. This pops up a color picker from which standard colors can be picked and/or custom colors defined.

The buttons **<Neutral>**, **<Acidic>**, **<Hydrophilic>**, **<Aromatic>**, **<Basic>**, and **<Hydrophobic>** provide a shortcut to select specifically the corresponding group of amino acids.

Pressing the **<Defaults>** button restores the default colors.

A number of options are designed to enhance the visualization of conserved parts in the alignment. They are specific to the *Sequence display 1* panel, and are grouped in the menu item **Alignment > Identity display**.

Select **Alignment > Identity display > Conserved blocks** to display the sequence positions that are conserved throughout the alignment in gray.

Select **Alignment > Identity display > Neighbor identity blocks** to display sequence positions in gray when at least one of the neighboring sequences has the same nucleotide at the corresponding position.

The two other options (**Alignment > Identity display > Identity with consensus** and **Alignment > Identity display > Difference with consensus**) are self-explanatory but require a consensus to be calculated first (see 8.4.7).

The zoom sliders of the *Sequence display 1* panel and the *Sequence display 2* panel can be used to zoom selectively in the horizontal and vertical direction. For detailed information on the use of zoom sliders, see 2.3.7. When the *Sequence display 1* panel is zoomed vertically, the *Dendrogram* panel, *Information fields* panel and *Similarities* panel are zoomed proportionally.

8.4.9 Editing an alignment

A multiple alignment as calculated by the software (see 8.4.6) can be edited via drag-and-drop of individual positions or sequence blocks. The manually edited sequence alignment is saved along with the alignment project and is used to base the global clustering on (see 8.4.10).

The *Sequence display 1* panel and *Sequence display 2* panel contain a cursor which is synchronized between both panels. Similar as in a text processor, the cursor always appears between two characters (bases or amino acids).

Using the mouse, select one of the corner positions of the block to define and while holding down the left

mouse button, drag the mouse pointer to define the desired block. Alternatively, hold down the **Shift**-key and click with the mouse to define the opposite corner of the selection. Note that a block can comprise a single or several sequences. The selection is visible as a black rectangle (see Figure 8.4.13).

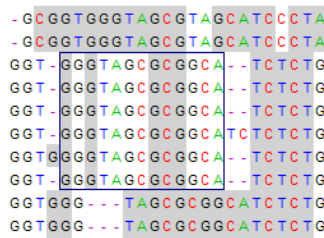


Figure 8.4.13: Selecting blocks of bases for drag-and-drop manual alignment.

A selection block can also be made using the keyboard, by holding down the **Shift**-key while pressing the arrow keys.



If necessary, the block can be moved over other bases at the left or right side. This will then force a gap to be introduced in the sequences up and down from the block, in order to preserve the original alignments left and right from the block and to align the block the way the user has forced it to.



A gap common for all sequences in the alignment project, i.e. spanning the complete alignment, will be automatically removed. As a consequence, nothing will happen if you try to realign a block of bases spanning the whole alignment.




Double-clicking a position in the alignment will select a continuous stretch (without gaps) of the clicked sequence.

For manual alignment editing, as well as for other actions performed on the alignment, a multilevel undo and redo function is available. The undo function can be accessed with **Edit > Undo** (, **Ctrl+Z**). The redo function is accessible through **Edit > Redo** (, **Ctrl+Y**).



Within the constraints of the dendrogram, the order in which sequences appear in the multiple alignment can be changed by repeatedly swapping dendrogram branches, as described in 8.4.12, until the sequences are listed in the desired order.

8.4.10 Calculating a global cluster analysis

To calculate a global clustering based on the sequences in the alignment, choose **Clustering > Calculate > Clustering (multiple alignment)...** (). This calls the *Clustering settings (multiple alignment)* dialog box.

When **Discard unknown bases** is disabled, the program will use a predefined cost table for scoring uncertain or unknown bases. For example, N with A will have 75% penalty, as there is only 25% chance that N is A. Y and C will be counted 50% penalty because Y can be C or T with 50% probability each. If this setting is enabled, all uncertain and unknown bases will not be considered in calculating the final similarity. The **Gap penalty** is a parameter which allows you to specify the cost the program uses when one single gap is introduced. This cost is relative to the score the program uses for a base matching, which is equal to 100%. The program uses 0% as default.

Under **Correction**, one can select the **Jukes and Cantor** correction [21], a **one parameter** correction for the evolutionary distance as calculated from the number of nucleotide substitutions. Alternatively, the **Kimura 2 parameter** correction [22] can be selected.

A selection can be made between the available clustering algorithms: **UPGMA**, **Ward**, **Neighbor Joining**, **Single linkage** and **Complete linkage**.

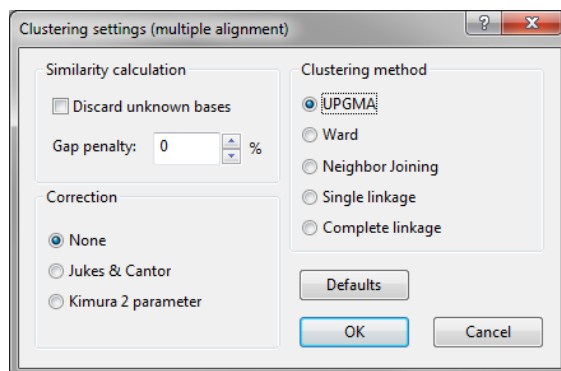


Figure 8.4.14: The *Clustering settings (multiple alignment)* dialog box.

Pressing the **<Defaults>** button resets all parameters to their default value.

The *Similarities* panel displays the global similarity matrix, on which tree in the *Dendrogram* panel is calculated.

8.4.11 Calculating a maximum parsimony cluster analysis

Selecting **Clustering > Calculate > Parsimony (multiple alignment)...** calls the *Maximum parsimony cluster analysis* dialog box (see Figure 8.4.15).

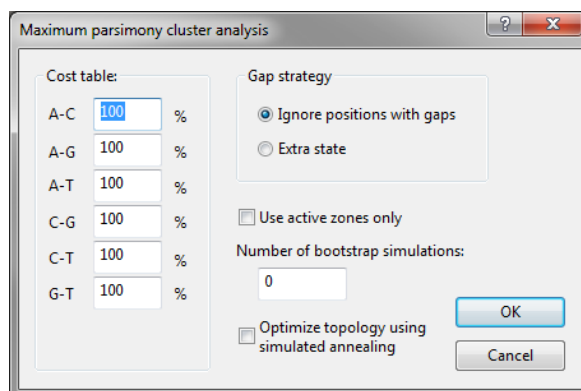


Figure 8.4.15: The *Maximum parsimony cluster analysis* dialog box.

This dialog allows you to specify a cost for each base conversion (mutation) in the **Cost table**. The default setting is 100 Gaps can be dealt with in two ways: the software can **Ignore positions with gaps**, or can consider gaps as an **Extra state**. In case gaps are ignored, every position that contains a gap in one or more sequences of the multiple alignment, will be excluded from the analysis. For every diverse sequences this may result in the omission of a considerable part of the sequence from the similarity calculation.

The check box **Use active zones only** is only applicable when a reference sequence is defined, and when certain zones on this reference sequence are excluded for analysis.

BioNumerics uses methods that are described in the literature to optimize the topology of parsimonious trees. An alternative method, which sometimes finds even more parsimonious trees, but which is considerable slower, is the mathematical principle of **Simulated annealing**.

In addition, BioNumerics can do a **Bootstrap** analysis on the parsimony clustering, for which you can enter the **Number of bootstrap simulations**. If zero is entered, no bootstrap values are calculated.



Enabling simulated annealing and at the same time entering a number of bootstrap simulations will increase the computing time dramatically. It is not recommended to combine these options.

8.4.12 Dendrogram display functions

A dendrogram is displayed from the moment a pairwise or global clustering is performed (see 8.4.6 and 8.4.10, respectively). Similar as in the *Comparison* window (see 13.3.1), several dendrogram display functions are available in the *Sequence alignment* window.

Entries can be selected from within the *Dendrogram* panel of the *Sequence alignment* window:

To select an individual entry, hold the **Ctrl**-key and click on a dendrogram tip (where a branch ends in an individual entry). Alternatively, choose **Clustering > Select branch into list** or right-click on the dendrogram tip and the menu item from the floating menu. Repeat this action to unselect the entry.

To select a cluster on the dendrogram at once, hold the **Ctrl**-key and left-click on a branch node. Alternatively, choose **Clustering > Select branch into list** or right-click on a branch and choose the menu item from the floating menu.

When a dendrogram node or tip is clicked on, a diamond-shaped cursor appears at that position. The average similarity between the entries at the cursor's place is shown in the upper left corner of the *Dendrogram* panel.

In some cases, it may be necessary to select the root of a dendrogram, for example if you want to (un)select all the entries of the dendrogram. In case of large dendrograms, selecting the root may be difficult using the mouse. With **Clustering > Select root**, the cursor is placed on the root of the dendrogram.

Two branches grouped at the same node can be swapped to improve the layout of a dendrogram or make its description easier: select the node where two branches originate and choose **Clustering > Swap branches**.



When position-based search results, such as sequence searches (see 8.4.18), mutation listings (see 8.4.19) or bookmarks (see 8.4.20), are mapped on the alignment, the program will warn about this when a dendrogram is recalculated or branches are swapped. When continuing with the action, the search results will be lost. This default behavior can be changed in the general settings for the alignment project.

The function **Clustering > Reroot tree**, only applies to neighbor joining trees in the *Sequence alignment* window. This clustering method produces trees without any specification as to the position of the root or origin (*unrooted* trees). Since users will often want to display such trees in the familiar dendrogram representation, the tree is to be rooted artificially. "Re-rooting" is usually done by adding one or more unrelated entries (so-called *outgroup*) to the clustering, and using the outgroup as root. The result is a *pseudo-rooted* tree.

8.4.13 Cluster significance tools

Similar as for comparisons, a number of cluster significance tools are available for alignment projects. For more background on these tools, see 13.3.7.

The *standard deviation* of a branch is obtained by reconstructing the similarity values from the dendrogram branch and comparing the values with the original similarity values. The standard deviation of the derived values versus the original values is a measure of the reliability and internal consistence of the branch. To calculate the error flags of the branches, choose **Clustering > Calculate error flags**. The error flags, i.e. the standard deviations of the dendrogram branches compared to the corresponding sections in the similarity matrix, are shown at each branch. The average similarity and the standard deviation at the position of the cursor is shown in the left upper corner. Choose **Clustering > Calculate error flags** again to remove the error flags.

The cophenetic correlation (see also 13.3.7), i.e. the correlation between the dendrogram-derived similarities and the matrix similarities, is a parameter to express the consistency of a cluster. This is calculated with **Clustering > Calculate cophenetic correlations**. The cophenetic correlation is shown at each branch, together with a colored dot, of which the color ranges between green, yellow, orange and red according to decreasing cophenetic correlation.

A bootstrap analysis is based on *sampling with replacement* (for more background information, see 13.3.7) and can only be calculated on a global clustering. To calculate a global clustering based on a multiple sequence alignment, choose **Clustering > Calculate > Clustering (multiple alignment)...** (🖨️). Selecting **Clustering > Bootstrap analysis...** calls a new dialog.

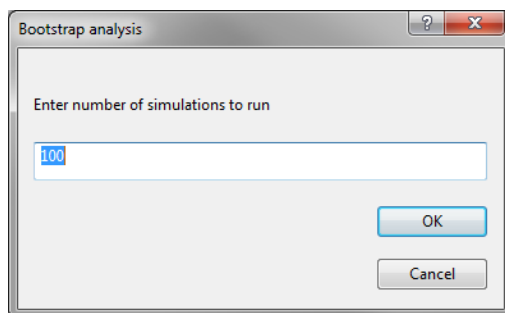


Figure 8.4.16: Specify the number of bootstraps.

The dialog prompts for the number of simulations (samplings). Pressing <OK> starts the analysis. The bootstrap values are shown at each branch, together with a colored dot, of which the color ranges between green, yellow, orange and red according to decreasing bootstrap value.

8.4.14 Matrix display functions

The similarity matrix is displayed in the *Similarities* panel, in default configuration located at the right hand side of the *Sequence alignment* window.

Initially, the matrix is displayed as differentially shaded blocks representing the similarity values. Similar as for the *Comparison* window, the interval settings for the shadings is graphically represented in the caption of the *Similarities* panel.



Figure 8.4.17: Adjustable similarity shading scale.

There are two ways to change the intervals for shading:

- Drag the interval bars on the scale; the matrix is updated instantly.
- Select **Options > Similarity matrix shades...** or right-click within the *Similarities* panel and select **Similarity matrix shades** from the floating menu. In the *Similarity shade limits* dialog box that appears, the maximum/minimum values for each interval can be entered as numbers, from low (top) to high (bottom).



If it is difficult to read the similarity values on the shaded background, you can remove the shades by entering 100% for each interval.

To export a tab-delimited text file of the similarity matrix, select **File > Export > Similarity matrix....** Select a location to save the text file when the program prompts for a destination path. If desired, the suggested name can be modified. The text file contains the similarity values with the entry keys as descriptors.

8.4.15 Printing and exporting a sequence alignment

When printing from the *Sequence alignment* window, BioNumerics first shows a print preview. This print preview shows the dendrogram, entry keys and sequence alignment in multi-line view and looks exactly as it will look on printed pages.

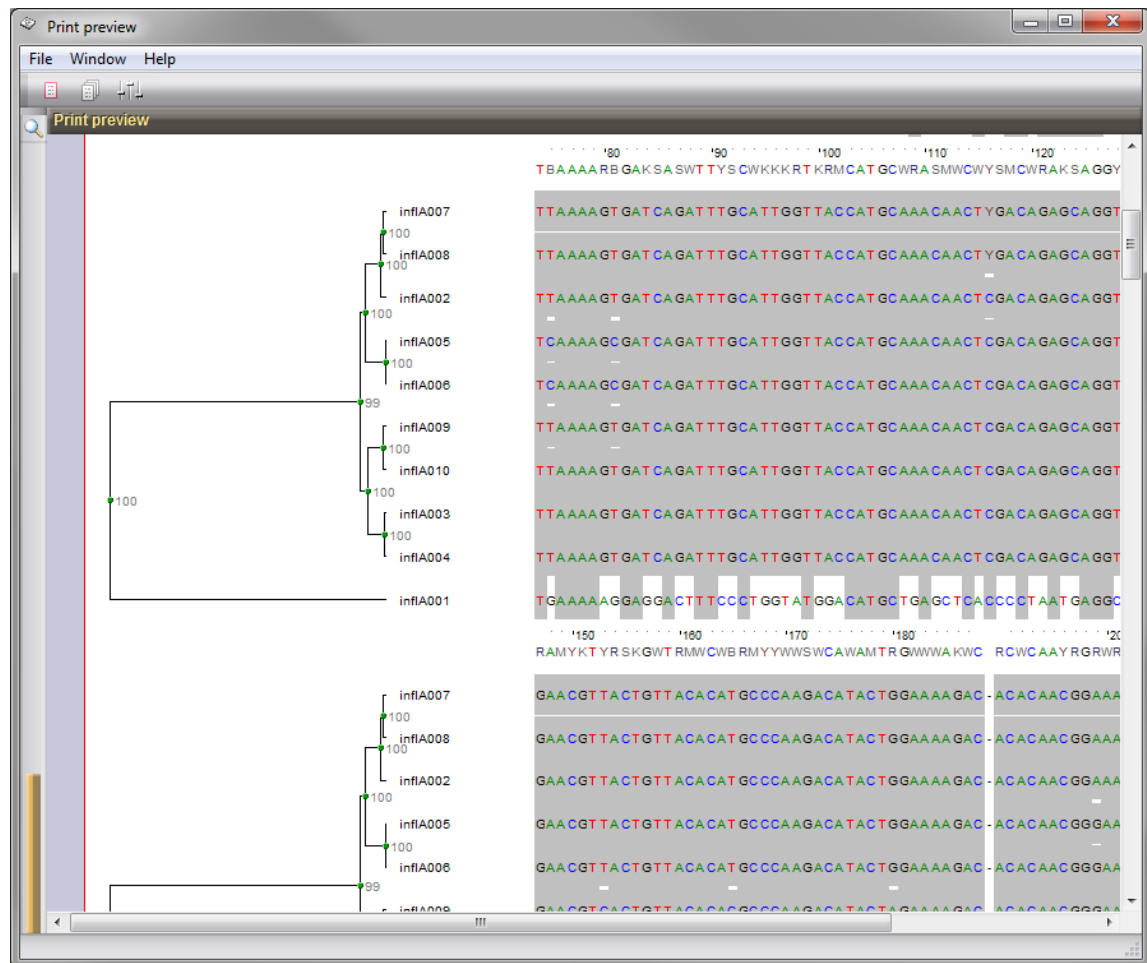


Figure 8.4.18: The *Alignment print* window.

It is possible to zoom in and out on a page using the zoom slider, located in default configuration on the left hand side of print preview (see 2.3.7 for zoom slider functions). When zoomed, the horizontal and vertical scroll bars allow you to scroll through the page.

On top of the first preview page, there are three small yellow slide bars. These slide bars represent the following margins, respectively:

- Left margin of the dendrogram;
- Right margin of the dendrogram;
- Left margin of the alignment;

Each of these slide bars can be shifted individually to reserve the appropriate space for the mentioned items. The image is printed exactly as it looks on the preview.

The menu command **File > Printer setup...** (🖨️) allows you to set the paper orientation, the margins, and other printer settings for the default printer.

With **File > Print selected pages** (📄) the selected pages are printed. Selected pages are indicated with a red border. Use **Ctrl+click** to select multiple pages.

Use **File > Print all pages** (📄) to print all pages at once.



When a part of the alignment is selected in the *Sequence alignment* window (see 8.4.9), only the selected part will be printed.

An alignment, part of an alignment or a calculated consensus sequence can be exported in different formats.

To export an alignment in text format, choose **File > Export > Formatted alignment...** This brings up the *Formatted sequence export* dialog box, with the same functionality as described for exporting multiple alignments from the *Comparison* window (see 8.3.14).

To export an alignment in a graphical format, select the part of the alignment that you would like to export and choose **File > Export > Selected alignment to clipboard (graphical)**. The selected part of the alignment is now copied to the clipboard and can be pasted in other applications as Windows metafile or enhanced metafile.

To export a consensus sequence in plain text format, select **File > Export > Consensus sequence to clipboard**. From the Windows clipboard, the consensus sequence can be pasted in other applications.

With the command **File > Export > Alignment to comparison** an alignment can be exported to the *Comparison* window, e.g. if the alignment should be used in a cluster analysis that needs to be compared with other analyses present in a comparison. This function calls the *Export to comparison* dialog box.

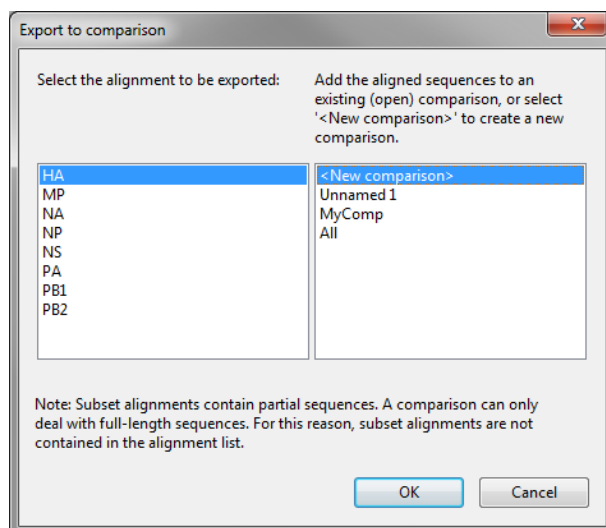


Figure 8.4.19: The *Export to comparison* dialog box.

In the list on the left-hand side, all alignments that can be exported will be displayed. Note that subsets will be excluded from the list, since the *Comparison* window can only deal with complete sequences.

The list on the right-hand side shows all comparisons that are currently open. In addition, an option "<New comparison>" will be listed and highlighted by default. If an open comparison has not been saved yet, it will be listed as "Unnamed#". With the default option, all entries from the alignment will be exported to a new comparison. When an existing open comparison is selected, aligned sequences will only be exported for the entries that the alignment and the comparison have in common. No new entries will be added to an

existing comparison.

Highlight the alignment that you want to export in the list on the left, highlight either an open comparison or "<New comparison>" from the list on the right and press <OK> to export the alignment.

8.4.16 Finding sequence positions in an alignment

You can have the cursor jump automatically to a certain position on a sequence in the alignment. This can be particularly useful e.g. when examining the occurrence of a certain SNP in a set of equal-length sequences. With *Edit > Find > Find position....*, the *Position search* dialog box is called.

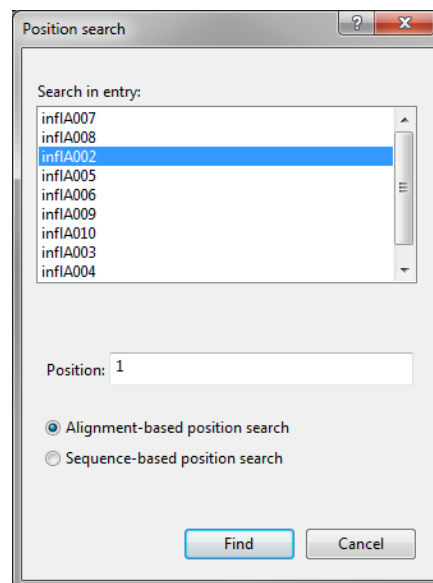


Figure 8.4.20: The *Position search* dialog box: search for a specific position on one of the entries.

The sequence on which the cursor currently resides is highlighted in the *Search in entry* list, but any other sequence can be selected as well. A position can be entered as a number. The alignment-based numbering (*Alignment-based position search*) or the individual sequence numbering (*Sequence-based position search*) can be used. Both are equivalent in case no gaps were introduced in the sequences.

When pressing the <Find> button, the cursor will jump to the corresponding position on the sequence. Note that the search position is the position *before* the cursor.

8.4.17 Sequence translation

BioNumerics can automatically translate an alignment of nucleotide sequences into amino acids according to a selected translation table and within a certain translation frame. The translated amino acid sequence is displayed in the sequence alignment.

To set the settings that will be used for the translation, select *Alignment > Translation > Define....* This calls the *Translation settings* dialog box.

From the list under *Select translation table*, the different translation tables, corresponding to variants of the standard genetic code, can be selected. By default, the standard code is used. The translation frame can be set under *Define translation frame*. Uncheck *Keep nucleotide sequences displayed* if you only want the amino acid sequences to be displayed, without the nucleotide sequences from which they originated.

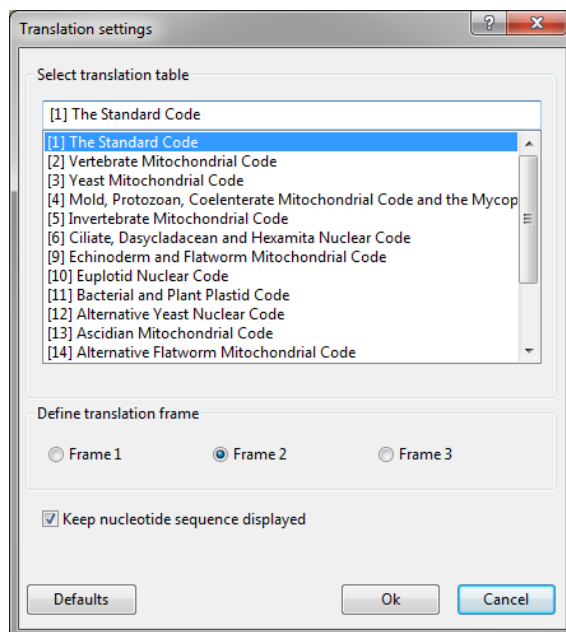


Figure 8.4.21: The *Translation settings* dialog box to select translation table and specify translation frame.

Choose **Alignment** > **Translation** > **Show/Hide** (🔍) to display/hide the translated amino acid sequence.

If the nucleotide sequences you are working with are never translated, for example in case of ribosomal RNA sequences, select **Alignment** > **Translation** > **None** to disable the translation into amino acids. In this case, no amino acid changes will be listed in the *Mutation listing panel* when an alignment is searched for mutations (see 8.4.19 on how to perform a mutation search).

8.4.18 Subsequence search

The complete alignment or any selection of sequences within the alignment can be searched for the occurrence of a subsequence. This subsequence can correspond to e.g. a restriction site, primer sequence, repeat pattern, or any other specific sequence you are interested in. The *Find sequence* dialog box is called with **Edit** > **Find** > **Find sequence...** (🔍).

By default, all entries present in the alignment project are highlighted in the **Entries to be searched** list. Any other selection of entries can be made from this list. To select all entries at once, press the <**Select all**> button.

Under **Sequence**, you can enter the subsequence to search for.

The **Search settings** are applicable to the current subsequence search:

- **Mismatches allowed:** the maximum number of mismatches one allows for a subsequence still to be considered as matching with a target sequence.
- **Allow gaps:** Whether or not you allow the subsequence to be interrupted by gaps.
- **Consider IUPAC codes:** Allows the search sequence to be matched with uncertain positions denoted as IUPAC unresolved positions (e.g. R, Y, etc., including N) and allows IUPAC code to be used in the search sequence. When unchecked, only A, T, C or G will be matched against the target sequence(s).
- **Reverse search:** Whether or not the invert-complemented sequence will be searched as well.

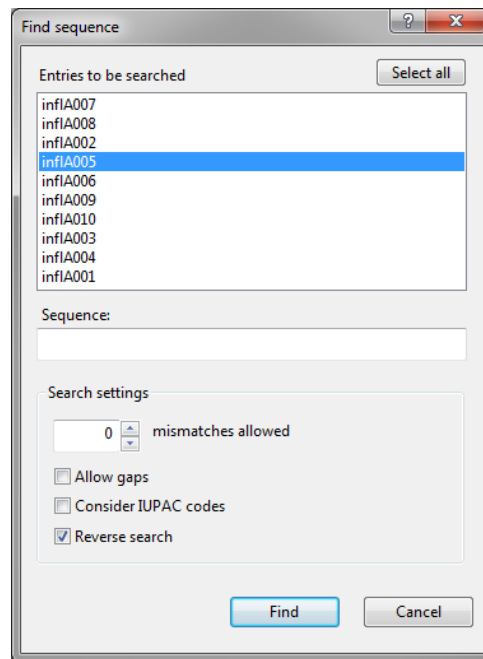


Figure 8.4.22: The *Find sequence* dialog box, to find a sequence in the alignment.

Pressing <**Find**> starts the search. The search results are displayed in the *Sequence search results* panel (see Figure 8.4.23). This grid panel lists every match of the entered subsequence, as defined in the search settings.

Sequence search results						
[1] ttcttggtcmg						
Count	Entry	Position entry	Position alignment	Direction	Match	Mismatch
1	inflA002	393	393-404	→	TTCTTGGTCMG TTTGTGCCCA	2
2	inflA008	393	393-404	→	TTCTTGGTCMG TTTGTGCCCA	2
3	inflA006	392	393-404	→	TTCTTGGTCMG TTTGTGCCG	0
4	inflA003	393	393-404	→	TTCTTGGTCMG TTCTTGGTCAG	0
5	inflA010	393	393-404	→	TTCTTGGTCMG TTTGTGGTCAG	0
6	inflA009	393	393-404	→	TTCTTGGTCMG TTTGTGGTCAG	0
7	inflA001	393	393-404	→	TTCTTGGTCMG TTCTTGGTCAG	0
8	inflA005	393	393-404	→	TTCTTGGTCMG TTTGTGGTCAG	0
9	inflA004	393	393-404	→	TTCTTGGTCMG TTTGTGGTCAG	0
10	inflA007	393	393-404	→	TTCTTGGTCMG TTCTTGGTCCA	1

Figure 8.4.23: The *Sequence search results* panel from the *Sequence alignment* window.

Entry displays the key of the sequence in which the match occurs. **Position entry** is the position on the individual sequence where the match occurs, while **Position alignment** uses the position in the alignment. **Direction** is either forward (a blue arrow pointing to the right) or reverse (a red arrow pointing to the left). The column **Match** shows the search sequence (top) matched with the target sequence (bottom) and **Mismatch** displays the number of mismatches occurring. The latter will always be lower than or equal to the number of mismatches specified in the *Find sequence* dialog box.

The sequence search results are listed in the order in which they occur in the alignment. The alignment is screened from top to bottom, left to right. With other words, from the first position on the first, second, third, etc. sequence to the second position on the first, second, third sequence, etc.

When clicking on a match listed in the *Sequence search results* panel the cursor automatically jumps to the

corresponding sequence block in the alignment (in the *Sequence display 1* panel) and to the corresponding block on the curves (in the *Sequence display 2* panel; if displayed).

If more than one sequence search was performed, the results of a previous search can be displayed again by selecting this search from the drop-down list in the toolbar of the *Sequence search results* panel (see Figure 8.4.24).

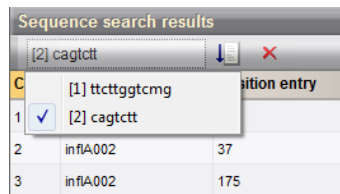


Figure 8.4.24: Drop-down list in the toolbar of the *Sequence search results* panel, displaying previous sequence searches.



Sequence searches are not saved along with the alignment project.

8.4.19 Mutation search

The mutation search tool is designed to detect mutations in individual sequences based on comparison with a consensus. This consensus can be derived from a single sequence or a set of sequences. Therefore, in order to perform a mutation search, a consensus sequence should first be calculated (see 8.4.7). The settings for defining the consensus determine the way mutations are defined. For example, if the **Minimal fraction of a specific nucleotide at a defined consensus position** (see 8.4.7) is set to 10%, all bases at a position that have more than 10% occurrence will contribute to the consensus: if 85% is T and 15% is C, the consensus will then be Y at that position. The mutation search algorithm will consider a sequence with a T or C at that position as NOT mutated. A sequence that has A at that position will be recorded as a mutation, since A is not contained in the consensus sequence. Using the right settings for the consensus sequence, the mutation search tool can be used for SNP discovery as well.

To search for mutations select **Mutations > Search...** (🔍). This calls the *Find mutations* dialog box.

By default, all entries present in the alignment project are highlighted in the **Entries to be screened** list. Any other selection of entries can be made from this list.

A checklist allows the **Types of mutations to be searched** to be selected. Each type of mutation (intergenic, synonymous, non-synonymous or indel) is displayed in a different color. Clicking the arrow button next to the color box allows you to pick a different color. The **Screen focus** determines whether only the selected alignment is screened or all sequence alignments present in the project.

If **Implement IUPAC code** is unchecked, any ambiguous position will be listed as a mutation. When checked, BioNumerics will consider the IUPAC nomenclature and score mutations in a "conservative" way. For example, for a position denoted as A in the consensus, any occurrences of R (A or G), M (C or A) or W (T, U or A) will not be scored as a mutation.

If you check **Map mutations in alignment viewer**, the positions of the mutations are indicated in the *Sequence display 1* panel and *Sequence display 2* panel using blocks of the corresponding color.

The mutation results are displayed in the *Mutation listing* panel (see Figure 8.4.26).

This grid panel lists all mutations that were found in comparison with the consensus. The column **Entry** shows the key of the entry where the mutation occurs. **Alignment** is the name of the alignment and **Position** the nucleotide position at which the mutation occurs. The type of the mutation (**Type**) can be silent, missense or indel and the color of the small square is as defined in the *Find mutations* dialog box. **NA change** is the

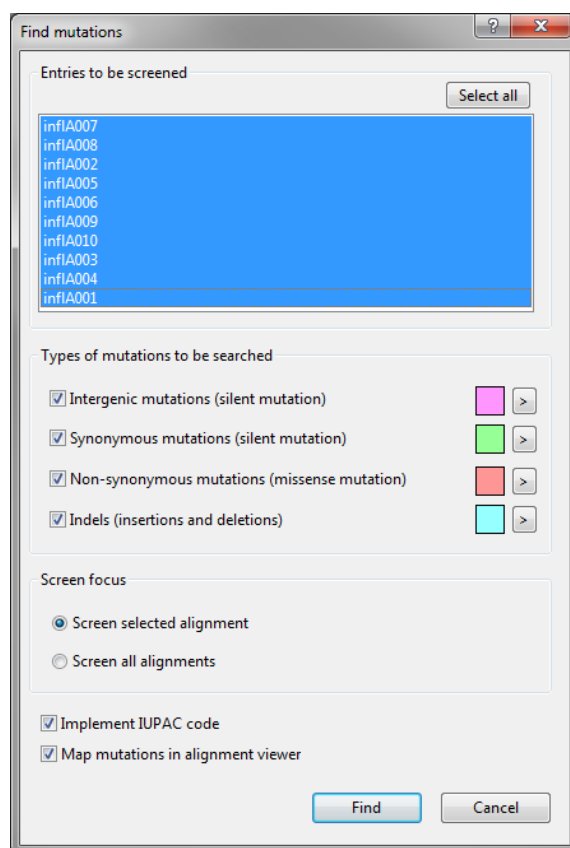


Figure 8.4.25: The *Find mutations* dialog box to specify settings for a mutation search.

Mutation listing						
[1] Mutation search						
Entry	Alignment	Position	Type	NA change	AA change	
inflA010	HA	405	missense	g → a	S → D	
inflA009	HA	405	missense	g → a	S → D	
inflA001	HA	405	missense	g → a	S → D	
inflA005	HA	405	missense	g → a	S → D	
inflA004	HA	405	missense	g → a	S → D	
inflA007	HA	405	missense	g → a	S → N	
inflA007	HA	412	missense	a → t	E → D	
inflA010	HA	415	silent	c → a		
inflA003	HA	420	missense	t → c	L → S	
inflA010	HA	420	missense	t → c	L → S	
inflA009	HA	420	missense	t → c	L → S	
inflA001	HA	420	missense	t → c	L → S	
inflA005	HA	420	missense	t → c	L → S	
inflA004	HA	420	missense	t → c	L → S	
inflA007	HA	420	missense	t → c	L → S	
inflA004	HA	424	silent	g → a		
inflA007	HA	433	silent	a → c		
inflA002	HA	441	indel			
inflA008	HA	441	indel			
inflA006	HA	441	indel			



Figure 8.4.26: The *Mutation listing* panel in the *Sequence alignment* window.

nucleotide change and **AA change** is the change in amino acid (if any). The mutations are listed in the order in which they occur in the alignment. The alignment is screened from top to bottom, left to right. In other words, from the first position on the first, second, third, etc. sequence to the second position on the first, second, third, etc. sequence, and so on.



If you select **Alignment > Translation > None** in the main menu, no amino acid changes will be shown in the *Mutation listing* panel after a subsequent mutation search (column **AA change** will be empty) and all mutations will be marked as either silent or indel.

When clicking on any of the mutations listed in the *Mutation listing* panel, the cursor will jump to the corresponding position on the alignment (in the *Sequence display 1* panel) and to the corresponding position on the curves (in the *Sequence display 2* panel; if displayed).

To scroll through the mutation list, select **Mutations > Jump to next** () . To return to a previous mutation, select **Mutations > Jump to previous** () .

If more than one mutation search was performed, a previous listing can be displayed again by selecting the **Mutation search** from the drop-down list in the toolbar of the *Mutation listing* panel (see Figure 8.4.27).

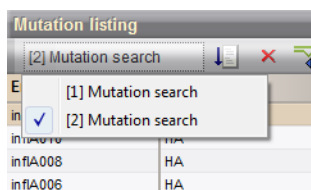



Figure 8.4.27: Drop-down list in the toolbar of the *Mutation listing* panel, displaying previous mutation searches.

To delete a mutation search, select it from the drop-down list and select **Mutations > Delete list** or press the button .

To export a tab-delimited text file of the mutation list, select **Mutations > Export list...** . Select a location to save the text file when the program prompts for a destination path and specify a name.

The mutation list is saved to a new character type experiment with **Mutations > Export mutations to character set...** . A new dialog pops up prompting for the character type name. After confirmation, a new *Comparison* window opens with the character set in focus.




Mutation listings are not saved along with the alignment project.

8.4.20 Defining bookmarks in a sequence alignment

A sequence position or sequence block, e.g. corresponding to a primer, probe, protein active site, etc., can be bookmarked in order to retrieve it easily.

To keep bookmarks organized, lists can be created to store related bookmarks in. Creating a new list is done with **Alignment > Bookmarks > Add new list...** () . BioNumerics prompts for the list name.

To create a bookmark, select a sequence position or sequence block in the alignment and select **Alignment > Bookmarks > Add bookmark...** () . The program will prompt for the bookmark name.



Bookmarks can also extend vertically and span multiple sequences. This can be useful, e.g., to define a region on all sequences to print or to create a subset.



Bookmarks should not necessarily belong to a bookmark list but can be defined directly. In that event, they are stored under **All bookmarks**.

To delete a bookmark from the list, click on it and choose **Alignment > Bookmarks > Delete selected bookmarks**.

Bookmark lists different from the one currently displayed can be selected from the drop-down list in the toolbar of the *Bookmarks* panel (see Figure 8.4.28).

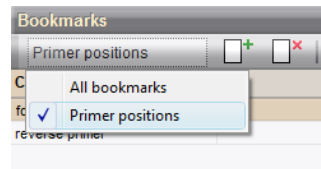


Figure 8.4.28: Drop-down list in the toolbar of the *Bookmarks* panel, displaying the available bookmark lists.

To delete a bookmark list, select it from the drop-down list choose **Alignment > Bookmarks > Delete list** (🗑️).



All bookmarks are saved along with the alignment project.

8.4.21 Primer analysis

8.4.21.1 Introduction

The primer analysis application in BioNumerics has been designed to calculate optimal primer and primer combinations, in function of various experimental parameters, for the amplification of a target region. The primer analysis tool can be launched from the *Sequence alignment* window or *Sequence editor* window (see 8.2.4) and requires the presence of the Sequence data module (SQ) in the BioNumerics configuration.

8.4.21.2 General primer analysis

When the primer analysis tool is launched from the *Sequence alignment* window, a possible input can be a set of (aligned) sequences, either nucleic or amino acid sequences, of which the primer analysis tool will automatically calculate a consensus sequence and derive therefrom primer and primer combinations in function of melting temperature, primer positions and degeneracy of the primers. This type of primer design can be launched in the *Sequence alignment* window with the command **Tools > Primer design...**

If one is only interested in finding primers and primer combinations spanning a specific region on the sequences, select this region in the *Sequence display 1* panel using the mouse. The selection will be transferred to the *Primer design* window and can be used to adapt the target region from. Alternatively, create a subset in the *Sequence alignment* window (see 8.4.4).

The *Primer design* window is divided into an upper panel (the *Sequence viewer panel*) and a lower panel, giving information concerning the primers and primer combinations found. In the *Sequence viewer panel*, all sequences are shown on a green background, with the consensus sequence depicted above.

With the zoom slider on top of the *Sequence viewer panel* one can zoom in and out on the sequence. Zooming can be done up to base level. The red vertical line indicates the cursor position on the sequence. If a selection was present in the *Sequence editor* window before launching the primer analysis tool, the selected sequence is highlighted with an orange background.

A base in the consensus sequence appears as A, C, G or T if this base is unanimously present in all sequences of which the consensus is derived. In all other cases, a IUPAC code in the consensus sequence represents the different bases present in the sequences. If a gap is present in a least one sequence, an asterisk is shown at this position in the consensus sequence.

Primers and primer combinations are searched in a target region for PCR amplification, also referred to as a *locus*. When a sequence selection is present, this selection will be the target region. The full sequence will



Figure 8.4.29: Zooming in on the sequence.

be taken as target when no sequence selection is present. Obviously, the default target region can still be modified:

To edit a target region select **Target > Edit...** A dialog box appears, asking for some specifications about the target design.

- With the **Full sequence** option checked, the whole consensus sequence will be specified as forward and reverse primer regions and will be used when searching for primers and primer combinations.
- Check the **Selection** option to search for primer combinations which will amplify a PCR product that spans a subsequence of the complete consensus sequence. The subsequence is specified by the start and stop positions in the two input boxes. If a selection is present in the *Primer design* window, the start and stop positions of the selection are automatically displayed in the start and stop input boxes. The target design around the selection region can be further specified in the *Extended selection options* dialog box. To launch this dialog, press the **<Extended selection>** button.

Pressing **<OK>** plots the target design on the sequence. The target design is plotted in the *Sequence viewer panel* below the consensus sequence. The region used to construct the design is shown in green. The asterisks in the consensus – if any – are excluded from the target region. The forward primer region is shown in red, and the reverse primer region is shown in blue (see Figure 8.4.30 for an example).



Figure 8.4.30: Target region adapted from a selection.

After having specified the forward and reverse primer regions on the consensus sequence, these primer

regions can be screened for fixed primer sequences, or one can let the software calculate optimal primers and primer combinations in function of various settings (see 8.2.4).

8.4.21.3 Discriminative primer design

When the primer analysis tool is launched from the *Sequence alignment* window, a second possible input can be a set of (aligned) sequences, either nucleic or amino acid sequences, of which one group is specified as the *positive* group, and one group as the *negative* group. With the primer analysis tool, primers and primer combinations can be generated, which will bind on any sequence of the positive group, but which will not bind on any sequence of the negative group, again in function of melting temperature, primer positions and degeneracy as specified by the user. This type of primer design is called *discriminative primer design* in BioNumerics and can be launched in the *Sequence alignment* window with the command **Tools > Discriminative primer design....**

Before launching the discriminative primer design tool, select the sequences you want to assign to the *positive* group. All non-selected sequences will automatically be assigned to the *negative* group.

If one is only interested in finding primers and primer combinations spanning a specific region on the selected sequences, select this region on the sequences in the *Sequence display 1* panel using the mouse. The selection will be transferred to the *Primer design* window and can be used to adapt the target region from. Alternatively, create a subset in the *Sequence alignment* window (see 8.4.4).

The *Primer design* window is divided into an upper panel (the *Sequence viewer panel*), and a lower panel, giving information concerning the primers and primer combinations found. In the *Sequence viewer panel*, sequences belonging to the positive group are shown on a green background, with the consensus sequence depicted above. The sequences of the negative group appear on a pink background, again with the consensus sequence annotated above.

With the zoom slider on top of the *Sequence viewer panel* one can zoom in and out on the sequence. Zooming can be done up to base level. The red vertical line indicates the cursor position on the sequence. If a selection was present in the *Sequence editor* window before launching the primer analysis tool, the selected sequence is highlighted with an orange background.

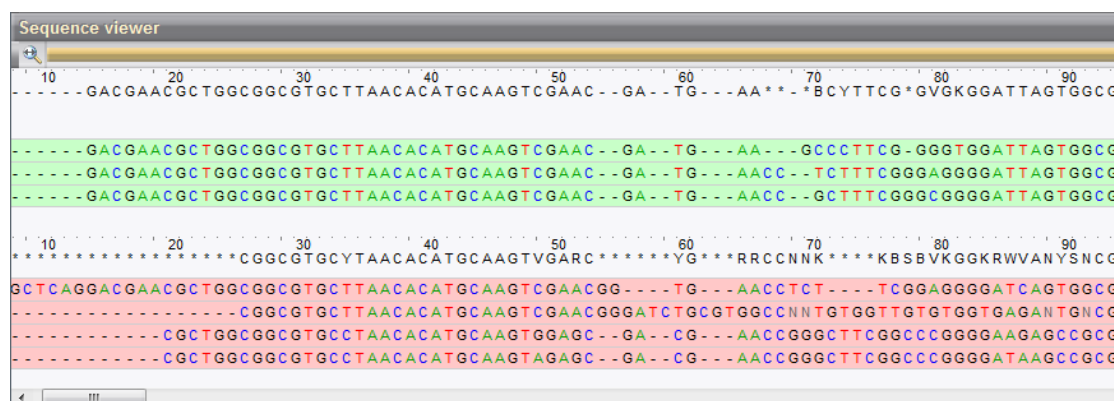


Figure 8.4.31: Zooming in on the sequence.

A base in the consensus sequence appears as A, C, G or T if this base is unanimously present in all sequences of which the consensus is derived. In all other cases, a IUPAC code in the consensus sequence represents the different bases present in the sequences. If a gap is present in a least one sequence, an asterisk is shown at this position in the consensus sequence.

Primers and primer combinations are searched in a target region for PCR amplification, also referred to as a *locus*. When a sequence selection is present, this selection will be the target region. The full sequence will

be taken as target when no sequence selection is present. Obviously, the default target region can still be modified:

To edit a target region select **Target > Edit....** A dialog box appears, asking for some specifications about the target design.

- With the **Full sequence** option checked, the whole consensus sequence will be specified as forward and reverse primer regions and will be used when searching for primers and primer combinations.
- Check the **Selection** option to search for primer combinations which will amplify a PCR product that spans a subsequence of the complete consensus sequence. The subsequence is specified by the start and stop positions in the two input boxes. If a selection is present in the *Primer design* window, the start and stop positions of the selection are automatically displayed in the start and stop input boxes. The target design around the selection region can be further specified in the *Extended selection options* dialog box. To launch this window, press the **<Extended selection>** button.

Pressing **<OK>** plots the target design on the sequence. The target design is plotted in the *Sequence viewer panel* below the positive consensus sequence. The region used to construct the design is shown in green. The forward primer region is shown in red, and the reverse primer region is shown in blue (see Figure 8.4.32 for an example). The asterisks and the gaps ("-" sign) in the positive consensus sequence - if any - are excluded from the target region. Target regions interrupted by gaps - as a result of the presence of gaps in all positive sequences - are linked with a "~" sign. Linked regions are considered as one continuous sequence. Target regions interrupted by one or more asterisks are not linked due to the presence of a nucleotide (or amino acid) in one or more positive sequences at the position of the asterisk. Such an ambiguous position is not suitable for primer location.



Figure 8.4.32: Target region adapted from the positive consensus sequence.

After having specified the forward and reverse primer regions on the consensus sequence, these primer regions can be screened for fixed primer sequences or one can let the software calculate optimal primers and primer combinations in function of various settings (see 8.2.4).

Chapter 8.5

Sequence databases

8.5.1 Restriction enzyme database

In each BioNumerics database, the **full-list** of enzymes is used by default when performing a restriction enzyme analysis in the *Sequence editor* window (see Figure 8.2.3). This full-list contains several thousands of enzymes, obtained from the REBASE website (<http://www.rebase.neb.com>), many of which are incompletely described or are not commercially available. Since it is usually not desired to have this complete list displayed when performing a restriction enzyme analysis in BioNumerics, the software allows the creation of new lists, containing a subset of enzymes of interest.

To view all enzymes present in the **full-list** choose *Database > Sequence databases > Restriction enzyme database...* in the *Main* window.

The *Enzyme Database Viewer* window pops up (Figure 8.5.1). The **full-list** - holding the entire REBASE list of enzymes - is automatically selected in the toolbar of the *Enzyme database panel*.


For each enzyme in the selected list, detailed information - if available - is displayed in the upper panel: name (*Name*), source organism (*Micro-organism*), recognition site and cleavage positions (*Recognition site*), methylated bases recognized (*Methylation site*), type of pattern (*Pattern type*) and type of cleavage (*Cleavage type*). The *Information panel* displays the full list of references and the commercial availability. Iso-schizomers and compatible enzymes, i.e. enzymes that produce the same sticky ends and are thus compatible in ligation reactions are displayed in the *Iso-schizomers panel* and *Compatible enzymes panel*, respectively.

The information of a selected enzyme is displayed in a separate window with *Edit > Show enzyme card...*

Detailed information (if available) is displayed: name (*Restriction enzyme*), source organism (*Micro-organism*), recognition site and cleavage positions (*Recognition site*), methylated bases recognized (*Methylation site*)

The *References panel* displays the full list of references, the panel below displays the commercial availability.

Iso-schizomers and compatible enzymes, i.e. enzymes that produce the same sticky ends and are thus compatible in ligation reactions are displayed in the *Iso-schizomers panel* and *Compatible enzymes panel*, respectively.

A new list, containing a subset of enzymes that are of interest, is created with *File > Create new list...* (). The new list is stored in the subdirectory `SequenceData \Enzymedata` of the database and is automatically selected in the toolbar of the *Enzyme database panel*. To display another list in the *Enzyme database panel*, select the list from the drop-down list.

A newly created list initially does not contain any enzymes. An enzyme selection can be created from any

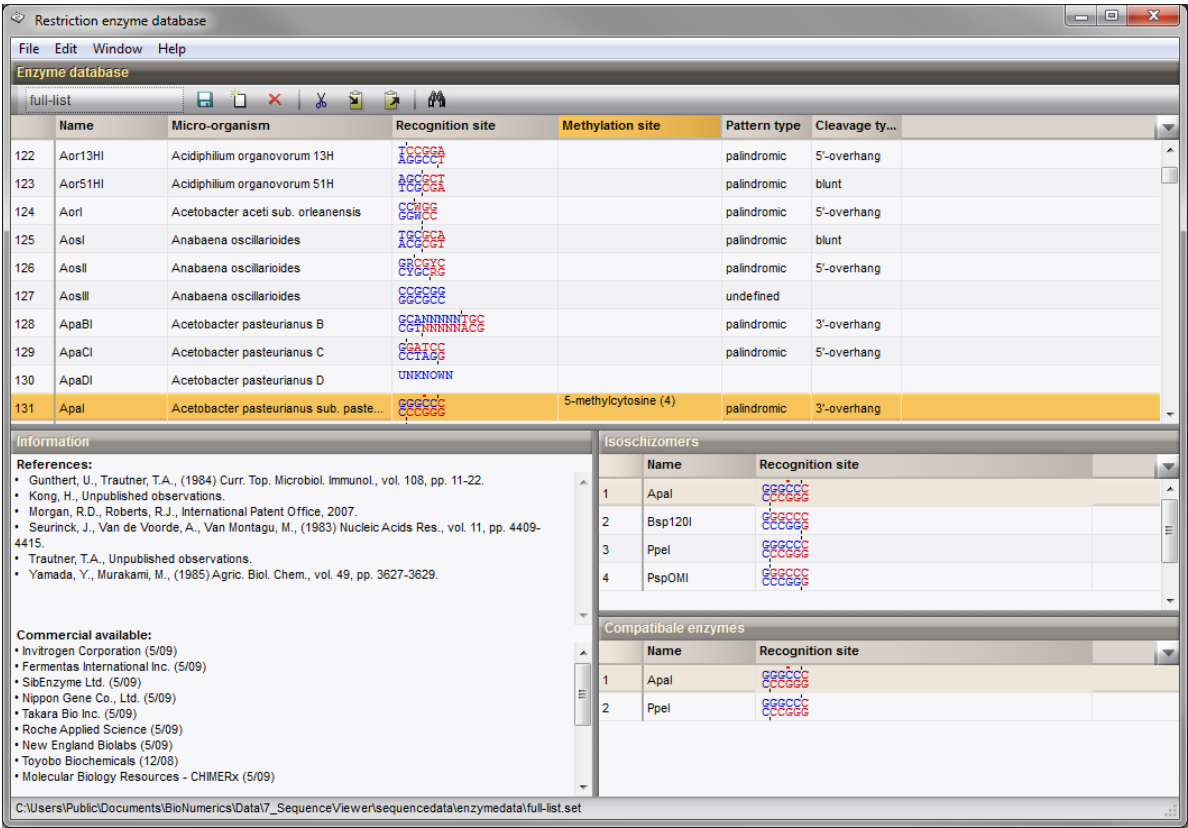


Figure 8.5.1: The *Enzyme Database Viewer* window.

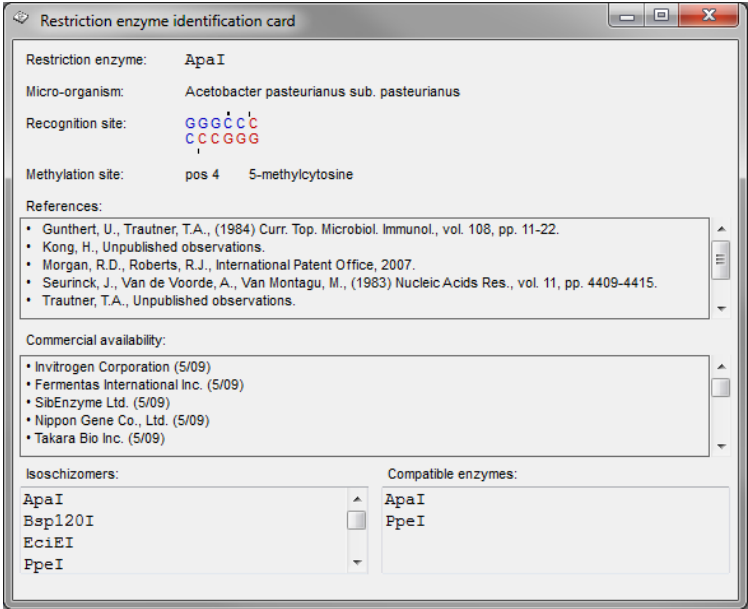



Figure 8.5.2: Identification card of a selected restriction enzyme.

list containing enzymes, and can be transferred to any other list.

A selection of restriction enzymes can be created using the **Ctrl**- and **Shift**-buttons.

Selected enzymes can be copied with *Edit > Copy selected enzymes* (, **Ctrl+C**). Alternatively, use the keyboard shortcut combination **Ctrl+C**.

A copied selection is pasted into a list with **Edit > Paste selected enzymes** (📋, **Ctrl+V**).

With **Edit > Cut selected enzymes** (✂, **Ctrl+X**) selected enzymes are removed from the list. Note that enzymes of the full-list cannot be removed.

A selection of restriction enzymes can also be created based on general criteria, like the number of bases in the recognition site, blunt or 5'/3' sticky ends, etc. The *Restriction enzyme filter* dialog box is called with **Edit > Find...** (🔍, **Ctrl+F**) (see Figure 8.5.3).

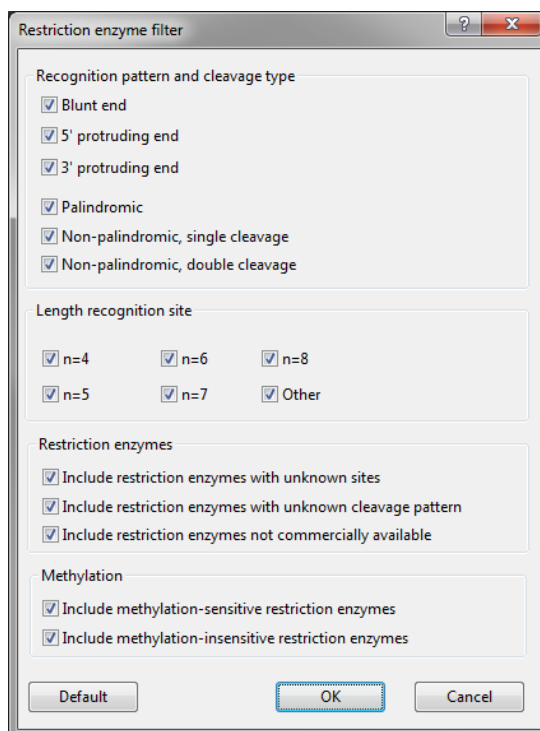


Figure 8.5.3: The *Restriction enzyme filter* dialog box: filtering settings for restriction enzyme selection.

The *Restriction enzyme filter* dialog box allows a number of parameters to be specified:

The *Cleavage type* can be **Blunt end**, **5' protruding end**, and/or **3' protruding end**. The *Recognition pattern* can be **palindromic**, **Non-palindromic single cleavage**, and/or **Non-palindromic double cleavage**.

The *Length recognition site* can be defined as **N=4**, **N=5**, **N=6**, **N=7**, **N=8**, and **Other**, or any combination of these.

Restriction enzymes with unknown sites and **unknown cleavage patterns** can be included or excluded from the search criteria. If the option **Include restriction enzymes not commercially available** is unchecked, only enzymes which are specified in the restriction enzyme database as being available from at least one commercial institution will be searched for.

Methylation-sensitive and **-insensitive restriction enzymes** can be included or excluded from the search criteria.

To restore the default settings press the **<Default>** button.

Restriction enzymes that fulfill the specified criteria in the currently displayed list are selected.

To remove a list from the database, select the list from the drop-down list in the *Enzyme database panel* and choose **File > Delete list** (✖). Please note that the full-list containing all enzymes cannot be deleted.

Saving a list to the database is done with **File > Save list** (💾, **Ctrl+S**) and **File > Save list as...**

The displayed list of enzymes is printed with **File > Print file...**

To close the *Enzyme Database Viewer* window select **File > Exit**.

8.5.2 Oligo nucleotide database

Each BioNumerics database has its own independent oligo nucleotide database. Oligo sequences generated with the primer design functionality can be transferred to the oligo nucleotide database as described in 8.2.4 and 8.4.21. Alternatively, one can manually add oligo sequences to the oligo nucleotide database independently from any primer project.

The oligo nucleotide database is called from the *Main* window with **Database > Sequence databases > Oligo database....**

Index	Name	Sequence	Length	Parent	Tm	GC-content	Deg...	dG5: 5'OH p...	dG3: 3'OH p...	dS: Thermo...	dH: Thermo...	dG: Total ...
4	L77616_for	gagtgacctgggtttttgctc	22	L77616 / Rice	63.1	54.5	1	-5.51	-6.29	-481.84	-174.20	-24.76
2	L77616_rev	ggcagattatcataccgcgaca	22	L77616 / Rice	62.0	50.0	1	-6.81	-6.39	-479.94	-173.00	-24.15

Figure 8.5.4: The *Oligo Database* window.

A new oligo is added to the database with **Oligo > Add...** (). This calls the *Edit oligo nucleotide sequence* dialog box.

Create oligo nucleotide sequence

Name:

Sequence:

Figure 8.5.5: The *Edit oligo nucleotide sequence* dialog box for adding or editing an oligo sequence.

Specify an oligo name in the **Name** text field box, and enter the sequence in the **Sequence** text box.



Although the sequence can be of undefined length, the oligo nucleotide database is not intended for the storage of long sequences.

The


meaning of the database fields in the *Oligo Database* window is the following:

- The **Index** is a unique number given to the oligo nucleotide. The index is generated automatically by BioNumerics.
- The **Name** field is used to characterize the oligo nucleotide sequence. When an oligo nucleotide sequence is added to the oligo database using the primer design functionality launched from the *Sequence editor* window (see 8.2.4), the default suggested **Name** consists of the name of the entry the primer is derived from and the orientation of the primer ("for" or "rev"), separated by a "_" sign. The default suggested **Name** of an oligo nucleotide sequence that is added to the oligo database from

a primer project launched from the *Sequence alignment* window (see 8.4.21), consists of the sequence type name and the orientation of the primer ("for" or "rev"), separated by a "-" sign. The default suggested name can be changed if desired.

- The **Sequence** field holds the oligo nucleotide sequence. The IUPAC code for degenerated sequences is accepted.
- The **Length** field displays the sequence length of the oligo nucleotide sequence in base pairs.
- The origin of the oligo nucleotide sequence is indicated in the **Parent** field. If the sequence is derived from a primer design launched from the *Sequence editor* window, the parent field holds the key of the linked database entry and sequence type. If the oligo nucleotide has been entered in the database independently from any primer project, or from a primer project launched from the *Sequence alignment* window, this field is left blank.
- The **Tm** value corresponds to the melting temperature of the oligo nucleotide sequence, calculated according the nearest-neighbor thermodynamics published by SantaLucia [34] using the default thermodynamic settings.
- The **GC-content** specifies the percentage G and C bases within the oligo nucleotide sequence. For degenerated sequences, the minimal and maximal possible GC-content is given.
- The degeneracy of the oligo nucleotide sequence is specified in the **Degeneracy** column. This is the number of combinations if all ambiguous bases are replaced with the respective possible combinations of bases given by the IUPAC code. A sequence with no ambiguous bases has a degeneracy of 1.
- The **dG5:5'OH pentamer free energy (kcal/mol)** gives the calculated 5'-OH pentamer free energy. The pentamer free energy of oligo ends gives an indication of the oligo binding stability at the ends. The dG5:5'OH value is calculated at standard 1 mol/l monovalent cation concentration at 37°C.
- The **dG3:3'OH pentamer free energy (kcal/mol)** gives the calculated 3'-OH pentamer free energy. The pentamer free energy of oligo ends gives an indication of the oligo binding stability at the ends. The dG3:3'OH value is calculated at standard 1 mol/l monovalent cation concentration at 37°C.
- The **dS:Thermodynamic entropy (cal/K.mol)** field holds the calculated standard thermodynamic entropy of the oligo nucleotide sequence. The dS value is calculated according to the nearest-neighbor thermodynamics published by SantaLucia using the default thermodynamic settings, and can only be calculated for non-degenerated sequences.
- The **dH:Thermodynamic enthalpy (kcal/mol)** column displays the standard thermodynamic enthalpy of the oligo nucleotide sequence. The dH value can only be calculated for non-degenerated sequences.
- The **dG:Total free energy (kcal/mol at 37°C)** field holds the total free energy of the oligo nucleotide sequence at 37°C. The dG value is calculated according to the nearest-neighbor thermodynamics published by SantaLucia using the default thermodynamic settings, and can only be calculated for non-degenerated sequences.

By default, the list is sorted according to the **Name** field. To sort the list according to one of the other columns, right-click with the mouse-pointer on the header of a column and select **Sort**.

A selected oligo nucleotide sequence can be edited with **Oligo > Edit...** .

With the **Ctrl**- and **Shift**-keys a selection of oligo nucleotides is made in the list.

Selected entries are deleted with **Oligo > Delete selected...** .

Choose **Oligo > Export selected to clipboard** to export the oligo nucleotide information of the selected entries to the clipboard.

To print the selected entries in the oligo nucleotide database choose **Oligo > Print selected...**

If a sequence is derived from a primer design launched from the *Sequence editor* window, the **Parent** field holds the key of the linked database entry and sequence type. To open the *Sequence editor* window of the linked database entry, select the oligo from the list and choose **Oligo > Open parent sequence...**

Selecting **Oligo > Calculate thermodynamic properties...** calls the *Oligo nucleotide sequence thermodynamics* dialog box (see Figure 8.5.6).

The dialog box is titled "Oligo nucleotide sequence thermodynamics". It contains the following fields and sections:

- Top strand sequence (5'-3'):** gagtgacctcgggtttttgctc
- Bottom strand sequence:** ctcactggagcccaaaacgag (with a "use complement" button)
- Length:** 22 bp
- Mismatches:** 0
- Experimental conditions:**
 - [Monovalent cation]: 50 mmol/l
 - [Mg⁺⁺]: 2 mmol/l
 - [top strand]: 0.05 μmol/l
 - [bottom strand]: 0.05 μmol/l
 - hybridisation temperature: 37°C
- Thermodynamic predictions:**
 - In 50 mmol NaCl and 2 mmol MgCl₂
 - dH = -174.2 kcal/mol
 - dS = -481.8 cal/K.mol
 - dG37 = -24.8 kcal/mol
 - Tm = 63.1°C
- Buttons:** Defaults, Print, Calculate, Exit

Figure 8.5.6: The *Oligo nucleotide sequence thermodynamics* dialog box for calculating thermodynamic properties.

At the top, the oligo nucleotide sequence is written in the upper edit box in the 5'-3' direction. The lower edit box has captured the complementary strand of the oligo nucleotide sequence and is written in the 3'-5' direction. This is a standard situation: the oligo nucleotide sequence binds on the template, which is the reverse complement of its own sequence. The template sequence is written in the 3'-5' direction for the sake of clearness in recognizing the sequence compatibility. The template is shown in the 5'-3' direction when selecting the 5'-3' radio button.

In the *Oligo nucleotide sequence thermodynamics* dialog box a calculation tool can be launched that provides the thermodynamic properties of oligonucleotide sequences. The thermodynamic values dS (standard thermodynamic entropy), dH (standard thermodynamic enthalpy) and dG37 (total free energy) are calculated according to the nearest-neighbor thermodynamics published by SantaLucia [34]. Where appropriate, nearest-neighbor thermodynamics for mismatched bases and dangling end corrections are calculated according to Bommarito et al. 2000 [10], Allawi and SantaLucia [2], Allawi and SantaLucia (1998a) [5], Allawi and SantaLucia (1998b) [4], and Allawi and SantaLucia (1998c) [3].

As the calculation of the dS and dG37 values are dependent on salt concentrations and oligo nucleotide strand concentrations (primer and template), thermodynamic settings can be specified by the user in the left lower part of the *Oligo nucleotide sequence thermodynamics* dialog box.

- The **[Monovalent cation]** concentration in the solution has a great influence on the charge of the DNA molecules. Melting temperature of the DNA molecules will be a function of the concentration in the solution. Higher monovalent salt concentrations lower the DNA melting temperature. The monovalent cation concentration is by default set to 50 mmol/l.
- **[Mg⁺⁺]** stands for the Mg²⁺ concentration in solution, which also influences the charge and melting

temperature of the DNA molecules. Higher Mg^{2+} concentrations lower the DNA melting temperature. The default value is 2 mmol/l.

- **[top strand]** reflects the oligo nucleotide strand concentration. Although the oligo nucleotide and template concentration in the solution does not have a great impact on hybridization of the DNA molecules, higher DNA concentrations lower oligo nucleotide melting temperatures weakly. This parameter is default set to 50 nmol/l.
- **[bottom strand]** corresponds to the template DNA concentration. This parameter is by default set to 50 nmol/l.
- By default, a **Hybridization temperature** of 37 degrees Celsius is taken, which will result in the calculation of a total free energy dG_{37} at 37 degrees Celsius. Entering another value here will affect the outcome of the calculated free energy. Higher temperatures will result in a lower free energy (less negative!) as a higher temperature influences the formation of duplexes negatively. Lower temperatures will result in higher free energies, which means more stable complexes. Note that the free energy should be a negative value: it represents the standard free energy change when a Watson-Crick duplex is formed. The duplex should show a lower entropy compared to the free oligonucleotide sequence alone, i.e., energy should be set free upon duplex formation.

Pressing the **<Defaults>** button restores the default settings.

The values are calculated and displayed in the right panel of the *Oligo nucleotide sequence thermodynamics* dialog box when pressing the **<Calculate>** button.

When using the default settings, the thermodynamic predictions correspond to the values that are shown in the information fields of the *Oligo Database* window. Changing the thermodynamic settings in the *Oligo nucleotide sequence thermodynamics* dialog box results in different thermodynamic properties.

The information displayed in the right panel is printed with the **<Print>** button.

The calculation tool offers the additional freedom that the user can enter any sequence in the upper edit box and calculate the thermodynamics properties without saving the sequence to the database.

The complement of the sequence can be fetched in the lower edit box by simply clicking the **<use complement>** button.

Mismatches can be entered between the oligo nucleotide sequence and its template in the edit boxes, for example by deleting a base and entering another instead.

However, the following should be noticed: only thermodynamic values for exact and one-mismatch NN-nearest-neighbors are available in the literature until now [34], therefore other data **cannot** be calculated **exactly**. This means the following:

Thermodynamic values can be calculated for the following exact duplex:

C	G	A	A	C	A	T	G	G	G	T	A	T	G	A	C	C	T	C	T
G	C	T	T	G	T	A	C	C	C	A	T	A	C	T	G	G	A	G	A

Also thermodynamic values of a template with one mismatch can be calculated:

C	G	A	A	C	A	T	G	G	G	T	A	T	G	A	C	C	T	C	T
G	C	T	T	G	T	A	C	T	C	A	T	A	C	T	G	G	A	G	A

Or even two non-adjacent mismatches are allowed:

```

C G A A C A T G G G T A T G A C C T C T
| | | | | | | | | | | | | | | |
G C T T G T A C T C A T A C T G A A G A

```

The thermodynamic values given with the calculation tool are a **rough estimation** if adjacent mismatches or gaps appear within the possible priming site: the calculations are only based on those positions displaying exact or single mismatch nearest-neighbors. For this mismatch, the thermodynamic values cannot be calculated exactly:

```

C G A A C A T G G G T A T G A C C T C T
| | | | | | | | | | | | | | | |
G C T T G T A C T A A T A C T G G A G A

```

Thermodynamic values of degenerated oligo nucleotide sequences cannot be calculated as the exact nearest-neighbors are not known at certain positions. An error message pops up when trying to calculate the thermodynamic values of degenerated sequences. For this duplex, the thermodynamic values cannot be calculated:

```

C G N A C A T N G G T A T Y A C D T C T
| | | | | | | | | | | | | | | |
G C T T G T A C C C A T A C T G G A G A

```

Within the thermodynamics calculator, formation of dangling ends can be taken into account for calculating the thermodynamic parameters of oligo nucleotide sequences. Dangling ends have to be considered in those situations where a duplex DNA is formed between two strands of unequal length. The fact that one strand extends the end of the other strand means there is a dangling end, which can influence the stability of the duplex significantly (mostly it is stabilizing).

Dangling ends have to be entered as follows:

```

d C G A A C A T G G G T A T G A C C T C T d
| | | | | | | | | | | | | | | |
A G C T T G T A C C C A T A C T G G A G A G

```

This example indicates that the upper strand binds to a template which extends the upper strand on the 5' as well as on the 3' end, more clearly written:

```

      C G A A C A T G G G T A T G A C C T C T
      | | | | | | | | | | | | | | | |
... A G C T T G T A C C C A T A C T G G A G A G ...

```

Other combinations are also possible:

```

d C G A A C A T G G G T A T G A C C T C T
| | | | | | | | | | | | | | | |
A G C T T G T A C C C A T A C T G G A G d

```

This may for example represent an adapter duplex with 3' end sticky ends, more clearly written as:

```

      C G A A C A T G G G T A T G A C C T C T
      | | | | | | | | | | | | | | | |
A G C T T G T A C C C A T A C T G G A G

```

8.5.3 Molecular weight markers

In the restriction enzyme analysis application of BioNumerics (see 8.2.3) it is possible to create hypothetical gels of fragment patterns. Next to the fragment patterns obtained from the selected enzymes, one or more *weight markers* can be displayed on the gel. The weight marker **Lambda-PstI** is the default weight marker. Other weight markers can be added to the database and selected as the default marker.

The list of molecular weight markers list is called with **Database > Sequence databases > Molecular weight markers....** The *Molecular weight markers* dialog box opens (see Figure 8.5.7).

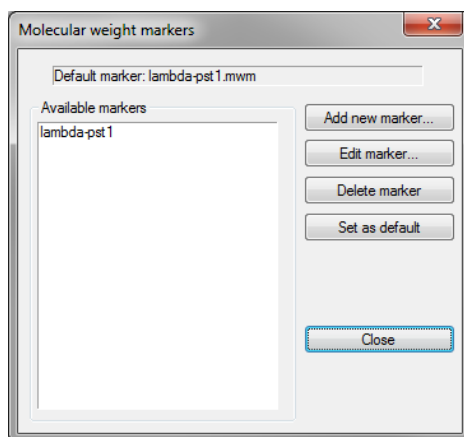


Figure 8.5.7: The *Molecular weight markers* dialog box.

This dialog displays all markers that are saved in the database. Initially, the **Lambda-PstI** marker is the only marker present in the database and is therefore automatically specified as the default marker.

To add a new marker to the database, click the **<Add new marker>** button. New markers are stored in the subdirectory SequenceData \Markerdata of the database, and are added to the list of available markers in the *Molecular weight markers* dialog box.

To edit the band information for a selected marker press the **<Edit marker>**. This opens the *Edit marker* dialog box (see Figure 8.5.8).

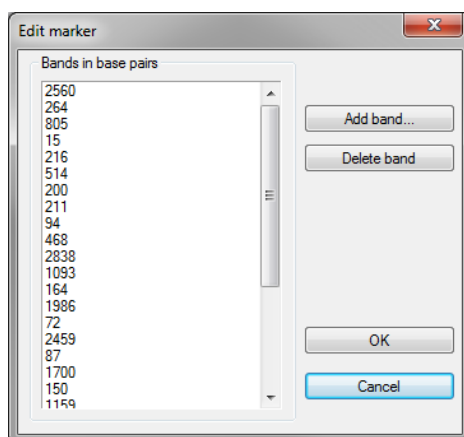


Figure 8.5.8: The *Edit marker* dialog box to edit bands for a selected marker.

This dialog box lists the available bands for the selected marker. With the **<Add band>** button, bands can be added to the marker (in bp). A band can be deleted from the marker by selecting it in the list and pressing **<Delete band>**.

A selected marker can be set as the default marker in the database by pressing the **<Set as default>** button. The default marker is automatically selected when adding a marker to a gel (see [8.2.3](#)).

A marker is deleted from the list with the **<Delete marker>** button.



The default marker cannot be deleted from the list.

Chapter 8.6

Pairwise chromosome comparisons

8.6.1 Introduction

The Chromosome comparison tools are part of the Genome Analysis Tools module (GA) and have been developed for large-scale comparison of sequences of unlimited length. The mathematical algorithms have been written in such a way that large score matrices, needed for the pairwise comparison of chromosomes, are circumvented and that calculation speed is maximized. Two types of chromosome comparisons can be created: a first type based on the full-length DNA sequences (further called "DNA-based chromosome comparisons") and a second type which only uses the information present in annotated coding sequence (CDS) features mapped on the sequences (further called "CDS-based chromosome comparisons").

Both DNA-based chromosome comparisons and CDS-based chromosome comparisons can be performed within the *Chromosome comparison window*. Whereas DNA-based analysis makes use of the full-length sequence data to perform matching between chromosomes, only coding sequences (mapped CDS features) are considered for the CDS comparative analysis. DNA-based seeds are screened for in nucleotide sequences, amino acid-based seeds are screened for in amino acid sequences obtained by translating the nucleotide sequences in amino acid sequences (proteins here) generated within the CDS-based chromosome comparisons.

The purposes of the chromosome comparison tools are diverse:

1. Full genome comparisons and clustering for evolutionary and population genetic studies;
2. Exploring new genomes by comparing to known ones (annotation);
3. Chromosome-wide comparisons to study the organization and structure of genomes (synteny): rearrangements, duplications, deletions, insertions, ...
4. Mutation analysis and gene selection analysis.

In order to understand the descriptions and settings in the next sections, we should consider the search algorithm implemented in the *Chromosome Comparison* window briefly and introduce some terms which will be used further on.

1. In a first round of screening for homologous stretches, the program looks up seeds. Seeds are small stretches of homology between two sequences (either at DNA or at translated level).
2. In a second screening, the program considers all seeds found, thereby looking if within a specified window size there is still a degree of homology found between the two sequences upstream or downstream of the specific seed (*Minimum x matches in window size y*). If the specified conditions are

fulfilled, the seed is selected for further analysis. This way, seeds located in regions of low homology (isolated seeds) are filtered out.

3. In a third screening round, the program considers the stretches of short homology, which have passed the second screening test, and extends these regions upstream and downstream until the identity within a specified window (*Stretch extension window size*) is below a specified value (*Minimum x matches in the stretch extension window size*). Once the upstream and downstream limits of the homology stretch are found, its length is tested against a minimal stretch length (*Minimal stretch length*). If the stretch's identity or length does not reach the minimal score defined, the stretch is not considered.

One of the primary concepts in fast sequence matching algorithms is the *seed*. In principle, seeds are small stretches of homology between two sequences which are further used to extend into longer alignments. Seeds can be exact matches (*consecutive matches*) or small patterns with matching positions (*non-consecutive matches*, also called "spaced seeds"). Mostly, seeds are denoted with "1" on a match position, and "0" or "*" on positions that do not require a match.

The well-known programs FASTA and BLAST for example use seeds with exact matches, with $n=11$ (eleven) for the default Blastn screening (seed = "11111111111"). Ma et al. [24] introduced in their Pattern-Hunter algorithm the use of spaced seeds, which enhanced the screening sensitivity significantly compared to the BLAST algorithm. The seed they used was "111010010100110111" with $n=11$. Meanwhile, many articles have been published on how to find the most optimal spaced seed patterns. A good overview can be found in Choi et al. [12] and Ilie & Ilie [20].

Within the *Chromosome Comparison* window, the algorithms have been developed to use seed- and spaced-seed-screening methods for screening the nucleic acid sequences, as well as for the derived translated sequences. According to our findings, the amino acid-based screening method is more sensitive and faster than the nucleic acid-based screening. Due to the wobble position of the codon, a translated sequence gains flexibility at every third position on the corresponding nucleotide sequence. In computational mathematics, an amino acid-based seed "111" (three consecutive hits) corresponds with a DNA-based seed with $n=9$, as the translated amino acids derive from a sequence with a length of 9 bases. However, due to the degenerated amino acid code (wobble position), the "111" amino acid-based seed will correspond in many cases with a DNA-based seed "110110110". This means that, to achieve a comparable sensitivity at DNA-level screening, one should use the DNA-based seed "110110110", which has only 6 matching positions.

As calculation time reduces enormously with longer seed lengths, the use of the "111" amino acid-based seed is much faster and equally sensitive compared to an analogous "110110110" seed used in DNA-based screening. Another advantage of the amino acid-based seed screening is, that seed lengths of $n=5$ and higher can easily be used, which would correspond to DNA-based seeds of lengths 15 and higher. Such seed lengths can never be achieved in DNA-based screenings due to computer memory limits. Creating DNA-based seeds with length $n=13$ is currently the upper limit for most computers. It is clear that the amino acid-based screening profits a lot from the very fast calculation times of those long seed lengths with $n > 15$.

Some good DNA-based seeds: 110110101000111 ($n=9$); 1101100011010111 ($n=10$); 111010010100110111 ($n=11$); 111001011001010111 ($n=11$) and 111010110100110111 ($n=12$).

Some good amino acid-based seeds: 11011 ($n=4$); 110111 ($n=5$) and 1011011 ($n=5$).

The smaller the seed is taken (n), the more accurate the screening method will be. However smaller seeds take much longer calculation times. Therefore, one will prefer to take seeds of sizes $n \geq 9$ for DNA-based seeds or $n \geq 4$ for amino acid-based seeds, which will speed up the calculation considerably. Choosing smaller seed sizes ($n < 9$ for DNA-based seeds and $n < 4$ for amino acid-based seeds) makes only sense when homology below 50% is expected.

8.6.2 Creating a new chromosome comparison project

In order to create a new chromosome comparison, genome sequences from e.g. FASTA files, text files or online from EBI or NCBI need to be imported into the database.

In the *Main* window, the *Chromosome comparisons* panel is displayed in default configuration as tabbed view with the *Comparisons* panel, *Decision networks* panel, *Alignments* panel, *Annotations* panel and *BLAST projects* panel in the bottom right part of the window. If desired, the configuration of the *Main* window can be customized as described in 2.3.4. In the manual, however, the default configuration will be used.

To create a new chromosome comparison project, select the entries to be included in the chromosome comparison, select the *Chromosome comparisons tab* in the *Main* window and select **Edit > Create new object...** (🟢). A name for the new chromosome comparison project is prompted for.

The new chromosome comparison project is added to the *Chromosome comparisons* panel in the *Main* window. The date on which the chromosome comparison project was created and last modified is displayed in the default information fields 'Creation date' and 'Modified date' respectively. When more than one chromosome comparison project is present, projects can be sorted and searched using the information present in the default or user-defined information fields. For a detailed explanation of the display options of the *Chromosome comparisons* panel and other grid panels, see 3.2.7.

To delete one or more chromosome comparison projects from the list, select the project and select **Edit > Delete selected objects...** (🔴).

Choose **Edit > Open highlighted object...** (🔍, **Enter**) to open a selected chromosome comparison project. The first time a chromosome comparison project is opened, it will open with the currently selected entries in the *Database entries* panel. As soon as a chromosome comparison project has been saved, selecting **Edit > Open highlighted object...** (🔍, **Enter**) will open the chromosome comparison project with the entries that were present when the chromosome comparison project was last saved.

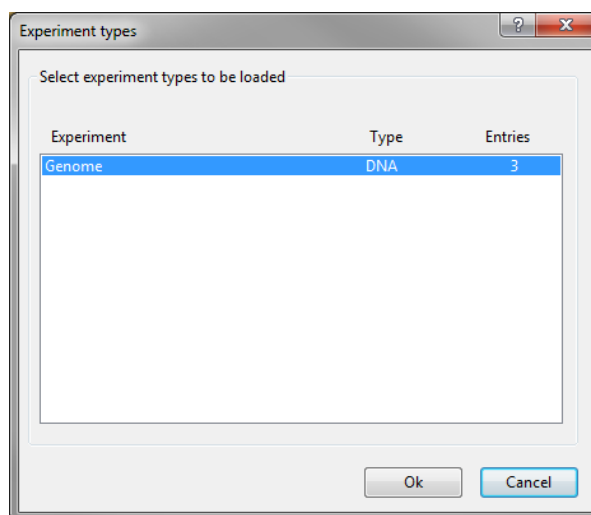


Figure 8.6.1: Select experiment(s) to be included in the chromosome comparison project.

The *Experiment types* dialog box displays a list of available sequence types and the number of associated entries. From this list, the user can select the experiment type(s) that should be included in the chromosome comparison project. Pressing **<OK>** opens the chromosome comparison project in the *Chromosome Comparison* window.

The *Chromosome Comparison* window (see Figure 8.6.2) displays several view panels: their function and arrangement is directed by the fact that a multiple chromosome comparison is an $n \times n$ matrix built up out

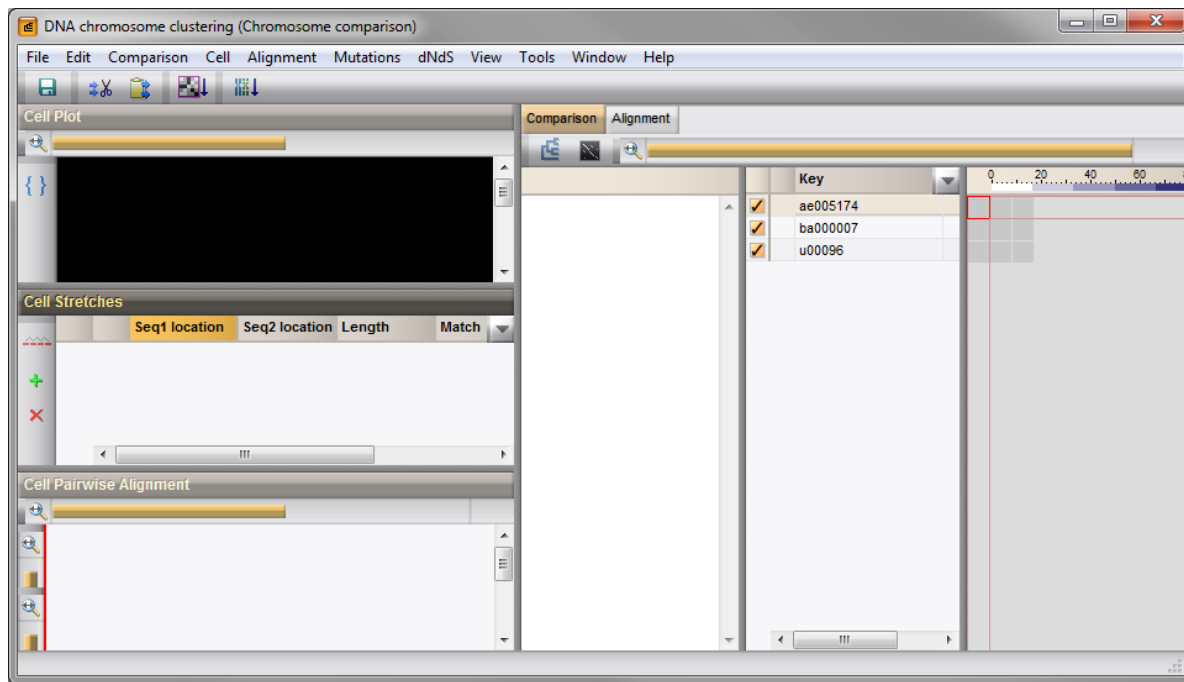


Figure 8.6.2: The *Chromosome Comparison* window.

of $n \times n$ pairwise comparisons. This means that the general overview of a chromosome comparison will be an $n \times n$ matrix (*Comparison* panel), with more detailed views being a selected single cell out of the matrix (a pairwise comparison) or a multiple alignment of $n - 1$ sequences against a template (*Alignment* panel). Related to the single cell view are three panels displaying pairwise comparison information: a dot plot view of the two selected sequences (*Cell Plot* panel), a listing of all stretches of homology found between the two sequences (*Cell Stretches* panel), and a graphical syntenic representation of the two sequences for selected regions of homology (*Cell Pairwise Alignment* panel).

The color of each specific cell indicates its calculation stage. A cell is gray if no calculation has been performed for this cell. If a cell is fully calculated, it will be depicted in green. Green shades indicate cell calculations in progress (if more than one seed was specified for the project). The left-to-right diagonal of the matrix contains those cells representing sequences compared to themselves. Note that calculation is executed in background and may be interrupted at any time. Upon interruption, settings and progress are saved, which allows the calculation to be resumed at stage of interruption.

8.6.3 Defining seed patterns and calculation of chromosome comparisons

When creating a new chromosome comparison (either a DNA-based chromosome comparison or a CDS-based chromosome comparison), the user will have to define one or more seeds, which will direct the screening method. The seeds will specify the sensitivity of the screening (shorter seeds are more sensitive) and the type of screening (DNA-based or amino acid-based). The screening algorithms in the chromosome comparison module accept combinations of seeds: the screenings are carried out successively using the selected seed patterns and newly found stretches are added to the stretch pool, if not yet present. Even for the DNA-based chromosome comparisons, mixing up DNA-based and amino acid-based seeds is allowed. It is clear that, for the CDS-based comparative chromosome mapping, only amino acid-based seeds are accepted, as here the screening is based on the translated regions of annotated CDS features.

Before running the chromosome comparison, the user can specify the project settings and the seeds to be

used for screening.

First, seeds can be specified in the general project settings with **Comparison > Calculate matrix...** (Figure 8.6.1). The seed settings can be modified in the *Seeds* tab of the *Project settings* dialog box (see Figure 8.6.3).

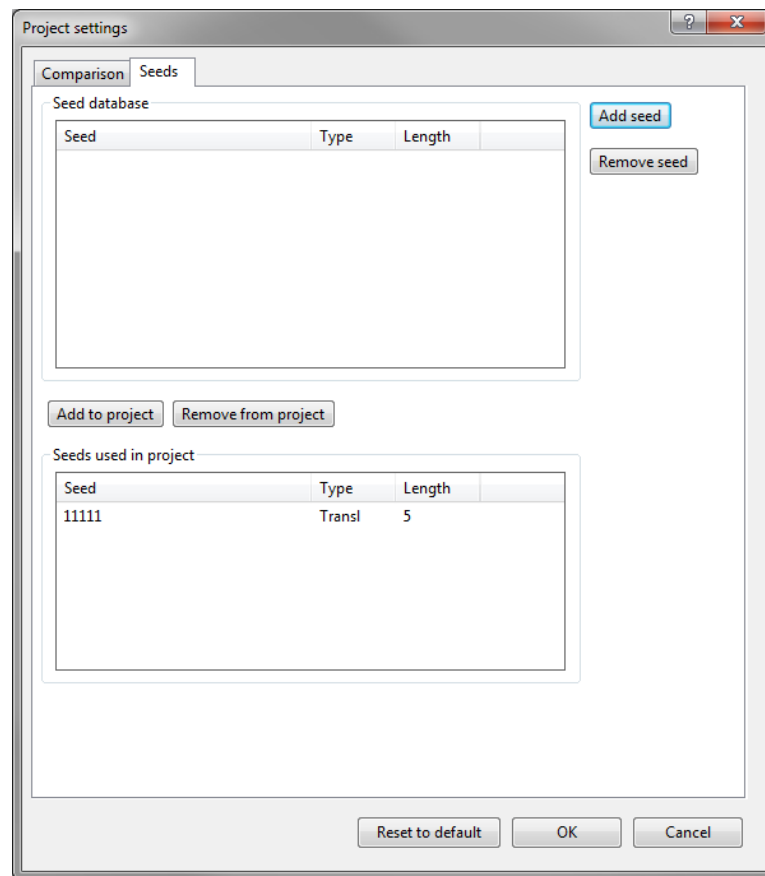


Figure 8.6.3: The *Project settings* dialog box: the *Seeds* tab.

The upper **Seed database** list displays the currently available seeds entered in the seed database. New seeds can be added to the database by pressing the **<Add seed>** button.

The seed editor will appear where you can enter a new seed pattern (see Figure 8.6.4).

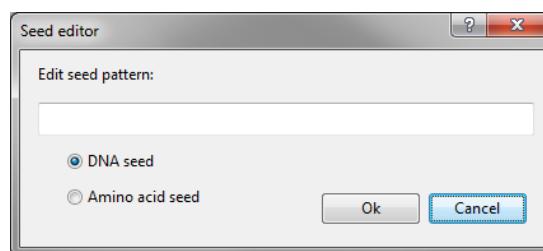


Figure 8.6.4: The *Seed editor* dialog box.

Enter “1” for match positions and “0” for positions which require no match. Other characters are not accepted.

As there are two screening methods possible in the chromosome comparison, namely based on the nucleotide sequences or based on translated sequences, each seed will have its corresponding screening method defined: DNA-based seeds and amino acid-based seeds.

In order to specify which type of seed has been entered, select **DNA seed** to link a DNA-based screening to

the entered seed, and select **Amino acid seed** to link the seed to an amino acid-based screening algorithm. Press **<OK>** to add the seed to the seed database.

A specific seed can be removed from the database by selecting it and pressing the **<Remove seed>** button.

The seed database is comparison-independent and thus will list all seeds created up to now in the different chromosome comparisons. The lower list **Seeds used in project** displays the seeds which are specific to the project currently opened. The seeds listed here specify which screenings have to be performed (DNA-based or amino acid-based) and which seed patterns have to be used. There is no limit set to the number of seeds to be used in the screening rounds.

Seeds listed in the seed database are added to the screening list by selecting the seed in the database and pressing the **<Add to project>** button. A seed can be removed from the screening list by pressing the **<Remove from project>** button.

New projects are initially defined with the standard seed “11111” for amino acid-based screening. If this is not the case, select the amino acid-based seed with pattern “11111” in the seed database and press the **<Add to project>**. If this seed is not present in the seed database, enter it with the **<Add seed>** button.

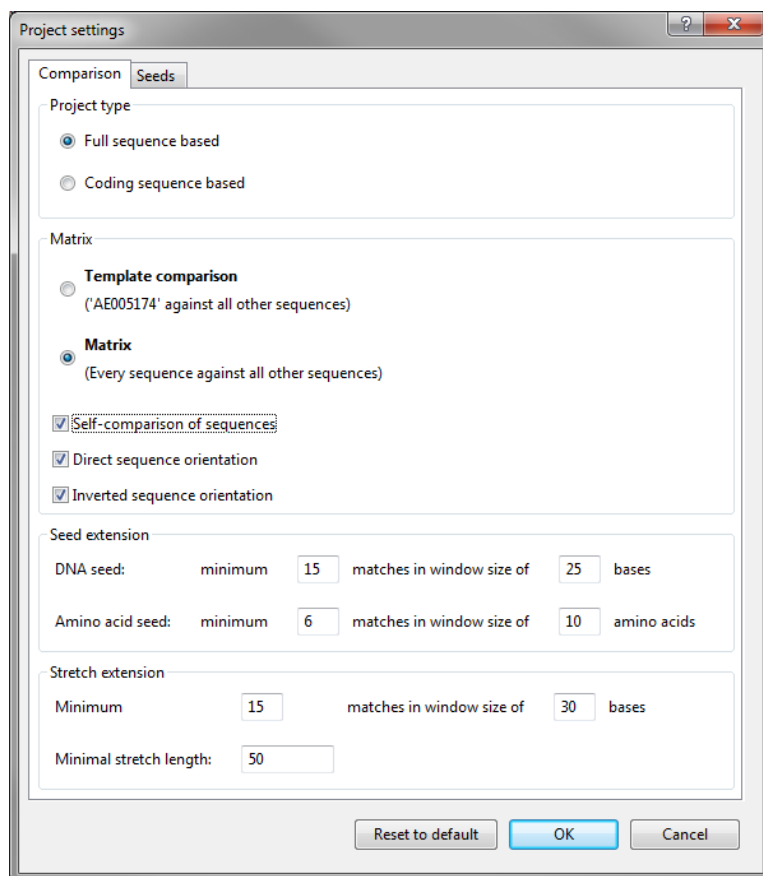


Figure 8.6.5: The *Project settings* dialog box: the *Comparison* tab.

In the *Comparison* tab, settings are grouped according to specifications, which will direct the calculation of the pairwise comparisons.

A **Project type** can be specified. In a **Full sequence based** chromosome comparison, DNA-based seeds are screened for in nucleotide sequences, amino acid-based seeds are screened for in amino acid sequences obtained by translating the nucleotide sequences according to the six possible reading frames. When **Coding sequence based** is selected, only amino acid seeds can be defined for the project. In case DNA-based seeds are defined, the program will produce a warning and remove the DNA-based seeds from the project.

Depending on the further aims of the project, the matrix of compared sequences can be calculated in different

ways:

- The full $n \times n$ matrix can be calculated, for example if phylogenetic studies are to be performed (check box **Matrix** checked).
- Only a selected row can be calculated, for example if a template-based alignment is to be performed (check box **Template comparison** checked). Here, the current selected sequence determines the template of matrix row calculation and alignment calculation.
- Furthermore, one can specify which sequence directions should be calculated. Only if the specific sequence orientation of each pairwise alignment is known, one can skip calculations of opposite orientations (either checking on **Direct sequence orientation** or **Inverted sequence orientation**, depending on the input sequences). Self-comparisons of sequences can be skipped, although these comparisons may also reveal interesting information (check box **Self-comparison of sequences** unchecked).

The **Seed extension** settings and **Stretch extension** settings concern the screening algorithm mathematics.

The **Seed extension** settings control the second step in the screening process: seed locations found in the first screening step are extended upstream and/or downstream within a given window size while counting the matches surrounding the seed location.

If the match count around the seed location does not reach the cut-off value **minimum matches** within a screening **window size**, then the seed is not selected for the third screening step.

Note that the **minimum match** and **window size** settings are separately defined for DNA-based seeds and amino acid-based seeds. If a coding sequence based project type is selected, only the **minimum match** and **window size** settings for amino acid-based seeds are enabled.

The **Stretch extension** settings direct the last screening step, in which the local alignments found in the second step are extended into alignment stretches.

Extension of alignment stretches occurs progressively upstream and downstream until the number of matches drops below the value **Minimum matches** in the specified **window size**. A **Minimal stretch length** will filter out alignment stretches which do not reach the minimal length specified.

Pressing <OK> starts the calculations. When the calculation is started, progression is shown in the status bar at the bottom of the chromosome comparison view. Moreover, the calculation progress is indicated on the gray rectangles in the matrix, turning green when calculated.

Pressing the <Stop> button will cause the program to stop the calculation. Project settings and pairwise comparisons that were finished are displayed in the similarity matrix and saved to disk. At this point, the project window, and even the program, can be closed.

The left-to-right diagonal of the matrix contains those cells representing the self-comparisons. Cells that will not be calculated (e.g. when the option self-comparison of sequences is unchecked) remain white boxes during and after calculations. The status bar at the bottom of the main window displays the calculation status of the current cell. A typical cycle of cell calculation is (1) formatting the first sequence by start of a new matrix column, (2) formatting the second sequence and (3) calculating the cell.

When all cells in the matrix are processed, each cell displays an identity score, with a corresponding color scale. The scale goes from black, corresponding with 100% identity, over blue towards white (0% identity). Initially, no clustering is calculated and no dendrogram is shown. This view, i.e., the matrix with identity scores, is the most global view that can be shown from the finished project. A second view, the matrix of dot plots, can be called by selecting **Comparison > Show dot plot view** (📊).

This view shows each cell from the matrix as a reduced dot plot, where the first sequence is plotted along the X-axis against the second sequence along the Y-axis (see Figure 8.6.6). If a dot plot is selected in the similarity matrix, the detailed matrix of dot plots is simultaneously displayed in the cell viewing panels.

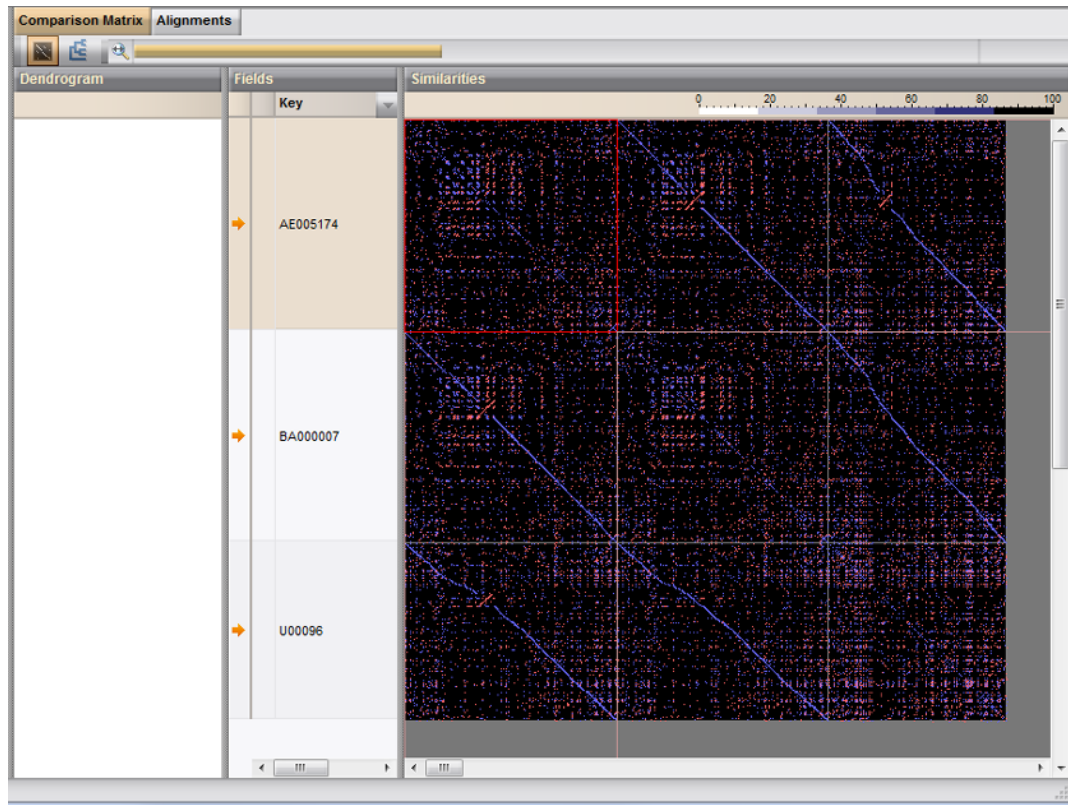


Figure 8.6.6: Matrix of dot plots view.

Within all dot plots, blue dots represent stretches of homology between both sequences in the direct orientation, whereas red dots represent stretches of homology between the first sequence (normal direction) and the second sequence in inverse orientation (dot colors can be user-specified, see Figure 8.6.10).

If chromosome comparisons are created from a considerable number of entries, the matrix dot plot can be zoomed in and zoomed out with the zoom slider at the top of the panel. One can show the matrix in a smaller view by adjusting the zoom sliders to the left (zoom out). The view can be enlarged by adjusting the zoom sliders to the right (zoom in).

A clustering of the entries, based upon the pairwise identity scores, can be calculated with **Comparison > Cluster matrix** (🔍).

A print preview mode where the dendrogram (if calculated), the entry list and the identity score matrix are shown, is called with **File > Print > Matrix...** (see Figure 8.6.7).

The print preview can be zoomed in or out by use of the zoom slider on the left.

Three yellow dashes are displayed at the top of the previewed page. They indicate the left margins of, respectively, the dendrogram figure, the entry key list, and the identity score matrix.

By moving the yellow dashes left or right, one can adjust the space for each of these columns, or even hide a specific column.



It may be that two dashes overlap each other, so that one column is hidden. To uncover the hidden dash and its column, move the upper dash to the right.

The information in the print preview can be scaled to fit the page size by pressing the 📐 button. Another user-defined scale can also be applied when pressing the 📏 button and entering the scaling factor. The default scaling factor is set to 100%. Also the font size can be increased and decreased by pressing the 🔍 or 🔍 button, respectively. Further, one can choose to print the dot plot information instead of the similarity values, therefore, press the 📄 button.

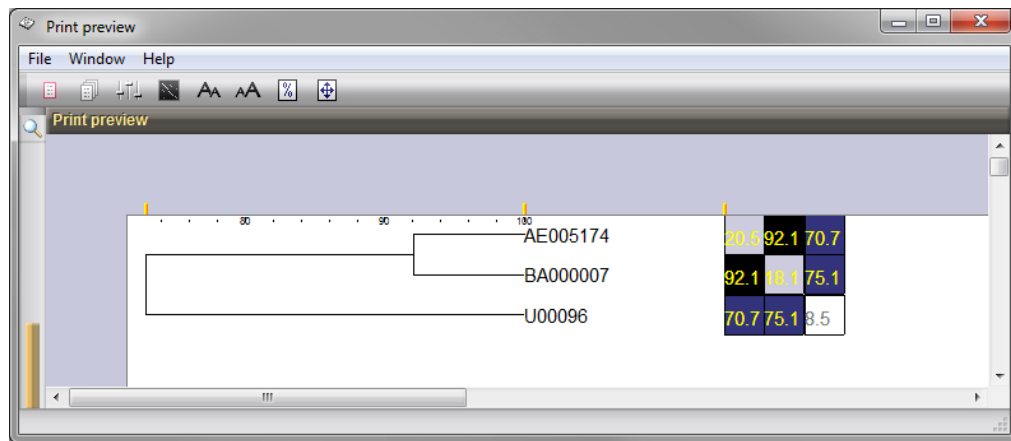


Figure 8.6.7: Print preview from the comparison matrix.

Choose **File > Printer setup...** (🖨️) to call the Windows printer setup dialog box, in order to select a specific paper orientation or a specific printer.

Selecting **File > Print selected pages** (🖨️) or **File > Print all pages** (🖨️) will print the selected/all pages as arranged in the print preview, whereas selecting **File > Copy page to clipboard** will copy the current page to the clipboard.

Closing the *Chromosome comparison matrix print* window is done by selecting **File > Exit**.

The chromosome comparison analysis results can be reset at any time by selecting **Comparison > Reset**. This action will permanently delete the existing similarity matrix.

A chromosome comparison that was saved can be called again at any time. Additional entries can be loaded to this chromosome comparison without the need to recalculate the full matrix. Pairwise chromosome comparisons that need to be calculated after reopening a chromosome comparison are indicated in the similarity matrix by a gray rectangle. The similarity values of previously calculated chromosome comparisons are directly loaded from the database. To complete the full matrix comparison, the missing comparisons of the matrix are calculated.

If a cell of the similarity matrix is selected either in the identity score matrix or in the dot plot matrix from the *Comparison* panel, corresponding pairwise sequence alignment information is displayed in the cell view panels. The *Cell Plot* panel displays a more detailed dot plot of the two selected sequence entries, the *Cell Stretches* panel lists all stretches of homology found between the two sequences and the *Cell Pairwise Alignment* panel presents the sequence alignment view (see Figure 8.6.8). Updating these panels may take a few seconds for large comparisons in case super stretches are switched on (see 8.7.1), as calculation of the super stretches has to be carried out. The progress of calculation is indicated in the status bar at the bottom of the window. The key information from the two selected sequences represented in the cell view panels is indicated in the status bar at the bottom of the window.

To allow a correct interpretation of the different cell view panels (the *Cell Plot* panel, *Cell Stretches* panel and *Cell Pairwise Alignment* panel), the following needs to be mentioned. By default, the items listed in the *Cell Stretches* panel are stretches of continuous homology found between the two concerned sequences (obtained after seed extension). This means that these stretches do not contain any gap when aligned, and have the same or inverted direction on the sequences. The corresponding dot plot displays the positions of all these stretches of continuous homology in an XY-plot. Along the X-axis is the first sequence plotted, along the Y-axis the second sequence. Blue spots indicate stretches with direct orientation on both sequences, red spots correspond to stretches with direct orientation on the first sequence and inverted orientation on the second sequence (default color settings).

Dots on the dot plot lying adjacent to each other are an indication of longer regions of homology between

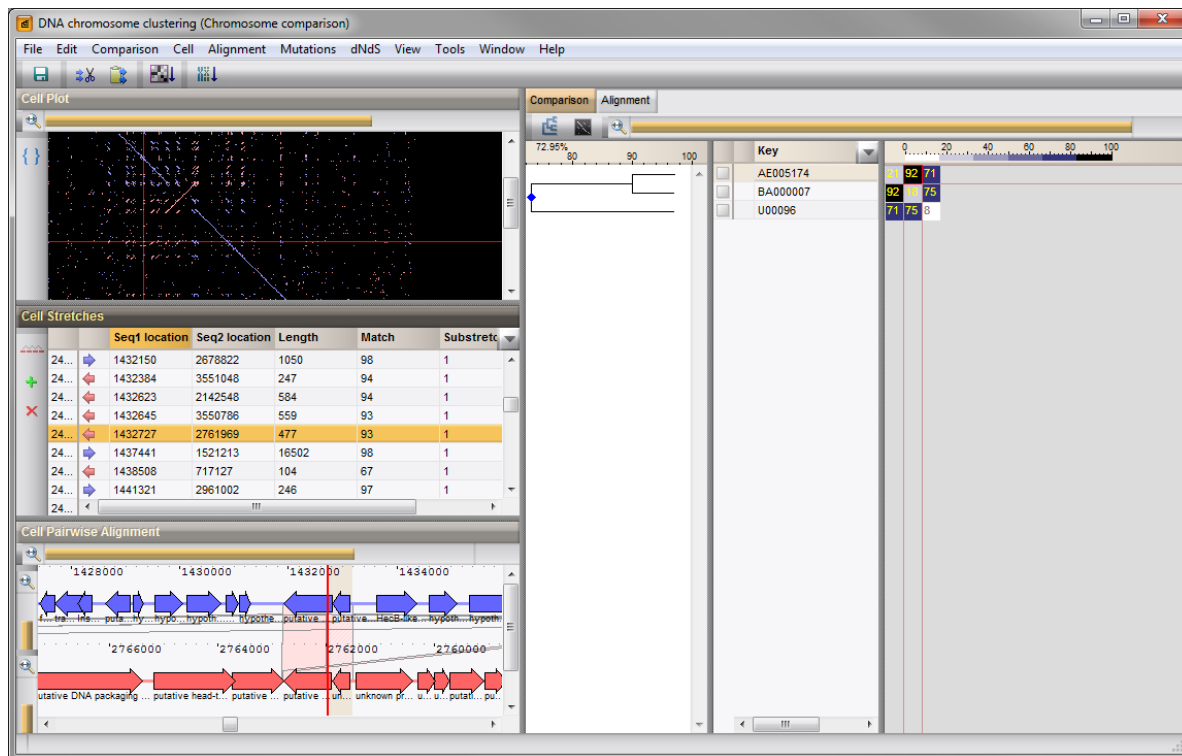





Figure 8.6.8: The *Chromosome Comparison* window after calculation of the pairwise similarity matrix.

both sequences. They indicate regions of discontinuous parallelism, i.e. stretches of continuous homology interrupted by gaps in one of the two sequences. If such regions of discontinuous parallelism occur, one can try to link the individual stretches into one block, i.e. into one stretch of discontinuous homology. Such blocks of discontinuous parallelism created by joining stretches of continuous homology are further denoted as *super stretches*.

Super stretches do have gaps in the alignment and can have direct and inverted orientation. A super stretch with direct orientation is exclusively built up by stretches with direct orientation, whereas a super stretch with inverted orientation consists of stretches with inverted orientation.

Within the cell view panels, all stretches are shown by default. With **Cell > Cell Stretches > Switch stretch/superstretches listing** () super stretches will be mapped. By default, both stretches and super stretches are plotted according to their orientation: blue dots indicate stretches of homology between both sequences in equal orientation, whereas red dots indicate stretches of homology between the first sequence in direct orientation and the second sequence in inverse orientation.

A spot within the dot plot panel can be selected by clicking with the mouse pointer on the spot. The selector will automatically jump to the nearest spot. A red X-Y cross indicates the current position selected on the stretch, whereas the gray X-Y selection indicates the boundaries of the selected stretch or super stretch (depending on selected stretch modus).

Within the *Cell Stretches* panel, stretches mapping on the second sequence in direct orientation are indicated with a  sign, those mapping on the second sequence in inverted orientation are marked with a  sign. Selecting a stretch or super stretch (depending on the selected stretch class modus) within the stretch list causes the dot plot to select and focus on the corresponding stretch in the dot plot and causes the *Cell Pairwise Alignment* panel to align the concerned sequences. The Pairwise Alignment view inverts the second sequence if a stretch is selected which maps on the second sequence in inverted orientation.

Within the *Cell Pairwise Alignment* panel, stretches can also be selected by clicking with the mouse on the mapped homology blocks. The corresponding stretch or super stretch will be selected within the stretch list

and the dot plot view will focus on the mapping position of the stretch.

When using the zoom slider (top of dot plot), focus is kept around the cursor position.

The stretch/super stretch list from the *Cell Stretches* panel can be sorted according to stretch positions, length, identity score and number of sub-stretches (for sorting super stretches) by selecting the corresponding field in the stretch list header, clicking the right mouse button, and selecting **Arrange by field**.

The stretches list can be exported with **File > Export > Stretch list...** The path, file name and the type of file used for exporting the data can be specified in this dialog box: in the "Tab-delimited file format", the columns are separated by tabs, whereas in the "Column-based file format", column text is lined using spaces. Use the tab-delimited format to export to programs such as Microsoft Excel or Access.

Within the *Cell Pairwise Alignment* panel, a selected stretch is drawn in hatched color (see Figure 8.6.9). The alignment can be zoomed in and zoomed out with the zoom slider at the top of the panel. Two other sliders "Distance cutoff" and "Identity cutoff" are present at the left of the *Cell Pairwise Alignment* panel in default configuration.

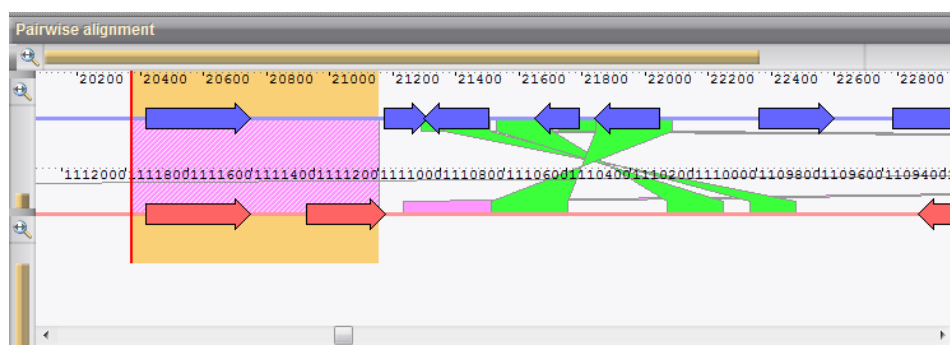


Figure 8.6.9: The *Cell Pairwise Alignment* panel. The stretch direction (blue/red) and inclusion state (purple/green) of the stretches are mapped.


The "Distance cutoff" slider indicates a distance in base pairs ranging from zero to the full sequence length. This distance value has to be seen as a tolerance scope for the degree of parallelism of the surrounding stretches to be plotted around the currently selected stretch. A low distance cutoff, for example 1,000 base pairs, will only plot stretches being parallel to the currently selected stretch and having a maximum shift of 1,000 base pairs. A high distance cutoff, for example half of the sequence length, will plot nearly every cross identity between the two sequences, which might sometimes result in a very complex figure.

The "Identity cutoff" slider indicates an identity score value between zero and 100 percent. This slider presents the minimum identity score of the stretches or super stretches used in the *Cell Pairwise Alignment* panel.

Display settings belonging to the *Cell Plot* panel and the *Cell Pairwise Alignment* panel are grouped and can be called with **View > Dot plot display settings...** This pops up the *Display settings* dialog box (see Figure 8.6.10).

The two color fields adjacent to **Stretches with same direction** and **Stretches with opposite direction** define the colors to be used when the mapping modulus of the stretch classes states to map the direction of the stretches. Stretches with opposite direction are those stretches for which the second sequence needs to be reverse-complemented in order to get a correct alignment.

The two color fields adjacent to the fields **Stretches included in alignment** and **Stretches not included in alignment** define the colors to be used when the mapping modulus of the stretch classes states to map the inclusion state of the stretches in the alignment. See 8.7 for further explanation of this item.

The stretch colors can be changed by pressing the  button at the right side of each color field.

If the check box **Use identity color codes for stretches** is checked, stretches will be mapped in a color code

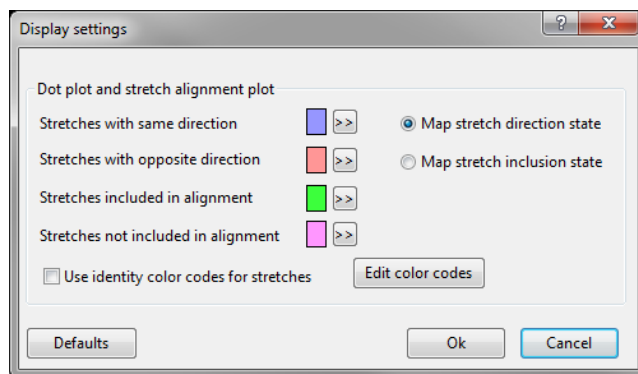


Figure 8.6.10: The *Display settings* dialog box.

range which reflects the identity score of the stretch. The color code ranges from the minimal stretch identity (as set within the stretch import settings, see 8.7) to 100%.

The color code range can be edited with the *Identity color codes* dialog box which is called by pressing the **<Edit color codes>** button (see Figure 8.6.11).



You may need to decrease the ***Identity cutoff threshold*** to reveal the colors corresponding to stretches with lower identities.

The plot view is printed with **File > Print > Dot plot...**

To copy the plot to the clipboard, choose **File > Export > Copy dot plot to clipboard**. The plot will be printed or exported to the clipboard as displayed on the screen.

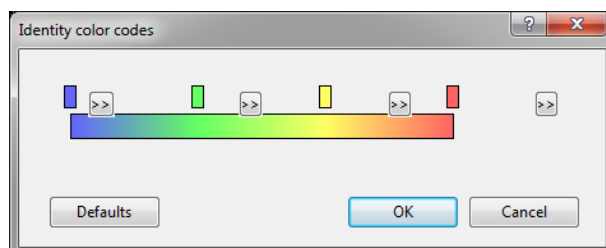


Figure 8.6.11: The *Identity color codes* dialog box.

This function overwrites the stretch direction (and stretch inclusion) mapping functions. Additionally, the same color code is shown within the *Cell Pairwise Alignment* panel.

The colors can be changed by pressing the **>>** button on top of the color scale.

Pressing the **<Defaults>** button, reset the color codes to their default colors.

Chapter 8.7

Multiple chromosome alignments

8.7.1 Introduction

The detailed information about pairwise homologies between sequences obtained within a multiple chromosome comparison can be used for calculating alignments between these sequences. When generating alignments of full genomes, one will notice fundamental differences with conventional alignments of coding sequences or proteins. Due to genome rearrangements, blocks of homology between related genomes may have lost fixed orientation and/or order or may have been repeated or deleted. To deal with these problems of rearrangements and to visualize the *syntenies* and *parallelisms* of these genomes in a comprehensible way, pointing out a sequence as guiding reference and aligning the other sequences against this reference is the most obvious approach. A drawback may be that conserved blocks present in an aligned subset of genomes but absent in the reference genome are not identified, as these blocks will not be represented. However, repeating the analysis with different reference genomes solves this problem. A second small shortcoming of alignment against a reference sequence is that alignments generated with different reference sequences may sometimes be inconsistent, i.e. that two genomic positions aligned in one reference system might not fall together anymore in another reference system. But again this problem is not of high relevance as these inconsistencies only occur with blocks of repeated sequences.

Within the chromosome comparison, the calculation of chromosome alignments is based upon the idea of joining stretches with a significant degree of parallelism into *super stretches*. The stretches found with the multiple chromosome comparison algorithm are stretches of continuous homology, i.e. aligned sequence stretches with no gaps. If several stretches of continuous homology concatenate one after another, they form a region of discontinuous homology. Within such regions of discontinuous homology, limited gaps should be allowed in order to get correct alignments of the concatenated blocks of continuous homology (stretches). In the chromosome comparison, concatenated stretches are referred to by the term super stretch. Locating the stretches into super stretches is a crucial step in setting up the multiple chromosome alignment. Therefore, the settings for joining stretches into super stretches are discussed in detail.



Calculating a multiple chromosome comparison requires the Genome Analysis Tools module (GA) to be present in your BioNumerics configuration.

8.7.2 Calculating a chromosome alignment

Within the *Alignment* panel (on top of right panel group), six panels are displayed in default configuration: the *Overview* panel (upper window panel), the *Tools* panel (middle window panel), the *Mutation analysis* panel, *Sequence search* panel, *Feature search* panel and *dNdS search* panel. The latter four panels are arranged as tabs in the bottom part of the window. Furthermore, the detailed information in the panels on the left (the *Cell Plot* panel, *Cell Stretches* panel and *Cell Pairwise Alignment* panel) is mutually linked to

the selected features in the *Alignment* panel on the right.

It is obvious that, before a chromosome alignment can be executed, the matrix data from the chromosome comparison should be calculated.



If a chromosome comparison will only serve as template for a chromosome alignment, the matrix does not need to be fully calculated. If the guiding sequence for the chromosome alignment is known, only the matrix row containing this guiding sequence as template should be calculated. In that event, select the row with the entry that should serve as guiding sequence, and in the general project settings, check **Template comparison**.

To start a new alignment, call the *Alignment Settings* dialog box with **Alignment > Create new alignment...** (see Figure 8.7.1).

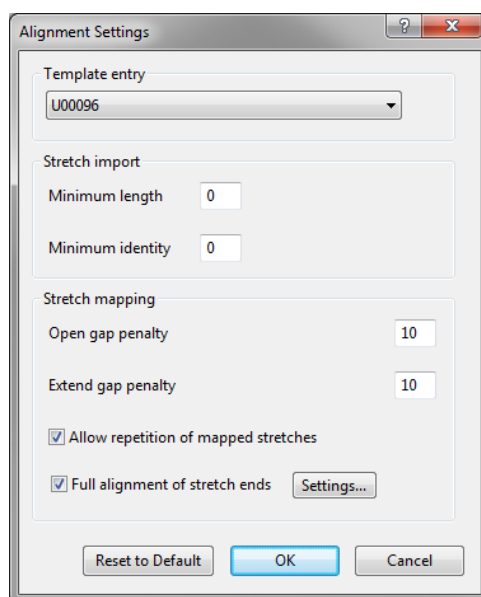


Figure 8.7.1: The *Alignment Settings* dialog box.

In general, multi-chromosome alignments are set up by pointing out a sequence as guiding reference and aligning the other sequences against this reference. In the further description of this section, this guiding sequence will be referred to as *template sequence* and the sequences aligned to the guiding sequence as *query sequences*.

In the *Alignment Settings* dialog box, the **Template entry** can be selected from the drop down list, displaying all entries from the chromosome comparison. The chosen entry will be used as template for alignment.

An additional filtering of stretches to be used in the alignment calculation can be set up with the settings displayed in the **Stretch import** criteria panel.

The minimal length for the stretches to be imported in the alignment can be entered in the field **Minimum length**, whereas a threshold identity score (as a percentage) can be specified in the field **Minimum identity**. It is obvious that these cutoff values cannot be more relaxed than the minimal stretch length and minimal stretch identity values entered in the stretch extension settings from the *Project settings* dialog box (see Figure 8.7.1), as these settings have discarded stretches not fulfilling the pairwise comparison calculation settings.

Under the **Stretch mapping** criteria, the general alignment settings can be specified. The **Open gap penalty** is the cost to create a gap between two stretches (if stretches are not on the same diagonal). The **Unit gap penalty** is the cost to increase an existing gap between two stretches. The unit gap cost is used to penalize larger gaps between two stretches.

The score of a stretch within the alignment algorithm is defined as the product of the stretch length with

the stretch identity (which is the stretch match count). In order to find the most optimal stretches upstream or downstream of a defined stretch X, surrounding stretches are weighted against stretch X with following calculation:

x_1 and y_1 are the XY-coordinates of the start positions of stretch X, len_1 is the length of stretch X

x_2 and y_2 are the XY-coordinates of the start positions of the next stretch to be added to the super stretch, len_2 is the length of this stretch

Upstream stretches:

$$gap_1 = |(x_2 - y_2) - (x_1 - y_1)|$$

$$gap_2 = \min(|x_2 + len_2 - x_1|, |y_2 + len_2 - y_1|)$$

$$maxd = gapcost.gap_1 + unitcost.gap_2$$

$$besthit = \frac{stretchscore}{\max(maxd, 1)}$$

Downstream stretches:

$$gap_1 = |(x_2 - y_2) - (x_1 - y_1)|$$

$$gap_2 = \min(|x_1 + len_1 - x_2|, |y_1 + len_1 - y_2|)$$

$$maxd = gapcost.gap_1 + unitcost.gap_2$$

$$besthit = \frac{stretchscore}{\max(maxd, 1)}$$

The stretch with highest weight will be the adjacent stretch to stretch X within a super stretch. The calculation of super stretches starts with the stretch with highest stretch score (seed). Progressive extension upstream and downstream of this stretch until no stretches can be added anymore, leads to the generation of the first super stretch. In a next round, the remaining free stretch with highest score is taken as next seed for generating the next super stretch. This process is repeated until no more super stretches can be generated. A higher gap cost penalty will prevent from making big gap jumps from one diagonal to another whereas a higher unit cost will prevent from making longer gap jumps along a diagonal. Note that the super stretch score is calculated by adding the scores of the stretches included in the super stretch.

If the check box **Allow repetition of mapped stretches** is checked, a query subsequence (Figure 8.7.2: sequence 2, subsequence C) that is aligned on one template subsequence (Figure 8.7.2: sequence 1, subsequence A) can also be aligned on another template subsequence (Figure 8.7.2: sequence 1, subsequence B). If not checked, the query subsequence would only be aligned with one of the two template subsequences (in this case with A, as the cost to align with B would be greater).

Additionally, settings for fine-tuning the alignments at the stretch ends can be defined in the *Full alignment settings* dialog box (see Figure 8.7.3). These settings are called from within the *Alignment Settings* dialog box by pressing the <Settings> button next to the **Full alignment of stretch ends** option.

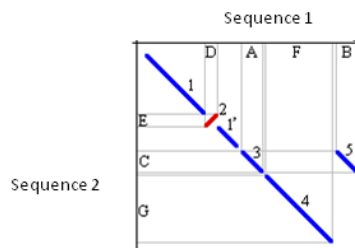


Figure 8.7.2: Illustration of stretch mapping criteria.

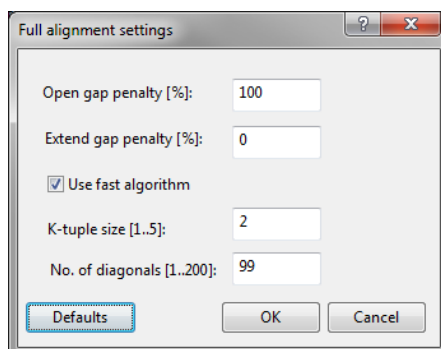


Figure 8.7.3: The *Full alignment settings* dialog box.

These specific settings are applied for the alignment of the sequence ends of stretches covering a range of 400 bp, starting from both ends of a stretch. Thus, if a stretch is smaller than 800 bp, these fine-tune alignment settings will be applied to the whole stretch. In most cases, the default alignment parameters will work fine. However, in case of large sequences with many overlapping regions and/or many uncertain positions, for example when aligning less homologous chromosomes, better results may be obtained by for example increasing the number of diagonals.

The default full alignment settings are 100% for **Open gap penalty** and 0% **Extend gap penalty**. The word size (**K-tuple size**) and **No. of diagonals** define how accurately the algorithm should work. The word size also relates to the alignment technique that is used, i.e. the algorithm of Needleman and Wunsch [30], which creates a lookup table of groups of bases to accelerate the alignment. Up to a certain extent, the smaller the word size is taken, the more accurate, but the slower the alignment. However, if the word size is set too small (e.g. the minimum of 1), the alignment may become less correct unless the number of diagonals is extremely high. The number can vary between 1 and 8. The suggested default setting is a word size of 2. The number of diagonals relates to the same algorithm of Needleman and Wunsch. The number of gaps the algorithm can create depends on the number of diagonals specified. The larger the number, the more accurate, but the slower the calculations. For sequences with poor similarity, e.g. the sequences near the stretch ends, the number can be increased. The number can vary between 1 and 100. The suggested default setting is 99 diagonals.

The **Use fast algorithm** option offers an interesting accelerated algorithm compared to the standard algorithm.

Pressing <OK> in the *Alignment Settings* dialog box will start the calculation in a background thread with indication of its status in the status bar at the bottom of the *Chromosome Comparison* window.

When the calculation is finished selecting **Cell > Cell Stretches > Map stretch on alignment** (+) maps the currently selected stretch or super stretch on the alignment.

Conversely, removing an aligned sequence can be done by selecting the corresponding region in the align-

ment and choosing **Cell** > **Cell Stretches** > **Remove stretch from alignment** (✖). The corresponding stretches will be unmapped.



Mapping stretches on the alignment may not always be possible: a stretch may be mapped only partially if it overlaps with another stretch which already belongs to an existing alignment block.

8.7.3 Alignment overview and Alignment detail panel

The *Overview* panel shows a listing of the aligned query sequences, plotted as horizontal graphs. On top is a scaling of the template sequence plotted together with a gray graph which represents the number of aligned sequences (consensus plot) along the Y-axis in function of the alignment position on the template sequence (X-axis). In the listing, graphs plot the identity score (Y-axis) of the aligned blocks of the query sequence in function of alignment position on the template sequence (X-axis). The red dots (standard color definition) indicate blocks of the aligned sequence which are inverted in the alignment, blue dots represent blocks of the aligned sequence in their original orientation in the alignment (see Figure 8.7.4). The alignment overview plot can be zoomed in and out with the zoom slider at the top of the panel, while the cursor (red line) is kept in focus.

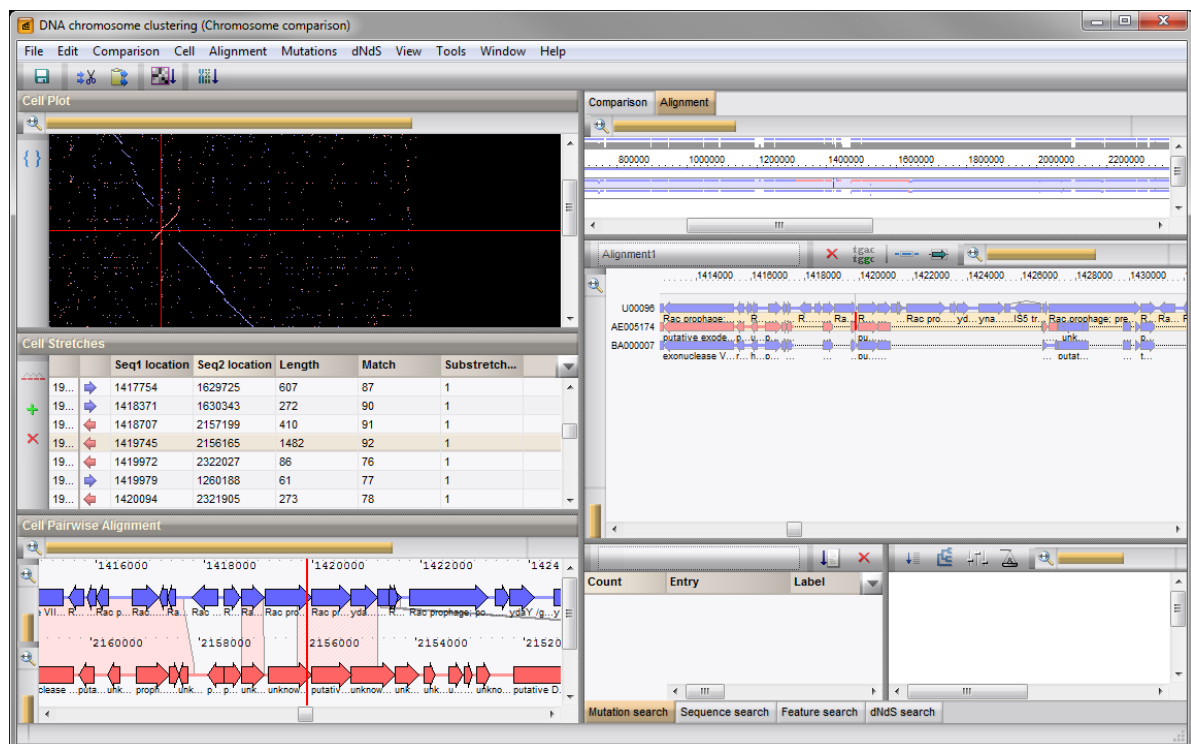


Figure 8.7.4: The *Chromosome Comparison* window with the *Overview* panel and *Tools* panel at the right.

Both in the *Overview* panel and *Tools* panel, a change in cursor position and a selection of sequences can be made with the mouse pointer. Making a selection or changing cursor position in the *Overview* panel updates the selection and cursor position in the *Tools* panel. Conversely, changing selection or cursor position in the *Tools* panel updates the selection in the *Overview* panel.



In the *Tools* panel, the cursor indication has a double function: the gray line indicates the alignment position whereas the red line indicates the current pairwise sequence selection (template plus one of the query sequences). This pairwise sequence selection is of importance for the dot plot functionality.

The alignment detail view can be zoomed in and out with the zoom slider at the top of the panel.

To change the alignment detail view from feature view (graphical) to text view (nucleic and amino acid information), select **Alignment > Show text view** (🔍). This text view does not use color indications for the different nucleotides. However, the same view can also be obtained in colored mode, by zooming with the zoom slider in the feature view. To switch back from text view to feature view, choose **Alignment > Show text view** (🔍) again.

Choose **Alignment > Selection > Snap to stretch** (📏) to activate the "Snap to stretch" option. Using this option, each change in cursor position results in selecting the respective stretch or super stretch, depending on which option is selected in the *Cell Stretches* panel.

With **Alignment > Selection > Snap to feature** (👉) the "Snap to feature" option is activated. With this option, each change in cursor position results in automatic selection of the respective feature.

The options discussed above are particularly useful when the interest is on a particular stretch or feature, as only this information will be printed or exported.

A selected stretch, feature or a specific region of the stretch alignment view can be printed by selecting **File > Print > Alignment...**

Display settings for the detailed alignment are available in the *Alignment display settings* dialog box which is called with **View > Alignment display settings...** (see Figure 8.7.5).

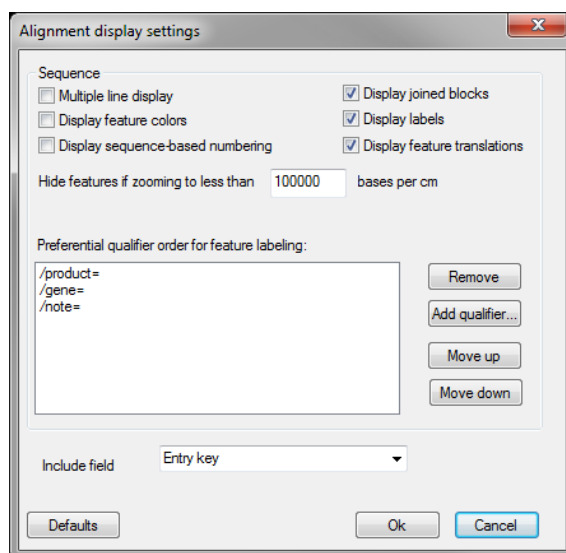


Figure 8.7.5: The *Alignment display settings* dialog box.

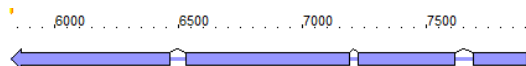
Multiple line display defines whether the sequence alignment view should be represented in a multiple line modus (checked) or a single line modus (not checked). In multiple line modus, sequence plots are continued on a next line when reaching the right margin of the output medium (screen/printer). This modus treats the sequence data in the same way as text is represented in a text processor. The multiple line display modus scrolls in the vertical direction. The single line modus plots the sequence figure(s) on a single continuous line and scrolls in the horizontal direction.

Display feature colors defines whether features should be displayed in the feature color as defined by the sequence layout (checked) or not (not checked). If no feature color is displayed, features are mapped in the color defining the stretch orientation (see further).

If the check box **Display sequence-based numbering** is checked, position numbering is indicated above each aligned query sequence. The position numbering refers to the original sequence positions. If this check box is unchecked, only the template sequence numbering will be shown.

The check box **Display joined blocks** defines whether features built up of joined sequence stretches (for example eukaryotic CDS features with exon-intron structures, disrupted genes...) should be mapped as

joined blocks (checked) or as one continuous block (not checked) representing start and end position of the feature. For example, the joined block modus will display a feature as



whereas the continuous block modus would display this feature as



Display labels defines whether feature labels should be displayed on the plot (checked) or not (unchecked).

With **Display feature translations** checked, CDS feature translations are shown underneath the coding sequences (at full zoom in), otherwise translation is not shown.

The drawing of features on the plot by higher zoom out levels can be eliminated by decreasing the cut-off value entered in the field **Hide features if zooming to less than**. This cut-off value relates to the printed plot as the zooming factor is expressed in bases per cm. A high cut-off value, for example 10,000,000 will result in feature drawing even by full zoom out, whereas a cut-off value of 0 will never draw the features.

The qualifier from the feature qualifier list that should be used for feature labeling, can be specified by choosing a **Preferential qualifier order for feature labeling**. For example, if the qualifiers `"/product=`", `"/gene=`" and `"/note=`" are defined as preferential qualifiers for labeling (in this order), then the label of choice will be the text defined under the qualifier `"/product=`" of the feature annotation. If a feature is not annotated with a `"/product=`" qualifier label, the next qualifier of preference is taken, namely `"/gene=`", and so on. The feature labeling adapted via the list of preferred qualifiers as defined here, overwrites the feature label specified in the sequence layout (see sequence editor). This means that, if no list of preferred qualifiers is defined, this feature label will be displayed. The list of preferred qualifiers for labeling can be edited: the **<Remove>** button removes the currently selected qualifier from the list and the **<Add qualifier>** button calls a dialog box, which allows you to select a qualifier out of the EMBL-GenBank feature annotation table. Pressing the **<OK>** button adds this qualifier to the list of preferred qualifiers for feature labeling. With the **<Move up>** and **<Move down>** button, you can change the order of the qualifiers within the list.

With the option **Include field**, a database information field can be added in front of each genome from the genome listing. This information field can be picked from a drop-down menu. Press **<OK>** to confirm the settings.

A print preview mode of the alignment is called with **File > Print > Alignment...** (see Figure 8.7.6).

The print preview can be zoomed in or out by use of the zoom slider on the left.

Choose **File > Printer setup...** (🖨️) to call the Windows printer setup dialog box, in order to select a specific paper orientation or a specific printer.

Selecting **File > Print selected pages** (📄), or **File > Print all pages** (📄) will print the selected/all pages as arranged in the print preview, whereas selecting **File > Copy page to clipboard** will copy the current page to the clipboard.

Note that the currently selected part of the sequence plot will be printed as displayed on the screen. This means that, if the plot is fully zoomed in up to base level display, printing will also be done at base level. Copying the sequence plot figure to the clipboard follows the same concept: the clipboard figure proportions (zoom, spacing...) will be as currently set on the screen.

Closing the *Chromosome comparison alignment print* window is done by selecting **File > Exit**.

After calculation of an alignment, two view types are available for the *Cell Plot* panel and the *Cell Pairwise Alignment* panel: either stretches are marked in function of their orientation (direction state) or they are marked in function of their use (inclusion state) in the alignment. If the orientation view modus is active, blue spots will be used for mapping stretches or super stretches with direct orientation on both sequences and red spots for mapping stretches/super stretches with direct orientation on the first sequence and inverted ori-

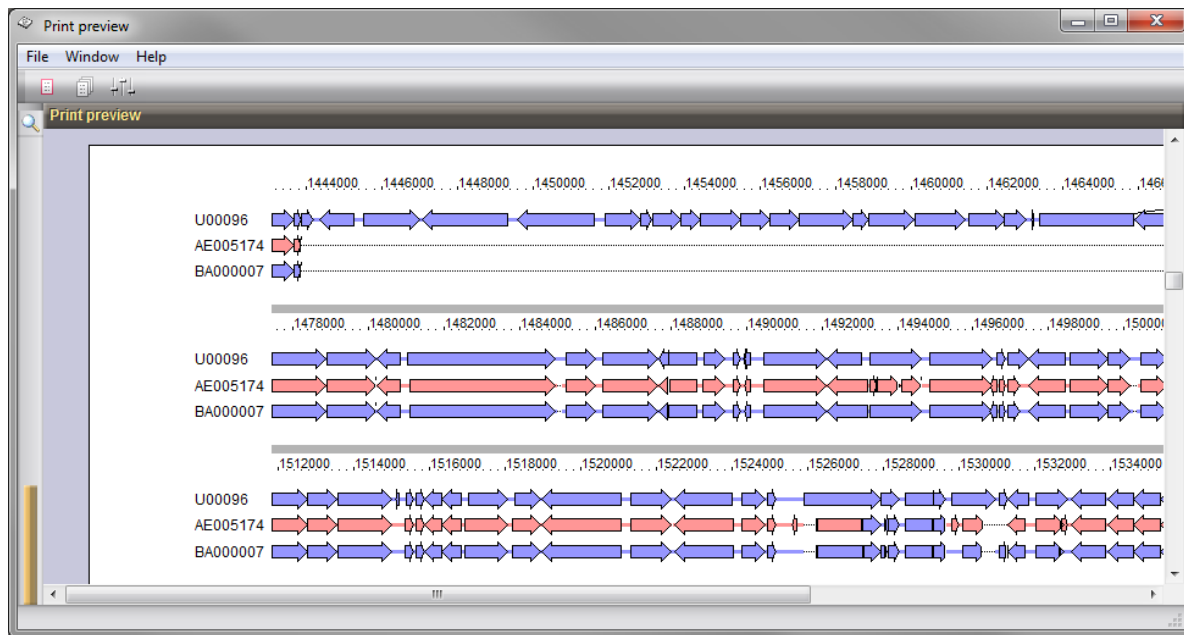


Figure 8.7.6: The *Chromosome comparison alignment print window*.

entation on the second sequence (default color definition). If the stretch inclusion modus is active, stretches or super stretches which are fully or partially used within the alignment will be mapped in green, whereas unused stretches/super stretches will be mapped in pink. It is clear that the stretch inclusion view is a very helpful tool for controlling the quality of the automated alignment.

Clustering of stretches along diagonals within the dot plot indicates regions of longer homology between the two selected sequences, and these regions of homology should be part of the alignment. This means that, in the stretch-inclusion dot plot view, such a clustering of stretches should be mapped in green, whereas the mostly smaller stretches of lower homology dispersed all over the two sequences are irrelevant and should be mapped in pink. However, it may happen that longer regions of homology are skipped in the alignment and appear as "not included" stretches within the stretch-inclusion plot. This observation may have two possible explanations.

- First, the region may be a repeated sequence and did not find a mapping position on the partner sequence as its matching position(s) is (are) already aligned with a copy (copies) of the repeat. In this case, the fact that these stretches are not used in the alignment is justified.
- Secondly, the automated alignment did not produce an optimal alignment and incorrectly skipped some regions of smaller homology (mostly smaller inversions in longer regions of homology). In this case, one should try other settings for the alignment in order to find a better detection of super stretches. Finding these regions of smaller homology to enhance the alignment of rearrangements can be achieved by specifying more stringent stretch extension settings, so that shorter super stretches are obtained.

Select **View > Dot plot display settings...** from the *Chromosome Comparison* window to switch between the view modus *Map stretch direction state* or *Map stretch inclusion state*.

8.7.4 Mutation analysis

In the *Mutation search* panel, the results of mutation search calculations are listed and visualized on the sequences. The mutation search function allows the detection of the following mutation types:

- Two types of silent mutations: these types of mutations cause a nucleotide change which does not lead to an amino acid change. A silent mutation localized within a non-coding sequence is called an *intergenic* mutation, whereas a silent mutation localized within a coding sequence is called a *synonymous* mutation.
- A *missense* or non-synonymous mutation is a translation change on the query sequence which leads to an amino acid change within the coding sequence located at this position.
- An *indel* mutation, which is either a deletion or insertion of a subsequence (or base).

To start a mutation search, choose **Mutations > Search...** (🔍). The *Find mutations* dialog box pops up (see Figure 8.7.7).

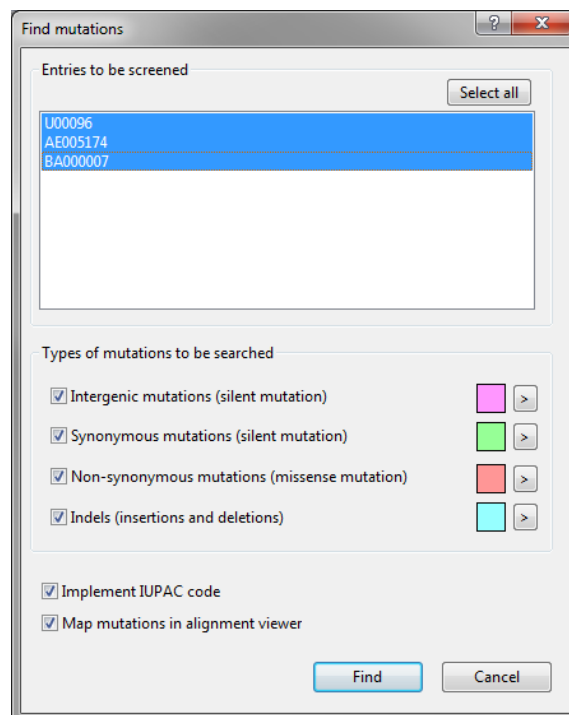


Figure 8.7.7: The *Find mutations* dialog box.

By default, all entries are selected within the *Entries to be screened* field. To restrict the screening, only select those entries you want to analyze. To select all entries again, press the <Select all> button.

The *Types of mutations to be searched* options list the four different mutation types described earlier. With the check boxes, combinations of mutation types to be searched for can be made. The color fields adjacent to each mutation type specify the color that will be used to label the base in the mutation view panel and in the detailed alignment view. The color fields can be edited by pressing the > button at the right side of the color fields.

The IUPAC code for ambiguous bases is supported in the mutation search function if **Implement IUPAC code** is checked. BioNumerics will then consider the IUPAC nomenclature and score mutations in a "conservative" way. For example, for a position denoted as A in the consensus, any occurrences of R (A or G), M (C or A) or W (T, U or A) will not be scored as a mutation.

With the option **Map mutations in alignment viewer** checked, the positions of the mutations in the Detailed alignment panel are color-labeled according to the mutation type. Alternatively, after calculation the mutations can also be color-labeled by pressing the <Map> button from the *Mutation analysis* panel.

Pressing the <Find> button starts the mutation search.

Count	Entry	Label	Position	Type	NA change	AA change	Quality
89321	AE005174	RNase II, ds RNA	2701795	silent	c → t		0.0428
89322	BA000007	RNase II	2701795	silent	c → t		0.0428
89323	AE005174	RNase II, ds RNA	2701984	silent	a → g		0.0529
89324	BA000007	RNase II	2701984	silent	a → g		0.0529
89325	AE005174	hypothetical protein	2702147	silent	t → c		0.0716
89326	BA000007		2702147	intergenic	t → c		0.0716
89327	AE005174	hypothetical protein	2702180	silent	c → g		0.0786
89328	BA000007		2702180	intergenic	c → g		0.0786
89329	AE005174		2702341	indel			1.0000
89330	BA000007		2702341	indel			1.0000
89331	AE005174	leader peptidase (sig...	2702389	missense	g → c	L → V	0.6667
89332	BA000007	signal peptidase I	2702389	missense	g → c	L → V	0.6667
89333	AE005174	leader peptidase (sig...	2702393	silent	a → t		0.6325
89334	BA000007	signal peptidase I	2702393	silent	a → t		0.6325
89335	AE005174	leader peptidase (sig...	2702465	silent	c → a		0.0945
89336	BA000007	signal peptidase I	2702465	silent	c → a		0.0945


Figure 8.7.8: The *Mutation search* panel after performing a mutation search.

The results for the mutation search function are listed in the *Mutation search* panel (see Figure 8.7.8).

At the left side of the panel, the mutation analysis listing contains the following information in the different columns:

- "Entry" indicates the entry key of the sequence on which the mutation is located.
- In the column "Label" the preferential qualifier used for feature labeling, as defined in the alignment display settings is displayed.
- "Position" gives the alignment position (template numbering) of the mutation.
- "Type" states the type of mutation: intergenic (silent) mutation, synonymous (silent) mutation, non-synonymous (missense) mutation or indels (insertions and deletions).
- "NA change" gives the nucleotide base change: the first letter is the nucleotide base on the template (reference) sequence, the second is the base on the query sequence.
- "AA change" gives the change at translation level as results from the nucleotide change. The first letter is the amino acid located within the translation product on the template sequence, the second letter is the amino acid on the query sequence.
- The quality score displayed in the field "Quality" is a measure of confidence. See Figure 8.7.9 for explanation of this quality factor.

The menu function *Mutations* > *Jump to previous* () jumps to the next mutation upstream from the current cursor position.

The menu function *Mutations* > *Jump to next* () will find the next mutation downstream from the current cursor position.



The jump function is directed by the mutation search settings: if for example only non-synonymous mutations have been searched for, jumping will occur between non-synonymous mutations whereas other type mutations (silent, indels) are not considered.

Within the *Mutation search* panel, the mutations can be sorted by each of the columns.

Right-click in one of the column headings and select **Sort** from the floating menu to sort the mutations according to the selected column.

The *Mutation analysis* panel is located next to the *Mutation search* panel and depicts a tabular view of alignment positions that resolve to mutations in one of the query sequences. The rows within the matrix

correspond to the query sequences, whereas the columns are the mutation-affected alignment positions. Only the mutation types that were selected to be calculated are displayed.

By default, the tabular mutation view is horizontally displayed. With the option **Mutations > Switch plot direction** (↕) the table can be inverted to vertical orientation. Clicking the same button again will show the mutation table horizontally again.

The mutation analysis permits to score the mutations. Mutation scoring settings (see Figure 8.7.9) are called with **Mutations > Call mutation score settings...** (⚙️).

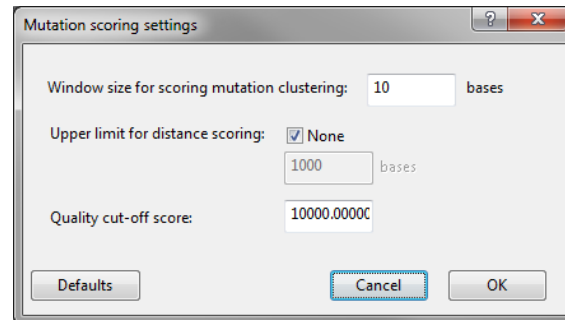


Figure 8.7.9: The *Mutation scoring settings* dialog box.

The **Window size for scoring mutation clustering** indicates the size of the window in which neighboring mutations around the scope mutation will be counted (count M) (the scope mutation is included to have $M \geq 1$). One single mutation located within a highly conserved region might be weighted as more important than several adjacent mutations in a more variable region.

The distance scoring is the smallest distance of the mutation position towards the stretch ends on which the mutation is located (distance D). The quality score of a mutation (Q) is then calculated as: $Q = \frac{M^2}{\sqrt{D}}$. Very small quality scores indicate a single mutation in a long stretch of very high sequence conservation ($M = 1$, $D > 1000$).

The **Quality cut-off score** is the maximal score a mutation may have to be displayed in the *Mutation analysis* panel.

The outcome of the mutation quality scoring is displayed in the "Quality" column of the *Mutation search* panel and can also be mapped in the tabular mutation view.

With the command **Mutations > Show/hide mutation quality** (🎨), the color code of the cells from the tabular mutation view can be changed to map the mutation quality. The color scale indicates the range of the mapped mutation quality.

Next to the mutation quality, the tabular mutation view also displays the number of genomes where a mutation is found on a specific mutation position on the template sequence.

Based on the calculated mutations, a cluster analysis (see Figure 8.7.10 for an example) can be calculated with **Mutations > Show/hide mutation clustering** (📊). This clustering is done according to a standard UPGMA algorithm.

Selecting a mutation within the *Mutation search* panel causes the detailed alignment view to jump to the alignment position where the mutation is located. The detailed alignment view can also be shown in text modus in order to have a classical alignment view of the sequences. To call this text modus, select **Alignment > Show text view** (📄).

The mutation list can be exported to a file with the menu option **File > Export > Mutation list**. A file will be created and can be loaded in e.g. Notepad or another text editor. The file name and the type of file can be specified in this dialog box: in the "Tab-delimited file format", the columns are separated by tabs, whereas in the "Column-based file format", column text is lined using spaces. Use the tab-delimited format to export



Figure 8.7.10: Detail on the calculated cluster analysis from the mutation results.

to programs such as Microsoft Excel or Access.

The sequential formatting of the mutation data is the same as in the *Mutation search* panel, namely mutation position (referred to the alignment), mutation type, entry key of the sequence on which the mutation is located, nucleotide change (referred to the template sequence), amino acid change (referred to the template sequence), gap position if indel mutation and the preferential qualifier defined as label.

If more than one mutation search was performed, a previous listing can be displayed again by selecting the mutation search from the drop-down list in the toolbar of the *Mutation search* panel.

To delete a mutation search, select it from the drop-down list and choose **Mutations > Delete current list** (✖).

8.7.5 Sequence search panel

Searching for a specific subsequence within the alignment is done with **Edit > Find subsequence...** (🔍). The *Find sequence* dialog box will appear.

By default, all entries present in the alignment project are highlighted in the **Entries to be searched** list. Any other selection of entries can be made from this list. To select all entries at once, press the **<Select all>** button.

Under **Sequence**, you can enter the subsequence to search for.

The **Search settings** are applicable to the current subsequence search:

- **Mismatches allowed:** the maximum number of mismatches one allows for a subsequence still to be considered as matching with a target sequence.
- **Allow gaps:** Whether or not you allow the subsequence to be interrupted by gaps.
- **Consider IUPAC codes:** Allows the search sequence to be matched with uncertain positions denoted as IUPAC unresolved positions (e.g. R, Y, etc., including N) and allows IUPAC code to be used in the search sequence. When unchecked, only A, T, C or G will be matched against the target sequence(s).
- **Reverse search:** Whether or not the invert-complemented sequence will be searched as well.

The subsequence search function matches the subsequence against the original full target sequences, not against the aligned sequences. This has the advantage that (1) all matches will be found, although some of

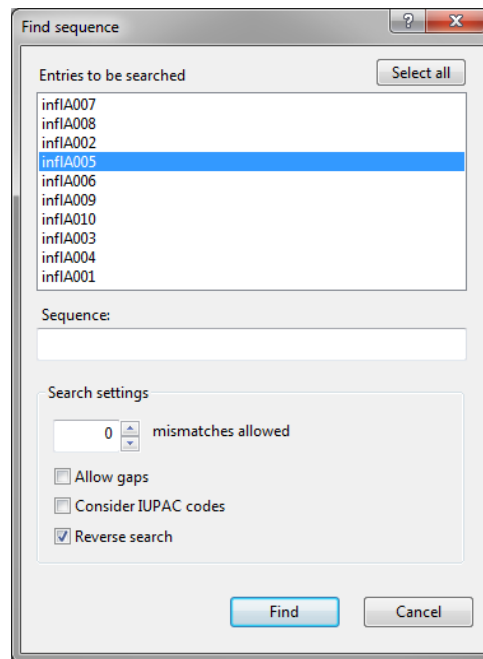


Figure 8.7.11: The *Find sequence* dialog box, to find a sequence in the alignment.

them may be absent (in this case, -1-1 will be displayed as alignment positions) or only partially present in the aligned sequences; (2) the position, match and orientation on the unaligned original entry is known; (3) presence and position on the aligned sequences can be traced back.

The results for the sequence search are listed in the *Sequence search* panel and will look like in Figure 8.7.12.

Sequence search results							
[1] gatgaatgatg							
Count	Entry	Position entry	Position alignment	Direction	Match	Mismatch	
1	U00096	608755	608755-608766	+	GAT GAAT GAT G GAT GAAT GAT G	0	
2	U00096	658294	658294-658305	+	GAT GAAT GAT G GAT GAAT GAT G	0	
3	U00096	1491230	1491230-1491241	+	GAT GAAT GAT G GAT GAAT GAT G	0	
4	U00096	1884378	1884378-1884389	+	GAT GAAT GAT G GAT GAAT GAT G	0	
5	U00096	3730474	3730474-3730485	+	GAT GAAT GAT G GAT GAAT GAT G	0	
6	AE005174	690970	608755-608766	+	GAT GAAT GAT G GAT GAAT GAT G	0	
7	AE005174	740208	658294-658305	+	GAT GAAT GAT G GAT GAAT GAT G	0	
8	AE005174	5189534	-1--1	+	GAT GAAT GAT G GAT GAAT GAT G	0	
9	AE005174	387764	335489-335500	+	GAT GAAT GAT G GAT GAAT GAT G	0	
10	BA000007	691188	608755-608766	+	GAT GAAT GAT G GAT GAAT GAT G	0	
11	BA000007	740427	658294-658305	+	GAT GAAT GAT G GAT GAAT GAT G	0	
12	BA000007	5046634	-1--1	+	GAT GAAT GAT G GAT GAAT GAT G	0	

Figure 8.7.12: The *Sequence search* panel.


A sequence match is described with the following characteristics:

- The column "Entry" indicates on which entry the match has been found.
- The original match position on the entry is given in the column "Position entry".
- The position on the alignment is given in the column "Position alignment". The position corresponds to the numbering on the template sequence, not with the numbering on the target sequence (get a full zoom to read the corresponding numbering on the target sequence).


- In the column "Direction", the direction of the match is indicated with an arrow: the ➡ sign indicates that subsequence and target sequence are kept in the same orientation for the produced match, whereas the ⇐ sign means that the subsequence had to be inverted in order to match the target sequence.
- The column "Match" displays the query sequence (on top) and the matching subsequence (bottom).
- In the column "Mismatch", the number of mismatches needed to produce the match is given. It is evident that *mismatches allowed* and/or *Allow gaps* in the search settings should be checked in order to produce matches containing mismatches.

By clicking on a sequence match, the matching sequence is selected in the detailed alignment panel.

If more than one sequence search was performed, a previous listing can be displayed again by selecting the sequence search from the drop-down list in the toolbar of the *Sequence search* panel.

To delete a sequence search, select it from the drop-down list and press the  button.

8.7.6 Feature search panel

Searching the alignment for a specific feature can be done with *Edit > Find feature...* (). The *Find features* dialog box will appear as shown in Figure 8.7.13.

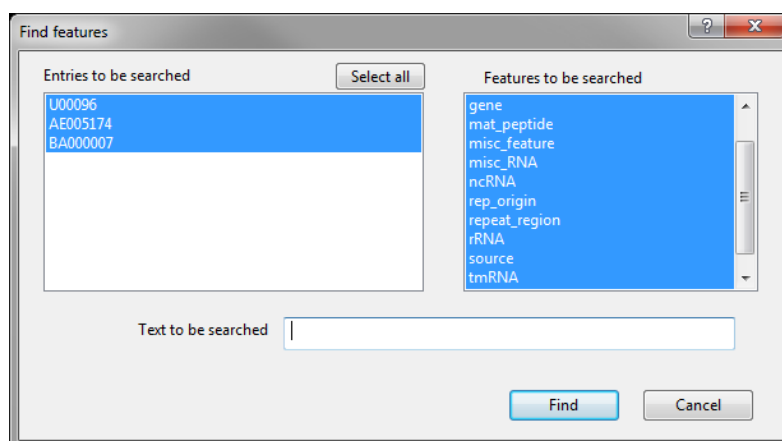


Figure 8.7.13: The *Find features* dialog box.

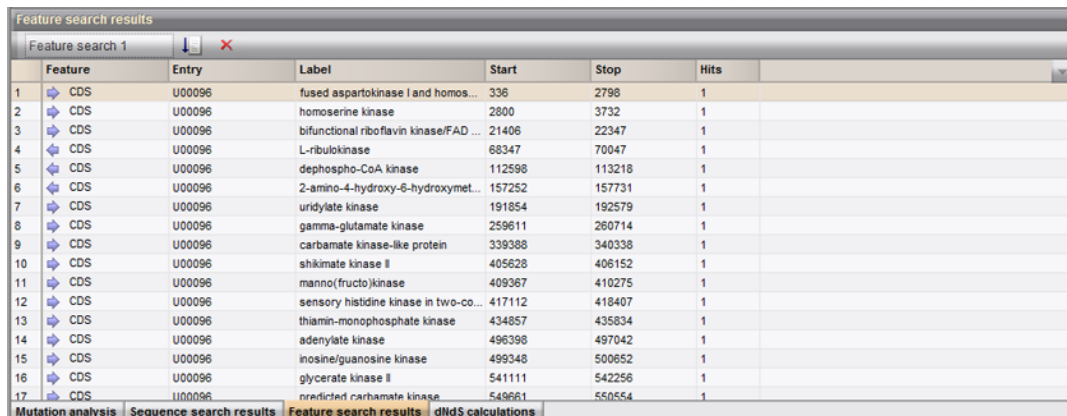
By default, all entries are selected in the listing *Entries to be searched*. If only one sequence or a subset of sequences has to be searched through, first select the subset. For a multiple selection, hold down the **Ctrl** key while clicking with the mouse. To select all entries, press the **<Select all>** button.

The feature types have to be selected within the listing *Features to be searched*. By default, all types are selected. A subset of feature types can be made by clicking on the feature types of interest and holding down the control key (for multiple selection). If no specific text has been entered in the field *Text to be searched*, all features of the target sequences that fulfill the feature type criteria as defined under *Features to be searched* will be fetched. On the other hand, if text is entered in the field *Text to be searched*, all target sequences are screened for features fulfilling both the feature type criteria and text-limiting criteria. Press the **<Find>** button to start the feature search. The feature search can be aborted by pressing the **<Cancel>** button.

Similar to the sequence search, the feature search function acts on the original full length sequences in order not to lose the information whether a searched item is not at all present on the sequences included in the project, or the item simply does not appear within the calculated alignment.

Pressing the <Find> button starts the calculations.

The results for the feature search function are listed in the *Feature search* panel (see Figure 8.7.14).



	Feature	Entry	Label	Start	Stop	Hits	
1	CDS	U00096	fused aspartokinase I and homos...	336	2798	1	
2	CDS	U00096	homoserine kinase	2800	3732	1	
3	CDS	U00096	bifunctional riboflavin kinase/FAD ...	21406	22347	1	
4	CDS	U00096	L-ribulokinase	68347	70047	1	
5	CDS	U00096	dephospho-CoA kinase	112598	113218	1	
6	CDS	U00096	2-amino-4-hydroxy-6-hydroxymet...	157252	157731	1	
7	CDS	U00096	uridylate kinase	191854	192579	1	
8	CDS	U00096	gamma-glutamate kinase	259611	260714	1	
9	CDS	U00096	carbamate kinase-like protein	339388	340338	1	
10	CDS	U00096	shikimate kinase II	405628	406152	1	
11	CDS	U00096	manno(fructo)kinase	409367	410275	1	
12	CDS	U00096	sensory histidine kinase in two-co...	417112	418407	1	
13	CDS	U00096	thiamin-monophosphate kinase	434857	435834	1	
14	CDS	U00096	adenylate kinase	496398	497042	1	
15	CDS	U00096	inosine/guanosine kinase	499348	500852	1	
16	CDS	U00096	glycerate kinase II	541111	542256	1	
17	CDS	U00096	predicted carbamate kinase	549661	550554	1	

Figure 8.7.14: The *Feature search* panel.

A feature hit is described with the following characteristics:

- Automatically, all the feature search results are given a continuous numbering in the "Count" column.
- The feature hit, defined within the features to be searched, is given in the column "Feature".
- The entry on which the feature was found is given in the column "Entry".
- The column "Label" displays the feature label as defined from the alignment display settings.
- The original feature position on the entry is given in the columns "Start" and "Stop".
- The number in the column "Hits" indicates over how many discontinuous aligned blocks (= separated by gaps) the feature item is spread. A hit number of 2 or more thus indicates that either the feature is repeated in the alignment if "allow repetition of aligned blocks" is checked in the alignment, or that the feature is aligned into several discontinuous blocks.

8.7.7 dNdS search panel

The alignment of homologous genes into gene clusters allows the detection of nucleotide substitutions at every position within these genes. The proportion of synonymous (silent) versus non-synonymous (amino acid change) nucleotide changes within such gene clusters gives outcome on evolutionary selection present on these DNA sequences. Some genes, or even gene clusters, may display a higher rate of non-synonymous mutations due to positive genetic selection.

As a nucleotide substitution will be a synonymous or non-synonymous mutation based on the translation encoded by the sequence spanning this nucleotide substitution, synonymous and non-synonymous mutations have to be looked up in sequence stretches encoding functional proteins (coding regions defined by CDS features). Therefore, the first step in calculating dNdS ratios will be the location of the gene clusters within the alignment. All CDS-encoding sequence regions located on the template sequence are extracted and mapped as gene cluster regions on the alignment. If a CDS region located on a query sequence partially or fully overlaps this gene cluster region and the query gene direction and translation is in frame with the template gene, it is added to the gene cluster. Based on the template gene cluster sequence, query gene cluster sequences are subsequently screened for observed and potential synonymous and non-synonymous nucleotide changes [31].

To launch a dNdS search, choose *dNdS* > **Calculate...** (📄). This calls the *Alignment Settings* dialog box (see Figure 8.7.15).

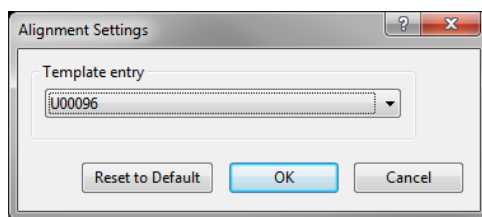


Figure 8.7.15: The *Alignment Settings* dialog box: select a template entry from the list.

The template entry for the dNdS calculations needs to be selected from the drop-down list. Pressing the **<OK>** button starts the calculation. The different gene clusters are searched through and subsequently the dNdS results are calculated. The progress of this calculation is shown in the status bar at the bottom of the *Chromosome Comparison* window.

The results of the calculation are given in the *dNdS search* panel (see Figure 8.7.16).


Count	Cluster	Entry	Start	Stop	Length	Label	Sd	Sn	S	H	ps	pn	ds	dn	ds/dn	ps/pn
1	1/4	AE005174	336	2796	2460		47.000	2.000	594.167	1885.8...	0.079	0.001	0.084	0.001	77.929	73.796
2	1/4	BA000007	336	2796	2460		47.000	2.000	594.167	1885.8...	0.079	0.001	0.084	0.001	77.929	73.796
3	2/6	AE005174	2800	3730	930		16.000	2.000	221.500	708.500	0.072	0.003	0.076	0.003	26.856	25.589
4	2/6	BA000007	2800	3730	930		16.000	2.000	221.500	708.500	0.072	0.003	0.076	0.003	26.856	25.589
5	3/8	AE005174	3733	5017	1284		18.000	1.000	302.000	982.000	0.060	0.001	0.062	0.001	60.945	58.530
6	3/8	BA000007	3733	5017	1284		18.000	1.000	302.000	982.000	0.060	0.001	0.062	0.001	60.945	58.530
7	4/10	AE005174	5233	5527	294		5.000	3.000	65.167	228.833	0.077	0.013	0.081	0.013	6.120	5.853
8	4/10	BA000007	5233	5527	294		5.000	3.000	65.167	228.833	0.077	0.013	0.081	0.013	6.120	5.853
9	5/13	AE005174	5682	6456	774		6.000	2.000	172.167	601.833	0.035	0.003	0.036	0.003	10.715	10.487
10	5/13	BA000007	5682	6456	774		6.000	2.000	172.167	601.833	0.035	0.003	0.036	0.003	10.715	10.487
11	6/15	AE005174	6528	7956	1428		20.000	4.000	345.000	1083.0...	0.058	0.004	0.060	0.004	16.295	15.696
12	6/15	BA000007	6528	7956	1428		20.000	4.000	345.000	1083.0...	0.058	0.004	0.060	0.004	16.295	15.696
13	7/17	AE005174	8237	9188	951		6.000	0.000	225.667	725.333	0.027	0.000	0.027	0.000	0.000	0.000
14	7/17	BA000007	8237	9188	951		6.000	0.000	225.667	725.333	0.027	0.000	0.027	0.000	0.000	0.000
15	8/20	AE005174	9305	9890	585		4.000	0.000	144.000	441.000	0.028	0.000	0.028	0.000	0.000	0.000
16	8/20	BA000007	9305	9890	585		4.000	0.000	144.000	441.000	0.028	0.000	0.028	0.000	0.000	0.000

Figure 8.7.16: The *dNdS search* panel.

The dNdS calculation results include the following columns:


- Within the column "Cluster", the first number indicates the gene cluster index, whereas the second number is the feature index of the feature present within the query sequence entry.
- The column "Entry" displays the entry key of the query sequence.
- The column "Start" indicates the gene cluster start position (alignment-based numbering).
- The column "Stop" indicates the gene cluster end position (alignment-based numbering).
- The column "Length" gives the length of the gene cluster region.
- The column "Label" shows information about the CDS features within the gene cluster (preferential qualifier labeling as defined within the alignment display settings; see Figure 8.7.5).
- The column "Sd" gives the number of observed synonymous substitutions within the gene cluster sequence.
- The column "Sn" gives the number of observed non-synonymous substitutions within the gene cluster sequence.

- The column "S" gives the number of potential synonymous sites within the gene cluster sequence.
- The column "N" gives the number of potential non-synonymous sites within the gene cluster sequence.
- The column "ps" gives the proportion of synonymous differences, calculated as S_d/S .
- The column "pn" gives the proportion of non-synonymous differences, calculated as S_n/N .
- The column "ds" gives the Jukes-Cantor correction for multiple hits of ps.
- The column "dn" gives the Jukes-Cantor correction for multiple hits of pn.
- The column "ds/dn" gives the ratio of synonymous to non-synonymous substitutions.
- The column "ps/pn" gives the ratio of the proportion of synonymous to the proportion of non-synonymous substitutions.

In the detailed alignment view, one can map the dNdS results on the sequences by clicking . A white/gray line in the center of the sequence indicates that this part of the sequences was included in the dNdS search function.

The dNdS search results can be exported to a file with the menu option **File > Export > dNdS list....** A file will be created and can be loaded in e.g. Notepad or another text editor. The file name and the file type can be specified in the next dialog: in the "Tab-delimited file format", the columns are separated by tabs, whereas in the "Column-based file format", column text is lined using spaces. Use the tab-delimited format to export to programs such as Microsoft Excel or Access.

If more than one dNdS search was performed, previous listings can be displayed by selecting the desired dNdS search from the drop-down list in the toolbar of the *dNdS search* panel.

To delete a dNdS search, select the dNdS search from the drop-down list and choose **dNdS > Delete current list** (.

Chapter 8.8

Genome annotation

8.8.1 Introduction

The annotation application in BioNumerics has been designed for annotating coding regions on sequences. Special mathematical algorithms have been developed in BioNumerics to perform annotation of chromosomes within minutes. The format, in which the annotation project is stored, is structured in such a way that the annotation can be edited at each stage. The annotation is stored unrelated to sequence position, so that inserting or deleting bases within the sequence does not lead to the loss of information. Partially annotated sequences or sequences with no annotation can be submitted to annotation, and existing annotations can be overwritten or not, as specified by the user. Annotation is performed through comparison of the *query* sequence, with existing annotated sequences, the *template*.

To be able to use the annotation application in BioNumerics, the Sequence data module (SQ) and the Genome Analysis Tools module (GA) need to be present in your BioNumerics configuration.

8.8.2 Creating a new annotation project

In the *Main* window, the *Annotations* panel is displayed in default configuration as tabbed view with the *Comparisons* panel, *Decision networks* panel, *Alignments* panel, *Chromosome comparisons* panel, *Annotations* panel and *BLAST projects* panel in the bottom right part of the window. If desired, the configuration of the *Main* window can be customized as described in 2.3.4.

To create a new annotation project, select the *Annotations tab* in the *Main* window and select **Edit > Create new object...** (➕). A name for the new annotation project is prompted for.

The new annotation project is added to the *Annotations* panel in the *Main* window. The date on which the annotation project was created and last modified is displayed in the default information fields 'Creation date' and 'Modified date' respectively. When more than one annotation project is present, projects can be sorted and searched using the information present in the default or user-defined information fields. For a detailed explanation of the display options of the *Annotations panel* and other grid panels, see 3.2.7.

To delete one or more annotation projects from the list, select the project and select **Edit > Delete selected objects...** (✖).

Choose **Edit > Open highlighted object...** (🔍, Enter) to open a selected annotation project. The first time an annotation project is opened, it will open with the currently selected entries in the *Database entries* panel. As soon as an annotation project has been saved, selecting **Edit > Open highlighted object...** (🔍, Enter) will open the annotation project with the entries that were present when the annotation project was last saved.

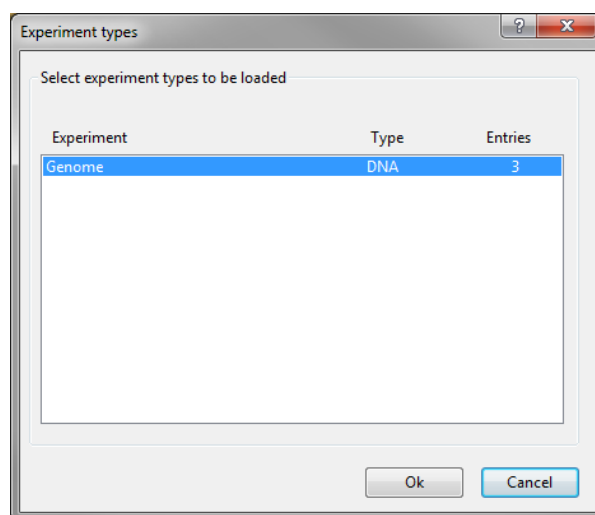


Figure 8.8.1: Select experiment(s) to be included in the annotation project.

The *Experiment types* dialog box displays a list of available sequence types and the number of associated entries. From this list, the user can select the experiment type(s) that should be included in the annotation project. Pressing **<OK>** opens the annotation project in the *Annotation* window.

8.8.3 The Annotation window

The *Annotation* window consists of three panels: the *Project sequences* panel, the *Sequence display* panel, and the *Pairwise comparison display* panel. All panels are dockable, which enables the user to customize the layout of the *Annotation* window according to personal preferences. See 2.3.4 for detailed information on the display options of dockable panels.

- *Project sequences* panel: For each selected entry/sequence type combination, a row is created in the *Project sequences* panel. This grid panel displays the entry information fields in tabular format, similar to e.g. the *Database entries* panel (for display options of grid panels, see 3.2.7).
- *Sequence display* panel: This panel displays a frame analysis overview of the query sequence (empty if no query sequence has been selected). Query coding regions and template hits that passed the annotation project criteria will be shown in this panel when the annotation project has been calculated.
- *Pairwise comparison display* panel: This panel is reserved for depicting the annotated query sequence. Parallelisms with the template sequences will be shown here (empty if the annotation project has not been calculated).

The upper part of the *Annotation* window contains the main menu and toolbars. The latter can be displayed or hidden according to your preferences. See 2.3.5 for display options of toolbars.

You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally.

8.8.4 Specifying a query sequence

Before the calculation of an annotation project can be started, one of the sequences in the *Project sequences* panel needs to be defined as the query sequence. Select this sequence from the list in the *Project sequences*

panel and choose **File > Set query sequence** (). The appointed sequence is considered as the query sequence in the annotation project and is preceded with a blue dot in the *Project sequences panel*.

The key and sequence type of the query sequence are displayed in the status bar in the bottom of the *Annotation* window. All other sequences in the *Project sequences panel* will serve as templates for the annotation of the query sequence in the annotation project.

The *Sequence display panel* shows the query sequence together with a frame analysis overview in the upper part of the panel. The **Standard Code** translation table is default used to analyze the query sequence in function of its six translation frames, but this can be changed in the *Frame analysis settings dialog box* (see 8.8.6). The three reading frames of the forward strand are mapped above the forward query sequence, the three reading frames of the reverse strand are mapped below the reverse query sequence. Open reading frames that fulfill the open reading frame settings (see 8.8.6) are plotted in gray on the query sequence, and in blue on the six reading frames. Features which have been imported, i.e. coding regions which were already mapped on the query sequence, are displayed in purple on the query sequence, and in blue on the six reading frames.

Within the lower part of the *Sequence display panel*, fields are depicted, corresponding to the coding regions that are mapped on the query sequence. The upper fields are related to the coding regions found on the forward orientation of the query sequence, the lower fields to the coding regions detected on the reverse orientation. The header of the fields is displayed in pink, indicating that the annotation project has not been calculated yet.

The zoom slider, located on top of the *Project sequences panel* in default configuration, allows zooming from full-length sequence view up to base level view. The red vertical line indicates the cursor position on the query sequence.

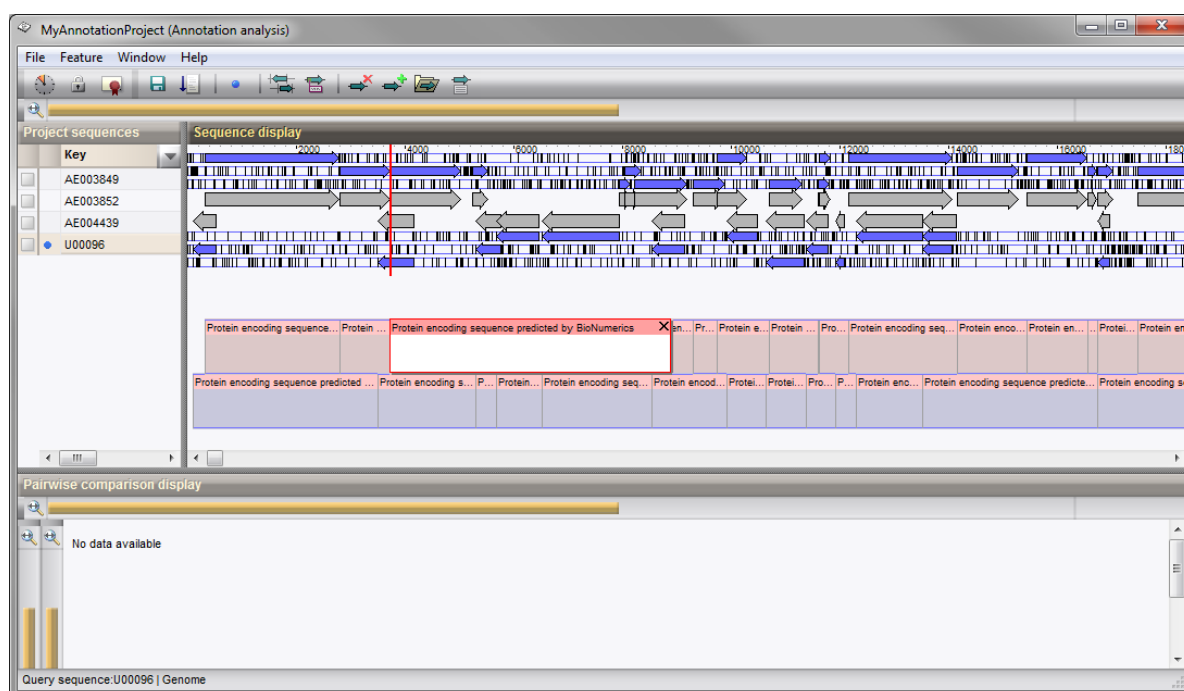


Figure 8.8.2: The *Annotation* window after specifying the query sequence.

8.8.5 Annotation steps

The sequence of calculation steps within an annotation run is the following:

1. In a first step, the six frames of the query sequence are screened for all possible coding regions (see 8.8.6).
2. Secondly, the possible query coding regions are screened against the coding regions that are mapped on the template sequences. Template coding regions showing any homology with query coding regions are retained and linked to the corresponding query coding region in a ranking based on *feature identity* and/or *chromosome synteny* scores (see 8.8.7).
3. Finally, query coding regions, for which template coding regions have been found showing a homology level which is above a threshold level set by the user, are annotated based on the descriptions available from those template coding regions. Which descriptions should be used can be defined by the user (see 8.8.8).

The settings comprised in these three steps are grouped in the *Project settings* dialog box. This dialog box can be called with **File > Run calculation...** (📄) (see Figure 8.8.3).

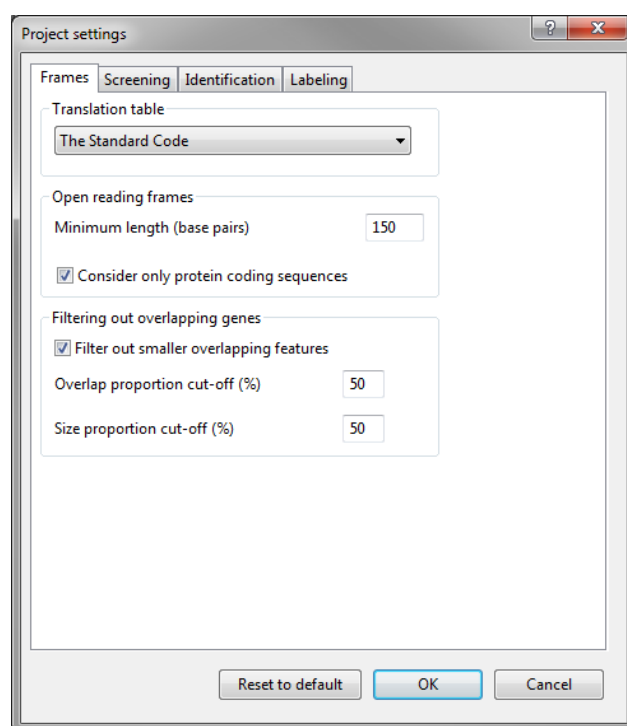


Figure 8.8.3: The project settings.

8.8.6 Frame analysis settings

In the first step of an annotation run, the six frames of the query sequence are screened for all possible coding regions. The settings that are used for this screening are grouped in the *Frames* tab of the *Project settings* dialog box (see Figure 8.8.3). Alternatively, these settings can be called with **File > Frame analysis settings...** (📄). This brings up the *Frame analysis settings* dialog box (see Figure 8.8.4).

Within the drop-down box **Translation table**, the translation table can be selected which should be implemented for looking up the coding regions (influences mostly the initiations of translation). By default, **The Standard Code** translation table is selected.

As one will mostly be interested in open reading frames with a considerable length, a size filtering needs to be specified in the *Open reading frames* panel. This **Size filtering** setting (expressed in base pairs) defines

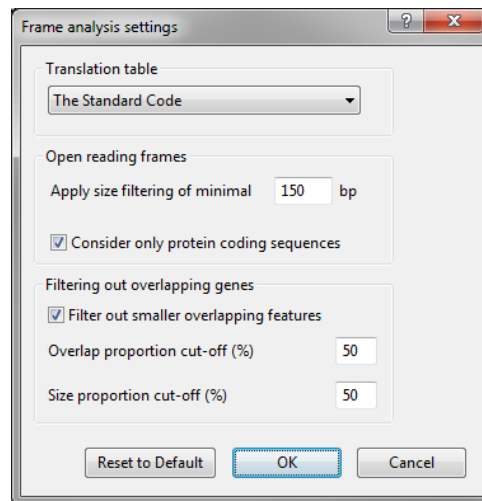


Figure 8.8.4: The frame analysis settings.

the minimal size which the open reading frame regions should have. If the option ***Consider only protein coding sequences*** is checked, the presence of an initiation codon is considered as a necessity for selecting open reading frames. As a consequence, the size filtering setting takes effect on the sizes of the *protein coding sequences (PCS)*, i.e. from initiation codon towards stop codon. If the option ***Consider only protein coding sequences*** is unchecked, the size filtering takes effect on the sizes of the *open reading frame regions (ORF)*, i.e. between two stop codons.

The option ***Filter out smaller overlapping features*** allows one to select the most favorable coding regions out of overlapping coding regions on opposite strands. A minimal ***Overlap proportion cut-off (in %)*** and ***Size proportion cut-off (in %)*** of the overlapping features should be specified when enabling this setting. With the ***Size proportion cut-off*** filter set lower than for example 50%, one can prevent that one of the two overlapping features having about the same size is removed (especially important with phages where overlapping genes are more abundant). With the ***Overlap proportion cut-off*** filter set above 50% for example, one prevents features having only a small overlap from being rejected. This filter selects in most cases the right coding regions, however small coding regions of about the same size overlapping each other may cause the filter to select the wrong regions. To avoid this, leave the filtering out (a more accurate selection based on homology screening can be made within the last annotation step, see 8.8.8).

Pressing the **<Reset to Default>** button resets the default settings in the *Project settings* dialog box.

Pressing **<OK>** in the *Project settings* dialog box saves the new settings for the annotation project to the database. If open reading frames were already plotted on the query sequence using different frame settings, a window pops up, asking the user to confirm the update action. Press **<OK>** to apply the new settings to the annotation project or press **<Cancel>** to cancel the update action.

8.8.7 Homology screening settings

A second set of options directing the annotation are those for the homology screening of the query coding regions against the template coding regions. The homology screening settings are grouped in the *Screening tab* of the *Project settings* dialog box (see Figure 8.8.3 and Figure 8.8.5). The homology screening algorithm implemented in the *Annotation* window is analogous to the algorithm implemented in the *Chromosome Comparison* window. More information about this algorithm and settings can be found in 8.6.1 and 8.6.3 respectively.

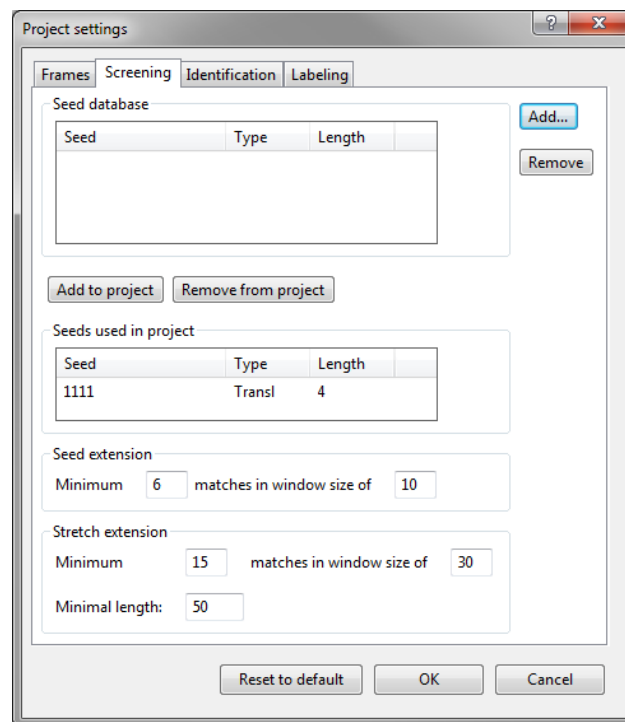


Figure 8.8.5: The *Project settings* dialog box: the *Screening* tab.

8.8.8 Feature annotation settings

The third set of options concerns the evaluation and annotation of the query coding regions selected in the previous round of screening. These settings are grouped in the *Identification* tab and *Labeling* tab of the *Project settings* dialog box. Alternatively, these settings can be called with the menu item **File > Feature annotation settings...** (🔍). This brings up the *Feature annotation settings* dialog box (see Figure 8.8.6).

- The settings grouped in the *Identification* tab define which coding regions will be accepted and mapped on the query sequence.
- The settings in the *Labeling* tab determine which labels will be introduced to the coding regions that are accepted and mapped on the query sequence.

The most obvious way to validate the identification of a query feature is to consider the **Feature identity** scores, derived from the pairwise alignments of the query feature with the template features. However, if a degree of chromosome parallelism can be found between query sequence and template sequence in a defined distance around the hit position on the respective entries, then additional information for the correctness of the identification is obtained. Therefore, **Chromosome syntenity** has been included as scoring factor for the annotation of the query features. The way the **Chromosome syntenity** scoring factor is generated, is by looking up feature hits appearing up- and downstream of the query feature **Q**. If these feature hits are also mapping in the vicinity of the template feature **T** (the feature with which the query feature is to be identified), then a **Chromosome syntenity** scoring factor is calculated in function of the number of such hits found.

The value entered under the field **Feature distance** in the *Syntenity determination panel* defines how far, in terms of feature units, the program will look up parallel hits, appearing at either side of the query feature **Q**. Note that if a feature distance equals n , then $2 \times n$ features are screened. The value **Sequence distance** defines the maximal distance, in base pairs, for which an adjacent hit may map away from the template feature **T** (on the same template of course) in order to be scored as "parallel hit". Cross mappings with distance values greater than the value set in **Sequence distance** are not considered to contribute to a "chromosome

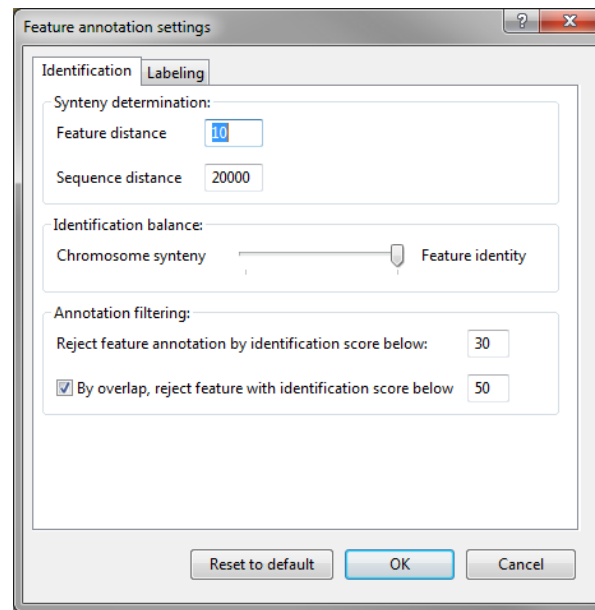


Figure 8.8.6: The feature annotation settings: The *Identification* tab.

parallelism” around the query feature **Q** and template feature **T**. The **Chromosome synteny** score is then calculated as the percentage of “parallel hits” found within two times the number of features defined by the **Feature distance** value (up- and downstream together).

The slider in the *Identification balance* panel allows the user to balance the impact of the **Feature identity** score and the **Chromosome synteny** score on the annotation. Moving the slider to the full right gives 100% weight to the feature identity score, excluding any effect of chromosome synteny scoring on the annotation. On the other hand, placing the slider to the full left will cause the annotation to be based only on chromosome synteny scoring: identity scores of the individual hits will have no contribution to the annotation. Slider positions in between are giving proportional weights to the identity and chromosome synteny scores. Note that, when using chromosome synteny scoring as additional identification weight, annotation can be based on a template feature giving the second best identity score or lower, as this decision may be supported by a better chromosome synteny.

The value indicated next to field **Reject feature annotation by identification score below** in the *Annotation filtering* panel is the percentage value which will reject those hits, for which the calculated score is not above this minimum value. These rejected hits will not be considered anymore within the annotation process. The last option within the lower panel concerns a cut-off value to filter out falsely identified query coding sequences, which show partial or full **overlap** with true query coding sequences. This cut-off value, which only takes effect on overlapping genes and clearly selects biological meaningful homologues (for example 50% or more), efficiently rejects falsely identified overlapping coding sequences, as in such cases only one of the overlapping genes will display clear identification with template features. On the other hand, in case of true overlapping genes (for example in many phages), both genes should be recognized by existing template coding sequences. In this case, both genes will pass the overlap cut-off value and will be mapped. Note that this filtering for false overlapping genes is much more efficient than the **Overlap proportion** filter applied in the open reading search routine (see Figure 8.8.4). It only takes longer calculation times.

The first set of settings, grouped in the *CDS labeling criteria* panel, define which description fields the query feature will adapt from the template feature. The feature description fields are called *qualifiers* in the EMBL-GENBANK format and are categorized through *qualifier keys* (e.g. “/product=”, “/gene=”, ...). Which description fields should be adapted from the template feature can simply be appointed by listing the qualifier keys which are of interest. The list box shows the current selection of qualifier keys for which the description fields should be adapted for annotation. More qualifier keys can be added to the list by

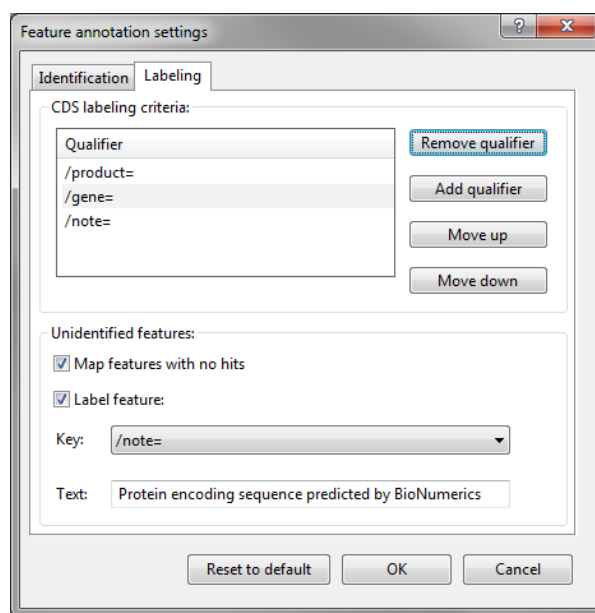


Figure 8.8.7: The feature annotation settings: The *Labeling* tab.

pressing the button **<Add qualifier>**. Qualifier keys which are not of interest can be removed from the list by selecting the qualifier key within the list and pressing the **<Remove qualifier>** button. The order of the qualifier keys within the list represents the preference for labeling the features: the description field from the first qualifier key within the list will be adapted as label for the feature on the plots. If this description field does not occur, then the field from the second qualifier key within the list will be taken, and so on. The preferential positions of the qualifier keys can be moved within the list by using the buttons **<Move up>** and **<Move down>**.

The settings grouped in the *Unidentified features panel*, concern the treatment of probable protein encoding regions for which no identification has been found (i.e. open reading frame regions showing one or more hits, but not contributing to an annotation of the feature as they did not pass the identification score cut-off value). The appearance of such cases is of course depending on the identification score and the homology screening cut-off values. The field **Map features with no hits** allows the user to decide whether unidentified features should be mapped as CDS features or not on the query sequence. Leaving the check button unchecked will cause the program to skip all unidentified features. If one decides to include unidentified features within the annotation, one can define a specific description to label these features: check the field **Label feature**, select a qualifier **Key** from the list and fill in a convenient description in the **Text** edit box.

Pressing the **<Reset to default>** button resets the default settings in both tabs of the *Feature annotation settings* dialog box.

Pressing the **<OK>** button in the *Feature annotation settings* dialog box saves the new settings for the annotation project to the database.

8.8.9 Calculating an annotation project

Pressing the **<OK>** button in the *Project settings* dialog box starts the calculation of the annotation project. During the calculations, the program shows the progress in the bottom of the window as a percentage and there is a green progress bar that proceeds from left to right.

Selecting **File > Interrupt calculation** will cause the program to stop the calculation. A calculation can be interrupted at any time. Processed data will not be lost and will be saved when saving the annotation project.

Upon a finished calculation, the *Sequence display* panel is updated (see Figure 8.8.8). For convenience of interpretation, one can zoom in or zoom out the *Sequence display* panel with the zoom slider located on top of this panel in default configuration. The zoom keeps the cursor position (red line) in focus.

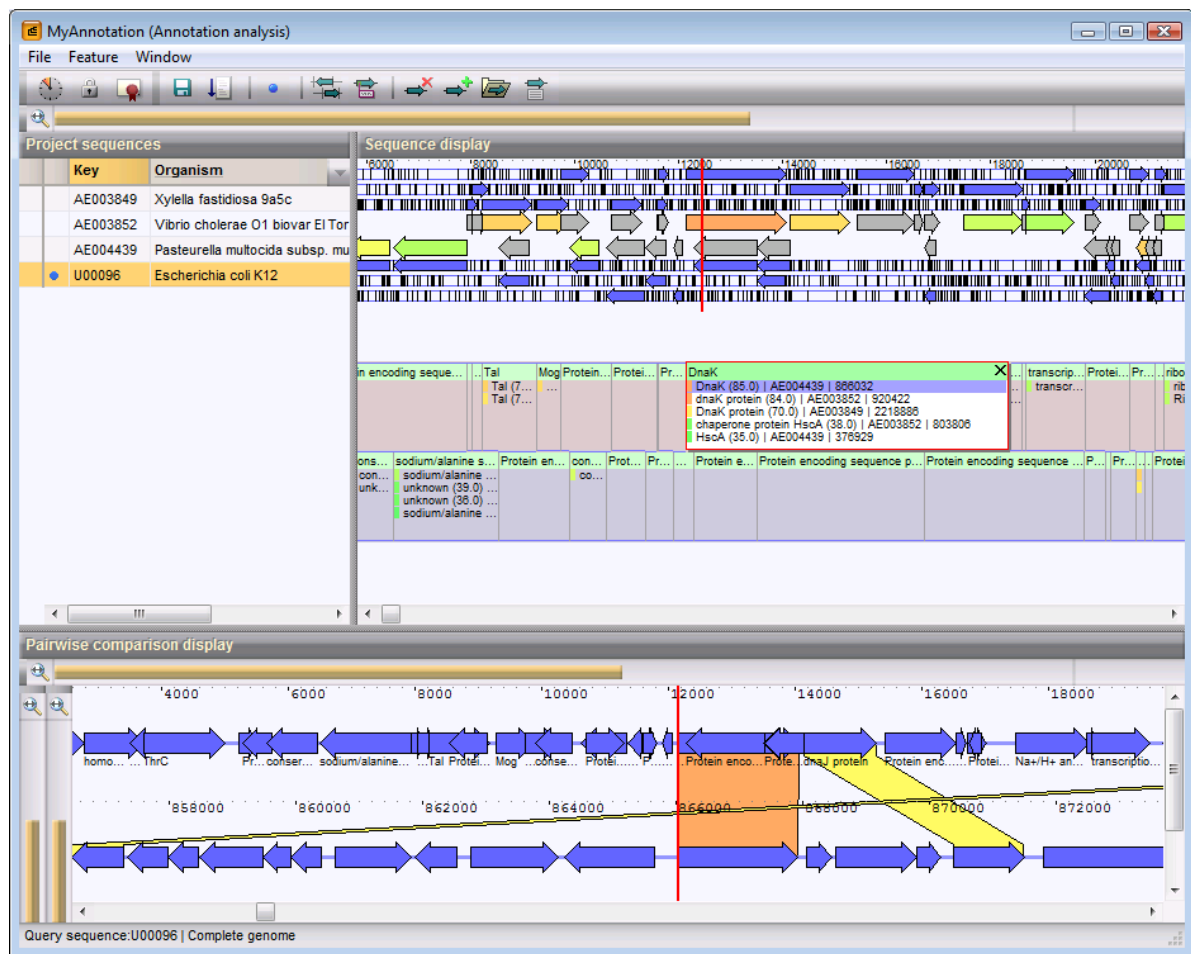


Figure 8.8.8: The *Annotation* window after finishing the calculations.

In the lower part of the *Sequence display* panel, the query coding regions and template hits that passed the annotation project criteria are displayed. The header of each coding region is colored in green, displaying the product description of the hit that is used for annotating the query sequence (default this is the description of the best scoring hit, but this can be changed by the user, see 8.8.11).

To view all hits of an open reading frame, select the open reading frame with the left mouse pointer: the field changes into a pop-up chart (see Figure 8.8.8). Within an information chart, hits found for that particular open reading frame are listed according their identification score. A color code indicates the quality of identification: the color range starts at red (100% identification score), goes over green (50%) and ends by blue (0%). Next to the color code stands the product description of the template feature. The identity score of the full query protein with the template protein is displayed next to the product description, followed by the **Key** of the template sequence, and ending with the start position of the template sequence.

When a hit is selected from an information chart with the mouse pointer, the query sequence (top) and template sequence (bottom) are plotted in the *Pairwise comparison display* panel, with focus around the selected template and query sequences (see Figure 8.8.8). Chromosome parallelism is mapped around the hit. The color of the blocks, depicting the parallel hits, represents the identification score of the hit (red (100%) going over green (50%) and ending by blue (0%)). The alignment in the *Pairwise comparison display* panel can be zoomed in and zoomed out with the zoom slider, located on top of this panel in default configuration. Two additional sliders "Distance cutoff" and "Identity cutoff" are present at the left

of the *Pairwise comparison display panel* in default configuration, allowing you to change the layout of the pairwise alignment.

- The **Identity cutoff** slider indicates an identity score value between zero and 100%. A slider position of e.g. 50% will only plot the features having an identity score of 50% or more.
- The **Distance cutoff** slider indicates a distance in base pairs ranging from zero to the full sequence length. This distance value has to be seen as a tolerance scope for the degree of parallelism of the surrounding features to be plotted around the currently selected feature. A low distance cutoff (e.g. 1,000 base pairs) will only plot the features being parallel to the currently selected stretch and having a maximum shift of 1,000 base pairs. A high distance cutoff, for example half of the sequence length, will plot nearly every cross identity between the two sequences, which might sometimes result in a very complex figure.

In the upper part of the *Sequence display panel*, all open reading frames that passed the annotation project criteria are plotted on the query sequence and on the six reading frames. The open reading frames are plotted in blue on the reading frames. On the query sequence, the open reading frames that show a hit with the template features, are colored using the color scale used for representing the identity score. The color is obtained from the hit that is used to annotate the query feature (default this is the best scoring hit, but this can be changed by the user, see 8.8.11). Open reading frames which did not show any hit with any of the features from the template sequences (at least for the given annotation settings) are plotted in gray on the forward and reverse query sequence. Features which have been imported, i.e. coding regions which were already mapped on the query sequence are displayed in purple; manually mapped features (see 8.8.11) are displayed with a turquoise color on the query sequence.

Mapped query coding regions can be selected by holding the **Shift-key** and clicking on the feature arrow on the query sequence. A detailed view of a query coding sequence with its respective hits, can be called by selecting the query coding sequence in the *Sequence display panel* and selecting **Feature > Identification details....** This brings up the *Annotation Detail* window, consisting of four panels: the *Dendrogram panel*, the *Fields panel*, the *Identity scores panel*, and the *Alignment panel* (see Figure 8.8.9).

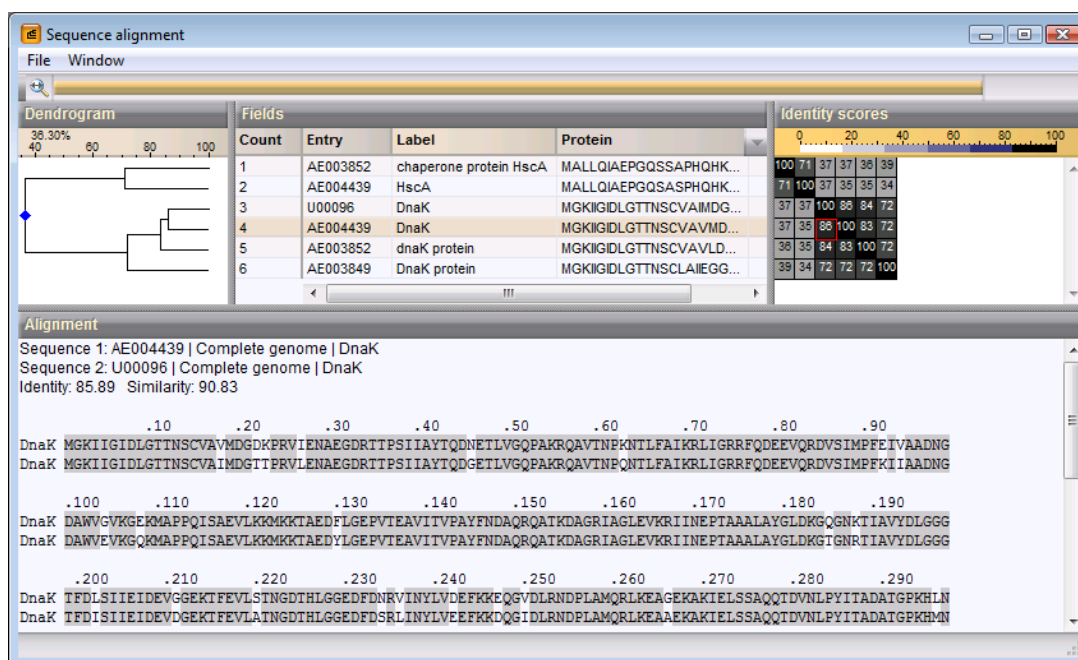


Figure 8.8.9: The *Annotation Detail* window, showing a detailed comparison of a single annotated CDS.



- *Fields panel*: This grid panel displays all hits that are present in the information chart of the selected query coding sequence (for display options of grid panels, see 3.2.7). The entry key of the hits are shown in the 'Entry' column, the product description of the hits are displayed in the 'Label' column, and the 'Protein' column holds the amino acid sequence information. An extra row is reserved for the query coding sequence: the name of the selected query sequence is displayed in the 'Entry' column and the product name and amino acid sequence of the hit that is used for annotation are displayed in the 'Label' and 'Protein' columns.
- *Identity scores panel*: This panel displays the identity score matrix. The matrix is displayed as differentially shaded blocks representing the scores. The interval settings for the shadings is graphically represented in the caption of the panel.
- *Dendrogram panel*: Displays the derived UPGMA clustering of all template hits with the query coding sequence.
- *Alignment panel*: This panel is reserved for depicting the amino acid alignment of the two sequences that represent the selected identity score in the *Identity scores panel*. Amino acids are highlighted if they are the same on both sequences. Sequence information of both protein sequences and the identity and similarity scores are shown in the upper part of this panel.

You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally.

The matrix, dendrogram, and grid panels can be enlarged or reduced in size using the zoom slider, located on top of these panels in default configuration.

8.8.10 General functions

If already calculated, one can reset an annotation project by selecting the menu item **File > Reset**. The information charts in the *Sequence display* panel will be emptied and the header of the fields is displayed in pink.

An annotation project can be saved with **File > Save project** (, **Ctrl+S**). All calculations are stored along. Selecting **Edit > Open highlighted object...** (, **Enter**) in the *Main* window, will open the selected annotation project with the entries that were present when the annotation project was last saved.

An annotation project can be closed with **File > Exit**. If unsaved data is present in the annotation project, a dialog box pops up, prompting to save the changes for the annotation project.

Sequences of selected entries in the *Main* window can be added to the annotation project with **File > Add template sequences....** The *Experiment types* dialog box opens, displaying a list of available sequence types for the selected entries. From this list, the user can select the experiment type(s) for the selected entries to be included in the annotation project. For each new selected entry/sequence type combination, an additional row is created in the *Project sequences* panel. When new template sequences are added to an annotation project, the header of the information charts is displayed in pink, indicating that the annotation was not calculated using the template sequences that are currently present in the annotation project.

With the option **File > Export annotation table...** the annotations can be exported to a text file.

The annotated sequence can be saved in the database with the option **File > Save annotated sequence as....** A dialog pops up prompting for the entry key and sequence type (see Figure 8.1.154). The default suggested settings can be changed if desired. When the annotations are saved with a sequence in the database, the annotations can be viewed in the *Annotation* panel of the *Sequence editor* window of this sequence (see Figure 8.8.10). More information about this panel can be found in 8.1.6.

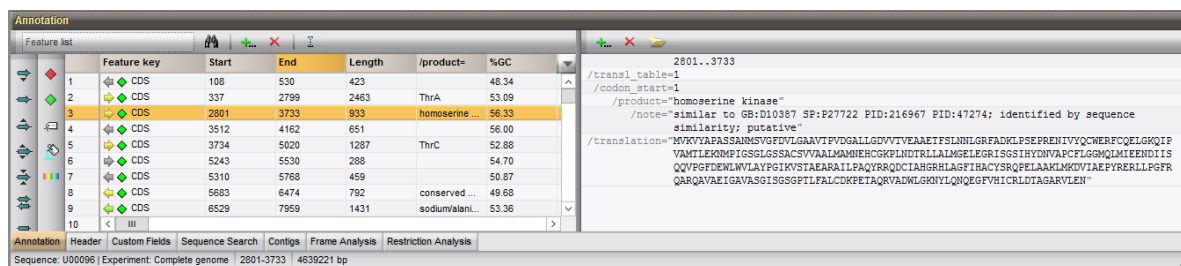


Figure 8.8.10: The Annotation panel in the Sequence editor window.

8.8.11 Editing an annotation

A mapped coding region can be removed by selecting the feature in the *Sequence display* panel and selecting **Feature > Delete**.

An open reading frame region which is not mapped on the query sequence, can be mapped by selecting the open reading frame in the reading frame in the upper part of the *Sequence display* panel and selecting **Feature > Add from selection....** Manually mapped features are displayed with a turquoise color on the query sequence.

Standard, the annotation description of the best scoring hit found within the template sequences is used for annotating the query sequence. If desired, next best scoring hits can also be selected for annotation: select an individual hit within the information chart and choose **Feature > Annotate > From selected hit**. The query feature is now annotated based on the selected hit and the product description of the selected hit is displayed in the header of the information chart. The product description of the selected hit is displayed with a blue color in the chart.

To restore the original annotation of a query sequence, select the feature in the *Sequence display* panel and choose **Feature > Annotate > From best hit**. The hit with highest identity score is again used to annotate the query sequence and the header information of the information chart is updated. The product description of the best scoring hit is displayed with a blue color in the chart.

The *Annotation feature editor* window keeps history of the editing of a selected feature. This window can be called with **Feature > Edit...** (see Figure 8.8.11).

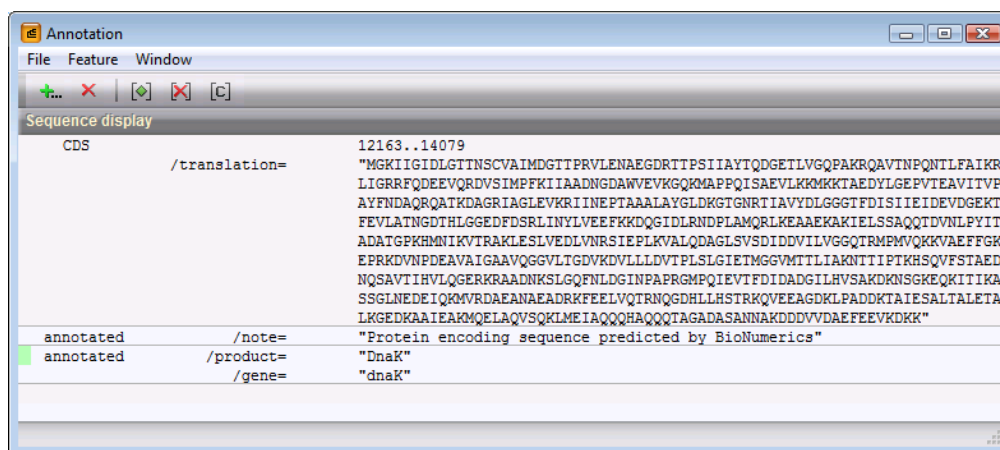


Figure 8.8.11: The Annotation feature editor window.

The amino acid sequence of the selected query feature is displayed in the upper part of the *Sequence display* panel. The qualifier groups are listed below the translation. The qualifiers that passed the labeling criteria (see 8.8.8) are contained in the qualifier groups. The green colored square within a qualifier group indicates

that the group is selected for annotating the feature. If no editing has been performed on a calculated annotation project, each feature will contain two qualifier groups: the initial qualifier group, and the qualifier group containing the qualifier descriptions of the best scoring hit. When another hit is selected to annotate the feature, a new qualifier group is automatically added to the feature qualifier history. Note that re-editing a qualifier group that has been edited before, does not result in the creation of a new qualifier group, instead this group is updated.

To force the creation of a new qualifier group from an existing qualifier group, use the copy function **Feature > Create copy of selected qualifier group**. The qualifier group is copied and is automatically marked with a green square.

A qualifier group can be deleted with the **Feature > Delete selected qualifier group**.

Which qualifier group should be used for annotation can be specified by selecting the qualifier group of interest and choosing **Feature > Set qualifier group as label**.

A qualifier description is added to a selected qualifier group using **Feature > Insert qualifier...** Selecting the intended qualifier and adding the description into the subsequent dialog boxes results into the duplication of the selected qualifier group and the insertion of the new qualifier in this group. The new qualifier group is automatically marked with a green square.

A qualifier description can be edited by double-clicking on the qualifier description text within the editor. The text appears highlighted and the description can be changed.

Qualifiers can be removed with **Feature > Delete selected qualifier**.

8.8.12 BLAST tools

BLAST functions are present in the *Annotation* window, allowing the user to perform a BLAST screening of query open reading frame regions against public databases. The results of the BLAST are imported into the annotation project and act as normal template hits. This means that alignments and cluster matrices can be viewed from template hits derived from BLAST runs. The only difference is that no chromosome parallelism can be defined from BLAST template hits, as the surrounding context of the BLAST results is not available. Therefore, no chromosome parallelism plot will be shown if a BLAST hit within an information chart is selected (the *Pairwise comparison display* panel will be empty when a BLAST template hit is selected in an information chart).

Using **Feature > Annotate > Automatic annotation of selected feature...** in the *Annotation* window, the currently selected query feature can be blasted against a database. This action calls the *BLAST search settings* dialog box, which is described in 8.9.2.7. The function runs in background and the submission progress is displayed in the status bar of the *Annotation* window.

With **Feature > Annotate > Automatic annotation of all unknown features...**, all mapped features which did not show any hit with the template features included within the project can be blasted against a database. This action calls the *BLAST search settings* dialog box, which is described in 8.9.2.7. The function runs in background and the progress of submission is displayed in the status bar of the *Annotation* window. For large sequences, it is recommended to run this function overnight.

Imported BLAST hits are marked with a colored diamond in the information charts, whereas template hits, which have been found on the template sequences included within the project, are marked with colored squares. The color code of the signs indicates the quality of identification (red (100%), going over green (50%) and ending by blue (0%). When a query open reading frame region has been screened against public databases via the BLAST-functions, the feature is marked with a purple attachment appearing at the bottom of the information chart (see Figure 8.8.12).

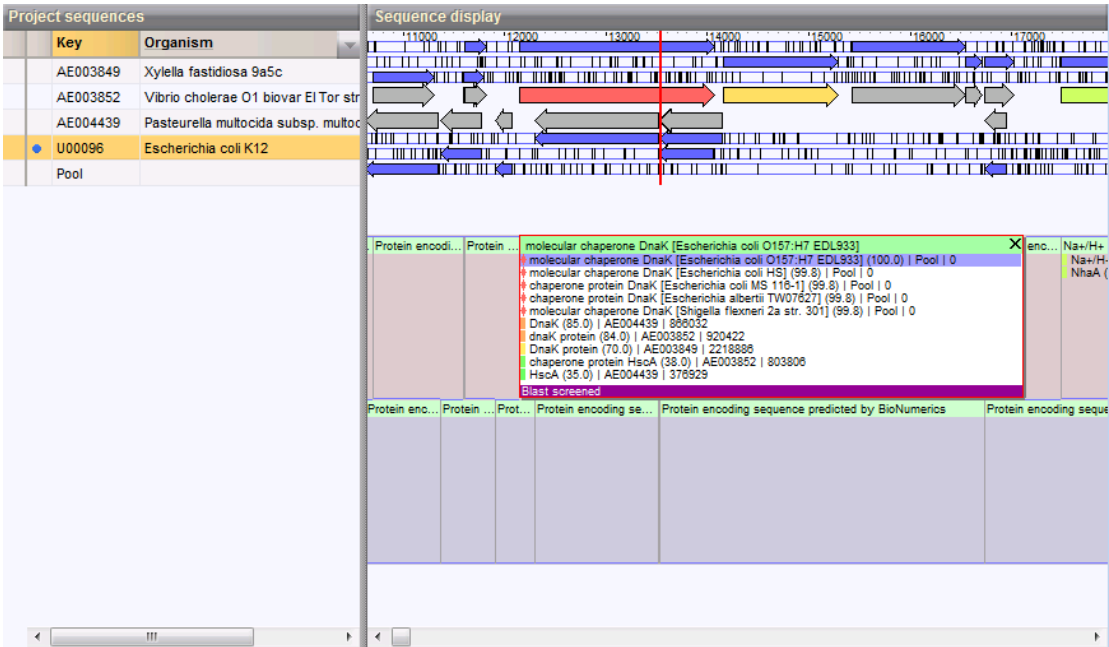


Figure 8.8.12: Blast screening of a selected feature.

Chapter 8.9

BLAST analysis

8.9.1 Introduction

In this manual, only some basic concepts and terms that are essential to understand the most important settings and parameters in the BLAST implementation in BioNumerics will be explained in brief. For in-depth documentation, we refer to the specialized literature [23].

BLAST or *Basic Local Alignment Search Tool* [7] is a fast sequence comparison algorithm used to search sequence databases for optimal local alignments to a query sequence. To optimize the speed, a substitution matrix is generated from words of length W . The initial search is done for a word of length W that scores at least T residues when compared to the query sequence. The T parameter determines the speed and sensitivity of the search. In the gapped-BLAST implementation, *word hits* are then extended in either direction, allowing gaps to be introduced, in an attempt to generate an alignment with a score exceeding a threshold value of S . One such gapped alignment is called a *high-scoring segment pair* (HSP). A sequence from the BLAST database that shows one or more HSPs with the query sequence is called a *hit*.

Before BLAST can be performed, the sequences in the database need to be converted into a special indexed format (BLAST database). The BLAST database is not updated when new sequences are added to the BioNumerics database. To update a BLAST database with new sequences, it has to be built again.

There are several different types of BLAST, depending on the type of sequences to compare:

- **Blastp**: compares an amino acid query sequence against a protein sequence database.
- **Blastn**: compares a nucleotide query sequence against a nucleotide sequence database.
- **Blastx**: compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.
- **tBlastn**: compares a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands).
- **tBlastx**: compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database. Due to the nature of tBlastx, gapped alignments are not available with this option.

Other adaptations of BLAST, such as PSI-BLAST (for iterative protein sequence similarity searches using a position-specific score matrix) and RPS-BLAST (for searching for protein domains in the Conserved Domains Database) perform comparisons against sequence profiles [25].

For the comparison of protein sequences or translated nucleotide sequences, a *Substitution Scoring Matrix* is required. Two families of matrices exist: *PAM matrices* and *BLOSUM matrices*:

- **PAM matrices** (Percent Accepted Mutation) are based on global alignments of closely related proteins. For example, the PAM1 is the matrix calculated from comparisons of sequences with no more than 1% divergence. Other PAM matrices are extrapolated from PAM1.
- **BLOSUM matrices** (BLOck SUBstitution Matrix) are based on local alignments. For example, BLOSUM 62 is a matrix calculated from comparisons of sequences with no less than 62% divergence. All BLOSUM matrices are based on observed alignments; they are not extrapolated from comparisons of closely related proteins. BLOSUM 62 is the default matrix in BLAST 2.0. Although it is tailored for comparisons of moderately distant proteins, it performs well in detecting closer relationships. A search for distant relatives may be more sensitive with a different matrix.

BLOSUM matrices with high numbers and PAM matrices with low numbers are both designed for comparisons of closely related sequences. BLOSUM matrices with low numbers and PAM matrices with high numbers are designed for comparisons of distantly related sequences.

With the advent of whole genome analysis and by using BLAST searches, the function of newly sequenced genes can be inferred, new members of gene families can be predicted, and evolutionary relationships can be explored. Additionally, this way the location and the function of protein-coding and transcription-regulation regions in genomic DNA can be obtained.



To be able to use the BLAST functionality in BioNumerics, the Sequence data module (SQ) needs to be present in your BioNumerics configuration.

8.9.2 Creating a new BLAST analysis

8.9.2.1 Background

The BLAST functionality is implemented in BioNumerics as a general sequence analysis tool, and as such the BLAST projects are accessible from within multiple sequence analysis windows:

- *BLAST analysis from the Main window* (see 8.9.2.2),
- *BLAST analysis from the Sequence editor window* (see 8.9.2.3),
- *BLAST analysis from the Sequence alignment window* (see 8.9.2.4),
- *BLAST analysis from the Chromosome Comparison window* (see 8.9.2.5), and the
- *BLAST analysis from the Annotation window* (see 8.9.2.6).

A description of the different ways to initiate a BLAST project from the different windows is given in the following sections.

8.9.2.2 BLAST analysis from the main window

In the *Main* window, the *BLAST projects* panel is displayed in default configuration as tabbed view with the *Comparisons* panel, *Decision networks* panel, *Alignments* panel, *Chromosome comparisons* panel and *Annotations* panel in the bottom right part of the window. If desired, the configuration of the *Main* window can be customized as described in 2.3.4.

To create a new BLAST project, click on the tab of the *BLAST projects* panel in the *Main* window and select *Edit > Create new object...* (🟢). A dialog appears, querying the name of the new BLAST project.

The new BLAST project is added to the *BLAST projects* panel in the *Main* window and the creation date and last modified date of the project are updated in the default information fields. When more than one BLAST project is present, projects can be sorted and searched using the information present in the default or user-defined information fields. For a detailed explanation of the display options of the *BLAST projects* panel and other grid panels, see 3.2.7.

To delete one or more BLAST projects from the list, select the project and select **Edit > Delete selected objects...** (✖).

Upon creation of a BLAST project, the *BLAST* window is automatically opened and the *BLAST search settings* dialog box is displayed, querying the BLAST search settings. Alternatively, choose **Edit > Open highlighted object...** (📁, **Enter**) to open a selected BLAST project. The first time the BLAST project is opened, it will open with the currently selected entries in the *Database entries* panel. If no entries were selected upon creation, an information dialog appears, indicating that at least one entry should be selected from the database. In this case, go back to the *Main* window, select the entries to be used in the BLAST project and re-open the project by selecting **Edit > Open highlighted object...** (📁, **Enter**). As soon as the BLAST project has been saved, selecting **Edit > Open highlighted object...** (📁, **Enter**) will open the BLAST project with the entries that were present when the project was last saved.

See 8.9.2.7 for more information on how to continue with the BLAST analysis, starting from the *BLAST search settings* dialog box.

8.9.2.3 BLAST analysis from the Sequence editor

When looking at detailed sequence information in the *Sequence editor* window, one may want to start a BLAST search on the complete sequence or on a selected subsequence.

To blast the complete sequence select **Tools > BLAST analysis...** without having any subsequence selected. This will launch the *BLAST search settings* dialog box. When a subsequence is selected in the *Sequence Editor* panel of the *Sequence editor* window, selecting **Tools > BLAST analysis...** will create a BLAST project for the selected subsequence only. This might be very useful e.g. to check a specific gene annotation derives from a whole genome assembly.

See 8.9.2.7 for more information on how to continue with the BLAST analysis, starting from the *BLAST search settings* dialog box.

8.9.2.4 BLAST analysis from the Alignment window

The *Sequence alignment* window is used for the calculation of multiple sequence alignments, subsequence searches and mutation analysis and can also be used to start a BLAST search on a selection within the alignment. Thereto, create a selection within the *Sequence display 1* panel by holding the left mouse button. This selection can span multiple entries. Once the selection is made, press **Tools > BLAST selected sequence(s)...** to launch the *BLAST search settings* dialog box where the search settings can be entered to start the BLAST analysis. Within the created BLAST project, the selection of each of the entries will be blasted as a separate entity.

See 8.9.2.7 for more information on how to continue with the BLAST analysis, starting from the *BLAST search settings* dialog box.

8.9.2.5 BLAST analysis from the Chromosome comparison window

The *Chromosome Comparison* window has been developed for large-scale comparison of sequences of unlimited length. From this window, BLAST searches can be launched on specific sequence selections. Sequence selections of parts of the sequences in the *Cell Plot* panel, the *Cell Pairwise Alignment* panel or

the *Alignment* panel are all synchronized, the way that making a selection in any of the panels and selecting **Tools > BLAST selected sequence(s)...** launches the BLAST analysis for the selection. The BLAST project is created and the *BLAST search settings* dialog box appears, asking for the search settings.

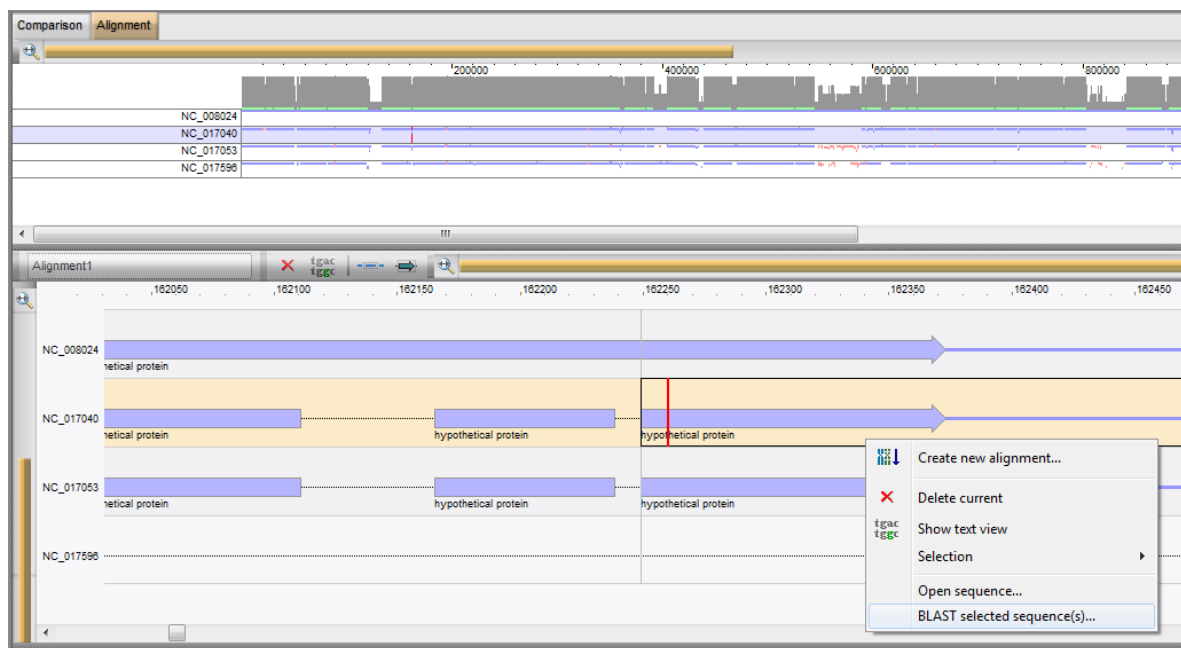


Figure 8.9.1: Start BLAST analysis from the multiple alignment view in the *Chromosome Comparison* window.

Typically, a selection will be made in the multiple alignment view. With the option **Alignment > Selection > Snap to feature** (🔗) enabled, clicking any of the CDS in the alignment view automatically selects the complete sequence spanning the CDS and makes it easy to launch the BLAST search on one feature. When a selection is made over multiple database entries, each entry selection will be blasted as a separate entity within the same BLAST project.

See 8.9.2.7 for more information on how to continue with the BLAST analysis, starting from the *BLAST search settings* dialog box.

8.9.2.6 BLAST analysis from the Annotation window

The *Annotation* window is specifically designed to annotate the coding regions on a query sequence. Within the *Annotation* window, two different types of BLAST searches can be launched. The BLAST functionality under **Feature > Annotate** allows the user to perform a BLAST screening of query open reading frame regions against the EMBL-GENBANK public databases. The results of the BLAST are imported into the annotation project and act as normal template hits. This means that alignments and cluster matrices can be viewed from template hits derived from BLAST runs. Alternatively, the BLAST functionality under **Feature > BLAST selected sequence (DNA)...** and **Feature > BLAST selected sequence (protein)...** refers to the BLAST projects and should be seen as a totally independent analysis, starting from the sequence selection in the *Sequence display* panel of the *Annotation* window. When selecting an ORF in one of the six reading frames, the sequence spanning the ORF is selected and the BLAST project can be launched for the selected subsequence by **Feature > BLAST selected sequence (DNA)...** or **Feature > BLAST selected sequence (protein)...** This opens the *BLAST search settings* dialog box.

See 8.9.2.7 for more information on how to continue with the BLAST analysis, starting from the *BLAST search settings* dialog box.

8.9.2.7 The BLAST search settings

Before BLAST analyses are launched, the BLAST program and BLAST screening database needs to be defined from the *BLAST search settings* dialog box. As already stated in the introduction, two main types of BLAST algorithms are available to perform sequence comparisons, i.e. the DNA-based queries and the protein-based queries.

Depending on the BLAST type, a BLAST program can be selected from the *BLAST search settings* dialog box.

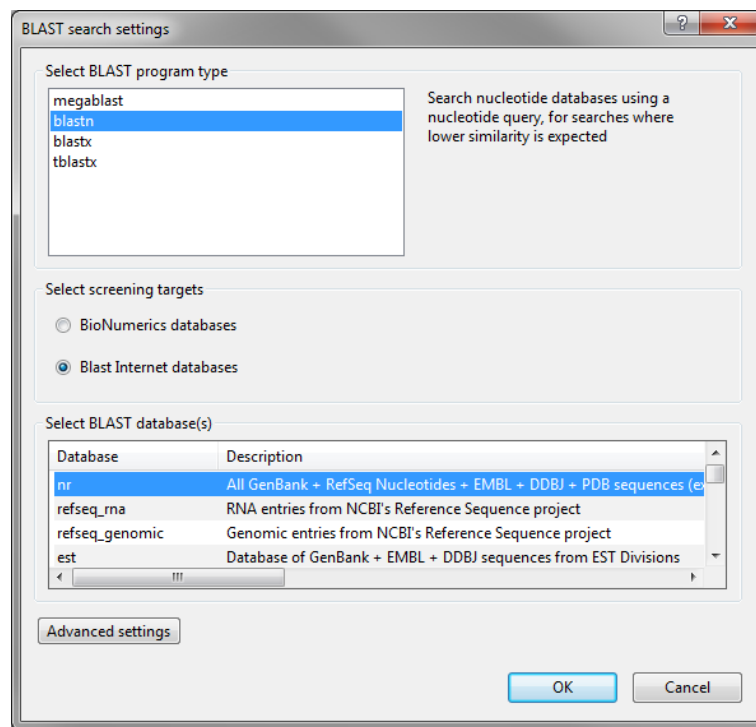


Figure 8.9.2: The *BLAST search settings* dialog box for DNA-based BLAST analysis.

It is important to realize that different nucleotide searches require different algorithms, each having its specific characteristics [29] [25]. Hereafter, an overview of the DNA-based queries (see Figure 8.9.2) is given:

- **Megablast** searches nucleotide databases using a nucleotide query, for searches where high similarity is expected. This algorithm is intended for comparing a query to closely related sequences as it is specifically designed to efficiently find long alignments between very similar sequences and works best if the target percent identity is high. As this is a very fast algorithm, this is the the best tool to use to find the high identity match to your query sequence. This algorithm finds similar sequences by breaking the query into short subsequences called words. Rather than requiring exact word matches as seeds for alignment extension (e.g. used by Blastn), discontinuous Megablast uses non-contiguous word within a longer window of template. In coding mode, the third base wobbling is taken into consideration by focusing on finding matches at the first and second codon positions while ignoring the mismatches in the third position. Searching in discontinuous Megablast using the same word size is more sensitive and efficient than standard Blastn using the same word size.
- **Blastn** searches nucleotide databases using a nucleotide query, for searches where lower similarity is expected. The Blastn algorithm also finds similar sequences based on words, but in contrast to Megablast, exact matches to the query (i.e. word hits) are used for the identification. The BLAST program then extends these word hits in multiple steps to generate the final gapped alignments. One

of the important parameters governing the sensitivity of BLAST searches is the length of the initial words, or word size as it is called. The most important reason that Blastn is more sensitive than Megablast is that it uses a shorter default word size (i.e. 11). Because of this, Blastn is better than Megablast at finding alignments to related nucleotide sequences from other organisms. The word size is adjustable in Blastn and can be reduced from the default value to a minimum of 7 to increase search sensitivity.

It is important to point out that nucleotide-nucleotide searches are not the best method for finding homologous protein coding regions in other organisms. Searches at the protein level i.e. by direct protein-protein BLAST searches or by translated BLAST searches are more sensitive because of the codon degeneracy, the greater information available in amino acid sequence, and the more sophisticated algorithms and scoring matrices used in protein-protein BLAST searches.

- **Blastx** searches protein databases using a translated nucleotide query to search for similar proteins to those encoded by a nucleotide query. Translated BLAST services are useful when trying to find homologous proteins to a nucleotide coding region. Blastx compares translational products of the nucleotide query sequence to a protein database and is particularly useful when the reading frame of the query sequence is unknown or it contains errors that may lead to frame shifts or other coding errors.
- **Tblastx** searches translated nucleotide databases using a translated nucleotide query to search for novel genes in error prone nucleotide query sequences as it gets around the potential frame-shift and ambiguities that may prevent certain open reading frames from being detected. Tblastx takes a nucleotide query sequence, translates it in all six frames, and compares those translations to the database sequences dynamically translated in all six frames. This makes the Tblastx program the slowest of the BLAST family, but effectively performs a more sensitive blastp search without doing the manual translation and allows find very distant relationships between nucleotide sequences. This type of search is computationally intensive and should be used only as last resort. Searching with large genomic queries is not recommended.

If translated sequence information is available e.g. a selected CDS in the *Annotation* window, a protein-based query can directly be started on the sequence selection.

The different protein-based queries (see Figure 8.9.3) include:

- **Blastp** searches protein databases using a protein query and is typically used to identify a query amino acid sequence and to find similar sequences in protein databases. Like other BLAST programs, blastp is designed to find local regions of similarity. When sequence similarity spans the whole sequence, blastp will also report a global alignment, which is the preferred result for protein identification purposes.
- **Tblastn** searches translated nucleotide databases using a protein query to find protein homologs in unannotated nucleotide data. A Tblastn search allows to compare a protein sequence to the six-frame translations of a nucleotide database and can be a very productive way of finding homologous protein coding regions in unannotated nucleotide sequences such as expressed sequence tags (ESTs; short, single-read cDNA sequences containing portions of transcripts from many uncharacterized genes) and draft genome records (HTG; unannotated coding regions obtained from draft sequences from genome projects), located in the BLAST databases est and htgs, respectively. Like all translating searches, the Tblastn search is especially suited to work with error prone data like ESTs and draft genomic sequences from HTG because it combines BLAST statistics for hits to multiple reading frames and thus is robust to frame shifts introduced by sequencing errors.

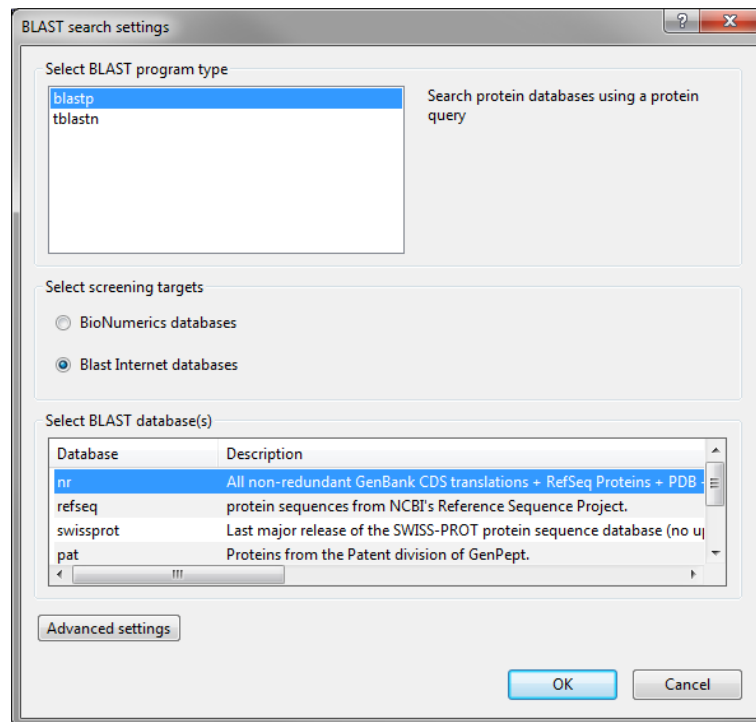


Figure 8.9.3: The *BLAST search settings* dialog box for protein-based BLAST analysis.

Once the BLAST program is selected from the *BLAST search settings* dialog box, the screening target should be specified. One has the option to use a BioNumerics BLAST database as screening target, or to use a BLAST database available as online repository on the internet.

When selecting a **BioNumerics database**, the available BLAST databases are listed in the dialog. Again, depending on the BLAST program of choice, the nucleic acid-based databases or the protein-based databases are displayed. See 8.9.4 for more information on how to create and manage BioNumerics BLAST databases.

Instead of using the local BioNumerics databases, one can also opt to use the **BLAST internet databases**. When selecting this option, the sequence and protein databases available for BLAST from the NCBI website (<http://blast.ncbi.nlm.nih.gov/>) are listed in the table. Multiple internet databases can be selected by holding the **Ctrl**-key.

Optionally, one can alter the **Advanced BLAST Settings** by pressing the <Advanced settings> button at the bottom of the dialog. This launches the *Advanced BLAST search settings* dialog box, where detailed BLAST settings for the different BLAST programs can be specified (see Figure 8.9.4).

In the *Advanced BLAST search settings* dialog box, BLAST parameter settings can be specified for the different BLAST program types [29][37].

The **Algorithm parameters** include:

- **Expectation value:** the expected number of random matches by chance, based on a random model. This setting specifies the statistical significance threshold for reporting matches against database sequences. The default value i.e. 10, means that 10 such matches are expected to be found only by chance, according to a stochastic model. Only matches with a statistical significance smaller than the expectation value will be reported. Lower expectation thresholds are more stringent, leading to fewer chance matches being reported.
- **Word size:** BLAST is a heuristic algorithm that works by finding word matches between the query and database sequences. BLAST uses these word matches to initiate extensions that might eventually lead to full-blown alignments. For nucleotide-nucleotide searches (i.e. Megablast and blastn) an exact

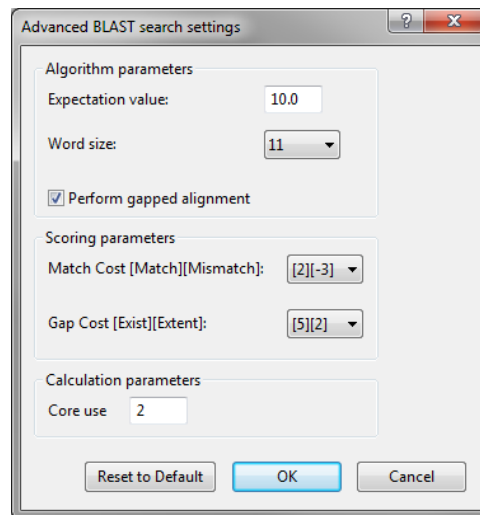


Figure 8.9.4: The *Advanced BLAST search settings* dialog box.

match of the entire word is required before an extension is initiated, so that one normally regulates the sensitivity and speed of the search by increasing or decreasing the word-size which can vary between 16 - 256 bp (default: 28) and 7 - 15 bp (default: 11), respectively. For other BLAST searches, non-exact word matches are taken into account based upon the similarity between words. The amount of similarity can be varied so one normally uses just the word-sizes 2 and 3 for these searches.

- **Perform gapped alignment option:** With this option checked, gaps can be introduced in the alignment as long as the sequence homology exceeds the threshold value.

Many nucleotide searches use a simple scoring system that consists of a **reward** for a match and a **penalty** for a mismatch. The (absolute) match/mismatch ratio should be increased as one looks at more divergent sequences. A ratio of 0.33 (1/-3) is appropriate for sequences that are about 99% conserved; a ratio of 0.5 (1/-2) is best for sequences that are 95% conserved; a ratio of about one (1/-1) is best for sequences that are 75% conserved. To ensure BLAST returns more reliable statistics for blastn searches, restraints were introduced on the allowed match/mismatch pairs and their associated gap existence and gap extension costs.

The **Scoring parameters** therefore include:

- **Match cost:** reward and penalty cost pairs for matching and mismatching bases in the alignment.
- **Gap cost:** gap existence and gap extension cost pairs for introducing and extending gaps in the alignment. Increasing the gap costs will result in alignments which a decreasing number of gaps introduced.
- **Scoring matrix:** assigns a score for aligning any possible pair of residues, and as such determines the overall alignment score. Different substitution matrices are tailored to detect similarities among sequences that are diverged by differing degrees. The matrix used in a BLAST search can be changed depending on the type of sequences you are searching with.

The **Calculation parameter** includes only one parameter, i.e. the **number of cores to use** for the calculation.

Press **<Reset to Default>** to load the initial default parameter values, press **<OK>** to update the advanced BLAST settings or press **<Cancel>** to close the *Advanced BLAST search settings* dialog box without altering any of the BLAST parameter settings.

With the BLAST program and BLAST database selected, the BLAST search can be launched in the *BLAST* window by pressing **<OK>**.

8.9.3 The BLAST window

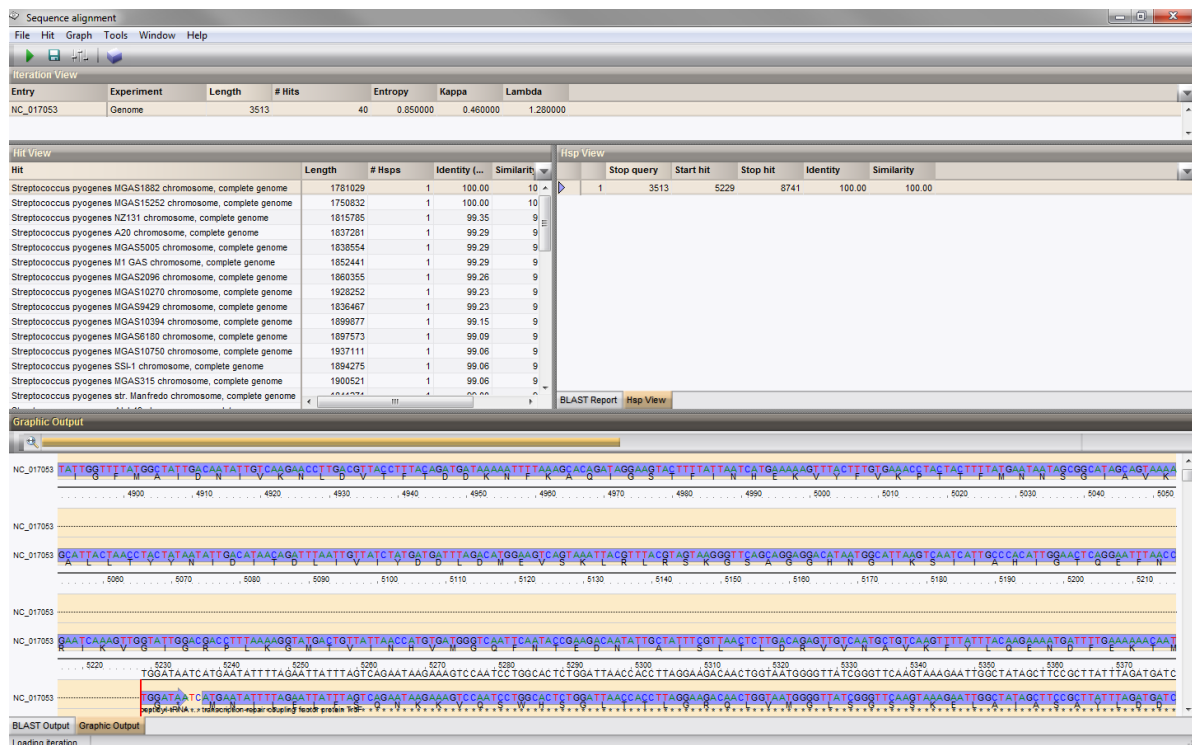





Figure 8.9.5: The BLAST window.


The BLAST window consists of multiple panels: the BLAST Report panel, the Iteration View panel, the Hit View panel, the Hsp View panel, the BLAST Output panel and the Graphic Output panel (see Figure 8.9.5). All panels are dockable, which enables the user to customize the layout of the BLAST window according to personal preferences. You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally. See 2.3.4 for detailed information on the display options of dockable panels.

- The BLAST Report panel provides an over view of the BLAST screening settings i.e. the program that was used to perform the search and the related program settings, and also the database that was used for screening.
- In the Iteration View panel, the entry information from the blasted sequence is displayed (entry key, experiment type and sequence length). Additionally, the number of returned hits, and the relative entropy value H (expressed in bits) is indicated, together with the kappa and lambda parameter values, two statistical parameters used in calculating BLAST scores and are used in converting a raw score (S) to a bit score (S'). At high values of H short alignments can be distinguished by chance, whereas at lower H values a longer alignment may be necessary [6].
- In the Hit View panel, the different BLAST hits are listed by decreasing identity values. For each hit, the GenBank header definition of the sequence is displayed in the Hit column, followed by the length of the full hit sequence, the number of High-scoring segment pairs (HSPs) per hit, and the identity and similarity value of the HSP with the highest alignment score in a given search.
- When selecting a hit with multiple Hsps, the detailed Hsp information is displayed in the Hsp View panel. In this view, the strand specificity is displayed next to the start and stop positions of the query sequence and the start and stop positions of the Hsps found on the hit sequence. Next to the position information, the sequence identity and similarity scores are indicated together with the bit



score, S' , derived from the raw alignment score S (calculated as the sum of substitution and gap scores), which can be used to compare alignment scores from different BLAST searches. One of the most important scores in this panel is the Expectation value or E value, representing the number of different alignments with raw alignment scores equivalent to or better than S as is expected to occur in a database search by chance. The lower the E value, the more significant the score and thus the alignment. Detailed alignment information i.e. the number of identical positions, the number of positive positions and the alignment length completes the information displayed in this panel.

- For any selected Hsp, the *BLAST Output* panel displays the alignment, preceded by the sequence identifier, the full subject definition line, and the number of identical residues in this alignment (Identities), the number of conservative substitutions (Positives), and if applicable, the number of gaps in the alignment. Beneath that, the actual alignment is shown, with the query on top, and the database match below. The numbers at left and right refer to the position in the sequence. One or more dashes (-) within a sequence indicate insertions or deletions. The line between the two sequences indicates the similarities between the sequences. The BLAST alignment from the *BLAST Output* panel can be printed by selecting **Hit > Print alignment...** (). This opens the *Data Viewer* window where the alignment can be copied to the clipboard, or a print job can be submitted.
- In case the match is available in the database, its alignment can be displayed in the *Graphic Output* panel. If the match originated from an online database, the alignment first needs to be loaded in the *Graphic Output* panel, by selecting **Graph > Fetch full hit sequence....** Again, the alignment is displayed with the query on top, and the database match, showing the annotation information, below. The numbers refer to the position in the database match sequence. One or more dashes (-) within a sequence indicate insertions or deletions. On top, conservative substitutions, as judged by the substitution matrix, are indicated with +. The asterisk between the two sequences indicates the similarities between the sequences, and just beneath the database match, the amino acid residue is shown.

A BLAST hit can be saved to the BioNumerics by selecting **Hit > Load full hit sequence in sequence editor...** (). This will automatically download the hit sequence from the online repository and opens it in the *Sequence editor* window. From the *Sequence editor* window, the sequence can then be saved to the database by selecting **File > Save** (, **Ctrl+S**) for later reference.

Once a blast analysis has completed it can be saved by **File > Save...** (). This opens the *Save BLAST object* dialog.

In this dialog, the name of the BLAST object can be specified and one has the option to save the BLAST object as a local copy or as a database object available to all database users.

To resubmit a modified BLAST search from within the same project, the BLAST search settings can be queried for and changed from the *BLAST search settings* dialog box by selecting **File > Settings...** () (see 8.9.2.7). The modified search can be launched by selecting **File > Run...** ().

Also from this window, the overview of the available nucleic acid-based and protein-based local and internet BLAST databases can be obtained. To this purpose, select **Tools > BLAST database tools...** to call the *BLAST Database Tools* dialog box. See 8.9.4 for more information on the creation and management of BLAST database.

File > Exit will close the *BLAST* window. In case the BLAST project was not yet saved, first the *Save BLAST object* dialog will appear again before closing the window.

8.9.4 BLAST databases

Two types of BLAST databases exist: the *nucleic acid databases*, containing DNA or RNA sequences and the *protein databases*, assembled by amino acid sequences which are specified by the translated codons.

Select **Database** > **Sequence databases** > **BLAST databases...** to manage the BLAST databases in the *BLAST Database Tools* dialog box (see Figure 8.9.6).

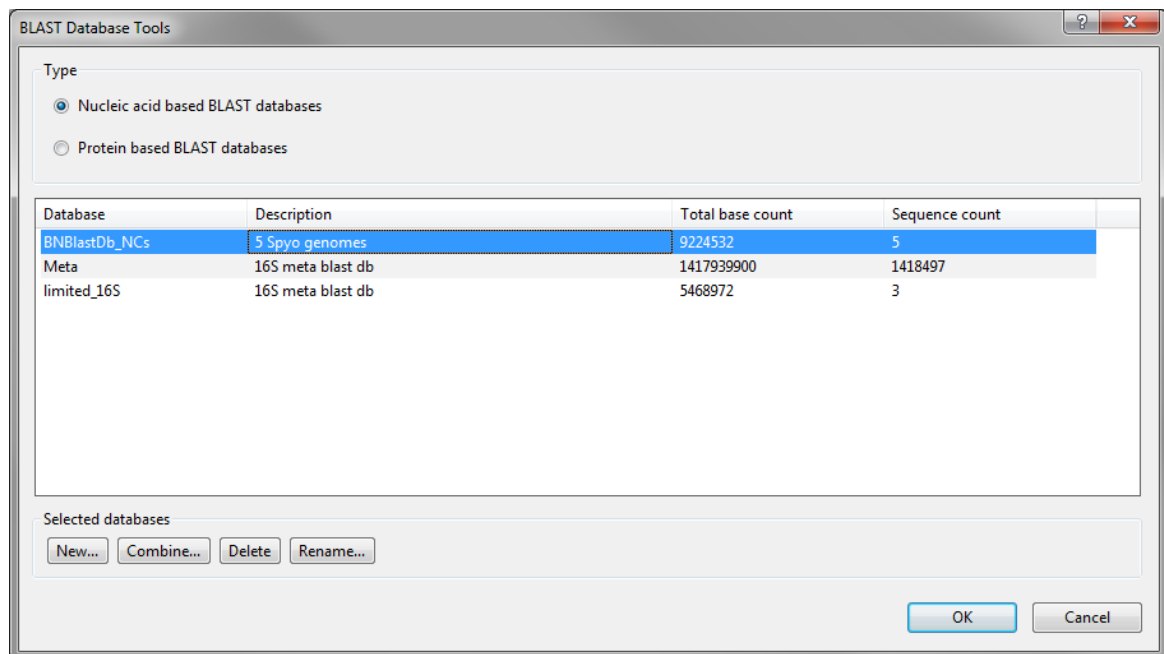


Figure 8.9.6: The *BLAST Database Tools* dialog box.

In the *BLAST Database Tools* dialog box, an overview of the existing nucleic acid and protein BLAST databases is shown. For each database type, a list is provided with the database name, the description, the total base count and the sequence count i.e. the number of sequence entities present in the database. From this window, new BLAST databases can be uploaded in the database, modified or deleted.

A new BLAST database can be created from a selection of entries in the database or from a FASTA file. To create a new BLAST database, select <New> in the *BLAST Database Tools* dialog box. This pops up the *Create BLAST database* dialog box (see Figure 8.9.7).

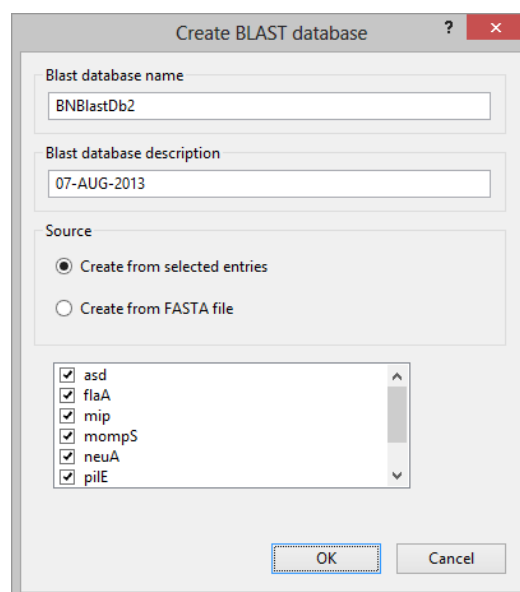


Figure 8.9.7: The *Create BLAST database* dialog box.

In this dialog the BLAST database name, a database description and the source can be specified. If entries

were selected in the database, both the options *Create from selected entries* and *Create from FASTA file* are available. In case no entry was selected, only the option *Create from FASTA file* is active.

To create a BLAST database from the entry selection, modify the database name and description, select the source, define which sequence experiment types should be included in the BLAST database and press **<OK>**. The window closes and the new database was added to the list.

To create a BLAST database from FASTA file, modify the database name and description, check FASTA file as source and browse for the FASTA file which contains the reference sequences you want to import as BLAST database. Press **<OK>** to create the BLAST database and to close the *Create BLAST database* dialog box. The new database is now added to the list in the *BLAST Database Tools* dialog box.

The name of an existing BLAST database can be altered at any time. Select the database and select **<Rename>** to call the *Rename BLAST database* dialog box.

In the *Rename BLAST database* dialog box, the current name of the BLAST database is displayed and the new name can be entered. The name cannot contain spaces. Press **<OK>** to close the dialog box and save the changes to the database list.

A new BLAST database can also be defined as a combination of existing BLAST databases. Thereto, select multiple BLAST databases by holding the **Ctrl**-key and press the **<Combine>** button. This pops up the *Rename BLAST database* dialog box again.

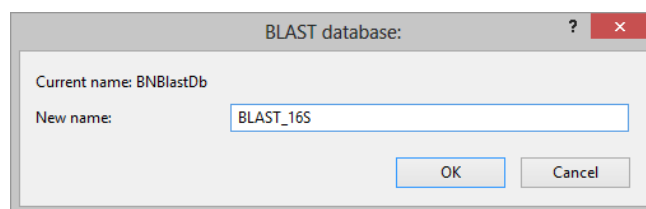


Figure 8.9.8: The *Rename BLAST database* dialog box.

In this dialog box, the current BLAST database name is displayed and the name of the new BLAST database can be entered. The new name cannot contain spaces. Press **<OK>** to create the combined database. The database is added to the list of available BLAST databases, and the description contains the names of the BLAST databases from which the new database was compiled.

A selected database can be deleted from the list by selecting **<Delete>**. A confirmation is then displayed. When confirming the deletion of the BLAST database, the database is removed from the BioNumerics BLAST repository.



When deleting a database that was also used as a source for a combined BLAST database, both the database itself, and all combined databases relying on this database will be deleted from the list of available BLAST databases. So before deleting databases, check whether this may have implications on other derived BLAST databases.

Press **<OK>** to close the *BLAST Database Tools* dialog box and save the changes to the BioNumerics BLAST databases. Press **<Cancel>** to close the *BLAST Database Tools* dialog box without saving any changes.

Chapter 8.10

Whole genome single nucleotide polymorphism analysis

8.10.1 An introduction to wgSNP analysis

8.10.1.1 Definitions

A Single Nucleotide Polymorphism (SNP) is a variation in a single nucleotide, which occurs at a specific position of the genome. SNPs are always defined with respect to a *reference sequence*. A SNP search or SNP analysis can therefore be regarded as a post-analysis on (aligned) sequences, in which SNPs are determined on one or more sample sequences, in relation to a reference sequence. When performed on whole genome sequences (WGS), we refer to this analysis as **whole genome SNP (wgSNP)** analysis.

In any SNP search, it is crucial that the reference and sample sequence are aligned, since this is the only way in which base calls per position can be compared meaningfully. However, the process of aligning sequences can be calculation-intensive and time-consuming. This is especially the case for sequences of full genome length, where large insertions, deletions or rearrangements can further complicate the procedure.

Instead, the wgSNP analysis in BioNumerics relies on all sequences being *collinear*, i.e. in the same frame and having the same length, which avoids a time-consuming sequence alignment. Within reasonable limits, this is enforced by the use of a *reference mapped* sequence type (see [8.10.1.2](#)). Collinearity can then be achieved by starting from the raw sequence reads (stored in a sequence read set experiment) and mapping these reads for each entry in the analysis to the same reference sequence.

The choice of a reference sequence in a wgSNP analysis is a very important one, since only genomic information that is in common between the reference sequence and the sample sequence will be included in the analysis. With other words, any gene, integron, plasmid, etc. that is present in the reference but not in the sample (or vice-versa) will be left out. In order to obtain the highest possible resolution in a wgSNP analysis, the reference should be as similar as possible to the sample sequences. This can be achieved by choosing the reference based on a screening method such as wgMLST.



Although same-length short DNA sequences (such as trimmed gene sequences) could in principle also be analyzed with the tool explained in this chapter, it is primarily designed for SNP analysis on whole genomes. For finding SNPs on sequences up to a few thousands bases, the mutation search in the *Sequence alignment* window is the recommended tool (see [8.4.19](#)).



For unaligned genome-length sequences, such as a set of reference genomes downloaded from an online repository, we refer to the SNP analysis in the *Chromosome Comparison* window (see [8.7.4](#)).



A completely different approach to SNP analysis is PCR-based SNP calling e.g. using TaqMan technology. The *SNP calling plugin* (see the separate plugin manual) was specifically developed for interpretation of data generated with the latter methodology.

In a SNP analysis, each sample sequence (obtained via mapping to the reference, see above) is compared to the reference sequence and all base differences are recorded. This total number of SNPs may consist for a large proportion of 'SNPs' from regions that have no or insufficient sequence coverage. Since these regions would strongly bias the results, they should be removed from the analysis. This is achieved via *SNP filters* (see 8.10.2), of which a wide variety is available in BioNumerics for discrimination of 'true' point mutations from e.g. insufficiently covered positions, sequencing artifacts, indels, etc..

Since SNP filters (potentially) remove sequence positions (i.e. for all sequences included in the analysis), it can occur that adding or removing a sample sequence results in the inclusion or exclusion of a SNP position. With other words, a set of SNPs that is retained after filtering is only relevant within the set of sample sequences it originates from. The way this is handled in BioNumerics is to make the SNP matrix available in a comparison as a character aspect of the original sequence experiment type (see 8.10.3). As the comparison defines the set of entries, storing the SNP matrix with the comparison reinforces this relationship.

8.10.1.2 Prerequisites

As indicated in 8.10.1.1, SNPs are always determined in relation to a reference sequence and all sample sequences should be in the frame of the reference sequence, so that positions can be compared. To ensure that this conditions are met, a SNP analysis in BioNumerics has following prerequisites:

- All sequences on which a SNP analysis against a certain reference sequence is performed, should be stored in the same sequence type. Conversely, if the same sequences are to be analyzed against a different reference sequence, a new sequence type should be created.
- The sequence type should be reference mapped, meaning that the option *Use reference sequence as mapping template (required for SNP analysis)* should be checked during creation of the sequence type (see 8.1.1). If this was not done at creation time, an existing sequence experiment type can be converted into a reference mapped sequence experiment type (see 8.1.2.2).
- The reference sequence should be specified in the sequence experiment type settings (see 8.1.2.3).
- Sequence reads should be mapped against this reference sequence.
- All entries should have an experiment for the sequence type, i.e. a full data matrix is needed.

These requirements are automatically checked off by the software: a SNP analysis can only be done on sequences stored in a reference mapped sequence type, for a selection of entries that all have an experiment for this sequence type. The "reference mapped" property further ensures that (1) a reference sequence is present and (2) that all sequences are of the same length.



Performing a SNP analysis requires the Genome Analysis Tools module (GA) to be present in your BioNumerics configuration.

8.10.1.3 Workflow

Below is a typical workflow for a wgSNP analysis in BioNumerics (see also Figure 8.10.1):

1. Choose a reference sequence. This might be a closed, fully annotated genome sequence (e.g. downloaded from an online repository), but could just as well be a de novo assembled sequence, consisting of multiple contigs (i.e., a draft genome). Selection of a reference sequence is a crucial step in any SNP analysis, as explained in 8.10.1.1.

2. Create a sequence experiment type and check the option *Use reference sequence as mapping template (required for SNP analysis)* (see 8.1.1).
3. Import the reference sequence into this new sequence experiment type. See 8.1.3 for an overview of sequence import options. When using a de novo assembled sequence as a reference, the recommended option is using **File > Save as...** (**Ctrl+Shift+S**) in the *Sequence editor* window (see 8.1.6.2.4), as this ensures that coverage information is copied along. The first sequence imported into a reference mapped sequence type will automatically be assigned as the reference sequence.
4. Map the sequence reads of the samples against the reference sequence to obtain consensus sequences for the samples, in the same frame as the reference sequence. The mapping can be done locally on your desktop computer via **Analysis > Sequence read set types > Map to reference** (see 9.5.3) or on the calculation engine after installation of the *WGS tools plugin*.
5. Filter out the relevant SNPs via a stored template (see 8.10.2). The results can be assessed and – if needed – further refined in the *SNP filtering* window.
6. Analyze the obtained SNP matrix in the *Comparison* window (see 8.10.3).

8.10.2 SNP filtering

8.10.2.1 Introduction

A potentially very large set of SNPs may be found between the reference sequence and the sample sequences in the analysis. In addition to true point mutations, this set may contain polymorphisms be due to sequencing artifacts (incomplete or insufficient coverage), large insertions (possible due to horizontal gene transfer), deletions or chromosomal rearrangements. For a phylogenetic analysis or for strain typing it is therefore very important to filter out only the relevant SNPs. BioNumerics offers this functionality through various SNP filters. These filters are contained in a template (see 8.10.2.4.4) and their effect can be assessed in detail in the *SNP filtering* window (see 8.10.2.4).

The *SNP filtering* window can be accessed in principally two different ways:


- Starting from the *Main* window, based on a selection of entries (see 8.10.2.2).
- From the *Comparison* window, starting from a reference mapped sequence type as the active experiment (see 8.10.2.3).

Both methods are essentially equivalent and which workflow to choose mainly depends on personal preferences.

8.10.2.2 Initiating a SNP filtering from the Main window

A SNP filtering can be performed on the sequence experiments of the selected entries – with or without opening the *SNP filtering* window – via the *SNP analysis* wizard:

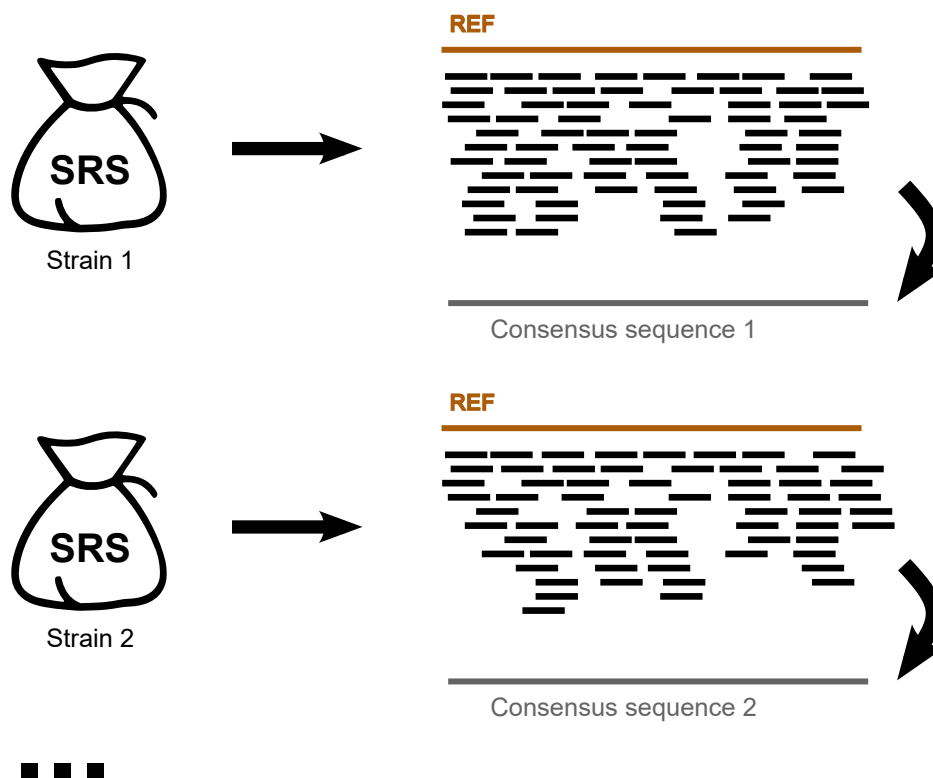
In the *Main* window, make a selection of entries to analyze and select **Analysis > Sequence types > Start SNP analysis....** This action opens the *Experiment type* wizard page (see Figure 8.10.2).

Alternatively, the *SNP analysis* wizard can be started via **File > Process...** () and the option **Start SNP analysis** (under *Sequence type*) in the *Process data* dialog box.

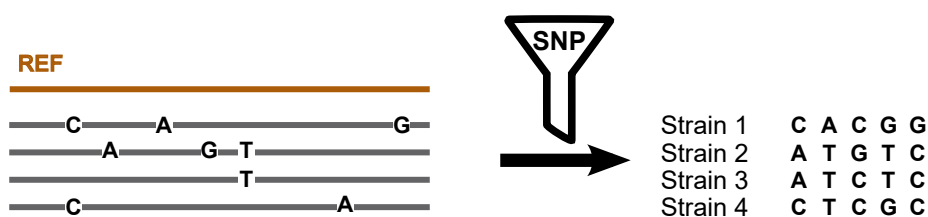
1. Select reference sequence & Import in new sequence type

REF

2. Map sequence reads against reference



3. SNP filtering on consensus sequences



4. Analyze SNP matrix in comparison

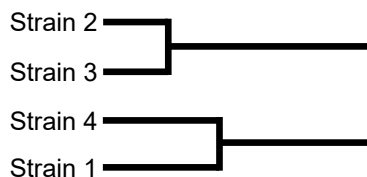


Figure 8.10.1: An illustration of the wgSNP workflow in BioNumerics.



The software checks if all selected entries have sequence data for at least one sequence type that has the option checked. If this is not the case, the error message "No reference mapped Sequence experiment type found with data for all selected entries" will pop up.

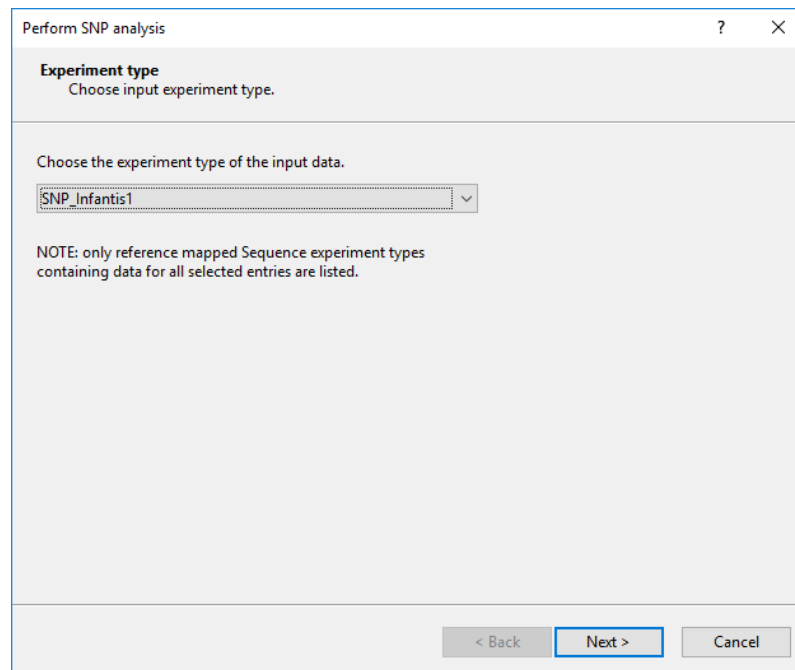


Figure 8.10.2: The *Experiment type* wizard page.

This dialog allows the user to select the sequence experiment type on which to perform the SNP analysis.

The drop-down list contains *exclusively* reference mapped sequence experiment types (see 8.1.1 and 8.1.2.2), for which a sequence is present *for each entry* in the selection. This ensures that (1) all sample sequences are in the same frame and (2) a complete data matrix is available.

Pressing <*Next*> will display the *Template* wizard page (see Figure 8.10.3).

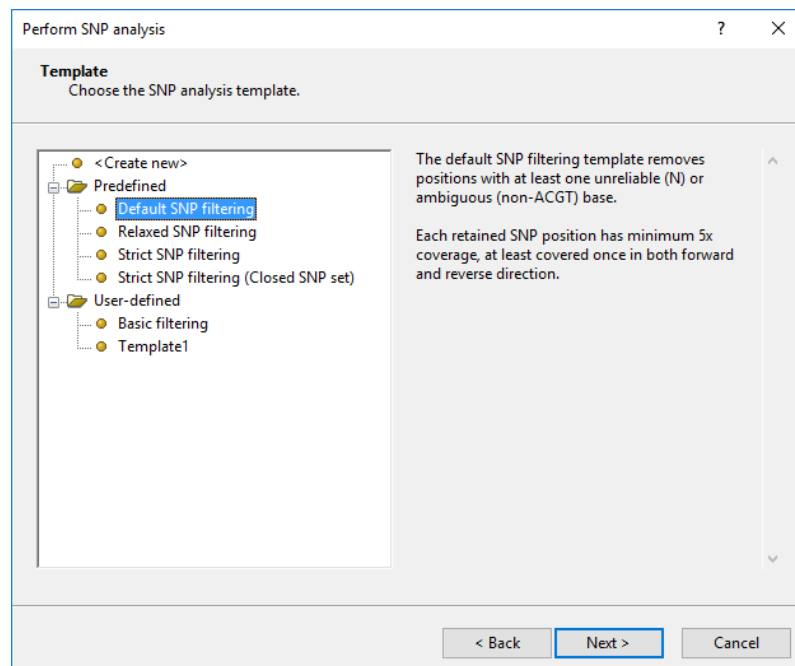


Figure 8.10.3: The *Template* wizard page.

From the tree control on the left, you can select the SNP analysis template to be applied on the data set. A SNP analysis template contains a set of SNP filters with their parameter values.

Four **Predefined** SNP templates are available. Additionally, it is possible to create your own SNP templates (see 8.10.2.4.4), which will then appear in the **User-defined** list. When a SNP template is highlighted, its description appears on the right-hand side of the dialog (see Figure 8.10.3).

With the **<Create new>** option selected, a *SNP filtering* window will open after completion of the *SNP analysis* wizard without any filtering applied and SNP filters can be added manually (see 8.10.2.4.2).

Pressing **<Next>** will display the *Analysis* wizard page (see Figure 8.10.4).

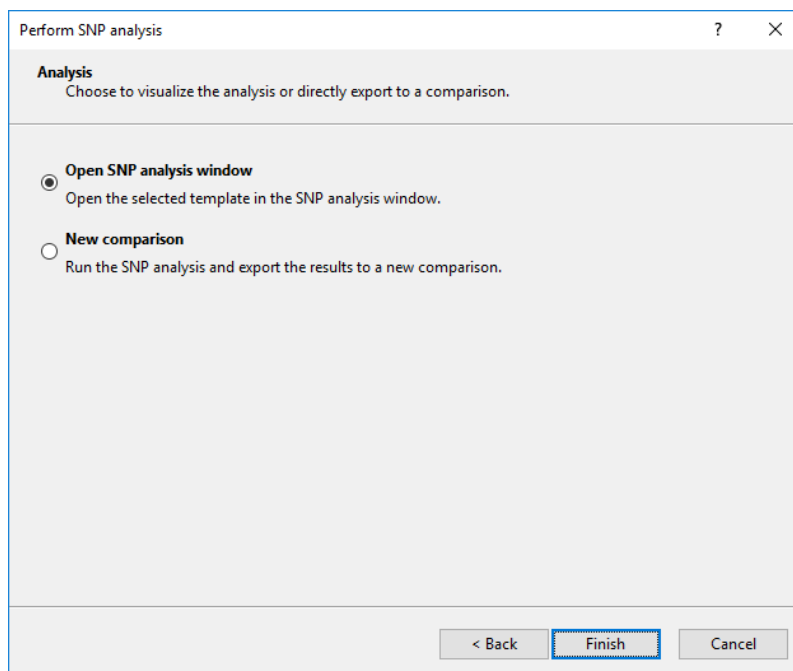


Figure 8.10.4: The *Analysis* wizard page.

Basically, two *Analysis* options are offered:

- **Open SNP analysis window:** The SNP analysis template selected in the previous step will be applied and the results displayed in the *SNP filtering* window, which allows a detailed assessment of the SNP filters and further refinement of the applied filters. This is the recommended option for most analyses.
- **New comparison:** The selected SNP analysis template will be applied "blindly" and the resulting SNP matrix is opened directly in the *Comparison* window. This is useful in a routine analysis, with a tried-and-true SNP analysis template.

Pressing **<Finish>** will open the *SNP filtering* window (see 8.10.2.4).

8.10.2.3 Initiating a SNP filtering from the Comparison window

A SNP analysis can be started from a comparison, in which the sequence data of a reference mapped sequence experiment type are displayed. The advantage of this approach is that more than one SNP matrix can be exported to the same *Comparison* window (see 8.10.2.4.5), so that the resulting cluster analyses can be compared more easily.

Similar as when the SNP analysis is started from the *Main* window, the user has the option to pass via the *SNP filtering* window:

In the *Comparison* window, select **Sequence > Open SNP window...** to call the *Template* wizard page (see Figure 8.10.5).



The command **Sequence > Open SNP window...** is only available when (1) the active experiment type is a reference mapped sequence type (see 8.1.1 and 8.1.2.2), (2) its <Default> aspect is selected from the corresponding drop-down list in the *Experiments* panel and (3) the actual sequence data are loaded in the *Experiment data* panel.



In order to perform a SNP analysis, all entries in the comparison should have a sequence for the active sequence type.

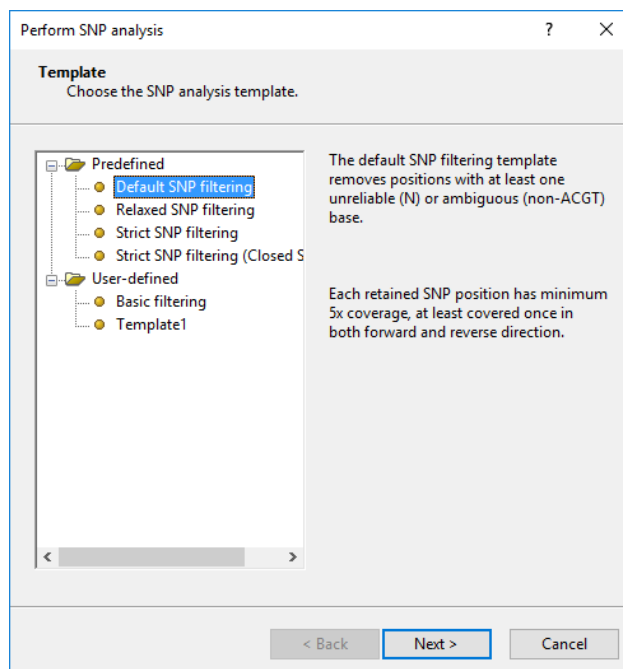


Figure 8.10.5: The *Template* wizard page.

This dialog offers the same SNP templates as available from the *Template* wizard page (see also 8.10.2.2).

Select a SNP template and press <*Next*> to open the *SNP filtering* window (see 8.10.2.4).

Alternatively, it is possible to apply a SNP template without opening the *SNP filtering* window and perform a cluster analysis on the obtained SNP matrix in a single action (see 8.10.3.2). This is probably the fastest method for SNP analysis in routine applications.

8.10.2.4 The SNP filtering window

8.10.2.4.1 Layout of the SNP filtering window

The *SNP filtering* window is shown in Figure 8.10.6. The purpose of this window is to evaluate SNPs within a set of sample sequences and to examine the effect of SNP filters and their parameters.

This window consists of following panels:

- The *Entries* panel shows all entries that are included in the SNP analysis, with all entry information fields. Two additional fields are present: 'Total' shows the raw number of SNPs (i.e. without any SNP filter applied) and 'Retained' shows the number of SNPs after applying all active SNP filters for the sample sequence.
- The *Filters* panel shows the list of SNP filters that are applied, with the 'Info' column showing additional information regarding the filter and applied settings (if applicable). This list is initially pop-

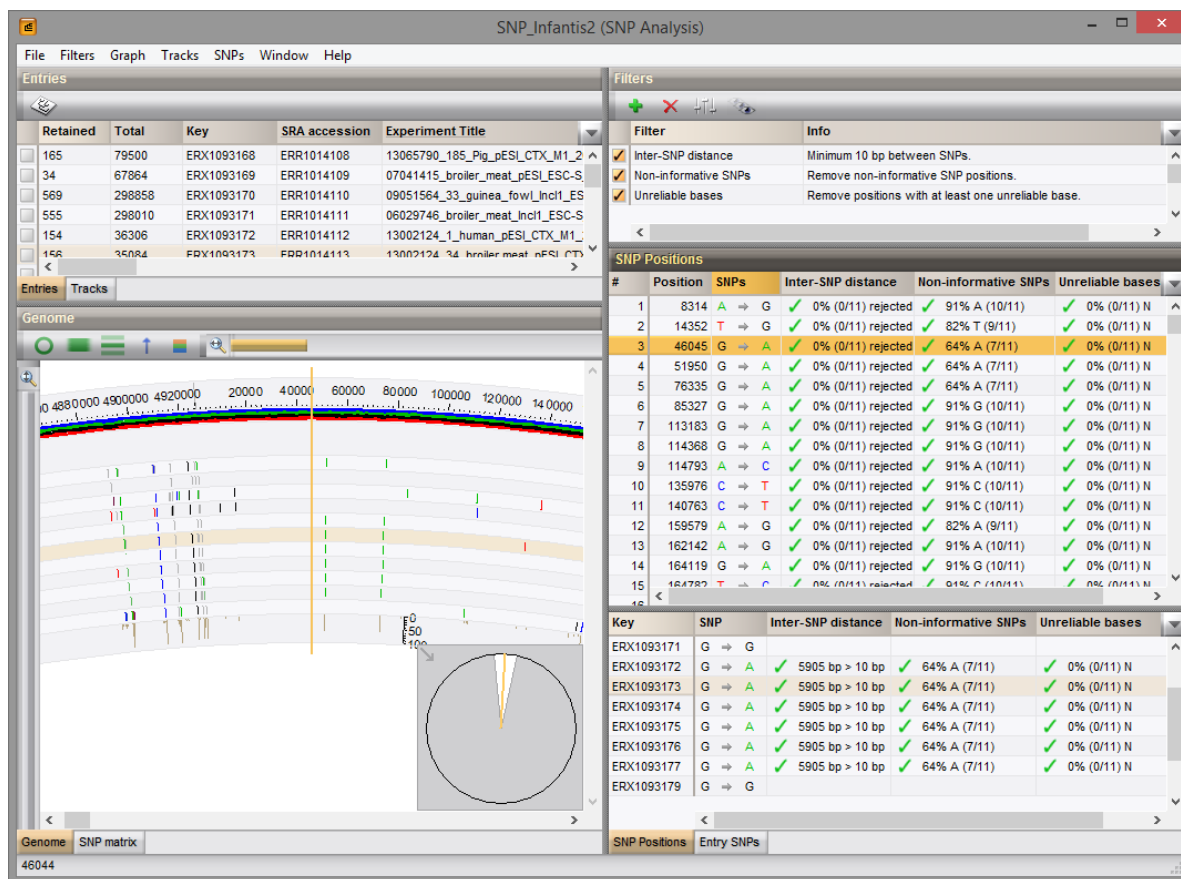




Figure 8.10.6: The *SNP filtering* window.

ulated from the SNP template, but SNP filters can be added or removed and their settings can be changed.

- The *SNP Positions* panel shows information on sequence positions where at least one SNP was detected. For each SNP filter that is listed in the *Filters* panel, a column is displayed with the filter's result on each position. The bottom of this panel shows a sub-panel with the details on the highlighted position, i.e. showing the base and filter results for all the sample sequences on that position.
- The *Entry SNPs* panel lists the SNPs for the highlighted entry in the *Entries* panel.
- The *Genome* panel shows the SNPs on a genome view and is very similar in functionality with the *Genome* panel in the *wgMLST quality assessment* window. Using **File > Export graph...**, the content of the *Genome* panel can be exported to one of several available graphics file formats.
- The *Tracks* panel in default view is displayed as a tab with the *Entries* panel. With this panel, you can determine which tracks are plotted in the *Genome* panel via the icon next to each track and rearrange their ordering using **Tracks > Move up** () and **Tracks > Move down** (). All SNP tracks can be toggled on or off at once with **Tracks > Toggle all SNP tracks** ().
- The *SNP matrix* panel shows the resulting SNP matrix, as it would be exported. The reference sequence is shown on top of the SNP matrix.

When the toggle **Filters > Toggle rejected SNP visibility** is unchecked () , the positions in the *SNP Positions* panel and the *Entry SNPs* panel will be limited to the retained SNPs, i.e. those SNPs that have passed the applied SNP filters. When the toggle is checked () the listed positions in both panels correspond to the total (i.e., unfiltered) SNP set.

Whenever possible, the cursor position is synchronized between the different panels. For example, if you click on a position in the *SNP Positions* panel, the details in the bottom part of the panel are updated and so is the *Genome* panel: the graph will show the position. Furthermore, the clicked position in the *SNP Positions* panel will appear highlighted in the *Entry SNPs* panel, *only* if the currently highlighted entry in the *Entries* panel has a SNP at that position.

Double-clicking a position in the details panel (bottom part of the *SNP Positions* panel) or in the *Entry SNPs* panel will open the *Sequence editor* window of the corresponding sequence, with this position highlighted. If a sequence assembly is available in BioNumerics, the  will be active and selecting **File > Open assembler** () will open the assembly.

8.10.2.4.2 Using SNP filters

SNP filters and their parameters are accessed via the *Filters* panel. All SNP filters in BioNumerics work on the unfiltered (Total) SNP set, so the order in which the SNP filters are applied does not influence the results.

Based on their mode of action, we can classify SNP filters in one of two groups:

- **Position-based SNP filters** decide whether or not a position should be removed from the SNP matrix, based on per-position statistics (calculated over all sample sequences in the data set) or information stored with the reference sequence look. Since a position is either retained or removed as a whole, applying only position-based SNP filters will never result in an incomplete data set, i.e. there will be no values missing from the final SNP matrix.
- **Sequence-based SNP filters** decide per sample sequence whether or not a SNP should be removed from the SNP matrix, based on criteria only involving the sample sequence. A sequence based filter potentially creates a SNP matrix from which data are missing. It is for this reason that all sequence-based SNP filters have an additional ***Position retention threshold***, to decide what to do with the position in case one or more sample sequences have a SNP removed.

By default, the SNP filters from the SNP analysis template are listed and activated in the *Filters* panel. SNP filters can be added, removed, activated and inactivated from this panel.

To add a SNP filter to the list, select **Filters > Add filter...** () . This action opens the *Add SNP filters* dialog box (see Figure 8.10.7).

In the top left part of the *Add SNP filters* dialog box, all available SNP filters are shown in a tree-like structure. A SNP filter may have one or more parameters, which are displayed for the highlighted SNP filter on the right hand side of the dialog. A short description of the filter's action is given in the bottom part.

Following SNP filters are available:

- **Coverage:**
 - ***Absolute coverage***: Removes SNPs where the sequence is not sufficiently covered. The ***Minimum coverage*** thresholds can be specified as absolute numbers (i.e. number of reads mapped at a given position) for ***Forward***, ***Reverse*** and ***Total*** (forward + reverse) coverage. For an explanation of the ***Position retention threshold***, see below.
 - ***Relative coverage***: Removes SNPs where the sequence is not sufficiently covered. The ***Minimum coverage*** thresholds can be specified as percentages of the median coverage for ***Forward***, ***Reverse*** and ***Total*** (forward + reverse) coverage. The median coverage is used for calculation of the percentage (as opposed to the maximum or average coverage) in order to be less sensitive for outliers. The ***Position retention threshold*** is explained below.

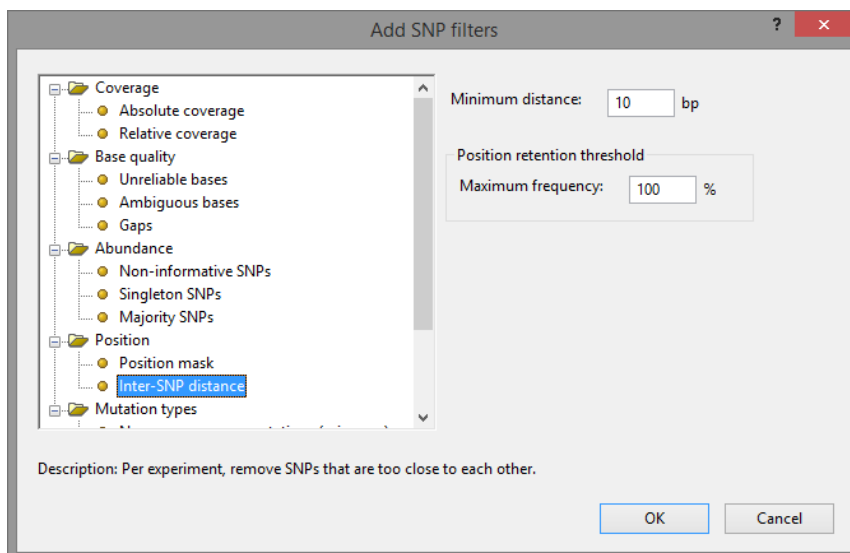


Figure 8.10.7: The *Add SNP filters* dialog box.

- **Gaps:** Removes positions with a percentage of gaps (i.e. bases that were not covered at all) that is higher than the specified **Maximum frequency**. For example, when entering a **Maximum frequency** of “0”, all positions with at least one gap will be removed.
- **Base quality:**
 - **Unreliable bases:** Removes positions with a percentage of bases that could not be called reliably (indicated with N) that is higher than the specified **Maximum frequency**.
 - **Ambiguous bases:** Removes a sequence position when the frequency of ambiguous bases exceeds the **Maximum frequency** threshold. Ambiguous bases are bases that cannot be uniquely called as A, T, G or C, i.e. all two-fold degenerated bases (R, Y, S, W, K or M), three-fold degenerated bases (B, D, H or V) and bases that could not be called at all (N). With the option **Take into account transitional bases** checked, IUPAC ambiguous bases that correspond to the base on the reference sequence or to an already present SNP will not be removed by the filter.
- **Abundance:**
 - **Non-informative SNPs:** Removes positions that are uninformative for discrimination between the entries in the analysis, i.e. positions that contain SNPs in comparison with the reference sequence, but where all sample sequences have the same base. This filter has no parameters.
 - **Singleton SNPs:** Removes singleton SNPs, i.e. positions where only one of the sample sequences has a different base call from the other sequences.
 - **Majority SNPs:** Removes a position when the majority base call is lower or equal to the specified **Maximum frequency**. With a **Maximum frequency** of 100%, this filter is identical to the **Non-informative SNPs** filter. With the same parameter slightly below 100% (to be more precise: $100(1 - \frac{2}{N}) \leq f_{max} < 100(1 - \frac{1}{N})$ with N the number of sample sequences), this filter behaves the same as the **Singleton SNPs** filter.
- **Position:**
 - **Position mask:** Removes positions based on a **Reference curve**, which can be selected from the drop-down list. Using an **Operator** (“>”: greater than, “<”: smaller than, “>=”: greater or equal, “<=”: smaller or equal, “=”: equal, or “!= ”: not equal) and a **Value** to compare to, the masking condition can be specified. See 8.10.2.4.3 on how to create reference curves for masking.

- **Inter-SNP distance:** Removes a SNP when another SNP occurs on the same sequence within a distance smaller than the specified **Minimum distance**. The rationale behind this filter is to remove evolutionary events that involve multiple nucleotides. The **Position retention threshold** is explained below.
- **Mutation types:**
 - **Non-synonymous mutations (missense):** Removes all mutations in intragenic regions that result in a change in amino acid sequence. This filter will be ignored in case the reference sequence has no annotation (the complete sequence will be considered as intergenic since no ORFs are found). The **Position retention threshold** is explained below.
 - **Synonymous mutations (silent):** Removes synonymous mutations, i.e. nucleotide changes in intragenic regions that do not result in a different amino acid after translation. This filter will be ignored in case the reference sequence has no annotation (the complete sequence will be considered as intergenic since no ORFs are found). The **Position retention threshold** is explained below.
 - **Intergenic mutations (silent):** Removes a position when it is located outside the Open Reading Frames (ORFs), as defined on the reference sequence. Should not be applied if the reference sequence has no annotation, since all positions will be removed in this case. This filter has no parameters.
 - **Transitions:** Removes transition mutations, i.e. the change of a purine nucleotide into another purine (A-G) or a pyrimidine nucleotide into another pyrimidine (C-T). The **Position retention threshold** is explained below.
 - **Transversions:** Removes transversion mutations, i.e. the change of a purine nucleotide into a pyrimidine or vice versa (A-C, A-T, G-C, and G-T). The **Position retention threshold** is explained below.

For all sequence-based SNP filters, a **Position retention threshold** can be specified additionally as a **Maximum frequency**. While the first parameter (if any) is determining whether a certain base is a SNP **per sequence**, the **Maximum frequency** determines what to do with the position **for all sample sequences in the analysis**: as soon as more sequences are filtered out then the **Maximum frequency** allows, the position is removed. See Figure 8.10.8 for an illustration.

Pressing **<OK>** in the *Add SNP filters* dialog box adds the SNP filter to the list in the *Filters* panel. By default, the filter is applied and the *SNP filtering* window updated accordingly. An extra column is added to the *SNP Positions* panel and *Entry SNPs* panel, indicating the effect of the filter.

The same filter can be added multiple times to the list. This can be useful, e.g. to combine several position masks.

One or more SNP filters can be removed by highlighting them in the list and selecting **Filters > Remove highlighted filters** (✖).

A SNP filter can be inactivated by unchecking its check box. This has in principle the same effect as removing the SNP filter, but the difference is that the effect of the inactivated SNP filter remains visible in the *Filters* panel and the *SNP Positions* panel, which helps to assess its effect. The command **Filters > Toggle rejected SNP visibility** (🔍) is also helpful to get a better understanding of the effect of individual SNP filters.

Filter settings for the highlighted filter can be accessed via **Filters > Filter settings...** (⚙️). If settings are available for the SNP filter, the *SNP filter settings* dialog box appears (see Figure 8.10.9).

Depending on the highlighted filters, different settings will be shown in this dialog. These settings are discussed for the *Add SNP filters* dialog box.

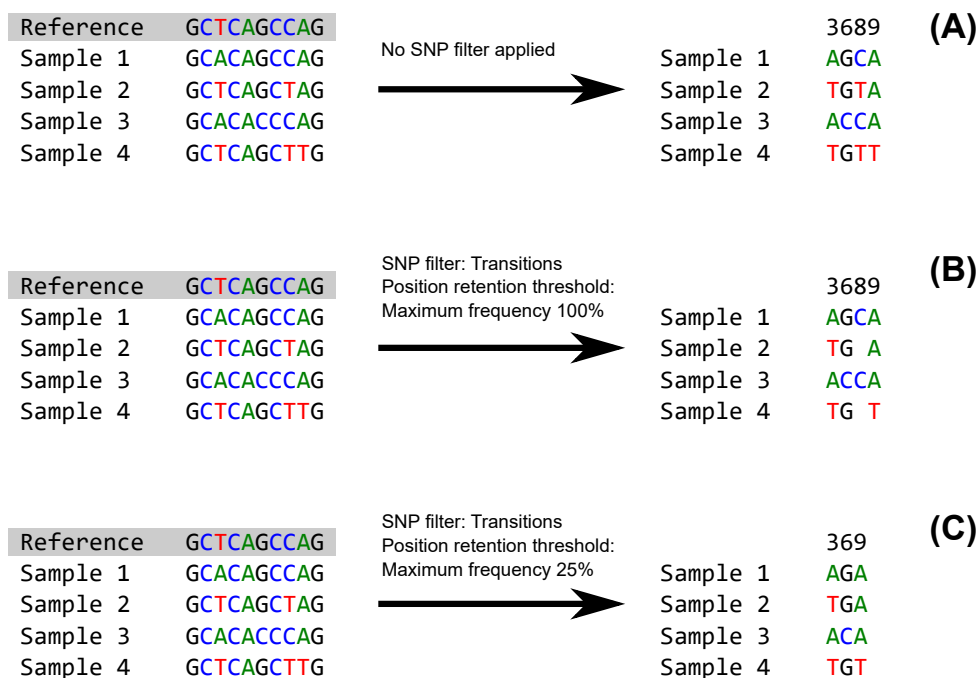


Figure 8.10.8: Illustrating the *Position retention threshold* for sequence-based SNP filters. **(A):** No SNP filter is applied and four positions have a SNP. **(B):** The sequence-based *Transversions* SNP filter is applied, which in this case removes the two C-T mutations on position 8. The *Maximum frequency* for the *Position retention threshold* is specified as 100% (i.e. no additional filtering applied on the position), resulting in values missing from the SNP matrix. **(C):** The *Transversions* SNP filter is applied, with a *Maximum frequency* for the *Position retention threshold* of 25%. This sequence-based filtering removes two out of four sequences on position 8, which is more than the specified 25%. Hence, position 8 is removed from the SNP matrix.

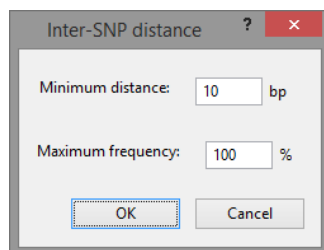


Figure 8.10.9: The *SNP filter settings* dialog box, here displayed for the *Inter-SNP distance* filter.

8.10.2.4.3 Position masks

Specific regions on the reference sequence can be left out of a SNP analysis via *position masks*, which are based on *reference curves*. In practice, the reference curve should first be calculated for the reference sequence, after which the reference curve can be selected from a *Position mask* SNP filter (see 8.10.2.4.2).

Reference curves are calculated in the *Sequence editor* window for the reference sequence. An easy way to access this information is by selecting *Edit > Open reference sequence* in the *Sequence type* window.

A reference curve that can be calculated for any reference sequence is a **non-uniqueness curve** (see 8.1.6.2.3). To have this curve calculated, select *Tools > Curves > Plot non-uniqueness...* in the *Sequence editor* window. A position mask based on a non-uniqueness curve allows to filter out problematic regions in a reference mapping.

A **custom mask curve** allows the creation a position mask that filters out or includes a predefined set of regions or positions. The start and end positions need to be defined in a mask file (see further). To create a mask curve, select **Tools > Curves > Create mask curve...**. This action calls the *Add mask curve* dialog box (see Figure 8.10.10).

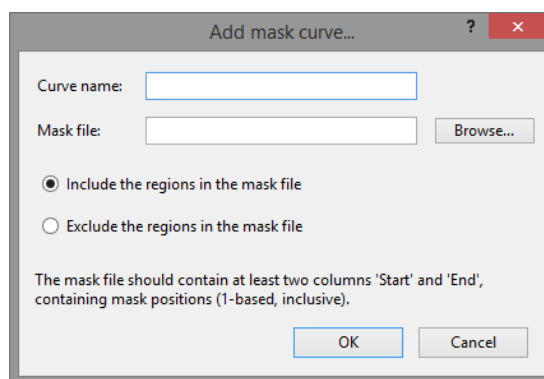


Figure 8.10.10: The *Add mask curve* dialog box.

Enter a **Curve name** that will be used in the custom curves list. If the curve already exists, the option will be provided to overwrite the existing curve.

A **Mask file** can be browsed for. A mask file is a CSV or tab-delimited text file with at least two columns 'Start' and 'End' that contain the start and end positions, respectively. Any additional columns in the mask file (if present) will be ignored.

With the option **Include the regions in the mask file** checked, the regions defined by the start and end positions (including the both positions) will be assigned a value of 100. All remaining positions in the sequence will be assigned a value of 0. The opposite is true when **Exclude the regions in the mask file** is checked.

Pressing <OK> will create the mask curve.

The mask curve will now be available in the *Add SNP filters* dialog box to base a **Position mask** filter on.

Optionally, the mask curve can be displayed in the *Sequence Viewer* panel with **Tools > Curves > [Curve name]**.

8.10.2.4.4 SNP templates

A set of SNP filters that provides satisfactory results can easily be re-applied to another data set if saved as a SNP analysis template. To save the current SNP filter list from the *Filters* panel (with the parameters as specified with each filter) as a template, select **File > Save filters as template...**

In the *Save analysis template* dialog box, a **Name** and **Description** can be entered for the template. The template will be saved as in the **User-defined** category.

The current SNP filters in the *Filters* panel can be overwritten by the set of filters loaded from a template via **File > Load filters from template...**

The *Load analysis template* dialog box contains an overview of all predefined and user-defined templates the user can choose from.

Deleting obsolete SNP templates can be done with **File > Remove filters template...**

The *Remove analysis templates* dialog box lists all available templates in the database. Highlight one or more templates and press <OK> to remove them. Please note that this action cannot be undone.

The current list of SNP filters can be exported to file in order to exchange the SNP analysis template between

databases, computers, users, etc. with **File > Export filters to file....**

In the *Export template* dialog box, the user can name the file and select its location by pressing **<Browse>**. Templates are saved as XML files.

To import a previously exported SNP analysis template, select **File > Import filters from file....**

In the *Import template* dialog box, browse for the location of the template file and press **<OK>**.

The SNP filters from the template file will now be loaded in the *SNP filtering* window. For future use, this set of SNP filters should be saved as a SNP analysis template.

8.10.2.4.5 Exporting a SNP matrix to the Comparison window

The final SNP matrix (i.e., those SNPs that are retained by all applied SNP filters) can be exported to the *Comparison* window with **File > Export to comparison...** (📁). When the SNP analysis was started from the *Main* window (see 8.10.2.2), a new *Comparison* window will be opened and the SNP matrix will be added as an aspect to the sequence type with the name **SNPs**.

With the SNP analysis initiated from the *Comparison* window (see 8.10.2.3), selecting **File > Export to comparison...** (📁) will open the *SNP Data name* dialog box.

The *SNP Data name* dialog box allows the user to specify a name for the sequence type aspect in which the SNP matrix will be saved. The default suggested name is **SNPs**.

The SNP matrix will always be added to the comparison from which the SNP analysis was started, even if several comparisons are open. By specifying a different name, several SNP matrices can be saved as different aspects for the same sequence type. This makes it easy to calculate and compare cluster analyses that start from different SNP templates.

8.10.3 Analyses on filtered SNP matrices

8.10.3.1 Introduction

After filtering, a SNP matrix is exported to the *Comparison* window (see 8.10.2.4.5). The rows in this SNP matrix correspond to the entries and the columns represent the positions where at least one SNP is retained after filtering. Each cell in the matrix therefore contains the base call for the entry at that specific position. Note that the SNP matrix can have missing values.

In the *Comparison* window, the SNP matrix is available as a *character aspect* of the original sequence type.

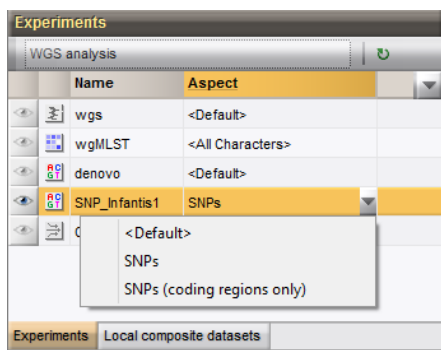


Figure 8.10.11: Detail of the *Experiments* panel, showing two SNP matrices as character aspects of a sequence experiment type.

SNP matrices have the same display options in the *Experiment data* panel as composite data sets (see 11.2.3). The most relevant visualization for SNPs and therefore the default one is to show character mappings and colors (see Figure 8.10.12).

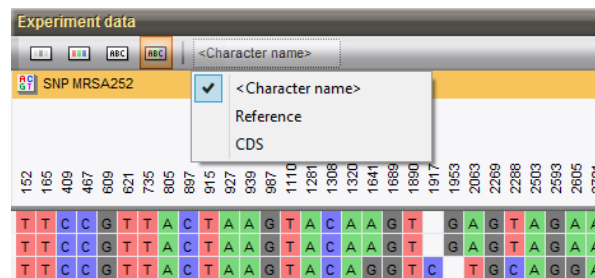


Figure 8.10.12: Detail of the *Experiment data* panel, illustrating the display options for SNP matrices.

By default, character names are shown in the header of the *Experiment data* panel. The character name corresponds to the position number on the reference sequence. Via the drop-down list in the caption (see Figure 8.10.12), the nucleotide on the reference sequence or the name of the CDS (taken from the annotation of the reference sequence, if available) can alternatively be displayed for each SNP position.

8.10.3.2 Calculating a clustering based on a SNP matrix

If a SNP filtering (see 8.10.2) was already performed, simply select the SNP character aspect from the 'Aspect' drop-down list (see Figure 8.10.11) and use **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**...

The *Similarity coefficient* wizard page that appears has the same set of coefficients as available for composite data sets (see 11.2.2 for more information). Only coefficients from the **Multi-state** category make sense for a cluster analysis on SNP data.

Alternatively, a SNP filtering and a cluster analysis on the obtained SNP matrix can be combined in a single action:

Highlight a reference mapped sequence type in the *Experiments* panel and select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**... The *Comparison settings* wizard appears (see Figure 8.10.13).

The only available parameter is the **Template** to use for **SNP filtering** (see 8.10.2.4.4 for more information). The selected SNP template will be applied, the resulting SNP matrix added as a character aspect of the sequence type and a dendrogram calculated on this information.

The similarity coefficient used is the **Categorical (SNPs)** coefficient (see 6.2.2). The **Scaling factor** is automatically optimized: it is set to "1" by default, to "10" if the data set contains more than 200 SNPs and to "100" with more than 2000 SNPs.

Pressing <Next> in the *Similarity coefficient* wizard page will display the second page, which groups the options related to the clustering algorithms. This step is discussed in 13.2.6.

Pressing <Finish> will calculate the dendrogram.



An error message will be produced in case the selected experiment type is not reference mapped. See 8.1.2.2 for instructions on converting an existing sequence type.

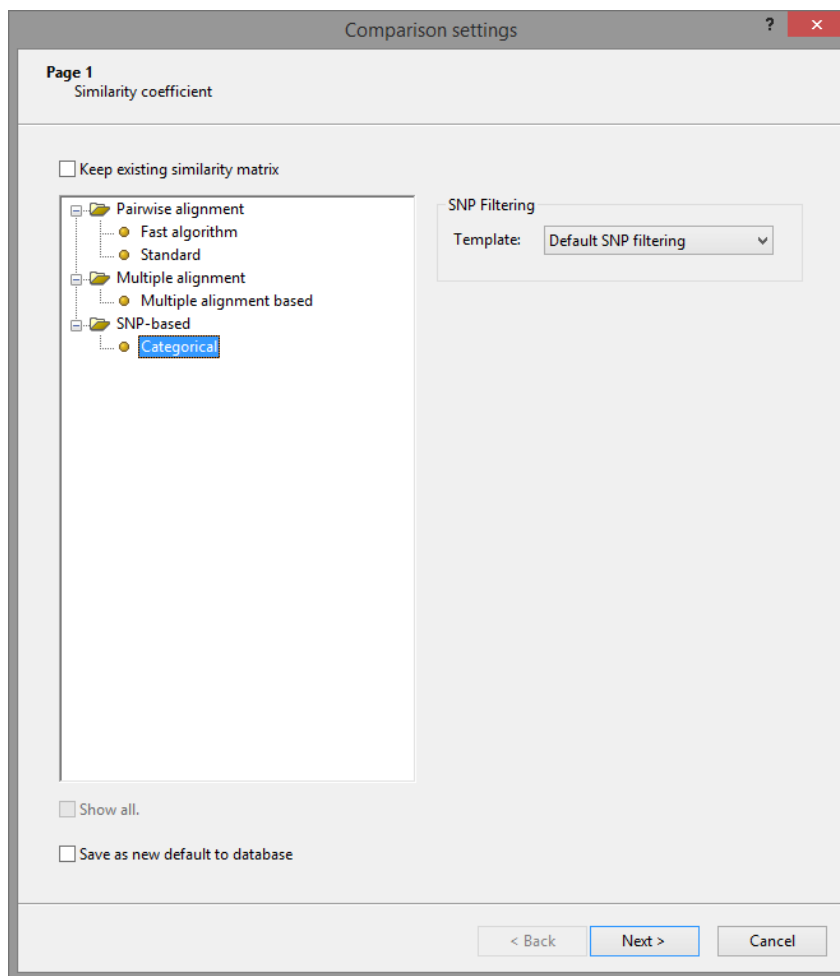


Figure 8.10.13: The *Similarity coefficient* wizard page, SNP-based clustering.

8.10.3.3 Exporting SNP data

In some scenarios, e.g. to run custom scripts on filtered SNP matrices, it might be useful to export SNP data as a character set.

In the *Comparison* window, a SNP matrix can be exported for the active experiment type with **File** > **Export** > **Export character data....** See [13.3.13](#) for more detailed information about this functionality.

Alternatively, the data displayed in the *SNP matrix* panel of the *SNP filtering* window can be exported using the column properties button (🔍) and selecting e.g. **Save content to file**.

Part 9

Sequence read sets

Chapter 9.1

Setting up sequence read set experiments


9.1.1 Introduction

A sequence read set in BioNumerics is designed to store large sets of short sequence reads, generated by high-throughput sequencers such as Roche/454[®] or Solexa/Illumina[®]. Sequence read sets typically need to be pre-processed first (e.g. demultiplexing, trimming, ...) and can then be assembled into sequences (see 18) or used for characterizing microbial communities via deep sequencing (see 19). Using keywords, sequence read sets can also be clustered directly (see 9.3).


Essentially, there are two ways to import sequence read sets:

- **As files:** The default import method (see 9.1.4) stores the sequence reads in the BioNumerics database, either in the source files directory or optionally in the relational database.
- **By link:** This import method (see the WGS tools plugin, Chapter Importing sequence read sets for the Calculation Engine) becomes available after installation of the *WGS tools plugin*. Only the link to the actual data is stored (e.g. accession number and download site), keeping the BioNumerics database lightweight. Storage by link is recommended to use sequence read sets with the calculation engine.

9.1.2 Creating a new sequence read set experiment

To create a new sequence read set type, highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** () to display the *Create a new experiment type* dialog box (see Figure 9.1.1).



Please note that, to be able to work with sequence read sets, the Sequence data module () needs to be present in your BioNumerics configuration.

In the *Create a new experiment type* dialog box, click on **Sequence read set type** and press <OK>. This will display the *Create new experiment type* dialog box (see Figure 9.1.2).

This dialog prompts you to enter a name for the new sequence read set experiment. Pressing <OK> will automatically add the created experiment type to the BioNumerics database.

General sequence read set experiment type settings are not queried for when creating the experiment type, so they need to be modified through the *Sequence read set experiment type settings* dialog box (see 9.1.3.2).

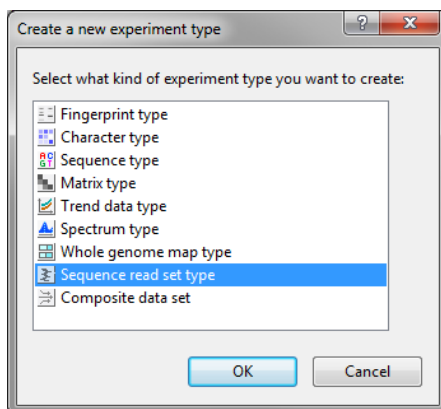


Figure 9.1.1: The *Create a new experiment type* dialog box.

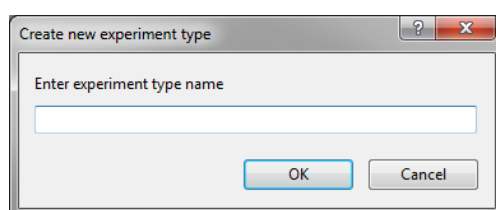



Figure 9.1.2: The *Create new experiment type* dialog box.


9.1.3 Editing a sequence read set experiment

9.1.3.1 Sequence read sets data type window

All settings that are relevant for a certain sequence read set experiment type can be accessed through its *Sequence read set type* window (see Figure 9.1.3). This window can be called from the *Main* window by clicking on the sequence read set experiment type in the *Experiment types* panel and selecting **Edit > Open highlighted object...** (, **Enter**) or simply by double-clicking on the sequence read set experiment type.

In the *Sequence read set type* window, the *Comparison settings* panel (top panel in default configuration) displays the comparison settings and the *Crosslinks* panel and *Attachments* panel (bottom) give an overview of the experiment cross-links and attachments, respectively.

9.1.3.2 General sequence read set experiment settings

Via **Settings > General settings...** (), the *Sequence read set experiment type settings* dialog box is called (see Figure 9.1.4).

When the option **Save in database** is checked, the read sets are stored in the connected database. Please note that this may fill up your database very quickly as these data sets are typically beyond the reach of small database systems e.g. Access or SQL Server Express. By default, the option **Save in database** is unchecked, which implies that the imported sequence read set data are stored as separate files in the Source files location, which is the general place to store source files such as TIFF images, CRV curve files, sequence trace files and also sequence read sets (see 3.7.2).

When importing sequence read sets, keyword profiles are being calculated for a range of keyword lengths ($n=3-9$). Although multiple keyword frequencies are calculated upon import, the default keyword length will be used for analysis e.g. calculating similarity values and visualization of the sequence read sets in the *Comparison* window. The **Default keyword length** can be defined from the drop down list. When the option

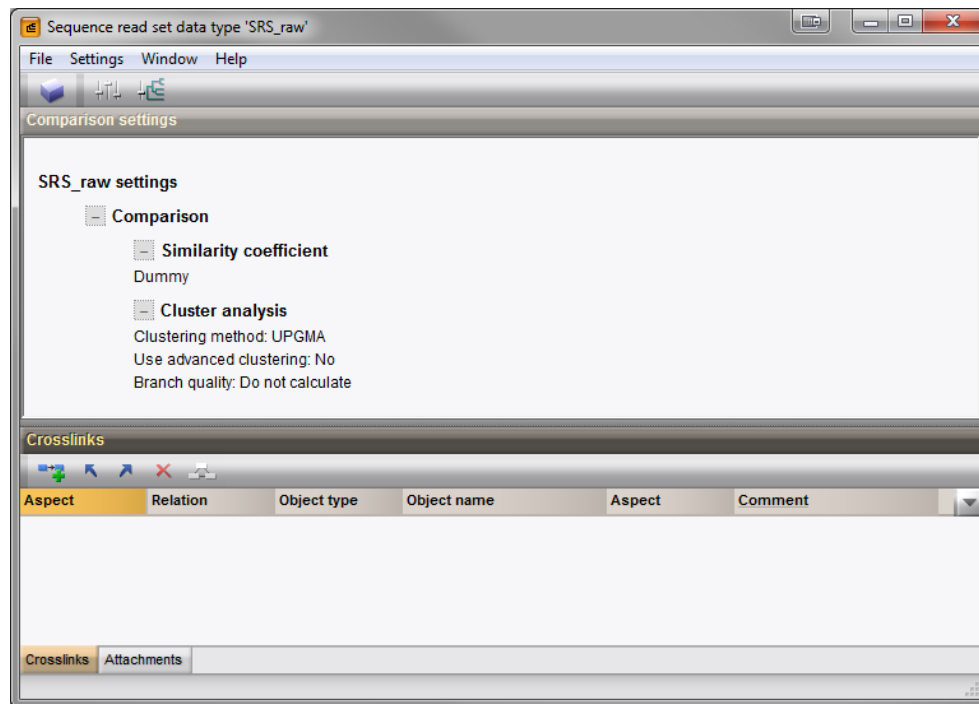


Figure 9.1.3: The *Sequence read set type* window.

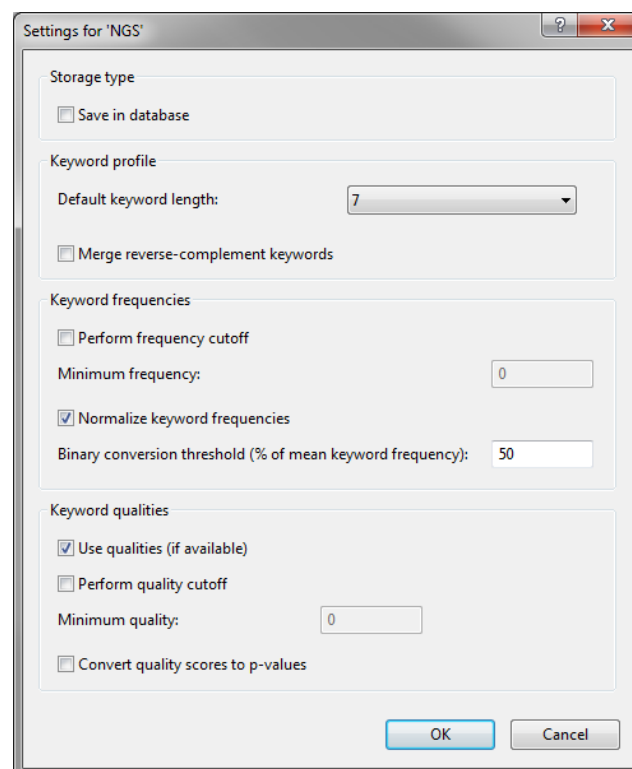


Figure 9.1.4: The *Sequence read set experiment type settings* dialog box.

to **Merge reverse-complement keywords** is checked, the frequencies of the keywords that are each others reverse-complement are merged, and displayed as a single keyword frequency.

When the option **Perform frequency cutoff** is enabled, only the frequencies exceeding the defined **Minimum frequency** are used for analysis. One can use the option **Normalize keyword frequencies** to normalize on

the total number of counts in the sequence read set i.e. each frequency is divided by the total number of counts in the sequence read set at hand. After normalization, the sum of the frequencies in one sequence read set is 1.

When calculating similarity scores between different sequence read sets based on presence/absence of frequencies, a **Binary conversion** needs to be done from numerical values to binary values, before one of these coefficients can be applied. Any character value above the **Binary conversion threshold** will be converted to presence. The **Binary conversion threshold** is expressed as a certain percentage of the mean keyword frequency.

When checking the option **Use qualities**, the individual base qualities of the nucleotides that determine the keywords are used to determine an average keyword quality. When taking into account these keyword qualities, a quality cutoff value can be set under **Perform quality cutoff**. This cutoff value is expressed as the absolute **Minimum quality**, defined as Phred score (range: 0-63).

With the option **Convert quality scores to p-values**, the log-scale Phred quality scores are converted to p-values and as such, taken into consideration when calculating e.g. similarity values between sequence read sets when quality values are taken into account. In this way, the large quality difference in Phred scores of e.g. 30 and 50, is dramatically reduced to a rather small quality difference when using the p-values.

9.1.3.3 Sequence read set comparison settings

In the *Sequence read set type* window, the comparison settings defined for the sequence read set are shown in the *Comparison settings* panel (see Figure 9.1.3). These settings can be accessed with **Settings > Comparison settings...** (🔧) in the *Sequence read set type* window, but also in the *Comparison* window. See 9.3 for a detailed explanation on cluster analysis of sequence read sets.

9.1.4 Importing sequence read sets as files

9.1.4.1 Introduction

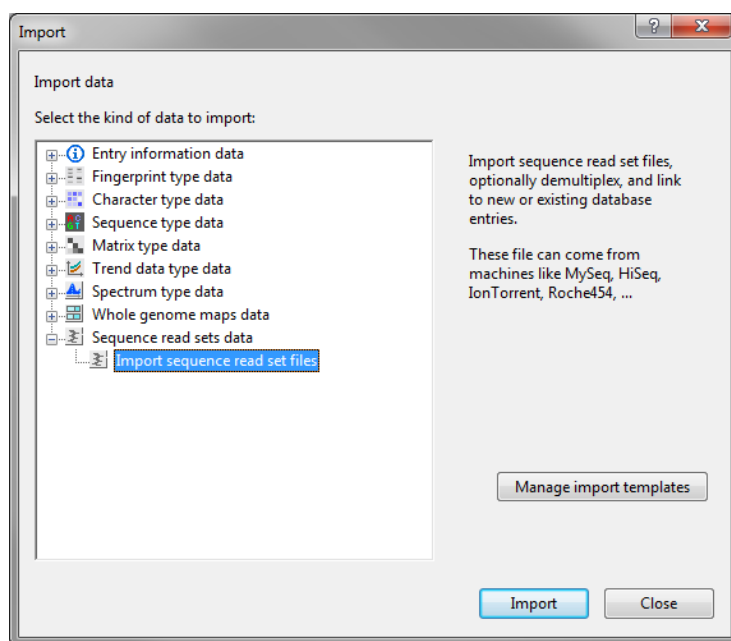


Figure 9.1.5: Importing sequence read sets via the *Import* dialog box.

When selecting the option to import sequence read sets to the BioNumerics database (see Figure 9.1.5), a multitude of different file types can be imported. Depending on the file extension and the content of the file, the software will automatically detect which file type is imported and how the import should be processed. Using this import functionality, sequence read sets can be imported from the following formatted files:

- Roche/454[®] sequence files, with extensions .fna (sequence information) and .qual (corresponding sequence quality information). These files are typically used for importing data derived from 454 GS systems (Roche Applied Science).
- FASTA files, with extensions .fasta, .fna, .ffn, .faa or .txt, and
- FASTQ files, with extensions .fq, .fastq or .txt. Both FASTA and FASTQ formatted files can be used to import any base space encoded data from e.g. the Genome Analyzer, MiSeq or HiSeq (Illumina); the SOLiD systems (Applied Biosystems); the Ion Torrent PGM (Life Technologies); the PacBio RS (Pacific Biosciences) and the HeliScope system (Helicos BioSciences).

Imported sequence read sets are linked to new or existing database entries. Optionally, the file name and multiplex identifier, if used, can be stored in entry information fields.

9.1.4.2 Importing sequence read sets from a FASTA file

The FASTA format is the most commonly used format for sequence read data without quality information. Each sequence in the FASTA file has a header line beginning with a >, followed by the sequence data which may span multiple lines. An example of two sequences in FASTA format is given here:

```
>gi|strain14587|Streptococcus pyogenes|emm gene|
ATAAGGAGCATAAAAATGGCTAAAAACAACACGAATAGACACTATTCGCTTAGAAAAAT
>gi|strain14516|Streptococcus pyogenes|emm gene|
ATAAGGAGCATAAAAATGGCTACCAACAACACGAATAGACACTATTCGCTTGAAAAAT
```

9.1.4.3 Importing sequence read sets from a FASTQ file

The FASTQ format is the most commonly used format for sequence read data including Phred quality information on the reads.

Each sequence in the FASTQ file has a header line beginning with a '@'. The second line contains the sequence data on a single line. Next, a second header line but now beginning with a '+' should be identical or empty, apart from the '+'. The next line contains the quality characters, representing the Phred quality of the corresponding nucleotide. The length of the sequence is identical to the length of the quality string. Quality scores are expected to be in the NCBI/Sanger format, using Phred scale quality encoding using ASCII 33 to 93.

An example of two sequences in FASTQ format is given here:

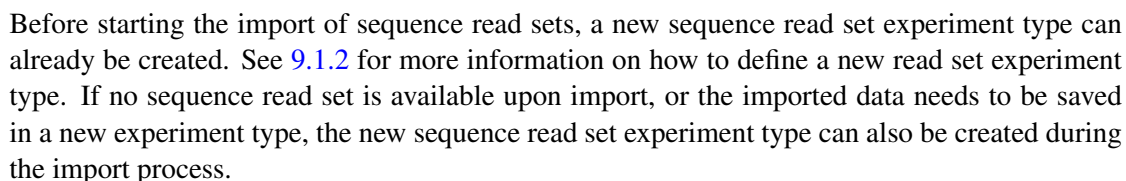
```
@strain59763|27Q4V:4:87
GACGTGGTGCTCTAATAGAGGAATAGATCTTTATAATAAATGGACAGAATCTAGCTCATGATCTAATGTTTACATCATT
+strain59763|27Q4V:4:87
2=====2=====288777-/04-///433*4))')33*33/03333-+++3-++))+++222/2)))$)++++++&
@strain59943|27Q4V:4:119
TGCGGCCGGGGCAGCCAGCGGTCCAGCATGGACAGGTCCGCCCCCGGGCGGGCAAACCAGAGTTTGTGCGGCTCCTTCGTCCA
+strain59943|27Q4V:4:119
88//+/++++&/8880858805525-1155055550550/----&)+(100+111+10555222*2-)-(++0/0()+)+(+
```

The sequence read data from 454 GS systems (Roche Applied Science), comes in two files per data set: the sequence data is saved in a FASTA file (extension .fna) and the corresponding quality scores, equivalent to the Phred score as known from Sanger sequencing, in a .qual file. The reads in the latter file have the entities in the same order as in the corresponding .fna file. For each base of the read, the quality value is given. The length of the sequence is identical to the length of the quality string.

```
>964853.9468.4865 length=115 uaccno=TRAESEQ01PRIG9
GCGTTTTTTCGTTTTTCGTTTTTCGTTGCGTTATAACCCAACTAAGCCGGAGGTAA
AAGGTAGTCTCTCAGACCTATGATTTTGATAAATTCATTGACTCTTCTCAGCGT
>268753.9768.2719 length=110 uaccno=TRAESEQ01PR08F
CGCAAAAACGCAAAACGCAACGCAACGACCAGCCTATGCGCCTGGTCTGTACACCGT
TGATCTGTCTCTTTCAAAGTTGGTCAGTTCGGTTCCTTATGATTGACCG
```

```
>964853.9468_4865 length=115 uaccno=TRAESQ01PRIG9
28 28 28 36 32 22 13 5 27 28 27 36 32 20 8 27 28 28 35 31 14 28 28 27 33 26 28 28
28 33 25 27 26 33 25 35 30 13 33 25 32 24 27 33 25 27 32 24 33 26 27 33 25 33 25 36
32 22 14 6 33 25 28 23 26 27 27 28 25 28 27 28 25 27 33 25 28 27 26 28 25 36 32 19
6 28 27 28 35 31 15 33 25 27 27 28 27 27 32 24 27 22 27 28 27 33 25 27 27 27 27 27
25 28 18
>268753.9768_2719 length=110 uaccno=TRAESQ01PRO8F
19 27 27 36 32 22 14 6 24 28 27 36 33 21 10 26 27 28 35 31 14 27 27 28 33 25 27 28
28 28 33 25 27 22 33 25 27 28 27 27 25 27 33 25 24 32 26 27 28 28 27 27 27 23 26 33
25 28 31 23 28 28 26 28 25 26 27 32 24 27 28 34 30 12 20 35 31 15 27 32 24 32 26 25
28 28 25 31 22 26 32 24 32 23 34 30 12 31 23 26 27 27 26 29 20 26 25 33 26 26
```

9.1.4.5.1 Background



9.1.4.5.2 Import sequence read sets: Input

Pressing the **<Browse>** button allows you to select the file(s) that you want to import. These files can be located on your computer, external drive or on a network location. Note that you can import multiple files at once. Just below the file list, a brief summary on the selected files is displayed and updated. This summary indicates how many files of a specific file format were found, and whether or not corresponding quality files (e.g. the .qual files for the .fna files for 454 data) or files containing paired-end data are found (e.g. Illumina paired-end data files, _1 and _2 files having the same file name). To obtain a detailed insight in which files will be linked to each other during import, one can launch the *File selection details* dialog box by pressing the **<Show details...>** button.

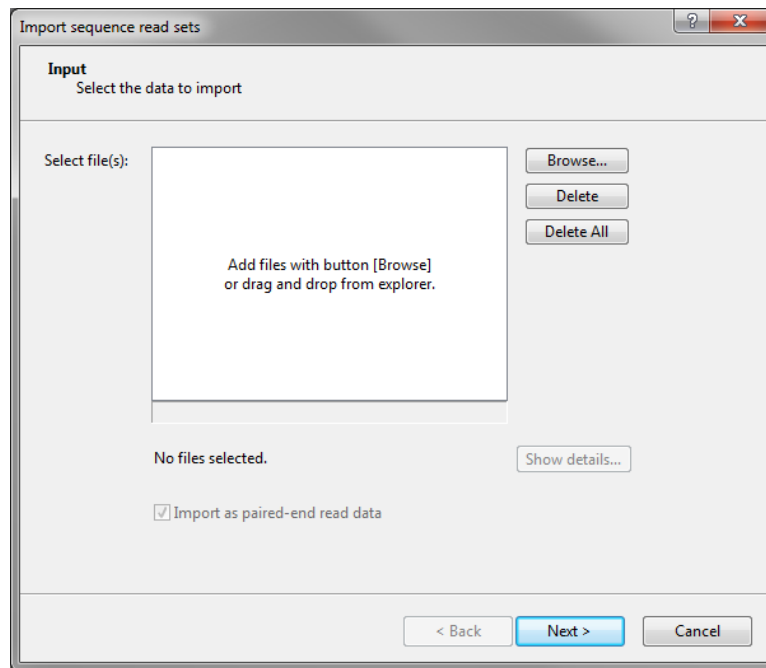


Figure 9.1.6: The *Import sequence read sets* wizard: *Input* wizard page.

Deleting one or multiple files from the import list can be done by selecting the items from the list and pressing the **<Delete>** button. By pressing the **<Delete all>** button, all files present in the import list are deleted at once.

Checking the option **Import as paired-end read data** ensures that the files are checked for the presence of paired-end data. Files that contain paired-end data are recognized by the same file name except for the last two characters, which should be **.1** and **.2**. If this option is checked, sequence reads will obtain the status of paired-end reads and this information is also saved to the experiment in the database.

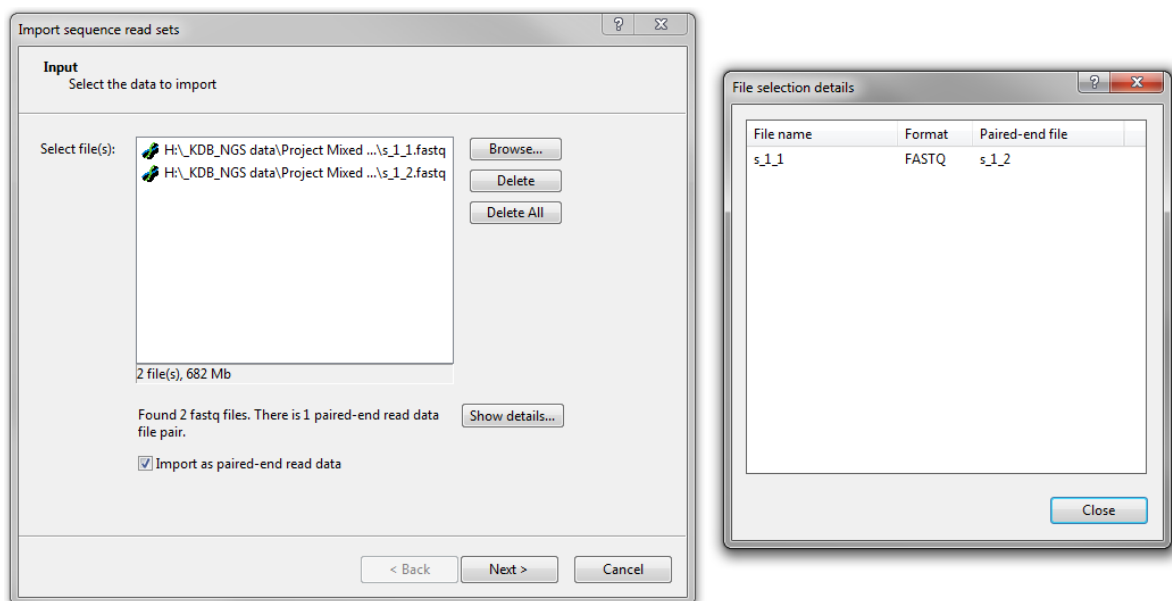


Figure 9.1.7: Import sequence read sets: File selection details for Illumina paired-end files.

For the import as presented in Figure 9.1.7, two paired-end Illumina files were selected. In the detailed information, each line indicates which files will be combined to create paired-end data.

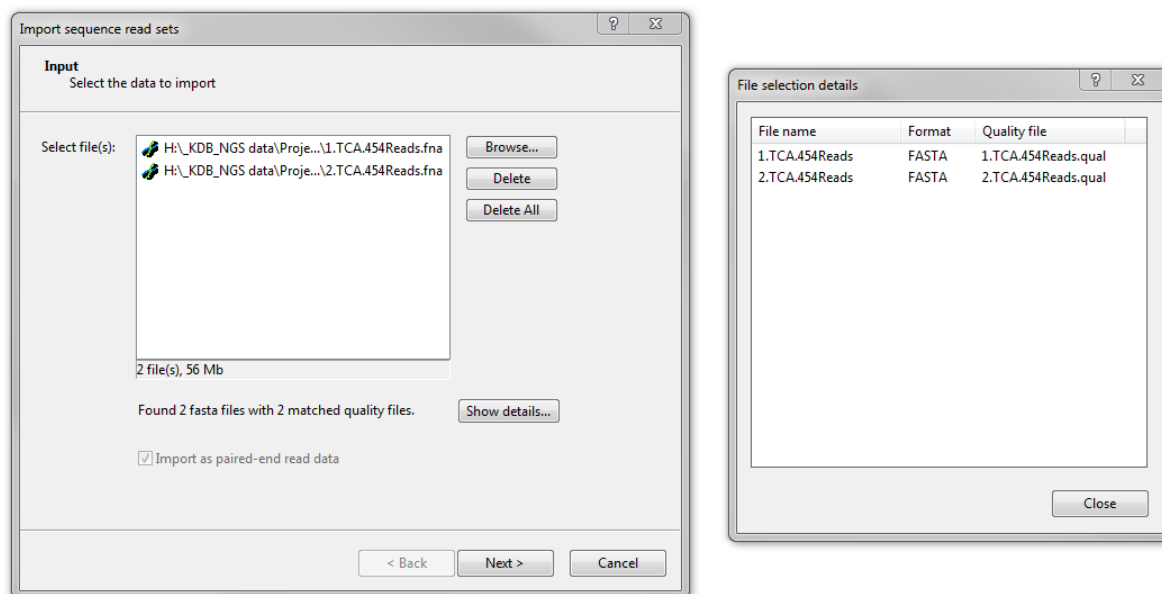


Figure 9.1.8: Import sequence read sets: File selection details for 454 .fna and .qual files.

For the import as illustrated in Figure 9.1.8, one .fna 454 file was selected. When a .fna file is selected, the software will automatically screen in the same folder for files having the same filename but with the .qual extension. If such a file is found, the files are linked upon import as displayed in the *File selection details* window.

9.1.4.5.3 Import sequence read sets: Qualities

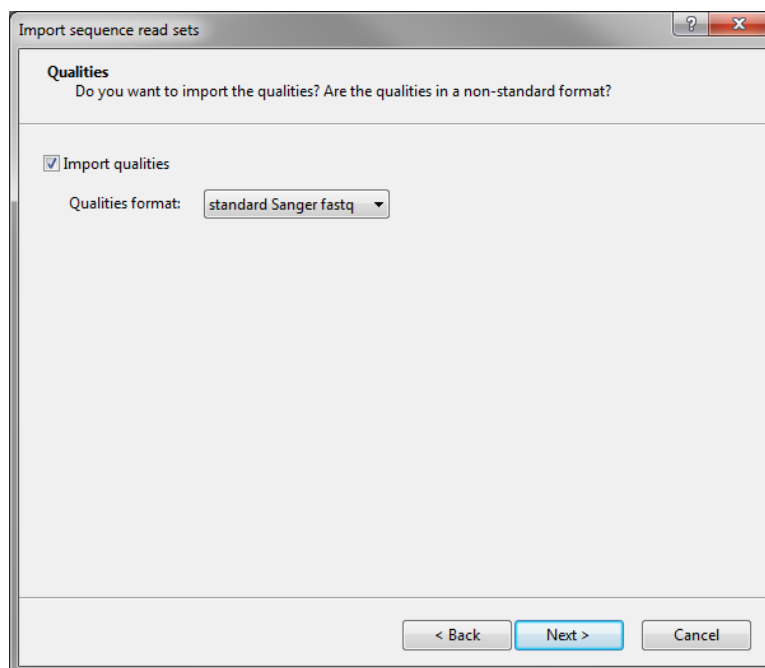


Figure 9.1.9: The *Import sequence read sets* wizard: *Qualities* wizard page.

In the *Qualities* wizard page (see Figure 9.1.9), you are requested to import the base qualities along with the sequence reads. When the option **Import qualities** is checked, the quality format used should be specified.

Three different quality conversions are present:

- Default, the Phred scale quality encoding using ASCII 33 to 93 is selected. This is also the standard encoding for FASTQ files from NCBI/Sanger or Illumina 1.8 and later.

For early Illumina FASTQ data formats, two more options are present in the drop down list:

- **Illumina pre version 1.8**, which uses a Phred scale ASCII 64 to 104, and
- **Illumina pre version 1.3**, which uses the early Illumina quality scale (quality values ranging from -5 to 40) using ASCII 59 to 104.

Changing any of the quality formats, changes the quality conversion formula that is used upon import of the sequence read set.

9.1.4.5.4 Import sequence read sets: Demultiplexing

During import, it is possible to demultiplex the data (see Figure 9.1.10). Multiplexing is used for sequencing multiple samples together within the same sequencing lane. The technique is typically used to sequence e.g. smaller bacterial genomes or to analyze multiple metagenomics samples. In general, it allows to analyze a much larger number of samples in a single run without drastically increasing cost and time or compromising on sequence coverage of the samples. Demultiplexing implies that a unique identifier tag, or barcode, that is specific for each library, is traced in the sequence reads, which are then filtered out accordingly, creating samples for individual downstream analyses.

Import sequence read sets

Demultiplexing
Do you want to demultiplex the sequences?

☒ Perform demultiplexing [Preview...](#)

Barcode location	Paired-end sequence
Location: At the beginning	Location: At the end
Bases before barcode: <input type="text" value="0"/>	<input type="text" value="0"/>
Size of barcode: <input type="text" value="8"/>	<input type="text" value="8"/>
Bases between barcode and sequence: <input type="text" value="0"/>	<input type="text" value="0"/>

Barcode filtering

Minimum number of sequences:

Minimum relative frequency:

< Back Next > Cancel

Figure 9.1.10: The *Import sequence read sets* wizard: *Demultiplexing* wizard page.

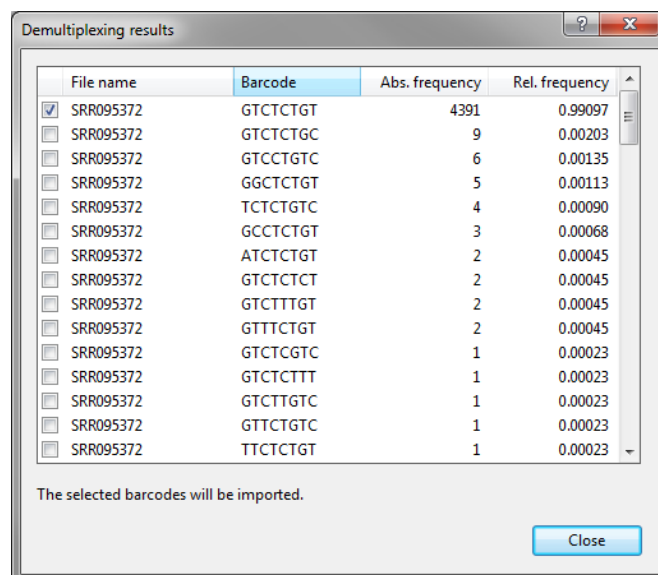
Leave the demultiplex option unchecked to start the import without demultiplexing the data.

If you are dealing with multiplexed data sets, check the option **Perform demultiplexing** and the demultiplex settings become active. Once the properties for the multiplex identifier tags are set, the read sequences will be screened and for each unique identifier tag, a sample will be created linking all sequence reads by their barcodes.

First, the **Barcode location** needs to be specified. The location can be set at the beginning or at the end of the read sequence. For paired-end data, one can also specify to have no barcode in a specific part of the linked paired-end reads. Barcodes have a fixed sequence length, and are followed by a linker sequence of fixed length as well. Optionally, a linker can be present before the barcode. All these sizes can be set in the fields **Size of barcode**, **Bases between barcode and sequence** and **Bases before barcode**, respectively.

As barcodes are defined by the barcode size, the software will also detect barcodes that differ in one or multiple bases from the actual sample barcode. This will typically result in real biological samples having relative read frequencies of 99% and some artificial samples having low relative read frequencies (see also Figure 9.1.11). As you don't want to save the latter samples to the database because they just represent sequencing errors within the barcodes and not actual samples, a system of **Barcode filtering** has been implemented.

Once the parameters are set, pressing <Next> will start the demultiplexing of the file and continues the import of the read files. However, before proceeding, it might be interesting to have a **Preview** of the demultiplexing itself. To open the demultiplexing preview, press the <Preview> button on top of the dialog page.



File name	Barcode	Abs. frequency	Rel. frequency
<input checked="" type="checkbox"/> SRR095372	GTCTCTGT	4391	0.99097
<input type="checkbox"/> SRR095372	GTCTCTGC	9	0.00203
<input type="checkbox"/> SRR095372	GTCTCTGC	6	0.00135
<input type="checkbox"/> SRR095372	GGCTCTGT	5	0.00113
<input type="checkbox"/> SRR095372	TCTCTGTC	4	0.00090
<input type="checkbox"/> SRR095372	GCCTCTGT	3	0.00068
<input type="checkbox"/> SRR095372	ATCTCTGT	2	0.00045
<input type="checkbox"/> SRR095372	GTCTCTCT	2	0.00045
<input type="checkbox"/> SRR095372	GTCTTTGT	2	0.00045
<input type="checkbox"/> SRR095372	GTTTCTGT	2	0.00045
<input type="checkbox"/> SRR095372	GTCTCGTC	1	0.00023
<input type="checkbox"/> SRR095372	GTCTCTTT	1	0.00023
<input type="checkbox"/> SRR095372	GTCTTGTC	1	0.00023
<input type="checkbox"/> SRR095372	GTTCTGTC	1	0.00023
<input type="checkbox"/> SRR095372	TTCTCTGT	1	0.00023

The selected barcodes will be imported.

Close

Figure 9.1.11: The *Demultiplexing results* dialog box.

When demultiplexing is completed, the *Demultiplexing results* dialog box will appear, indicating the file name of the file that contains the multiplexed data, the detected barcodes, and the absolute and relative frequencies of the reads assigned to a specific sample. When no barcode filtering is applied, all demultiplexing results are selected to be saved to the database. When a **Minimum number of sequences** or **Minimum relative frequency** threshold is defined in the **Barcode filtering**, only the barcoded samples surpassing these thresholds are selected to be imported in the database.



When you already have a fully annotated Excel or text file of your sample data including the barcode information, a good practice is to import this information first in the BioNumerics database, and link the detected barcode information identified during import to the database field that contains your barcode. By doing this, only a limited selection of demultiplexed sequence read sets present in one file will be imported or updated during import.

As a result of the demultiplex analysis, the demultiplexed samples will be saved to the database as individual entries and can be further used to a variety of applications such as genome assembly, mutation searches, MLST analysis or metagenomics analysis.

9.1.4.5.5 Import sequence read sets: Import template

To specify which sample data and metadata is saved in the database, one needs to create an import template. Once created, this import template remains available in the database. Import templates can be edited at any time and can even be exported as an xml file and imported in other databases or shared by colleagues. In this way, import templates are proven to be very valuable when routinely importing updated data files or similar data formats. More information on import template management can be found in [3.3.5.4](#).

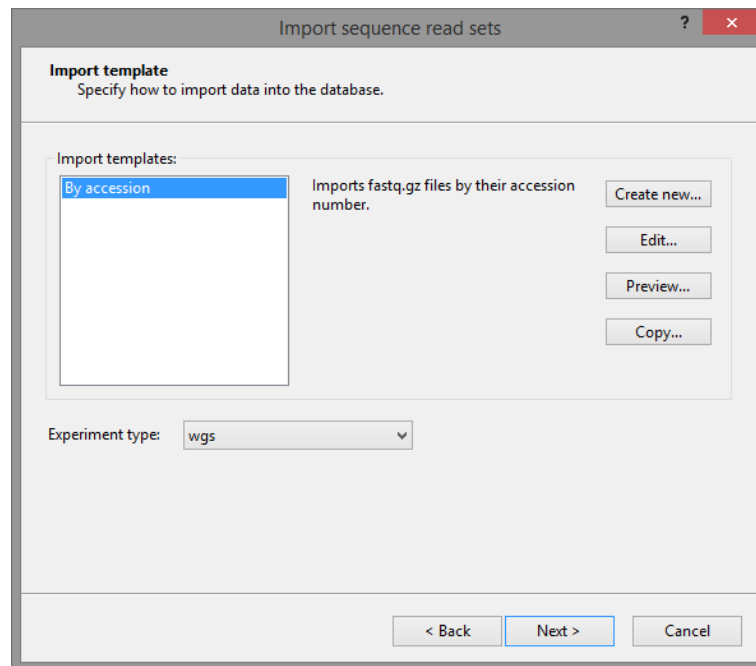


Figure 9.1.12: The *Import sequence read sets* wizard: *Import template* wizard page.

In the following section, we will only briefly discuss the creation of a new import template, typically used for the import of sequence read sets. From the *Import template* dialog page, one can create a new import template or edit an existing template. When editing a template, only two data sources are available: Barcode and File name. If demultiplexing was performed, the barcodes detected in the files can be saved to a database information field as well as the file name.

Typically all rows in the grid can be associated with a new or existing entry information fields. Initially the rows are not linked to any information in the database (the **Destination type** and **Destination** for all rows is set to **<None>**) (see Figure 9.1.13). Specifying a *destination* for one or more selected rows can be done by pressing the **<Edit destination>** button or by double-clicking the *Source type*. This action pops up a new dialog box prompting for the new destination for the selected row(s).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

The information of the selected rows can be linked to (see Figure 9.1.14):

- A **Sequence read set data type**.
- The default information field **Key**.

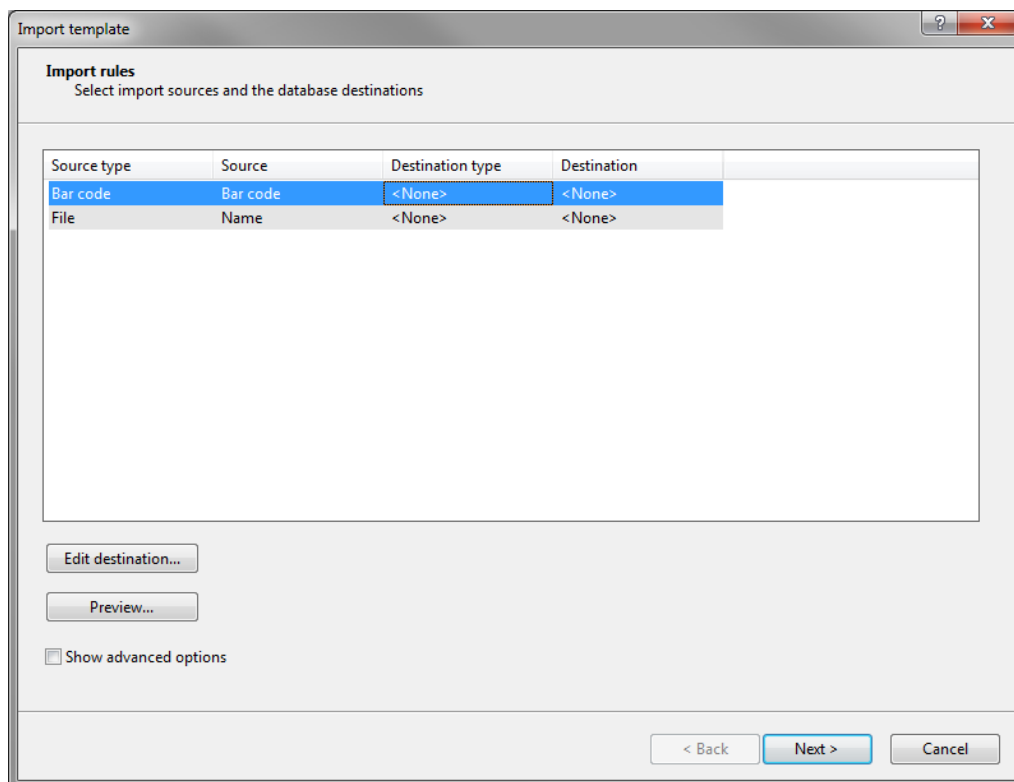


Figure 9.1.13: Import sequence read sets: Import template rules.

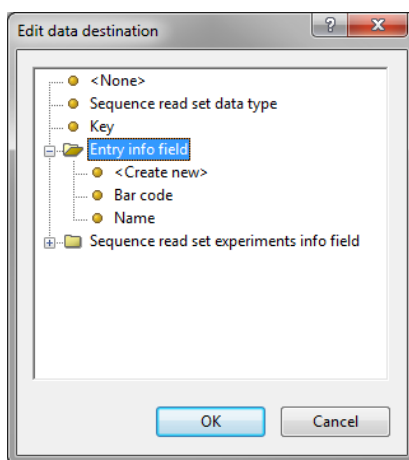


Figure 9.1.14: Import sequence read sets: Import template rules: Edit data destination.

- A new or existing non-default entry information field (select the *<Create new>* option or an existing field under the topic *Entry info field*, respectively).

If a row is linked to a new entry information field, a new dialog box pops up when confirming by the *<OK>* button. This new dialog box prompts for the entry information field name. A default name is suggested by the software, but can be overwritten if desired. Pressing the *<OK>* button creates the entry information field in the database, and updates the information in the *Destination type* and *Destination* columns in the grid.

Once the import template is defined, the experiment type where the data should be saved is to be defined (see Figure 9.1.12). The existing sequence read sets are displayed in the drop down menu. At this stage of the import, there is also the possibility to create a new sequence read set experiment. If this option is

selected, a name for the new experiment type is queried for when pressing <Next>.

9.1.4.5.6 Import sequence read sets: Database links

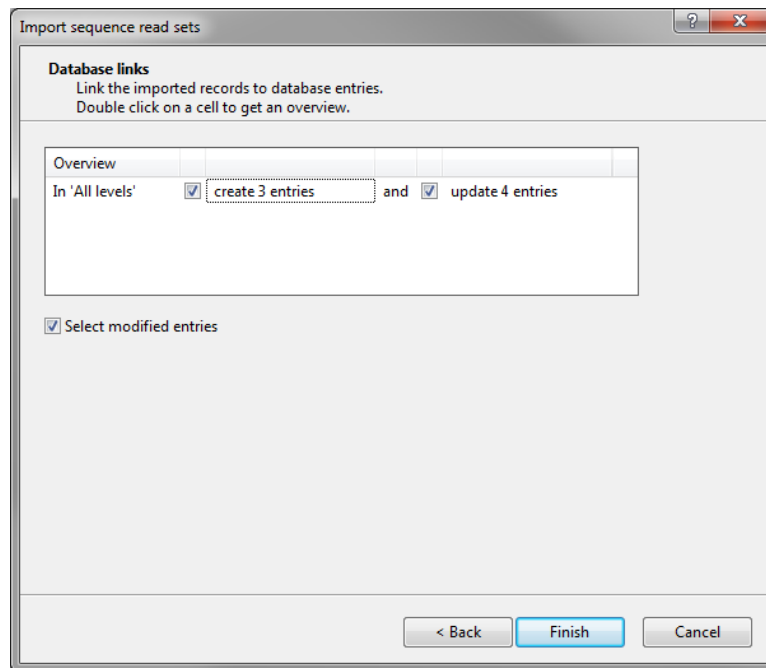


Figure 9.1.15: The *Import sequence read sets* wizard: *Database links* wizard page.

The *Database links* wizard page allows you to have an overview of the entries that will be created and/or updated in the database. At this point, you can still define that you only want to create new entries and not alter anything on data already present in the database or vice versa. When in doubt, double-clicking on the create or the update cell will give you a list of the entries that will be created or updated, respectively. Double-clicking on one of the entry keys opens the corresponding *Entry* window. By default, the check box **Select modified entries** is checked, which implies that after import, entries that were created or updated will be selected in the *Main* window. Select <Finish> to start the actual import of the data into sequence read set experiments.

Chapter 9.2

The sequence read set experiment card

To open the *Sequence read set experiment* window, click the colored dot of a linked sequence read set experiment type.

When no sequence read set is present for a specific key/sequence type combination, no colored dot is present in the *Experiment grid* of the BioNumerics database. A new sequence read set can be imported from **File** > **Import...** (📁, Ctrl+I).



Detailed information on the import of sequences in the BioNumerics database can be found in [9.1.4](#).

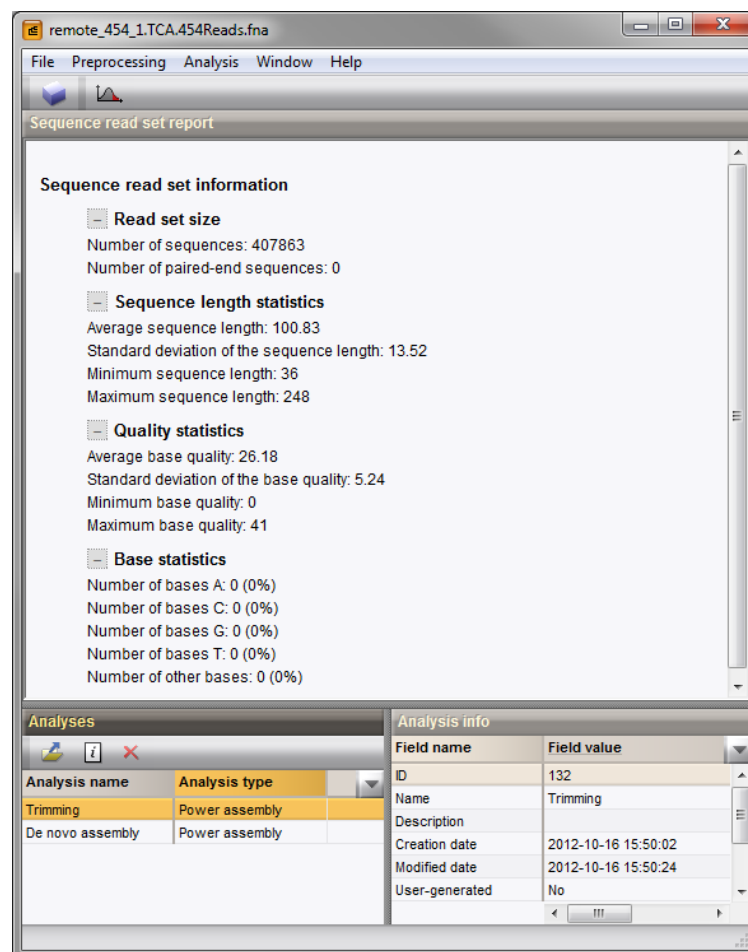


Figure 9.2.1: The *Sequence read set experiment* window.

In the *Sequence read set experiment* window, a summary of the characteristics of the sequence read set is displayed in the *Sequence read set report* panel. Four different information types are covered:

- *Read set size*: indicating the number of sequence reads and whether they are single-end or paired-end,
- *Sequence length statistics*: indicating the average, standard deviation, minimum and maximum sequence length,
- *Quality statistics*: indicating the average, standard deviation, minimum and maximum base quality, and
- *Base statistics*: the number of bases A, C, G, T and other IUPAC callings for ambiguous bases.

On a more detailed level, it is very interesting to consult the predefined charts concerning the average read quality distribution, the base distribution, the read length distribution, read quality distribution by %GC ... These charts provide insight in the sequence quality and the possible presence of sequencing artifacts in the run in a very quick and easy way. From these preliminary insights, assessments can be made for the required preprocessing steps before starting the actual analysis. Charts can be created in one click by selecting **Analysis > Charts and statistics...** (⌘, F7). This pops up the *Create chart* dialog box, where a full range of existing chart templates are listed (see Figure 9.2.2).

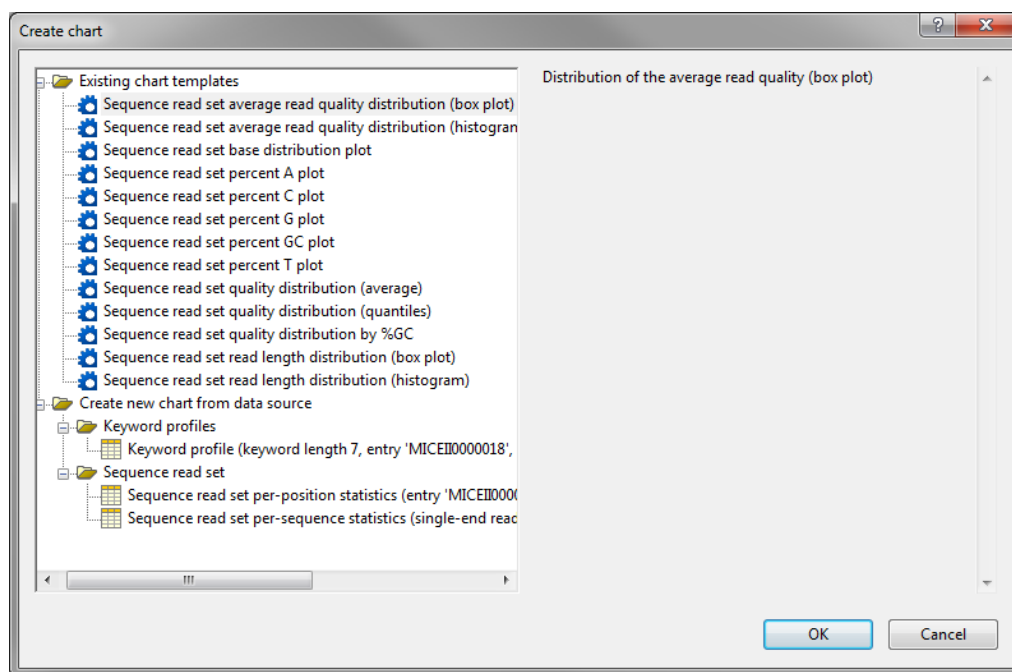


Figure 9.2.2: Lists of existing chart templates in the *Create chart* dialog box.

Selecting any of the charts and pressing <OK> will automatically create a dedicated chart upon the read information from the underlying sequence read set. See Figure 9.2.3 for an example of a base distribution plot generated by the chart template *Sequence read set base distribution plot*.

The second panel in this window, the *Analyses* panel, lists all the preprocessing actions and analyses that have been performed on this sequence read set. For any of the selected analyses, detailed analysis information is updated in the right-hand side panel.

Typically, sequence read sets contain the raw or preprocessed data to start whole genome assembly or metagenomics analysis. Both analysis types require specific preprocessing actions on the raw data. All preprocessing actions are grouped together under **Preprocessing**. The items listed here can clearly be divided in two groups. On one hand, the typical data preprocessing steps as preparation for genome assembly are:

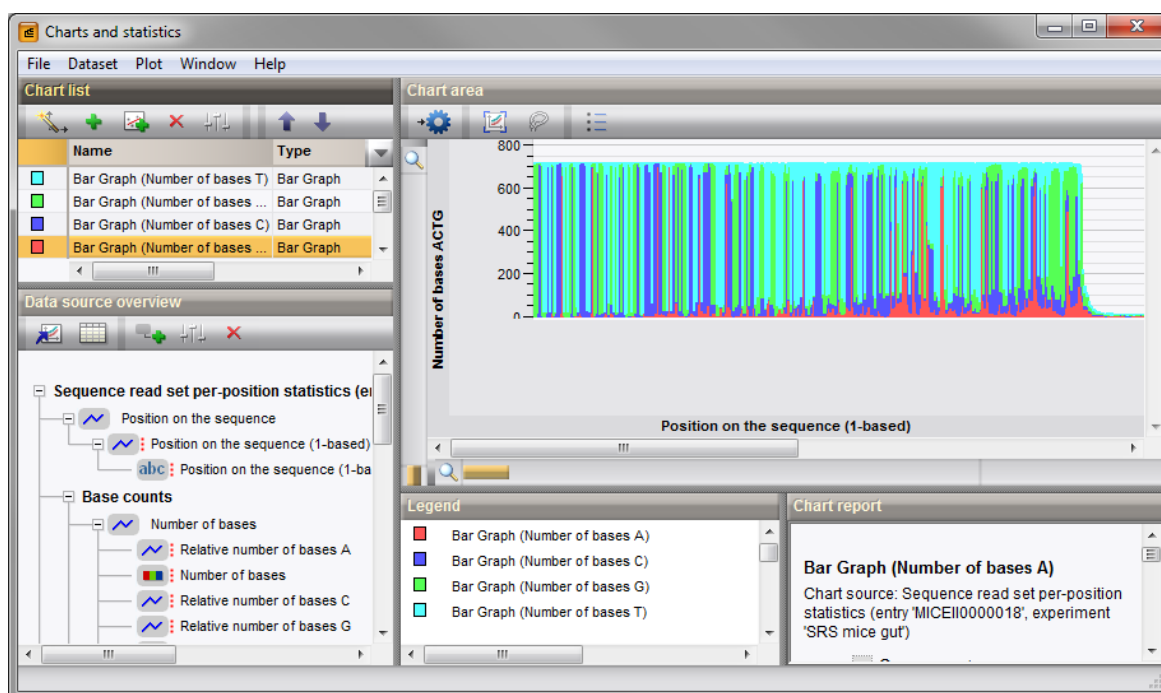


Figure 9.2.3: Example of a base distribution plot generated by an existing chart template.

- *Demultiplexing* the read set, if this was not done during the import procedure (9.4.1),
- *Splitting 454 paired-end reads* (9.4.2), and
- *Trimming* the read set based on sequence content, read quality and read length (9.4.3).

On the other hand, the typical data preprocessing steps for metagenomics analysis are listed:

- *Chimera detection* filters out the sequences originating from two different source strains and as such not representing real biologic sequences (9.4.4),
- *Primer removal* removes forward and reverse sequencing primers (9.4.5), and
- *Sequence selection*, based on the mapping position of the reads in the reference alignment. Criteria can be absolute i.e. based on specific start and/or end positions in the reference alignment, or based on the minimum and maximum sequence length spanning the alignment. Instead of using the absolute criteria, one can also opt to do the selection based on a relative criterion i.e. a specified percentage of sequences is retained after evaluating the start and/or end positions in the reference alignment and the minimum and maximum sequence length spanning the alignment (9.4.6).


In addition to chimera detection, primer removal and sequence selection, general trimming (9.4.3) is also suggested as preprocessing step for metagenomics data.


Within BioNumerics, sequence read sets are the data repositories to start genome assembly or metagenomics analysis. As such, once the read set is trimmed and preprocessed, typical analyses that can be initiated on the read set include:


- *De novo assembly* of bacterial genomes (9.5.1),
- *(Whole genome) resequencing assembly* based on a reference genome (9.5.2),
- *Single sample diversity analysis* to assess the alpha diversity of a metagenomics sample (9.5.5), and

- *Reference taxonomy-based identification* of all the sequences in the metagenomics sample (9.5.6).

The aforementioned analyses can easily be started from the *Sequence read set experiment* window by selecting one of the analyses from the **Analysis** menu. Each of these preprocessing pipelines and analyses is further discussed in detail in 9.4 and 9.5, respectively.

Once any of the preprocessing actions or analyses is executed, the analysis is added to the list in the *Analyses* panel. The selected analysis can be re-opened in its specific analysis window, either the *Power assembly* window or the *Metagenomics* window, by selecting **File > Open selected analyses** (). If necessary, the analyses can be modified and re-run in these application windows.

To display the analysis report of the selected power assembly analysis, press **File > Open selected analysis reports** (). The report contains the detailed information on all the actions that were executed in the analysis pipeline: the input data, parameter settings, summary histograms and results are displayed. Selecting **File > Copy report (text)** or **File > Copy report (html)** will copy the report information to the clipboard in text format or HTML text format, respectively. The content of the tables in the report can be copied individually by selecting **File > Copy table (text)** or **File > Copy table (html)** in the context menu. Selecting **Open in separate window** opens the table in a separate *Report table* window where you can copy and edit the selected rows.

A selected analysis can be removed from the *Analyses* panel by selecting **File > Remove selected analyses** ().

Imported sequence read sets can be exported to a FASTA or FASTQ file by selecting **File > Export > To fasta file...** or **File > Export > To fastq file...**, respectively. Within the export window that pops up, one can modify the directory to export the data to, and provide a file name for the exported sequence file.

Similar as for a character and sequence type experiment, the option **File > Remove this experiment...** is available, making it possible to delete the sequence read set experiment from the database. Deleting the sequence read set is an irreversible operation.

Chapter 9.3

Cluster analysis of sequence read sets

9.3.1 Sequence read set comparison settings



Please note that cluster analysis of sequence read sets requires both the Sequence data module (SQ) and the Tree and network inference module (TN) to be present in your BioNumerics configuration.

To calculate a cluster analysis on a sequence read set experiment, select the sequence read set type to analyze in the *Experiments* panel of the *Comparison* window, and choose **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**... The *Comparison settings* wizard appears (Figure 9.3.1).

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient.

The representation on the left provides an overview of the available coefficients. Depending on the selected coefficient, relevant settings are displayed at the right. Sequence read sets have a very limited number of parameter settings that can be addressed during cluster analysis. The coefficients are subdivided in two categories: **Numerical** and **Binary** similarity coefficients. Within the **Numerical** category the Pearson and Cosine correlation coefficients are listed, as well as the distance-based Euclidean and Manhattan distances. As binary correlations, Dice and Jaccard are available, as well as the Simple matching algorithm.

Numerical coefficients treat the calculated keyword profiles on the sequence read sets as numbers.

The **Pearson correlation** (or Pearson product-moment correlation) is calculated as:

$$r_{j,k} = \frac{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k)}{\sqrt{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2 \sum_{i=1}^n (x_{i,k} - \bar{x}_k)^2}}$$

with

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

and

$$\bar{x}_k = \frac{1}{n} \sum_{i=1}^n x_{i,k}$$

Hereby, n denotes the number of keywords in the keyword profile and $x_{i,j}$ and $x_{i,k}$ the i^{th} keyword frequency

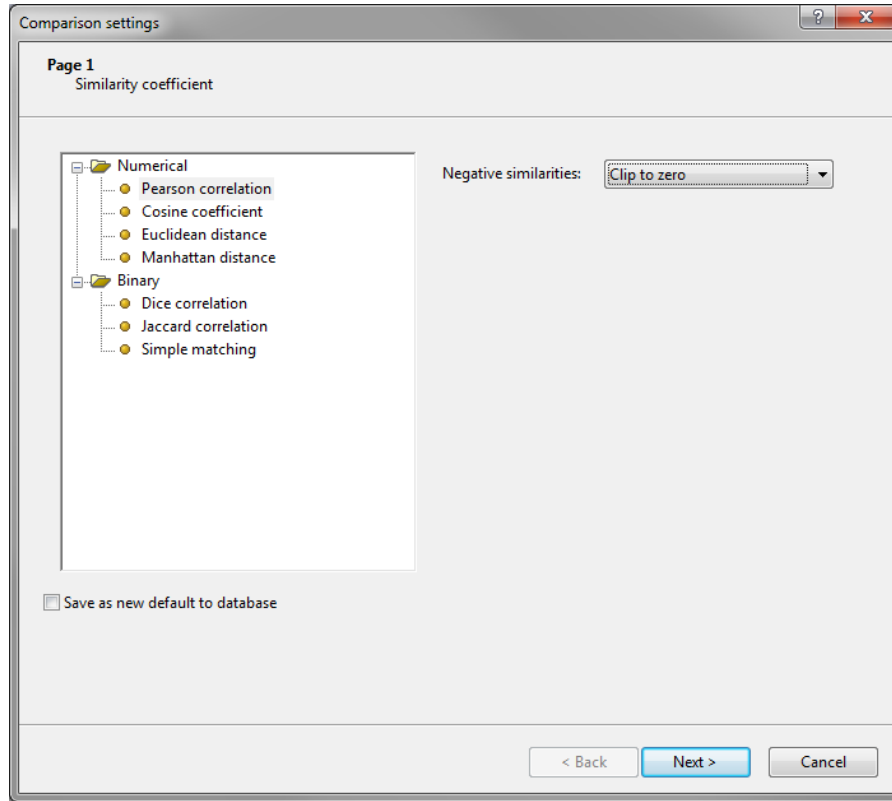


Figure 9.3.1: The *Similarity coefficient* wizard page for sequence read sets.

of entries j and k , respectively. r_i is the frequency range of keyword i .

The related ***Cosine coefficient*** is calculated as:

$$C_{j,k} = \frac{\sum_{i=1}^n x_{i,j} x_{i,k}}{\sqrt{\sum_{i=1}^n x_{i,j}^2 \sum_{i=1}^n x_{i,k}^2}}$$

Note that only for the ***Numerical*** Pearson correlation, one additional parameter is displayed at the right:

- ***Negative similarities*** can be dealt with in different ways. If ***Clip to zero*** is selected, negative similarity values will be replaced by zero (no correlation). When ***Unchanged*** is set, the program will calculate with the negative values. ***Absolute value*** will treat negative and positive similarity values in the same way. ***Negative similarities*** values can only be obtained with the ***Pearson correlation*** coefficient, therefore the option is not available when any of the other coefficients is selected.

Distance-based coefficients are a subcategory of the ***Numerical*** coefficients. Instead of a matrix of similarity values, application of any these coefficients will result in a matrix of distances.

The ***Euclidean distance*** coefficient is calculated as:

$$\Delta_{j,k} = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_{i,j} - X_{i,k})^2}$$

with $X_{i,j}$ and $X_{i,k}$ the i^{th} scaled keyword frequency of entries j and k , respectively. Keyword frequencies are scaled by dividing them by the frequency range or a fixed distance factor.

The ***Manhattan distance*** coefficient is calculated as:

$$M_{j,k} = \frac{1}{n} \sum_{i=1}^n |X_{i,j} - X_{i,k}|$$

For a **Binary** coefficient, a character can only have two states: presence or absence (1 or 0). The binary conversion threshold to convert the numerical keyword frequencies to binary characters is defined from the general sequence read set experiment settings (see also 9.1.3).

Using these binary characters, the **Jaccard** coefficient is calculated as:

$$S_J = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

with N the total number of characters. N_A , N_B and N_{AB} are the number of characters that are present for entry A , entry B and both A and B , respectively.

The **Dice** coefficient is calculated as:

$$S_D = \frac{2N_{AB}}{N_A + N_B}$$

The **Simple matching** coefficient is calculated as:

$$S_{SM} = \frac{N_{AB} + N_{ab}}{N}$$


N_{ab} is the number of characters that are absent for both A and B .


Jaccard correlation and **Dice correlation** are very related to each other, whereas **Simple matching** is fundamentally different. The Jaccard and Dice coefficients only consider "scoring characters" being two present characters in both data sets, whereas the simple matching coefficient also considers two absent characters as scoring.


If a similarity matrix already exists for the selected experiment, an option **Keep existing similarity matrix** appears. When checked, the previously calculated similarity matrix will be used.

On *Page 2* of the dialog box, the available cluster analysis details are displayed. More information on the numerous details for cluster calculation can be found in 13.2.6. Once the cluster analysis is executed, the resulting dendrogram and similarity matrix are shown for the selected sequence read set experiment type.

9.3.2 Sequence read set display settings

Different characteristics of the sequence read sets can be visualized in the comparison. Before using the different display functions, make sure that the sequence read set data is shown in the *Experiment data* panel by pressing the  button next to the experiment name in the *Experiments* panel.

Initially, the sequence read sets are displayed through their keyword view (see Figure 9.3.2), based on keywords of 4 bp. These keywords represent any combination of 4 nucleotides, for each keyword, the keyword abundances are displayed for the sequence read set at hand. For each entry a band intensity profile over the keywords is displayed, where the low abundant and high abundant keywords are displayed as white and black bands, respectively. The keywords themselves are indicated as colored blocks. When zooming in on these keywords, the blocks are replaced by the actual nucleotides: A, C, T and G. At any time you can display the keyword view by selecting **ReadSets > Keyword view (length 4)** .

Select **ReadSets > Keyword view (length 4)**  to display the position count view (see Figure 9.3.3). In this view, the bar heights on each read position are proportional to the nucleotides called over all reads.

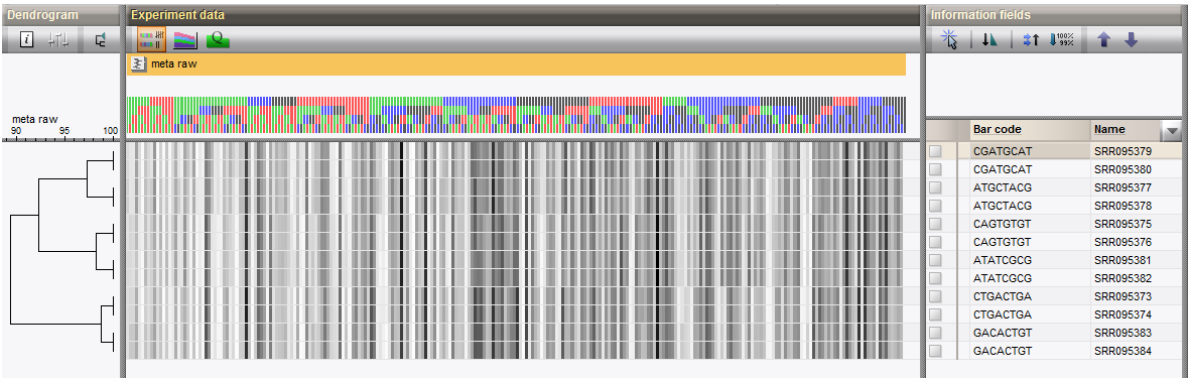


Figure 9.3.2: The sequence read set display settings: the keyword view.

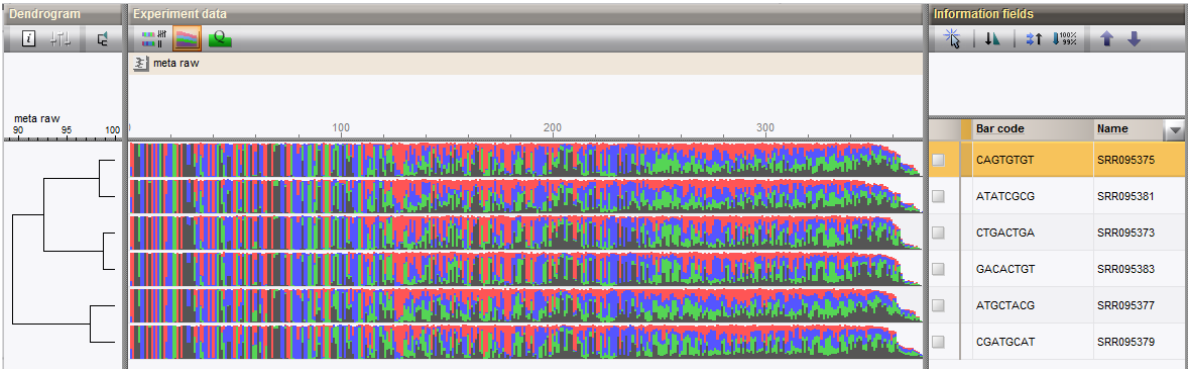



Figure 9.3.3: The sequence read set display settings: the position count view.

Typically, bars having only one color, refer to primer positions that were not (yet) trimmed off or are very conserved regions in the sequence. The tail at the end appears when not all the reads in the sequence read set are having the same read length.



Figure 9.3.4: The sequence read set display settings: the quality view.

Select **ReadSets > Keyword view (length 4)** () to display the quality view (see Figure 9.3.4). In this view, the bar height of the green bar indicates the average nucleotide quality on each read position. The quality scores are in Phred scale, typically between 0 and 63.

The keyword view, as well as the position count view can also be shown in normalized mode, meaning the views are compensated for the number of reads within one sample. To toggle between normalized and absolute keyword frequencies, select **ReadSets > Compensate views for number of reads**.

Chapter 9.4

Preprocessing sequence read sets

9.4.1 Demultiplexing

Multiplexing is used for sequencing multiple samples together within the same sequencing lane. The technique is typically used to sequence e.g. smaller bacterial genomes or to analyze multiple metagenomics samples. In general, it allows to analyze a much larger number of samples in a single run without drastically increasing cost and time or compromising on sequence coverage of the samples. Demultiplexing implies that a unique identifier tag, or barcode, that is specific for each library, is traced in the sequence reads, which are then filtered out accordingly, creating samples for individual downstream analyses. When starting the analysis from the *Main* window, you can select the input sequence read set experiment type that should be used for the analysis (see Figure 9.4.1). All available sequence read set experiment types are listed in the drop down.

First, the **Barcode location** needs to be specified. The location can be set at the beginning or at the end of the read sequence. For paired-end data, one can also specify to have no barcode in a specific part of the linked paired-end reads. Barcodes have a fixed sequence length, and are followed by a linker sequence of fixed length as well. Optionally, a linker can be present before the barcode. All these sizes can be set in the fields **Size of barcode**, **Bases between barcode and sequence** and **Bases before barcode**, respectively.

As barcodes are defined by the barcode size, the software will also detect barcodes that differ in one or multiple bases from the actual sample barcode. This will typically result in real biological samples having relative read frequencies of 99% and some artificial samples having low relative read frequencies (see also Figure 9.1.11). As you don't want to save the latter samples to the database because they just represent sequencing errors within the barcodes and not actual samples, a system of **Barcode filtering** has been implemented.

To start the demultiplexing analysis, select an output experiment type from the drop down list and press **<OK>**.

9.4.2 Split paired-end reads

This preprocessing action takes read sequences from Roche/454[®] paired-end runs that were imported as single-end reads, and creates paired-end read sequences. The split position is determined by aligning the adapter sequence to the read. When the adapter does not match any part of the read sequence, the sequence is left as a whole. If the adapter matches but is too close to the beginning and/or the end of the read sequence, only the parts that are long enough are retained, thus yielding a single-end read sequence, or no sequence at all. Sequence qualities are split accordingly. When starting the analysis from the *Main* window, one additional dialog page is displayed where you can select the sequence read set experiment type that should be used for the analysis (see Figure 9.4.2). All available sequence read set experiment types are listed in the

Demultiplexing analysis

The demultiplexing analysis splits up a sequence read set according to the multiplex identifiers contained in the sequence reads. The multiplex identifiers are removed from the sequences, along with the linker sequences used (if any).

Input experiment type
 Input experiment type: SRS mice gut

Barcode location for the first end
 Barcode location: At the beginning
 Number of bases before the barcode: 0
 Size of the barcode: 8
 Number of bases between barcode and sequence: 0

Barcode location for the second end
 Barcode location: At the end
 Number of bases before the barcode: 0
 Size of the barcode: 8
 Number of bases between barcode and sequence: 0

Frequency filtering
 Minimum number of sequences per barcode: 1
 Minimum relative frequency: 0 %

Output experiment type
 Output experiment type: SRS mice gut trimmed

OK Cancel

Figure 9.4.1: The *Demultiplexing analysis* dialog box.

drop down. Select the required sequence read set experiment type and select *<Next>* to proceed.

From this page of the dialog, the adaptor sequence and minimum sequence length to retain a single-end read can be defined (Figure 9.4.3).

First, the *Adaptor sequence(s)* to search for when determining the split position between the two ends of the paired-end read can be set. The available adaptor sequences are: the Roche/454 Flx[®] palindromic adaptor GTTGAACCGAAAGGGTTTGAATTCAAACCTTTCGGTTCCAAC, and the Roche/454 Titanium[®] adaptor TCGTATAACTTCGTATAATGTATGCTATACGAAGTTATTACG. If the latter is selected, also the reverse complement is used for analysis.

The *Minimum sequence size* is the minimum sequence length (bp) a splitted single-end sequence should have in order to be saved in the output sequence read set. If a paired-end read is splitted into one single-end read surpassing the threshold and one smaller than the threshold size, the latter is omitted from the analysis and the paired-end read is split into one single-end read only.

In this page of the wizard, the settings for aligning the adaptor sequence against the read are specified (Figure 9.4.4).

The alignment parameter settings include:

- The *Match score*: Score for two identical bases.
- The *Mismatch score*: Score for two non-identical bases.
- *Allow gaps in the adaptor*: Whether or not to allow gaps in an adaptor sequence; with *Open gap score*

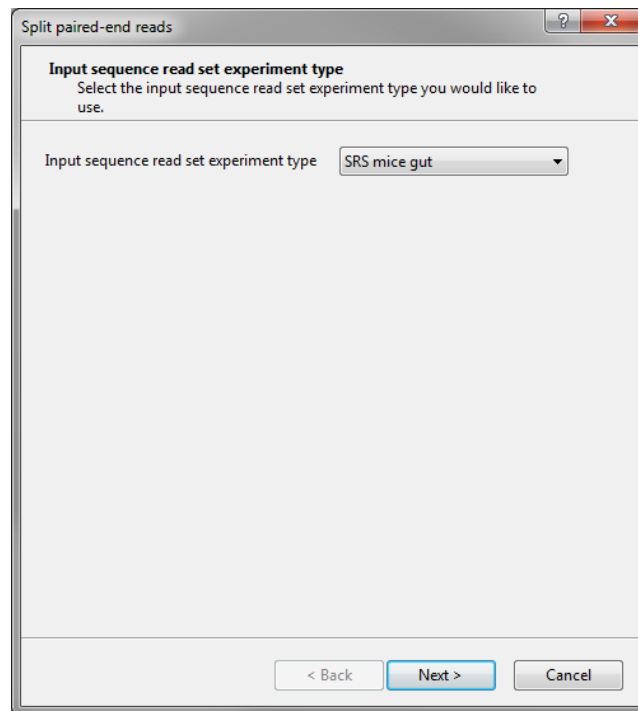


Figure 9.4.2: The *Split paired-end reads* dialog box: Input sequence read set experiment type.

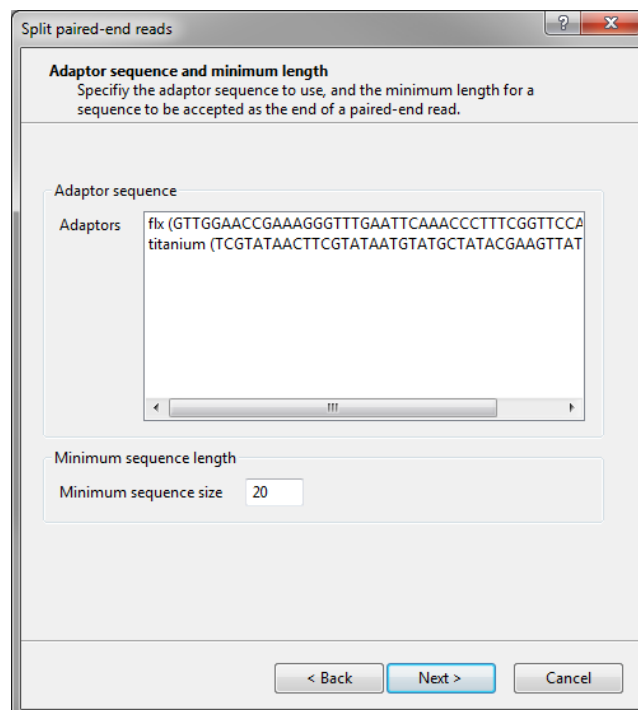
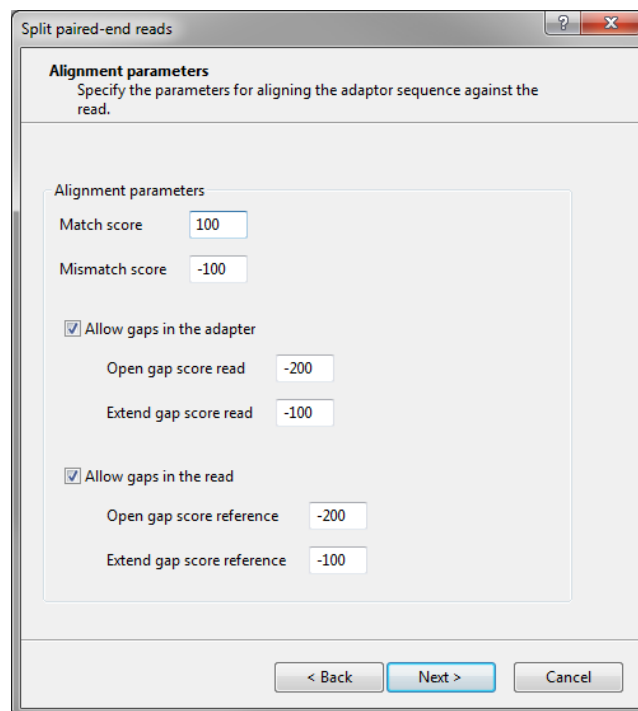


Figure 9.4.3: The *Split paired-end reads* dialog box: Adaptor sequence and minimum length.

read: Penalty score for introducing a gap in an adaptor sequence, and *Extend gap score read*: Penalty score for extending an existing gap in an adaptor sequence.

- *Allow gaps in the read*: Whether or not to allow gaps in a read sequence; with *Open gap score reference*: Penalty score for introducing a gap in a read sequence, and *Extend gap score reference*: Penalty score for extending an existing gap in a read sequence.



The dialog box is titled "Split paired-end reads" and contains a section for "Alignment parameters". The instructions state: "Specify the parameters for aligning the adaptor sequence against the read." The parameters are as follows:

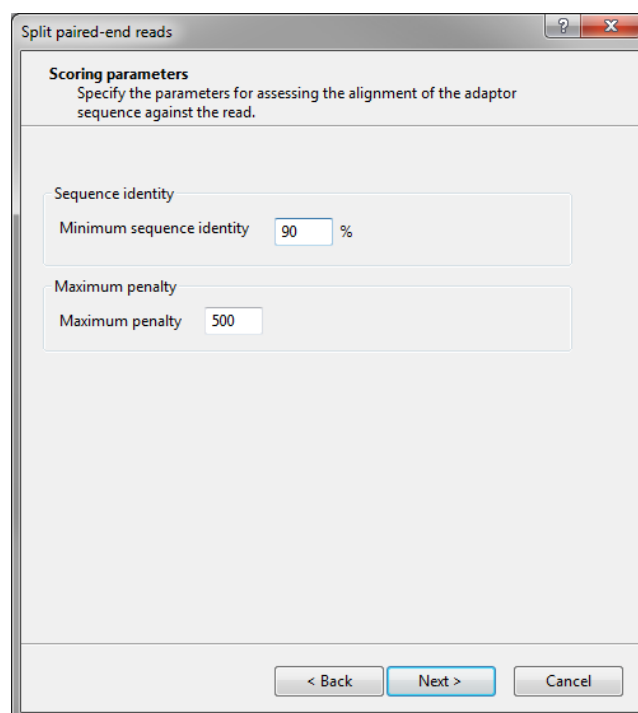
Parameter	Value
Match score	100
Mismatch score	-100
Allow gaps in the adapter	<input checked="" type="checkbox"/>
Open gap score read	-200
Extend gap score read	-100
Allow gaps in the read	<input checked="" type="checkbox"/>
Open gap score reference	-200
Extend gap score reference	-100

At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.4.4: The *Split paired-end reads* dialog box: Alignment parameters.

Press <Next> to proceed.

After calculation of the alignment of the adaptor sequence against the reads, the alignments are assessed based on the following parameters (Figure 9.4.5):



The dialog box is titled "Split paired-end reads" and contains a section for "Scoring parameters". The instructions state: "Specify the parameters for assessing the alignment of the adaptor sequence against the read." The parameters are as follows:

Parameter	Value
Sequence identity	
Minimum sequence identity	90 %
Maximum penalty	
Maximum penalty	500

At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.4.5: The *Split paired-end reads* dialog box: Scoring parameters.

- The *Minimum sequence identity*: minimum sequence identity for an alignment to be acceptable.

- The *Maximum penalty*: maximum penalty for an alignment to be acceptable. The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Press **<Next>** to continue.

At the last page, select an output experiment type from the drop down list (Figure 9.4.6). All sequence read set experiments present in the database are listed here. Select the experiment type to export the splitted sequence read set to, and press **<Finish>** to start the preprocessing of the paired-end reads.

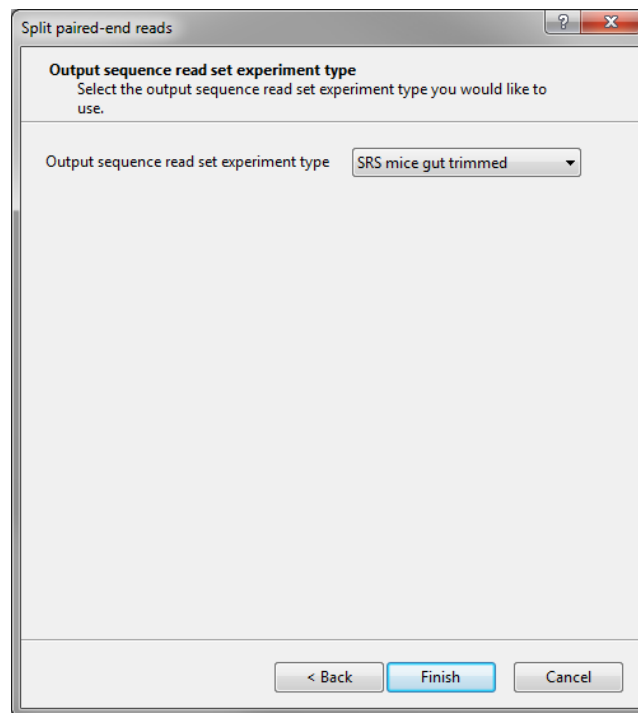


Figure 9.4.6: The *Split paired-end reads* dialog box: Output sequence read set experiment type.

For this analyses, a power assembly project is created. The project consists of a combination of predefined actions, each having their specific parameter settings. Detailed information on the parameter settings can be found in 18.5.2.4.

9.4.3 Trimming

Predefined trimming actions provide means to filter the sequence read sets and retain only those reads that meet the quality standards or structural expectations. The trimming actions can be based on the length and the quality scores of the read sequences, the polyA or polyGC content, ... When starting the analysis from the *Main* window, one additional dialog page is displayed where you can select the sequence read set experiment type that should be used for the analysis (see Figure 9.4.7). All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select **<Next>** to proceed.

Trimming is very specific, as such, one can decide whether or not to include a specific component in the trimming pipeline by simply checking the box in front of the description of the trimming actions. When unselecting a specific trimming option, the parameters become greyed and the trimming action will not be included in the trimming pipeline.

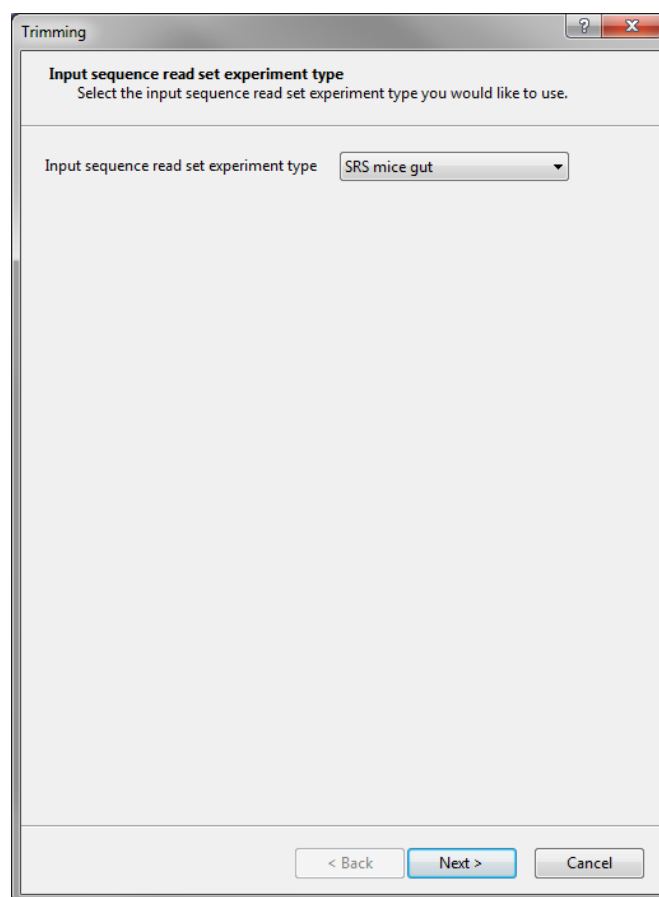


Figure 9.4.7: The *Trimming* dialog box: Input sequence read set experiment type.

The first page of the *Trimming* dialog box queries for the parameters for the structural trimming.

Structural trimming includes:

- *Remove reads with long homopolymers*: removes reads with long homopolymers i.e. strings of identical bases. The maximum number of identical bases can be set as *Maximum homopolymer length*.
- *Remove polyA reads*: removes all reads with too many A bases, expressed as percentage A over the read. The threshold is defined as *Maximum %A*.
- *Remove polyGC reads*: removes all reads with a read %GC below the minimum and exceeding the maximum %GC thresholds set as *Minimum %GC* and *Maximum %GC*.

Press <*Next*> to proceed to the Overall quality trimming options.

The overall quality trimming actions decide whether or not the read is retained in the sequence read set based on overall read quality (Figure 9.4.9).

From the *Sequence exclusion* criteria, reads can be excluded from the read set based on their minimum read quality and average read quality. Thresholds values are entered as *Exclude reads with minimum quality below* and *Exclude reads with average quality below* values. From the *Base replacement* settings, the *Replace bases by N when the quality is below* threshold can be set. In this case, the read is retained in the data set if the quality of only one base drops below the threshold but the base call is replaced by N. Press <*Next*> to continue.

The tail quality trimming acts on the tails, i.e. the end of the reads and uses read quality criteria (Figure 9.4.10).

Trimming

Structural trimming
Specify the parameters for trimming based on the sequence content.

☒ Remove reads with long homopolymers

This action removes all read sequences with long homopolymers.

Homopolymer length threshold
Maximum homopolymer length

☒ Remove polyA reads

This action removes all read sequences with too many bases A.

Maximum number of bases A
Maximum %A %

☒ Remove polyGC reads

This action removes all read sequences with a %GC that is too low or too high.

Thresholds for number of bases C or G
Minimum %GC %
Maximum %GC %

< Back Next > Cancel

Figure 9.4.8: The *Trimming* dialog box: Structural trimming.

Three tail quality actions are defined:

- *Raw quality tail removal*: removes the bases in the tail of the read as long as the base quality is below the *Minimum tail quality*.
- *Windowed average tail removal*: calculates the average base quality in a fixed window size before the last base of the read. If the average quality is below the *Minimum windowed average quality*, the last base is cut from the read and the window is shifted towards the beginning of the read by one position. Trimming the bases in the tail continues until the windowed average quality exceeds the threshold.
- *Rolling average tail removal*: calculates the average base quality in a fixed window size starting at the beginning of the read. If the average quality is above the *Minimum rolling average quality*, the base at the right of the window is retained and the window is extended by one position towards the end of the read. From the moment the average quality drops below the threshold, all bases after the window are trimmed off.

Press <**Next**> to continue.

The length trimming trims read sequences based on their read sequence length. Reads that are too short i.e. shorter than the number of bases defined as *Short read exclusion* parameter are removed from the data set (Figure 9.4.11). Long reads are shortened according to the *Long read restriction*. Press <**Next**> to continue.

On the last page, select an output experiment type from the drop down list (Figure 9.4.12). All sequence read set experiments present in the database are listed here. Select the experiment type to export the trimmed sequence read set to, and press <**Finish**> to start the trimming pipeline.

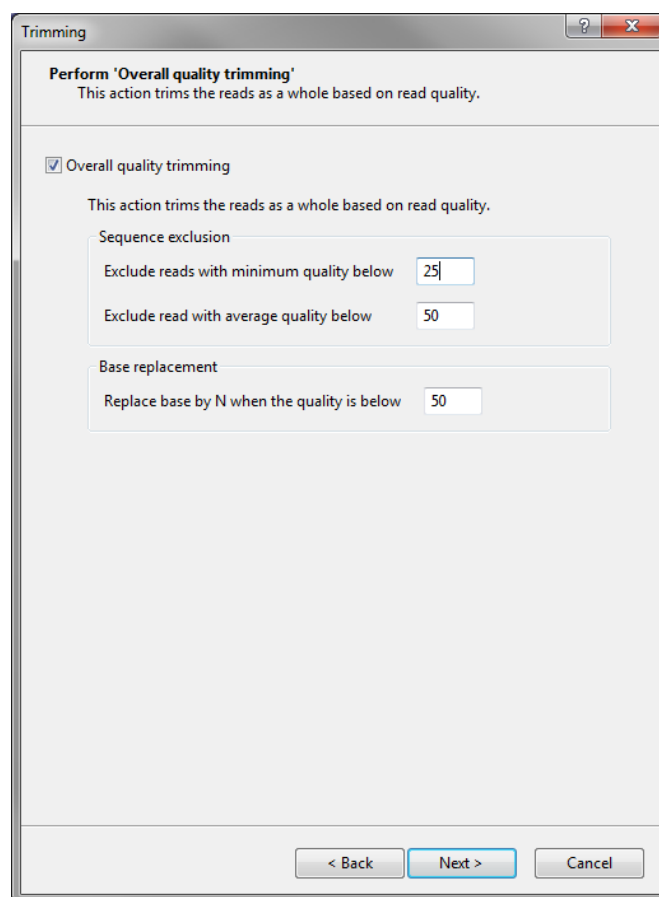


Figure 9.4.9: The *Trimming* dialog box: Overall quality trimming.

For this analyses, a power assembly project is created. The project consists of a combination of predefined actions, each having their specific parameter settings. Detailed information on the parameter settings can be found in [18.5.3](#).

9.4.4 Chimera detection

A common source of 16S sequence artifacts is the formation of chimeric sequences during PCR amplification of the 16S genes. Chimeras are formed when an aborted extension product from an earlier cycle of PCR can function as a primer in a subsequent PCR cycle. Studies have indicated that $\pm 5\%$ of the sequences within curated collections are anomalous or suspect, with chimeras accounting for the majority of problematic sequences [8]. Experimental measurements of chimera formation during PCR co-amplification of 16S rRNA sequences from cloned 16S genes or from mixed bacterial genomic DNA have indicated chimera formation rates of over 30%. Multiple factors including pairwise sequence identity between 16S rRNA genes, number of PCR cycles, and relative abundance of gene-specific PCR templates have been shown to influence chimera formation [39] [38] [1].

Although chimera formation rates can be lowered experimentally, no method has been shown to eliminate these artifacts entirely. Hence, the ability to recognize chimeric sequences is critical in using 16S sequences to profile microbial communities. Several computational methods have been used to identify chimeric sequences. We integrated the chimera-detection algorithm, Chimera Slayer [17] from the mothur tool [35], which can be applied to large datasets, performs well on short sequences, and is sensitive to chimeras between closely related 16S genes.

Chimera detection on a sequence read set can be initiated in different ways:

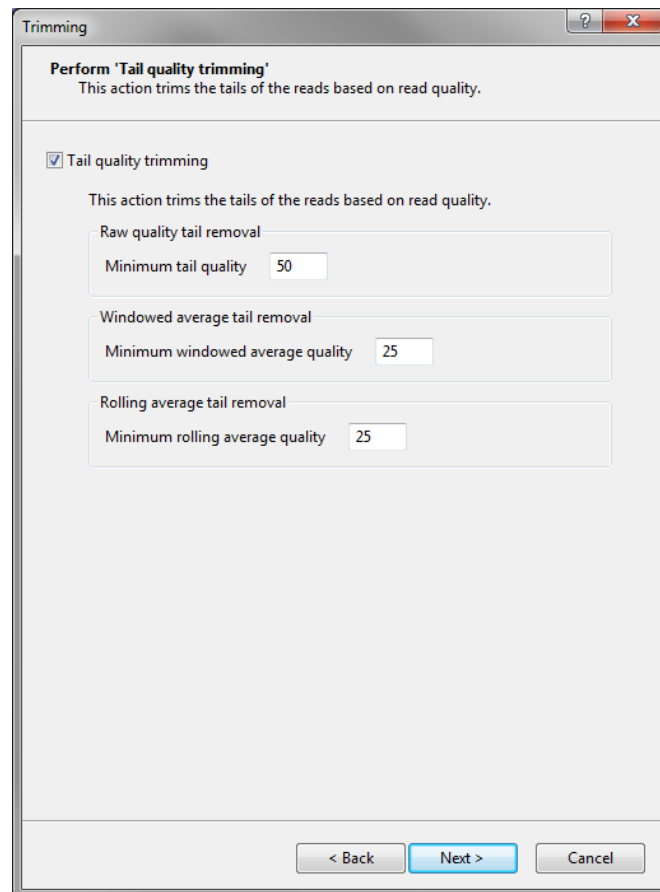


Figure 9.4.10: The *Trimming* dialog box: Tail quality trimming.

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis** > **Sequence read set types** > **Chimera detection**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing** > **Chimera detection**; or
- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File** > **New project...** : **Chimera detection** in the *Create metagenomics project* dialog box.

When launching a chimera detection analysis, the *Chimera detection* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <**Next**> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

The chimera detection reads the sequence read set to be preprocessed and a reference alignment from the database e.g. the Silva-based reference alignment, and outputs potentially chimeric sequences. Finally, the non-chimeric sequences are saved to the output sequence read set experiment in the database.

In the **Chimera screening and comparison settings**, the following parameters can be defined:

- For the *Chimera detection procedure* one has the option to do a self-comparison, or to compare the read sequences to a standard.
 - The *Self-comparison* does not need a reference alignment but will use the more abundant reads from the sample to check all the reads in the sequence read set.

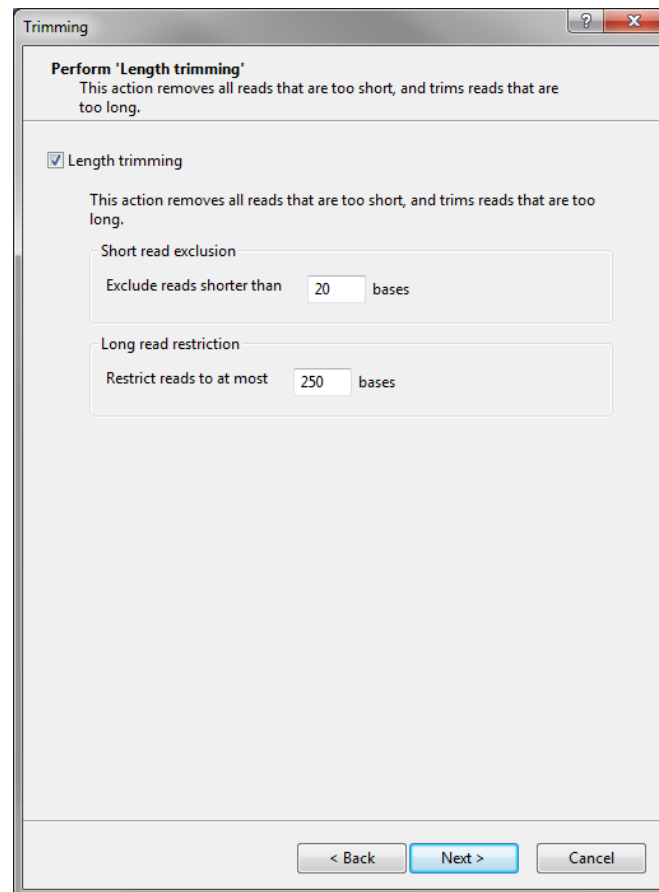


Figure 9.4.11: The *Trimming* dialog box: Length trimming.

- The *Comparison to a standard* will compare the reads from the sequence read set to the sequences from the reference set to decide whether or not the read is identified as a chimera. When this option is used, select the *Reference alignment* to be used from the *Reference alignment* drop down box at the bottom of the dialog. When the reference alignment is defined, the alignment settings can be defined from the *Alignment details* dialog after selecting **<Alignment details ... >**. In the *Alignment details* dialog box, the dedicated settings to align the reads from the sequence read set to the reference alignment can be defined. These settings include the search method to find the closest template for each read, the method to create a pairwise alignment between the read and the de-gapped template sequences, and the settings for the alignment assessment.
 - * There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - * After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively.

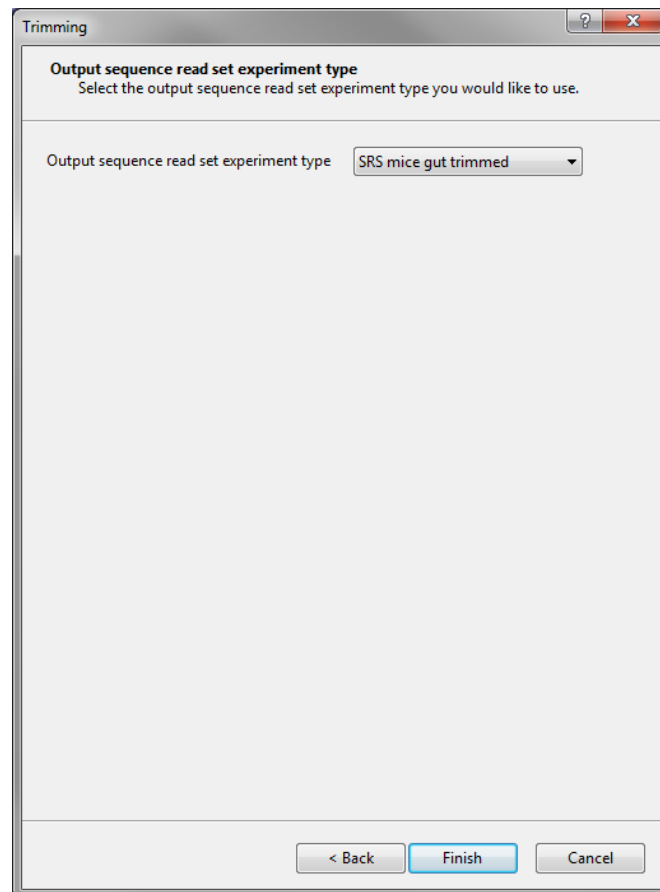


Figure 9.4.12: The *Trimming* dialog box: Output sequence read set experiment type.

The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.

- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- * Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Press **<OK>** to save the modified settings to the project. Press **<Cancel>** to return to the *Chimera detection* dialog box without altering the alignment settings.

- In the settings for the *Screening window*, the *Window size*, i.e. the window size for searching for chimeras, and the *Step size*, i.e. how many base pairs a window is moved each time while screening for chimeric sequences, can be defined.
- The parameters under *Sequence comparison* are used while screening the read sequences for a number of potential parents. The different parameters include:

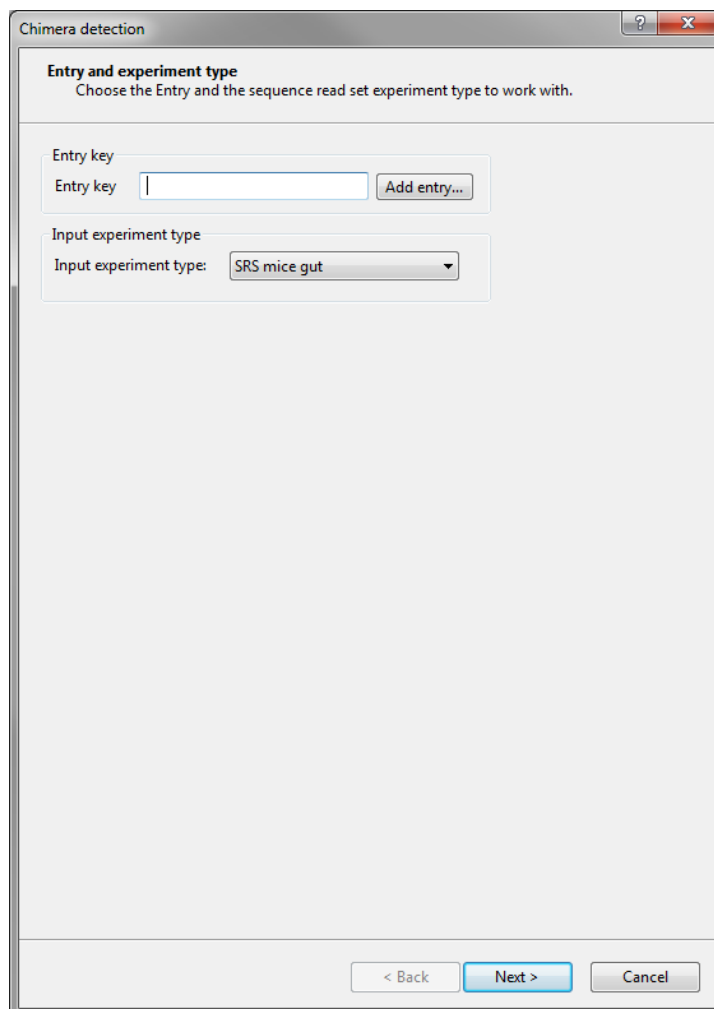
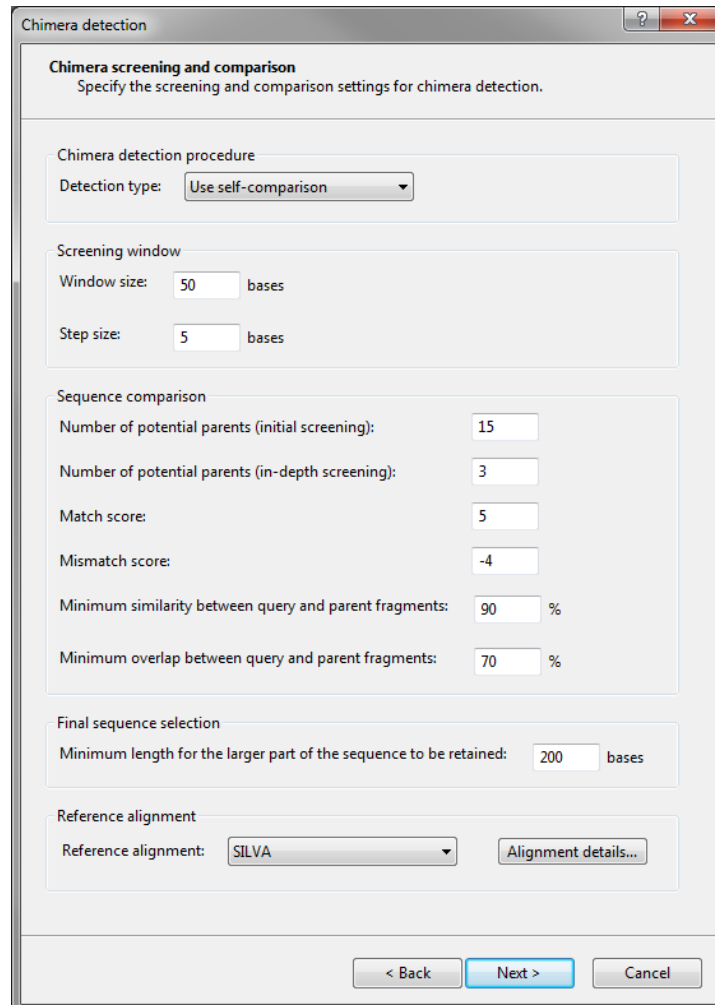


Figure 9.4.13: The *Chimera detection* dialog box: Entry and experiment type settings.

- *Number of potential parents (initial screening)*: how many potential parent sequences each read sequence should be compared with.
 - *Number of potential parents (in-depth screening)*: how many potential parent sequences to investigate from the best rated matches.
 - *Match score* and *Mismatch score*: score and penalty values to reward a matched base and penalize a mismatch base, respectively, while screening potential parent sequences.
 - *Minimum similarity between query and parent fragments (%)*: sequence similarity threshold between the read and the parent fragments.
 - *Minimum overlap between query and parent fragments (%)*: coverage threshold of the closest matches found in the read and the parent fragments.
- The only parameter available under *Final sequence selection* is the *Minimum fragment length for a sequence to be retained*. Chimeric sequences are trimmed to include only the longest piece. To be retained in the data set, this piece should be longer than the minimum fragment length.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press **<Finish>** to start the actual preprocessing.

See [19.3.1.1](#) for more details on the chimera detection parameter settings.



Chimera detection

Chimera screening and comparison
Specify the screening and comparison settings for chimera detection.

Chimera detection procedure
Detection type: **Use self-comparison**

Screening window
Window size: **50** bases
Step size: **5** bases

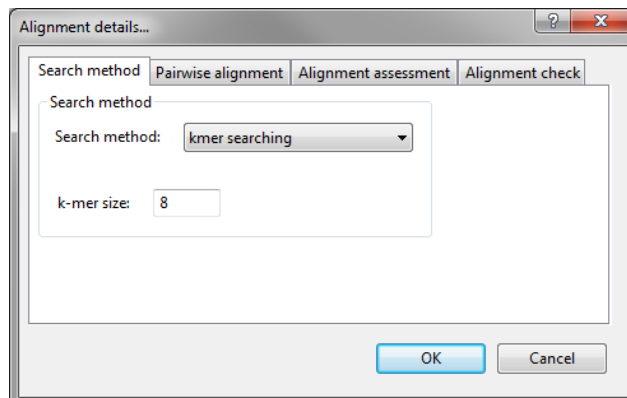
Sequence comparison
Number of potential parents (initial screening): **15**
Number of potential parents (in-depth screening): **3**
Match score: **5**
Mismatch score: **-4**
Minimum similarity between query and parent fragments: **90** %
Minimum overlap between query and parent fragments: **70** %

Final sequence selection
Minimum length for the larger part of the sequence to be retained: **200** bases

Reference alignment
Reference alignment: **SILVA** **Alignment details...**

< Back Next > Cancel

Figure 9.4.14: The *Chimera detection* dialog box: Chimera screening and comparison settings.



Alignment details...

Search method Pairwise alignment Alignment assessment Alignment check

Search method
Search method: **kmer searching**
k-mer size: **8**

OK Cancel

Figure 9.4.15: The *Alignment details* dialog box: Search method settings.

9.4.5 Primer removal

This preprocessing step will enable you to trim off primer sequences. The forward primer is defined as the forward sequencing primer. So if you are using the 16S rRNA primers 27F and 338R to generate sequences, but you are sequencing off from the 338R end of the fragment, you would list 338R as the forward primer and 27F as the reverse. Note that forward and reverse primers can be degenerate using standard IUPAC

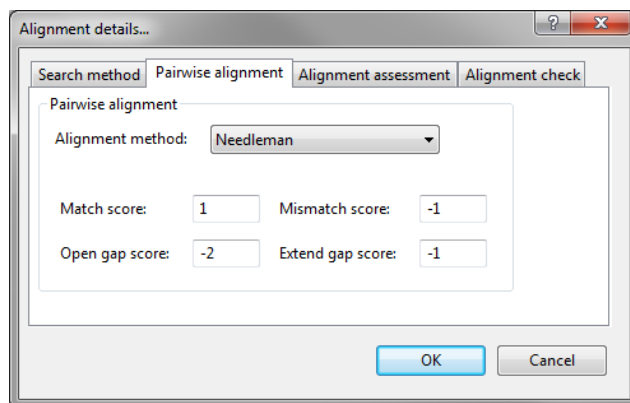


Figure 9.4.16: The *Alignment details* dialog box: Pairwise alignment settings.

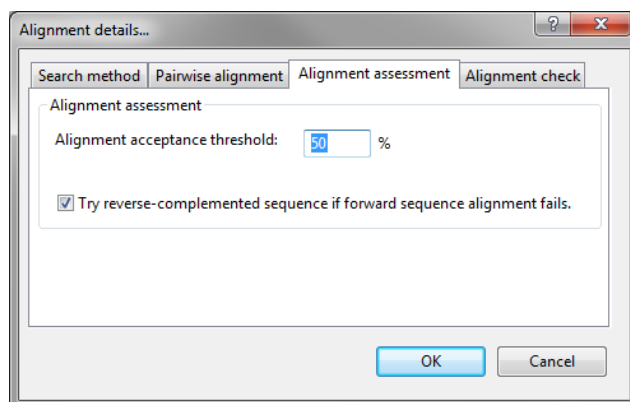


Figure 9.4.17: The *Alignment details* dialog box: Alignment assessment settings.

nomenclature. Primer oligos can be entered both as upper or lowercase letters. It has been shown that sequencing errors in the PCR primer region of a sequence correlate highly with poor sequence quality. Therefore, default settings only allow exact matches. However, one can define a number of mismatches against the primer sequences to avoid this strict screening and allow for inexact matches.

Primer removal on a sequence read set can be initiated in different ways:

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis > Sequence read set types > Primer removal**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing > Primer removal**; or
- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File > New project... : Primer removal** in the *Create metagenomics project* dialog box.

When launching a primer removal analysis, the *Primer removal* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <Next> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

The primer removal imports the sequence read set to be preprocessed and only saves the trimmed reads to the output sequence read set in the database.

Primers to be trimmed off can be selected from the lists in the *Forward primers* and *Reverse primers*. Multiple primers can be selected by holding the **Ctrl** button. On top of the dialog, the *Maximum number of*

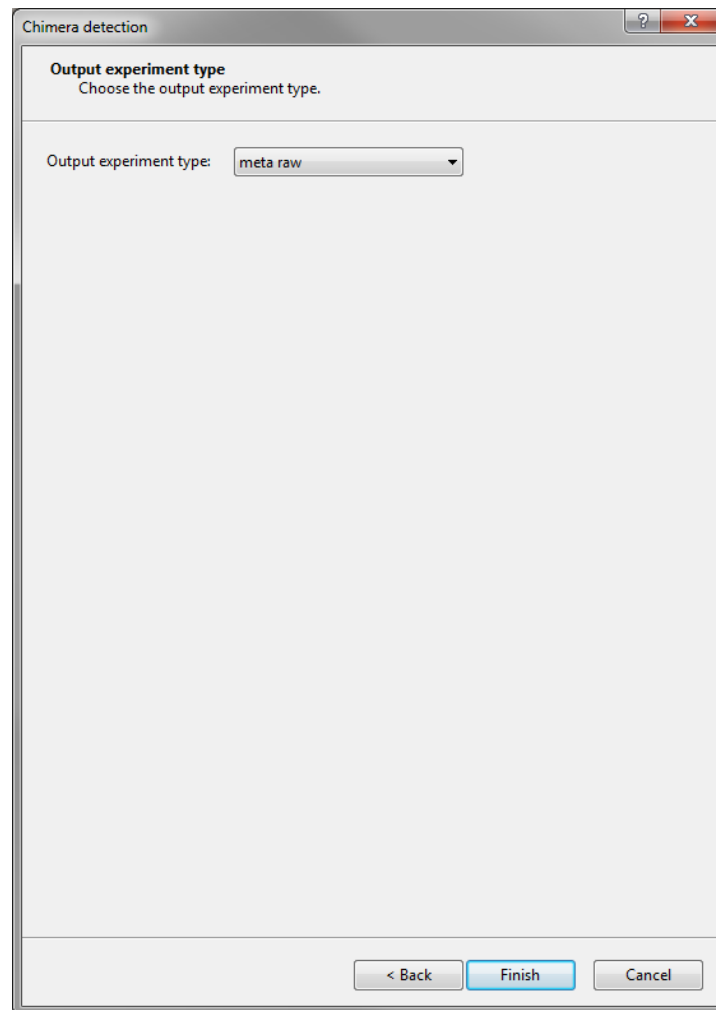


Figure 9.4.18: The *Chimera detection* dialog box: Output experiment type settings.

mismatches in the primer can be defined, allowing for inexact primer matches in the reads. Press **<Next>** to proceed.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press **<Finish>** to start the actual preprocessing.

See [19.3.2.1](#) for more details on the primer removal parameter settings.

9.4.6 Sequence selection

This preprocessing option performs sequence selection based on certain user defined criteria such as start and end position, and based on minimum and maximum length. The sequence selection functionality is based on the mothur command `screen.seqs` [35].

Sequence selection on a sequence read set can be initiated in different ways:

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis > Sequence read set types > Sequence selection**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing > Sequence selection**; or

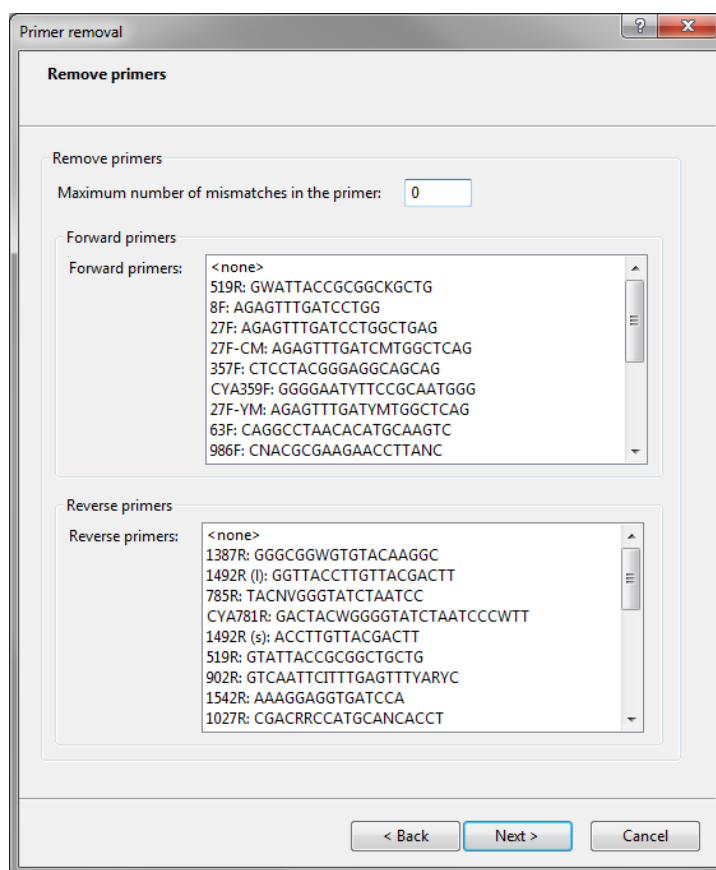


Figure 9.4.19: The *Primer removal* dialog box: Remove primers settings.

- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File** > **New project...** : *Sequence selection* in the *Create metagenomics project* dialog box.

When executing a screening of the read sequences, the *Sequence selection* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <Next> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

Different *Selection methods* can be checked and combined in the *Sequence selection* dialog box to create a custom sequence selection based on the alignment outcome.

Some reads from the data set may not align in the same region as most of the reads that are analyzed. In a situation where read alignments started at deviating alignment positions, the following options can be used:

- *Select sequences based on start position*: reads starting the alignment after the start position entered in the dialog box, will be omitted from the output read set.
- *Select sequences based on end position*: reads ending the alignment before the end position entered in the dialog box, will be omitted from the output read set.

In some pyrosequencing studies, the reads will differ in length. Trimming these reads within the expected window of minimum and maximum size can be done by selecting the options:

- *Select sequences based on minimum length*: reads spanning an alignment region shorter than the minimum length entered in the dialog box, will be omitted from the output read set.

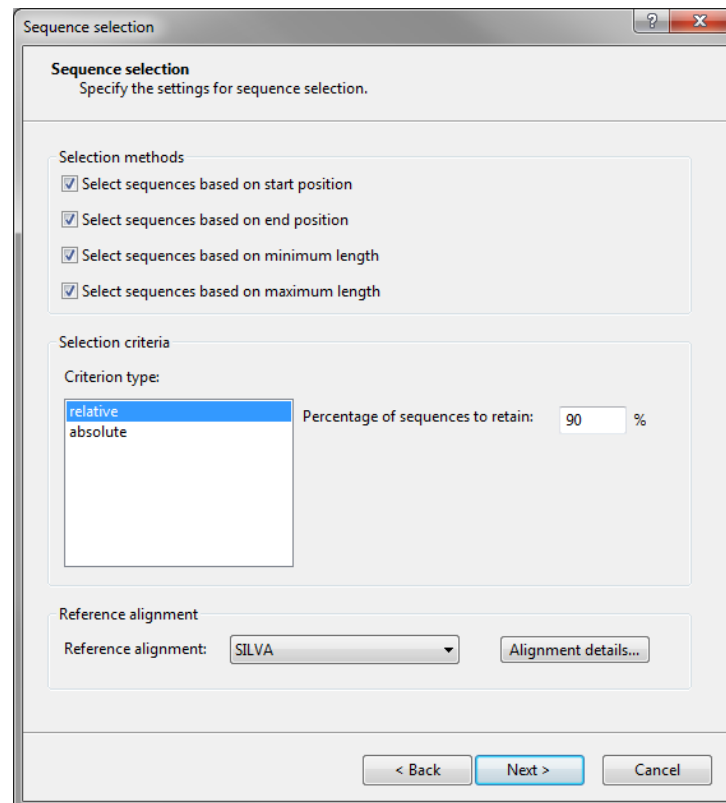


Figure 9.4.20: The *Sequence selection* dialog box: Sequence selection settings.

- *Select sequences based on maximum length*: reads spanning an alignment region larger than the maximum length entered in the dialog box, will be omitted from the output read set.

The *Selection criteria* can be defined in an *absolute* manner, as threshold values entered in the dialog box, or in a *relative* one. When using the relative criteria, the *percentage of sequences to retain* needs to be entered in the dialog box. The percentage of sequences to retain is evaluated for the combination of selection methods checked in the dialog, meaning that e.g. when the percentage to retain is set at 90%, the sequence read set will be trimmed off by 10% of the reads, using all the selected criteria.

All positions and lengths refer to positions and lengths on the reference alignment selected from the *Reference alignment* drop down box at the bottom of the dialog. When the reference alignment is defined, the alignment settings can be entered from the *Alignment details* dialog after selecting **<Alignment details ... >**. In the *Alignment details* dialog box, the dedicated settings to align the reads from the sequence read set to the reference alignment can be defined. These settings include the search method to find the closest template for each read, the method to create a pairwise alignment between the read and the de-gapped template sequences, and the settings for the alignment assessment.

- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.

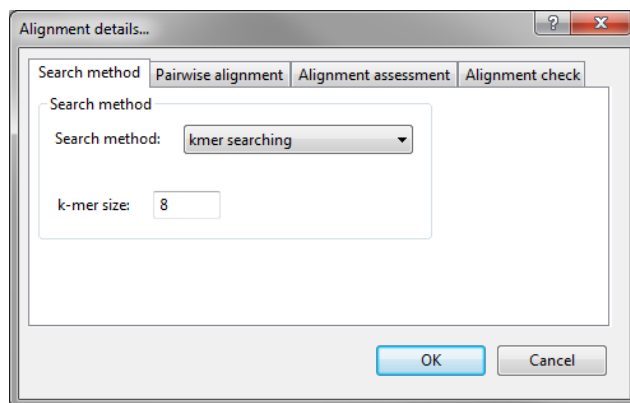


Figure 9.4.21: The *Alignment details* dialog box: Search method settings.

- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
 - *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.

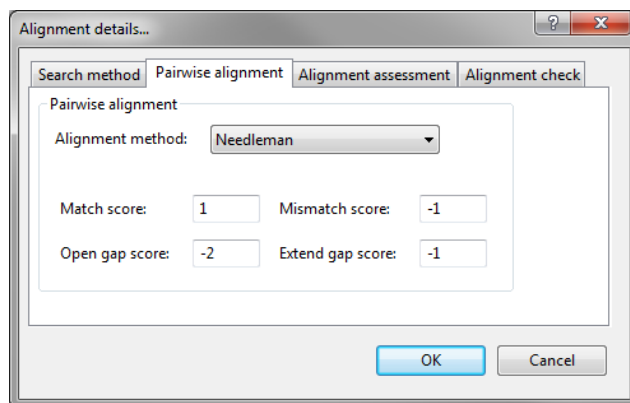


Figure 9.4.22: The *Alignment details* dialog box: Pairwise alignment settings.

- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Press <OK> to save the modified settings to the project. Press <Cancel> to return to the *Sequence selection* dialog box without altering the alignment settings.

Press <Next> to proceed.

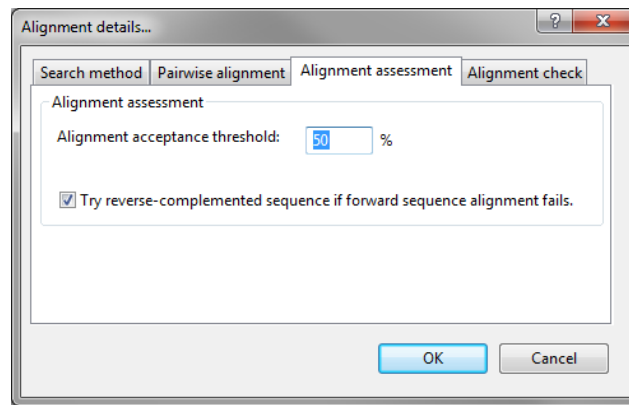


Figure 9.4.23: The *Alignment details* dialog box: Alignment assessment settings.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press **<Finish>** to start the actual preprocessing.

See [19.3.3.1](#) for more details on the sequence selection parameter settings.

Chapter 9.5

Sequence read set analyses

9.5.1 De novo assembly

This analysis will build a set of contigs (called de novo targets) out of the reads from the sequence read sets, and map the reads against them. Together with the target sequence(s), also the target coverage matrix, the target quality matrix and the target sequence quality are calculated. Finally, for every target sequence, an assembly view is created, allowing a visual inspection of the assembly and mapping results. The core of the de novo assembly actions has been implemented through the third-party tools Velvet and Ray.

- Velvet [43], a widely adopted de Bruijn graph-based assembly program, uses both single and paired-end reads, and uses coverage information to resolve repeat regions. The final output from Velvet is the assembly file that is directly visualized in the *Assembly* panel of the Power Assembler. See 18.7.5.5.1 for more information.
- Ray [9] is a parallel short-read assembly program developed to assemble both single and paired-end reads obtained from a combination of sequencing platforms. Detailed information on this assembly program can be found under 18.7.5.5.2.

When starting the analysis from the *Main* window, one additional dialog page, compared to the dialog when starting from the sequence read set, is displayed where you can select the sequence read set experiment type that should be used for the analysis (see Figure 9.5.1). All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select <Next> to proceed.

The next step is to select the de novo assembly algorithm that should be used in the analysis (Figure 9.5.2). Select one of the algorithms, Velvet or Ray, and continue by selecting <Next>.

When importing the reads into sequence read sets, the software flags the reads as being single-ended or paired-ended. This information can be used by the de novo assembly algorithms. Depending on the read types, one can check the boxes in front of *Use the single-end reads in the data set* or *Use the paired-end reads in the data set*, or both (Figure 9.5.3). When paired-end reads are used, the expected insert length should be determined by entering the insert length as a custom value, providing the *Insert size* and *Insert size deviation*. Both de novo algorithms can also use the paired-end information without the specific insert information. In this case, select the expected insert length to be determined automatically. Press <Next> to proceed in the wizard.

The *De novo assembly settings* are important to guide the de novo assembly as they are used within repeat solving problems and discard low-coverage nodes from the final contig determination (Figure 9.5.4).

- The information on the *Expected coverage* is used for repeat resolving and can be:

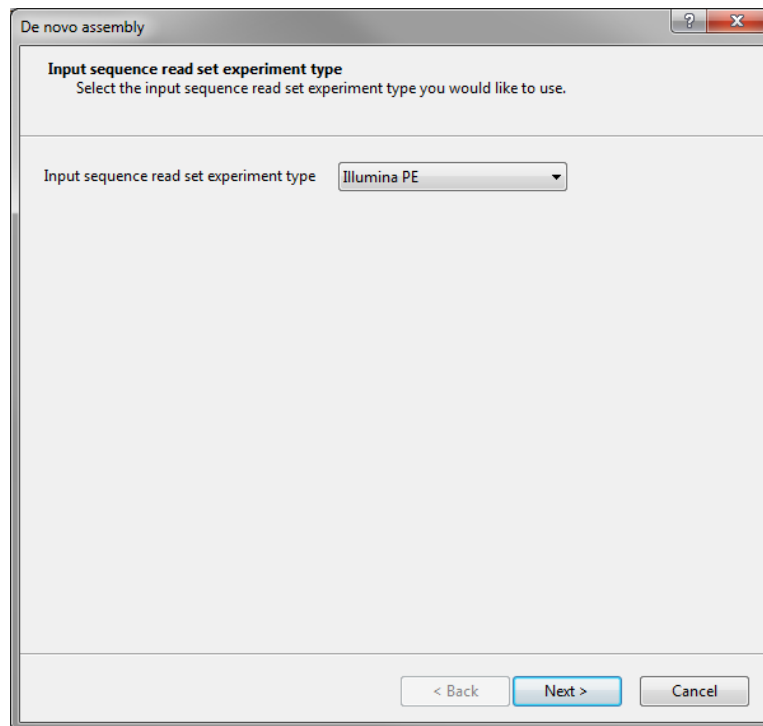


Figure 9.5.1: The *De novo assembly* wizard: Input sequence read set experiment type.

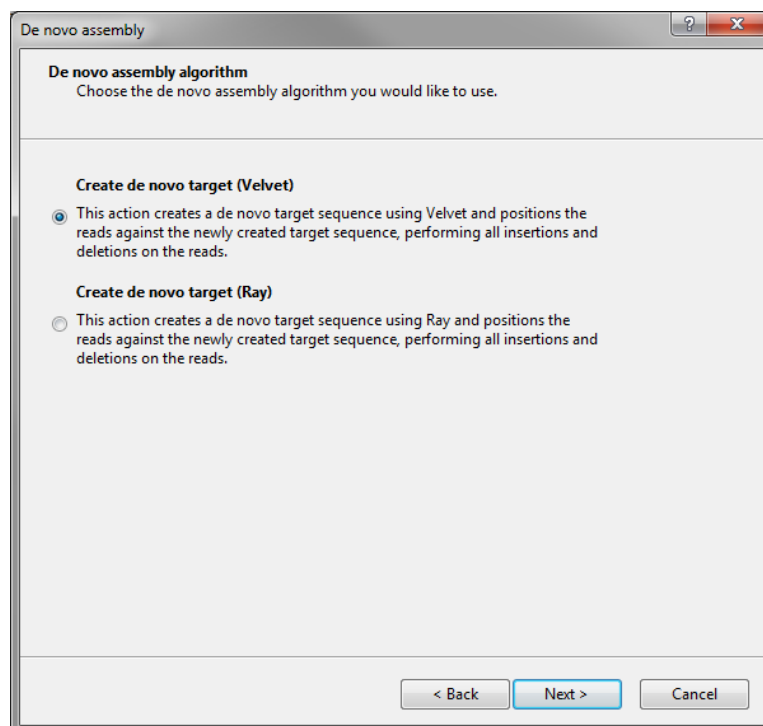


Figure 9.5.2: The *De novo assembly* wizard: De novo assembly algorithm.

- omitted from the analysis by selecting the option *don't use*.
- based on a custom coverage value by selecting the option *use custom value*. In this case, the *Custom expected coverage* should be entered.
- determined by the de novo assembly algorithm itself, by selecting the option *determine automatically*.

The screenshot shows a window titled "De novo assembly" with a subtitle "Read libraries for de novo assembly" and the instruction "Specify the read libraries to use for the de novo assembly." The window contains two sections: "Single-end reads" with an unchecked checkbox "Use the single-end reads in the data set.", and "Paired-end reads" with a checked checkbox "Use the paired-end reads in the data set." Below the paired-end reads section, there is a dropdown menu for "Expected insert length" set to "determine automatically", a text input for "Insert size" with the value "200", and a text input for "Insert size standard deviation" with the value "20". At the bottom of the window are three buttons: "< Back", "Next >", and "Cancel".

Figure 9.5.3: The *De novo assembly* wizard: Read libraries for de novo assembly.

The screenshot shows a window titled "De novo assembly" with a subtitle "De novo assembly settings" and the instruction "Specify the parameters that guide the de novo assembly procedure." The window contains two sections: "Expected coverage" with a dropdown menu set to "determine automatically" and a text input for "Custom expected coverage" with the value "10", and "Coverage cutoff" with a dropdown menu set to "determine automatically" and a text input for "Custom coverage cutoff" with the value "10". At the bottom of the window are three buttons: "< Back", "Next >", and "Cancel".

Figure 9.5.4: The *De novo assembly* wizard: De novo assembly settings.

- The information on the **Coverage cutoff** is used to exclude low-coverage nodes, and as such has an error correcting effect. The coverage cutoff can be:
 - omitted from the analysis by selecting the option *don't use*.
 - based on a custom coverage value by selecting the option *use custom value*. In this case, the

Custom coverage cutoff should be entered.

- determined by the de novo assembly algorithm itself, by selecting the option *determine automatically*.

Press <Next> to proceed to the alignment parameter settings.

In this page of the wizard, one can decide on creating a gapped or ungapped alignment and define the scores for opening or extending gaps in the alignment (Figure 9.5.5).

Figure 9.5.5: The *De novo assembly* wizard: Alignment parameters (read to de novo target).

The alignment parameter settings include:

- The **Match score**: Score for two identical bases.
- The **Mismatch score**: Score for two non-identical bases.
- **Allow gaps in the reads**: Whether or not to allow gaps in a read sequence; the **Open gap score read** is the penalty score for introducing a gap in a read sequence, and the **Extend gap score read** is the penalty score for extending an existing gap in a read sequence.
- **Allow gaps in the reference**: Whether or not to allow gaps in a reference sequence; the **Open gap score read** is the penalty score for introducing a gap in a reference sequence, and the **Extend gap score read** is the penalty score for extending an existing gap in a reference sequence.

Press <Next> to proceed.

After calculation of the alignment, the alignment quality is checked. Thereto, the following parameters are available (Figure 9.5.6):

- The **Minimum sequence identity**: minimum sequence identity for an alignment to be acceptable.

Figure 9.5.6: The *De novo assembly* wizard: Mapping assessment.

- The **Maximum penalty**: maximum penalty for an alignment to be acceptable. The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.
- The **Minimum overlap**: minimum overlap between the two sequences for an alignment to be acceptable.

Press **<Next>** to continue.

In the evaluation of the alignment, the paired-end read information can be taken into account when enforcing the paired-end read constraints. When doing so, the **Expected inter-read distance** and the **Maximum distortion of inter-read distance** can be defined (Figure 9.5.7). Press **<Next>** to proceed.

When calculating the quality matrix over the assembly, both coverage and sequence quality of the mapped reads are taken into account. The balance between both, i.e. the relative importance of the coverage and sequence quality, can be influenced by changing the **Target quality threshold** (Figure 9.5.8). Select **<Next>** to proceed.

Once the assembly is final, the target sequence can be called. Based on the coverage information, base calling thresholds can be defined (Figure 9.5.9).

- **Minimum coverage**: Minimum coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a 'N'.
- **Gap threshold**: Minimum frequency of a gap before that position is considered as a gap in the consensus sequence.
- **Single base threshold**: Minimum frequency of the most frequent base before this base is considered the unique base at a certain position.
- **Double base threshold**: Minimum frequency of the two most frequent bases before these bases are considered the two possible bases at a certain position.

The screenshot shows a window titled "De novo assembly" with a "Paired-end reads" section. The text says "Specify the way paired-end reads should be treated when evaluating their alignment." Below this is a "Paired-end handling" section with the text "When some of the reads to align are paired-end reads, and the average distance between them is known, this information can be used to improve the alignment." There is a checked checkbox labeled "Enforce paired-end read constraints". Below this are two input fields: "Expected inter-read distance" with the value "170" and "Maximum distortion of inter-read distance" with the value "20". At the bottom are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.5.7: The *De novo assembly* wizard: Paired-end reads.

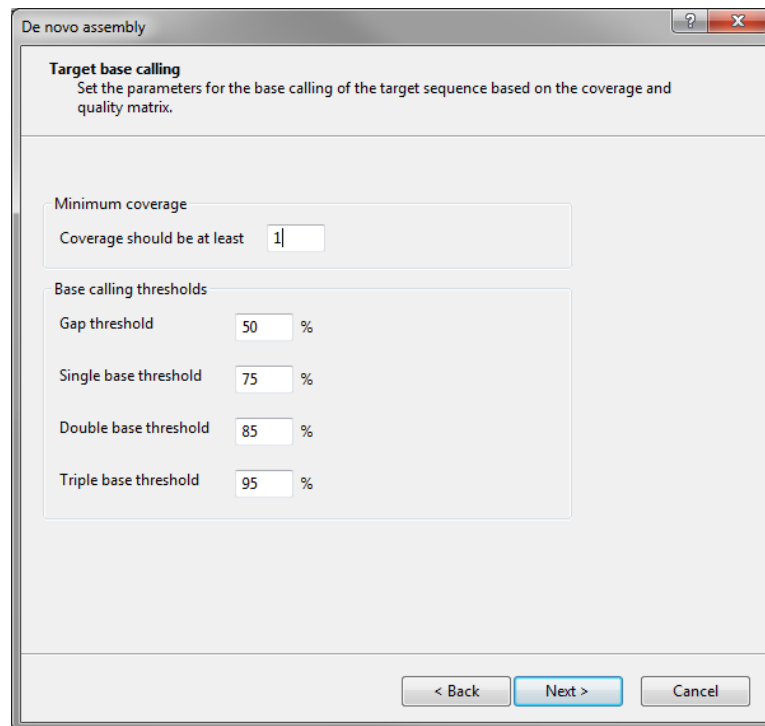
The screenshot shows a window titled "De novo assembly" with a "Target quality determination" section. The text says "Set the coverage/sequence quality balance in the calculation of the quality matrix." Below this is a "Coverage/sequence quality balance" section with the text "The target quality threshold determines the importance of the coverage in relation to the sequence quality in the calculation of the quality matrix." There is an input field labeled "Target quality threshold" with the value "10" and a "%" symbol. At the bottom are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.5.8: The *De novo assembly* wizard: Target quality determination.

- **Triple base threshold:** Minimum frequency of the three most frequent bases before these bases are considered the three possible bases at a certain position.

Press <Next> to proceed to the last dialog page.

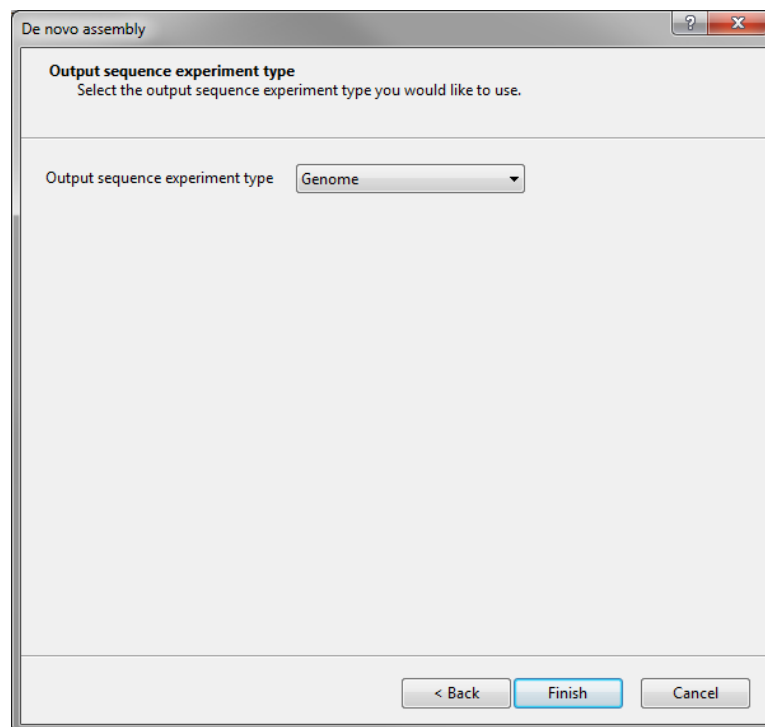
At the last page, select an output experiment type from the drop down list (Figure 9.5.10). All sequence



The screenshot shows a window titled "De novo assembly" with a "Target base calling" section. The instructions state: "Set the parameters for the base calling of the target sequence based on the coverage and quality matrix." There are two main input areas: "Minimum coverage" with a text box containing "1" and the label "Coverage should be at least", and "Base calling thresholds" which includes four rows: "Gap threshold" (50 %), "Single base threshold" (75 %), "Double base threshold" (85 %), and "Triple base threshold" (95 %). At the bottom are buttons for "< Back", "Next >", and "Cancel".

Figure 9.5.9: The *De novo assembly* wizard: Target base calling.

experiments present in the database are listed here. Select the experiment type to export the target sequence to, and press <**Finish**> to start the actual de novo assembly.



The screenshot shows a window titled "De novo assembly" with an "Output sequence experiment type" section. The instructions state: "Select the output sequence experiment type you would like to use." There is a dropdown menu labeled "Output sequence experiment type" with "Genome" selected. At the bottom are buttons for "< Back", "Finish", and "Cancel".

Figure 9.5.10: The *De novo assembly* wizard: Output sequence experiment type.

For each of the analyses, a power assembly project is created. The project consists of a combination of predefined actions, each having their specific parameter settings. Detailed information on these parameter

settings can be found in [18.5](#).

9.5.2 Resequencing assembly

This analysis will build a set of contigs (called targets) out of the reads from the sequence read sets by trying to position the reads against the reference sequences, and creates target sequence records from this positioning information. Together with the contigs, also the coverage matrix, the quality matrix and the sequence quality of the mappings are calculated. Finally, for every contig sequence, an assembly view is created, allowing a visual inspection of the assembly and mapping results. All mapping actions can handle more than one reference sequence at the same time.

The predefined mapping actions come in two flavors, depending on the application.

- In the predefined action **Create target (no inserts)**, it is assumed that the target sequence follows exactly the same frame as the reference sequence. In this case, the target sequence is supposed to be identical to the reference sequence, apart from individual base calls and small deletions. The alignment of the reads with respect to the references can be gapped or ungapped, but all insertions and deletions are performed on the reads.
- In the predefined action **Create target (with inserts)**, the reference sequence is a more distant relative of the target sequence, and not only the reproduction of the reference sequence is the goal, but also the local structural variation between reference and target is of interest. The alignment procedure of the reads with respect to the references should allow gaps. Insertions are performed both on the reads and on the reference sequence, thus building the target sequence out of it.

When starting the analysis from the *Main* window, one additional dialog page, compared to the dialog when starting from the sequence read set, is displayed where you can select the sequence read set experiment type that should be used for the analysis (see [Figure 9.5.11](#)). All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select **<Next>** to proceed.

After selecting the sequence read set to be used for the analysis, the next step is to define the template genome that will act as assembly template. Press **<Add...>** to select the appropriate entry from the database and, under **Input reference experiment type**, select the sequence experiment type from the drop down list that contains the genome information ([Figure 9.5.12](#)). Continue by selecting **<Next>**.

Select the resequencing assembly algorithm that should be used in the analysis ([Figure 9.5.13](#)) and continue by selecting **<Next>**.

In this page of the wizard, one can decide on creating a gapped or ungapped alignment and define the scores for opening or extending gaps in the alignment ([Figure 9.5.14](#)).

The alignment parameter settings include:

- The **Match score**: Score for two identical bases.
- The **Mismatch score**: Score for two non-identical bases.
- **Allow gaps in the reads**: Whether or not to allow gaps in a read sequence; the **Open gap score read** is the penalty score for introducing a gap in a read sequence, and the **Extend gap score read** is the penalty score for extending an existing gap in a read sequence.
- **Allow gaps in the reference**: Whether or not to allow gaps in a reference sequence; the **Open gap score read** is the penalty score for introducing a gap in a reference sequence, and the **Extend gap score read** is the penalty score for extending an existing gap in a reference sequence.

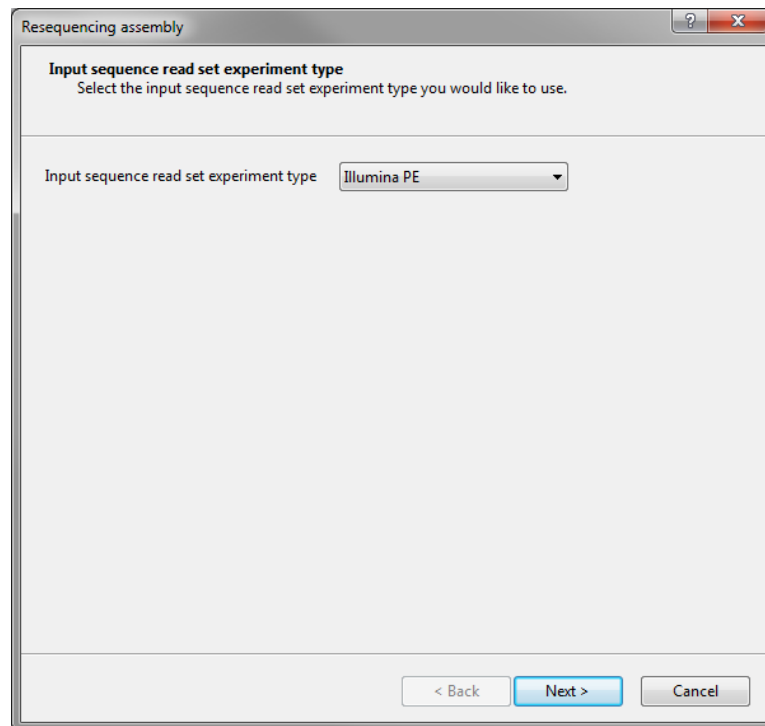


Figure 9.5.11: The *Resequencing assembly* wizard: Input sequence read set experiment type.

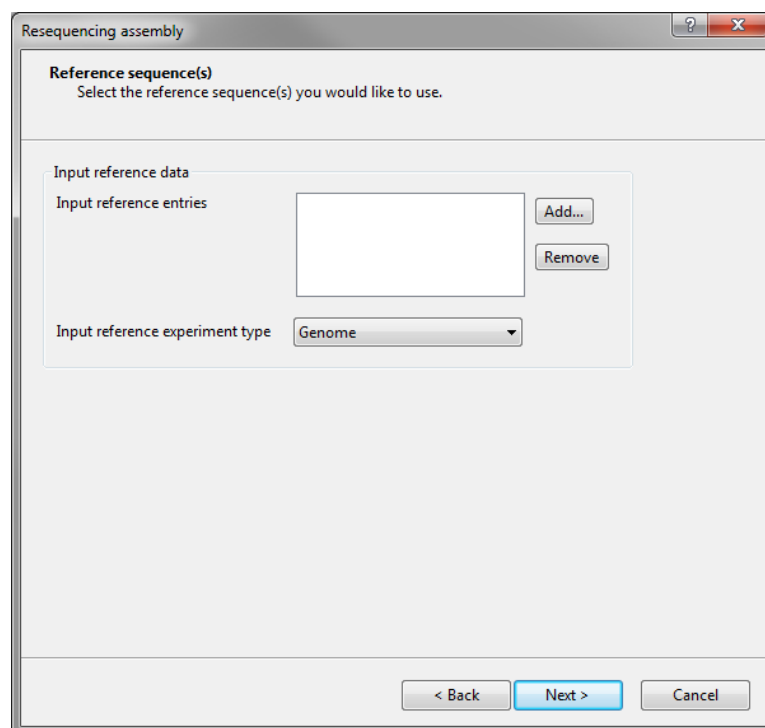


Figure 9.5.12: The *Resequencing assembly* wizard: Reference sequence(s).

Press <*Next*> to proceed.

After calculation of the alignment, the alignment quality is checked. The following parameters are available (Figure 9.5.15):

- The *Minimum sequence identity*: Minimum sequence identity for an alignment to be acceptable.

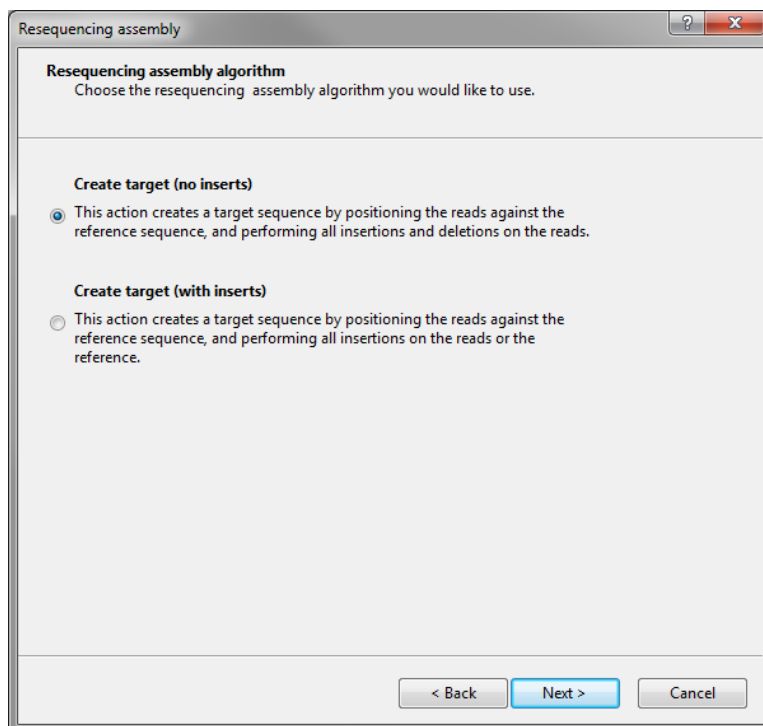


Figure 9.5.13: The *Resequencing assembly* wizard: Resequencing assembly algorithm.

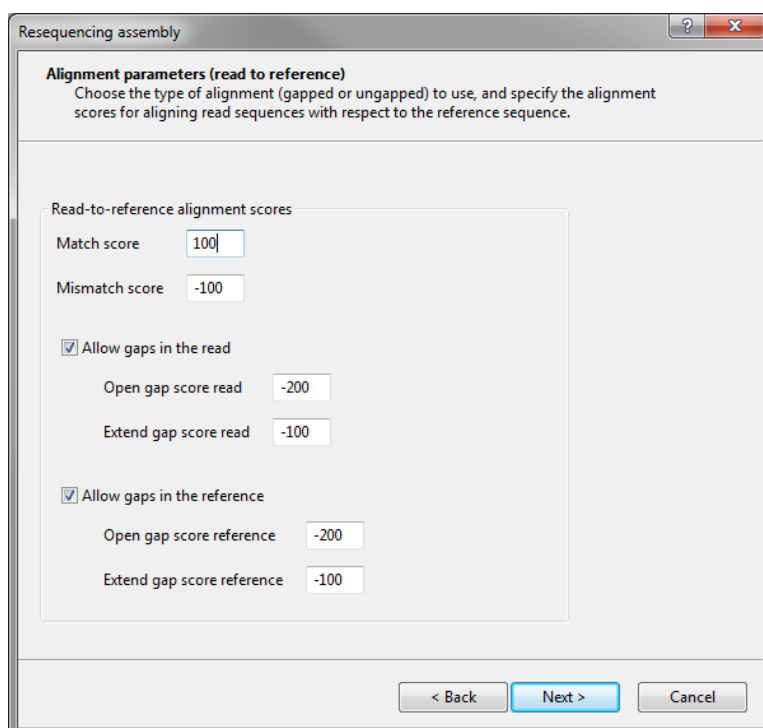


Figure 9.5.14: The *Resequencing assembly* wizard: Alignment parameters (read to reference).

- The **Maximum penalty**: Maximum penalty for an alignment to be acceptable. The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.
- The **Minimum overlap**: Minimum overlap between the two sequences for an alignment to be accept-

Resequencing assembly

Mapping assessment
Specify the parameters that determine whether an alignment is acceptable or not.

Sequence identity

Minimum sequence identity %

Maximum penalty

The total penalty is the sum of all negative contributions to the alignment score. This score should not exceed the penalty threshold.

Maximum penalty

Minimum overlap

For a reliable alignment, the overlap region between the two sequences should be big enough.

Minimum overlap %

< Back Next > Cancel

Figure 9.5.15: The *Resequencing assembly* wizard: Mapping assessment.

able.

Press *<Next>* to continue.

Resequencing assembly

Paired-end reads
Specify the way paired-end reads should be treated when evaluating their alignment.

Paired-end handling

When some of the reads to align are paired-end reads, and the average distance between them is known, this information can be used to improve the alignment.

☒ Enforce paired-end read constraints

Expected inter-read distance

Maximum distortion of inter-read distance

< Back Next > Cancel

Figure 9.5.16: The *Resequencing assembly* wizard: Paired-end reads.

In the evaluation of the alignment, the paired-end read information can be taken into account when enforcing the paired-end read constraints. When doing so, the *Expected inter-read distance* and the *Maximum*

distortion of inter-read distance can be defined (Figure 9.5.16). Press <Next> to proceed.

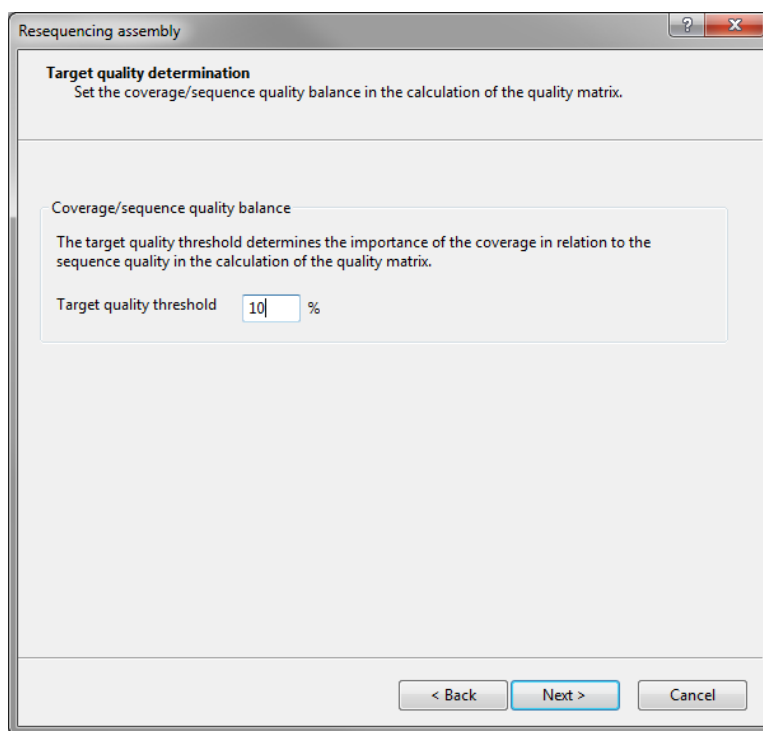


Figure 9.5.17: The *Resequencing assembly* wizard: Target quality determination.

When calculating the quality matrix over the assembly, both coverage and sequence quality of the mapped reads are taken into account. The balance between both, i.e. the relative importance of the coverage and sequence quality, can be influenced by changing the **Target quality threshold** (Figure 9.5.17). Select <Next> to proceed.

Once the assembly is final, the target sequence can be called. Based on the coverage information, base calling thresholds can be defined (Figure 9.5.18).

- **Minimum coverage:** Minimum coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a gap.
- **Gap threshold:** Minimum frequency of a gap before that position is considered as a gap in the consensus sequence.
- **Single base threshold:** Minimum frequency of the most frequent base before this base is considered the unique base at a certain position.
- **Double base threshold:** Minimum frequency of the two most frequent bases before these bases are considered the two possible bases at a certain position.
- **Triple base threshold:** Minimum frequency of the three most frequent bases before these bases are considered the three possible bases at a certain position.

Press <Next> to proceed to the last dialog page.

At the last page, select an output experiment type from the drop down list (Figure 9.5.19). All sequence experiments present in the database are listed here. Select the experiment type to export the concatenated target sequence to, and press <Finish> to start the actual resequencing assembly.

The screenshot shows a window titled "Resequencing assembly" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area is titled "Target base calling" with a subtitle "Set the parameters for the base calling of the target sequence based on the coverage and quality matrix." Below this, there are two sections. The first section, "Minimum coverage", contains a label "Coverage should be at least" followed by a text input field containing the number "1". The second section, "Base calling thresholds", contains four rows of settings: "Gap threshold" with a value of "50", "Single base threshold" with a value of "75", "Double base threshold" with a value of "85", and "Triple base threshold" with a value of "95". Each value is in a text input field followed by a percentage sign. At the bottom of the window, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.5.18: The *Resequencing assembly* wizard: Target base calling.

The screenshot shows a window titled "Resequencing assembly" with a standard Windows-style title bar. The main content area is titled "Output sequence experiment type" with a subtitle "Select the output sequence experiment type you would like to use." Below this, there is a label "Output sequence experiment type" followed by a dropdown menu currently showing "Genome". At the bottom of the window, there are three buttons: "< Back", "Finish" (highlighted in blue), and "Cancel".

Figure 9.5.19: The *Resequencing assembly* wizard: Output sequence experiment type.

For each of the analyses, a power assembly project is created. The project consists of a combination of predefined actions, each having their specific parameter settings. Detailed information on these parameter settings can be found in [18.5](#).

9.5.3 Map to reference

Using *Analysis* > *Sequence read set types* > *Map to reference*, sequence read sets from the selected entries are mapped against a reference sequence. The resulting sequences can then be used in e.g. wgSNP analysis (see 8.10). Calling this command opens the *Map to reference* dialog box.

The *Map to reference* dialog box contains several tabs. The *Input tab* will be displayed initially (see Figure 9.5.20).

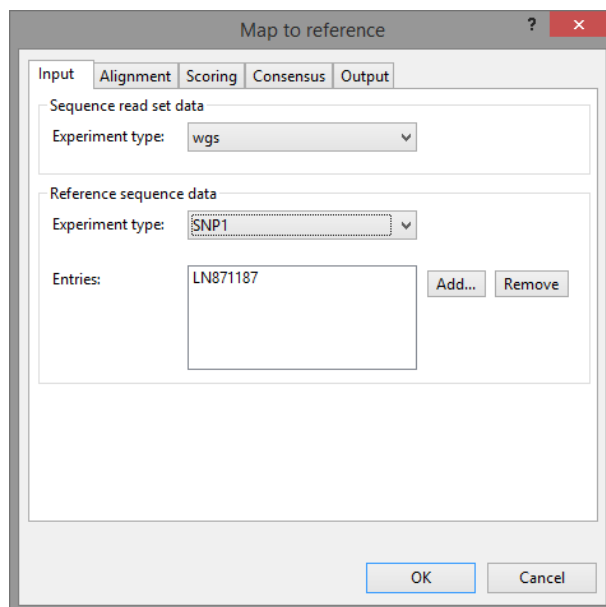


Figure 9.5.20: The *Map to reference* dialog box, *Input tab*.

In this tab, the input experiments should be specified. The map to reference action always works on the selected entries and on the sequence read set experiment type specified under *Sequence read set data*.

The reference sequence(s) to map against can be selected under *Reference sequence data*. If a reference mapped sequence type (see 8.1.1 and 8.1.2.2 for more information) is selected from the *Experiment type* drop-down list, the corresponding reference sequence will automatically be selected in the *Entries* list. Alternatively, one or more entries can be selected manually using the <Add> and <Remove> buttons.

In the *Alignment tab*, settings for the alignment algorithm can be specified (see Figure 9.5.21).

Checking *Perform gapped alignment* will allow gaps in the read sequences as well as gaps in the reference sequence(s).

The *Alignment scores* include:

- **Match score:** Score for two identical bases.
- **Mismatch score:** Score for two non-identical bases.
- **Open gap score read:** The penalty score for introducing a gap in a read sequence.
- **Extend gap score read:** The penalty score for extending an existing gap in a read sequence.

The latter two parameters are disabled when *Perform gapped alignment* is unchecked.

The *Scoring tab* contains the settings to assess the initial alignment (see Figure 9.5.22).

Following *Score assessment* parameters are available:

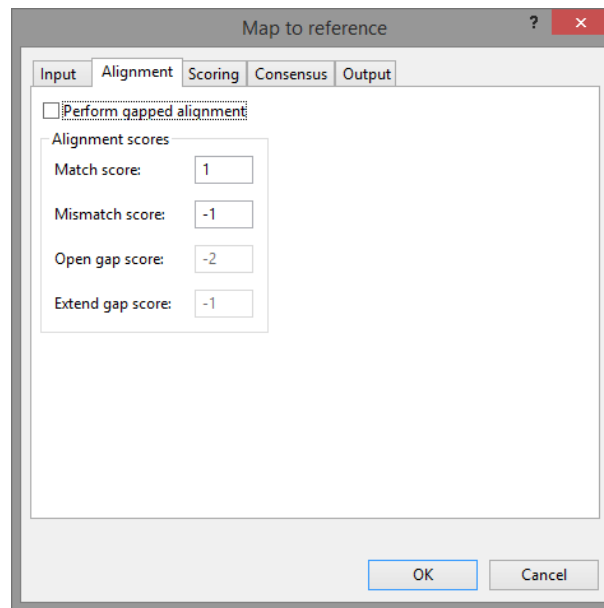


Figure 9.5.21: The *Map to reference* dialog box, *Alignment* tab.

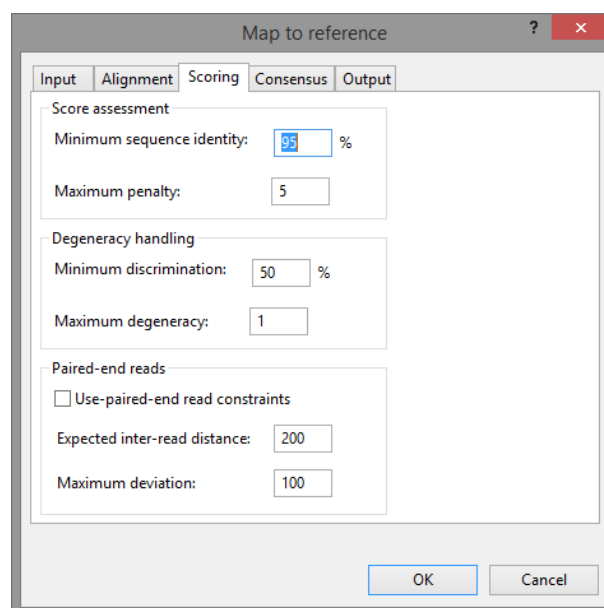


Figure 9.5.22: The *Map to reference* dialog box, *Scoring* tab.

- **Minimum sequence identity:** Minimum sequence identity for an alignment to be acceptable.
- **Maximum penalty:** Maximum penalty for an alignment to be acceptable. The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score.

Following **Degeneracy handling** parameters are available:

- **Minimum discrimination:** Minimum mapping discrimination allowed. The mapping discrimination quantifies the difference in sequence identity between the best position and the second best position.
- **Maximum degeneracy:** Maximum mapping degeneracy, i.e. the maximum number of positions where a read can be mapped on the reference sequence.

In the evaluation of the alignment, the paired-end read information can be taken into account by checking the *Use-paired-end read constraints* option. When doing so, the *Expected inter-read distance* and the *Maximum deviation* can be defined.

The *Consensus tab* groups the settings for calculating a consensus sequence based on the final alignment (see Figure 9.5.23).

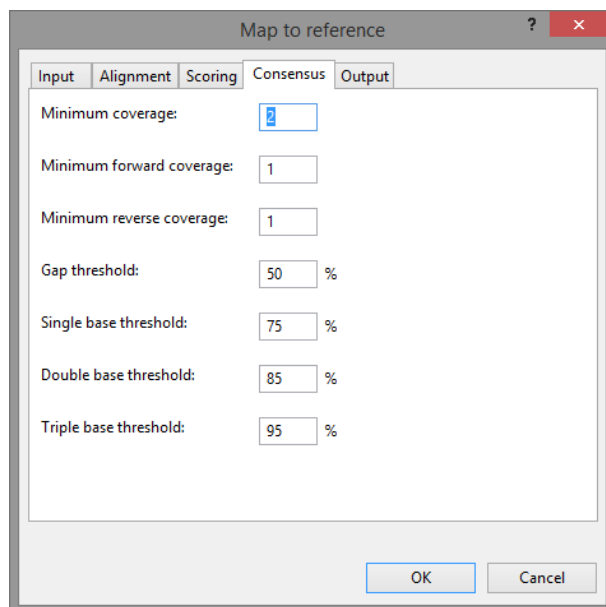


Figure 9.5.23: The *Map to reference* dialog box, *Consensus tab*.

Following options are available:

- **Minimum coverage:** Minimum coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a gap.
- **Minimum forward coverage:** Minimum forward coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a gap.
- **Minimum reverse coverage:** Minimum reverse coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a gap.
- **Gap threshold:** Minimum frequency of a gap before that position is considered as a gap in the consensus sequence.
- **Single base threshold:** Minimum frequency of the most frequent base before this base is considered the unique base at a certain position.
- **Double base threshold:** Minimum frequency of the two most frequent bases before these bases are considered the two possible bases at a certain position.
- **Triple base threshold:** Minimum frequency of the three most frequent bases before these bases are considered the three possible bases at a certain position.

In the *Output tab*, the experiments should be specified in which the consensus sequences should be stored (see Figure 9.5.24).

The sequences will always be saved for the selected entries on which the map to reference action is ran.

The default option is *Save to experiment type of reference*, i.e. the same experiment type will be used as specified for the reference sequence in the *Input tab*. Alternatively, check *Manually specify output experiment type* and use the buttons to add one or more experiment types to the list on the right.

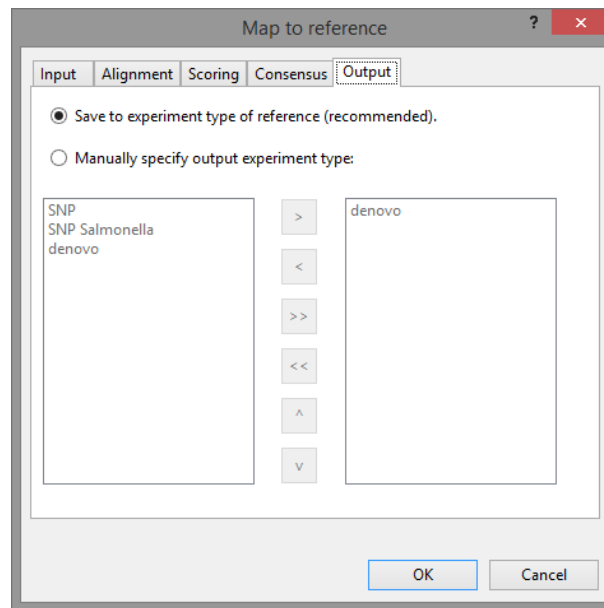


Figure 9.5.24: The *Map to reference* dialog box, *Output* tab.

9.5.4 Run custom template

A custom Power Assembler template can be run on sequence read sets from the selected entries via **Analysis** > **Sequence read set types** > **Run custom template...**. This action opens the *Template selection* dialog box (see Figure 9.5.25).

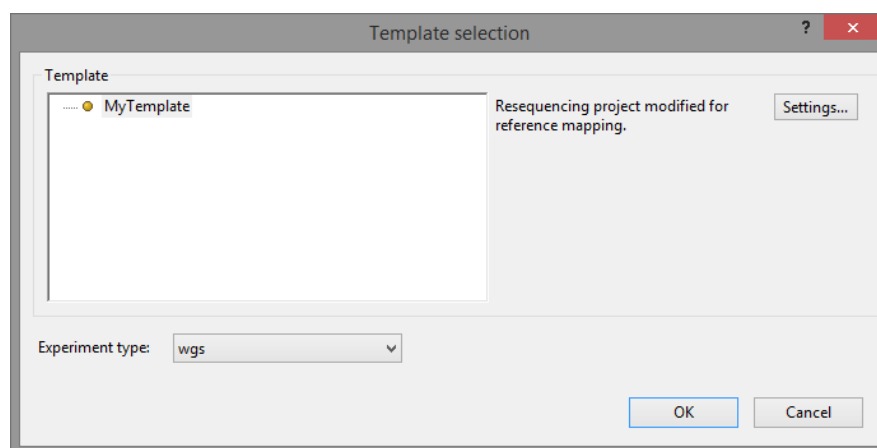


Figure 9.5.25: The *Template selection* dialog box.

From the **Template** list, a Power Assembly template can be selected. A description (if available) will appear on the right. The list contains the default sequence read set analyses (de novo assembly, resequencing assembly and map to reference) and any custom Power Assembly template present in the database. Pressing <**Settings**> will show the runtime parameters for the selected template.

The sequence read set experiment type on which to run the Power Assembly template can be picked from the **Experiment type** drop-down list.



A custom Power Assembly template is generated in the *Power assembly* window via **File** > **Store pipeline as template...** (see 18.3.2.2).

9.5.5 Single-sample diversity analysis

In this analysis, the alpha-diversity of a single sample is assessed. For the operational taxonomic units (further called OTUs) obtained, the within-sample diversity, the community evenness, the community richness and the community diversity indices are calculated (see 19.4.2).

For this analysis, BioNumerics makes use of the *mothur* [35] project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). BioNumerics uses the flexibility of the algorithms incorporated in *mothur* and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results.

When starting the analysis from the *Main* window, one additional dialog page, compared to the dialog when starting from the sequence read set, is displayed where you can select the sequence read set experiment type that should be used for the analysis. All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select **<Next>** to proceed.

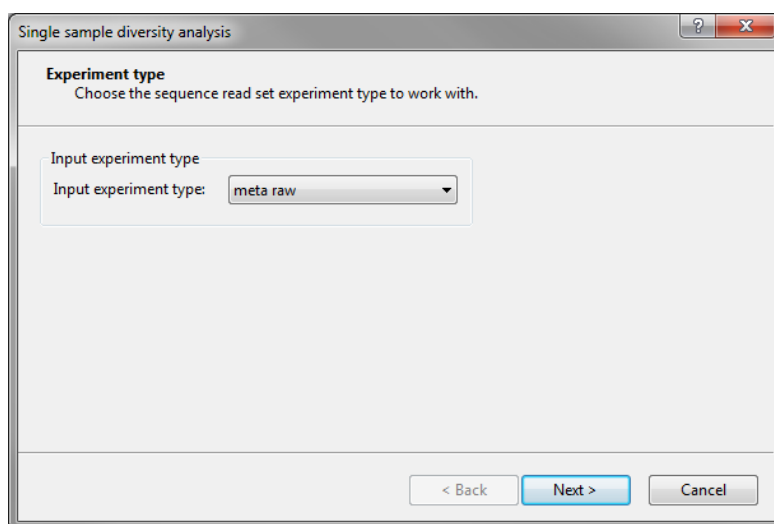


Figure 9.5.26: The *Single-sample diversity analysis* dialog box: Experiment type settings.

Similarly, when starting the analysis from within the *Metagenomics* window, another additional dialog page, compared to the dialog when starting from the sequence read set, is displayed. In this dialog, the entry information and the input sequence read set experiment type need to be specified. Press **<Add entry>** to select the entry to analyze. This opens the *Select entry* dialog box where the entry can be highlighted in the database list and press **<OK>** to return to the *Single sample diversity analysis* wizard where the sequence read set experiment type to be used for the analysis can be selected from the drop down list with available sequence read set experiment types present in the database. Select **<Next>** to proceed.

When starting the metagenomics analysis from the *Sequence read set experiment* window, the first page of the *Single sample diversity analysis* wizard asks for the OTU determination settings. When starting the analysis from the *Main* window or the *Metagenomics* window, this will be the second page of the wizard. The OTUs can be determined three ways. The first two options are similar to the ones described under *Identification against a taxonomic database* (see 9.5.6). Additionally, the third option will define OTUs based on sequence clustering and a similarity cutoff value on these sequence clustering results.

- The option **Determine OTUs by sequence clustering** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. Based upon the OTUs defined from this cluster analysis, the diversity analyses are calculated.
- The option **Determine OTUs by taxonomic identification** will determine the unique sequences in the sequence read set, identify the sequences against the reference taxonomy and create OTUs from the

The screenshot shows a window titled "Single sample diversity analysis" with a question mark icon and a close button. The main heading is "Entry and experiment type" with the instruction "Choose the Entry and the sequence read set experiment type to work with." Below this, there is a section for "Entry key" with a text input field and an "Add entry..." button. Another section for "Input experiment type" shows a dropdown menu currently set to "meta raw". At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.5.27: The *Single sample diversity analysis* wizard: Entry and Experiment type settings.

The screenshot shows a window titled "Single sample diversity analysis" with a question mark icon and a close button. The main heading is "OTU determination" with the instruction "Choose the way the operational taxonomic units are defined." Below this, there is explanatory text: "OTUs can be determined in two ways: either from the sequence data itself (by performing a sequence clustering), or from the phylotypic levels in a taxonomic database, once the sequences have been identified. In case the OTUs are determined by sequence clustering, a consensus taxonomy can be calculated for each OTU." There are three radio button options: "Determine OTUs by sequence clustering." (which is selected), "Determine OTUs by taxonomic identification.", and "Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs." At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 9.5.28: The *Single sample diversity analysis* wizard: OTU determination settings.

taxonomic identification results. Based upon the taxonomically defined OTUs, the diversity analyses are calculated.

- The option ***Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs*** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. For each of the reads in the cluster the taxonomic identification is calculated and based on these results, the cluster consensus taxonomy is defined. The OTUs derived from this consensus taxonomy are then used as input to calculate the diversity analyses.

Select one of these options and press <*Next*> to proceed.

When taxonomic identification is involved, the taxonomic identification settings require the taxonomic reference database to be selected from the drop down list, and the start and end levels of the taxonomic identification for the current analysis need to be specified. Select <*Next*> to proceed.



See [19.1.2](#) for more information on the taxonomic reference database.

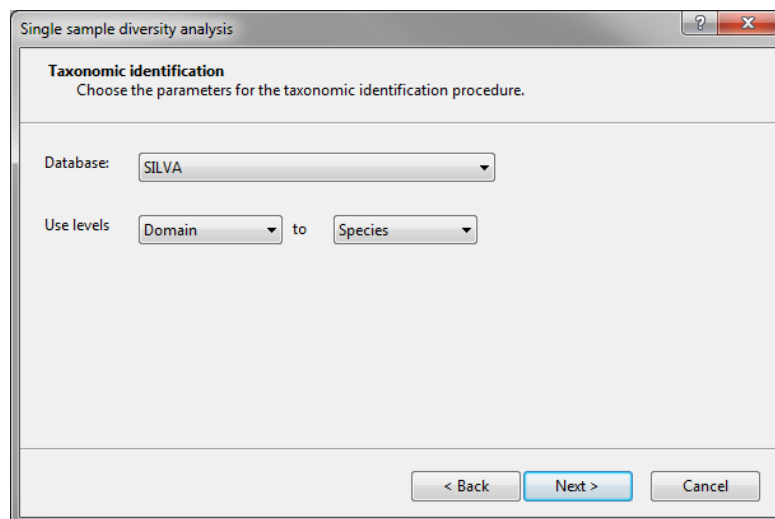


Figure 9.5.29: The *Single sample diversity analysis* wizard: Taxonomic identification settings.

When OTU determination is based on sequence clustering, these settings need to be defined in the *Single sample diversity analysis* wizard: Sequence clustering options. From this dialog, the sequence identity threshold needs to be set. The sequence identity threshold is a cutoff value applied to the clustering of the reads. Using this threshold, only sequences with a small distance will be clustered together, and not the complete matrix. This may result in a significant speedup of the cluster analysis involved in the OTU preparation. Next to the threshold, the reference alignment needs to be selected from the drop down list. Select **<Next>** to proceed.



See [19.1.1](#) for more information on the reference alignment.

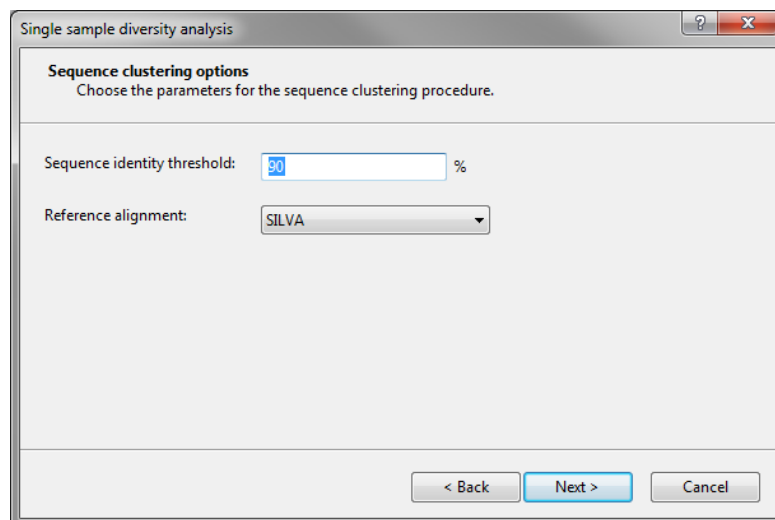


Figure 9.5.30: The *Single sample diversity analysis* wizard: Sequence clustering options.

When launching the single sample diversity analysis, one can specify the set of calculators that should be used in the analysis from the *Single sample diversity analysis* wizard: Diversity calculators settings. The options *Use default calculators* uses a set of calculators defined upon installation. The different calculators activated as default can be visualized by selecting the option *Use default calculators* and pressing the **<Custom calculators...>** button. This opens the *Custom Calculators* dialog box where the different summary diversity calculators, the collector curve calculators and the rarefaction curve calculators can be

viewed.

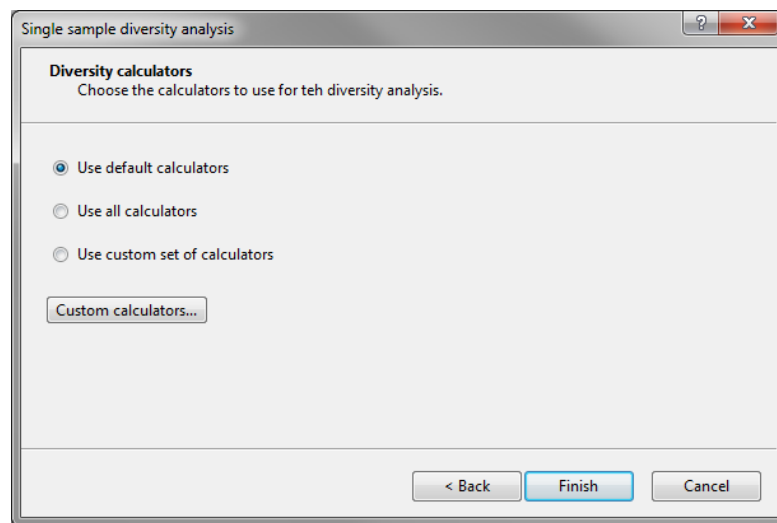


Figure 9.5.31: The *Single sample diversity analysis* wizard: Diversity calculators settings.

When the default selection of calculators is not satisfactory, one should choose the option *Use custom set of calculators*. In this case, opening the *Custom Calculators* dialog box by pressing **<Custom calculators...>** allows to select/unselect a custom set of summary diversity, collector curve and the rarefaction curve calculators that will be used in the current analysis.

A third option is to *Use all calculators*. When checking this option, all calculators displayed in the *Custom Calculators* dialog box are calculated in the analysis at hand. Press **<Finish>** to complete the dialog and create the analysis in the *Metagenomics* window.

When calculating a single sample diversity analysis, the different summary diversity calculators, the collector curve calculators and the rarefaction curve calculators can be viewed from the *Custom Calculators* dialog box. Initially, the selection shows the default activated calculators. When using a custom set of calculators, these calculators need to be defined from the same dialog by (un-)selecting the check boxes in front of the different calculators. Press **<OK>** to update the calculator selection in the analysis or press **<Cancel>** to leave the dialog without making any modifications on the predefined calculators.

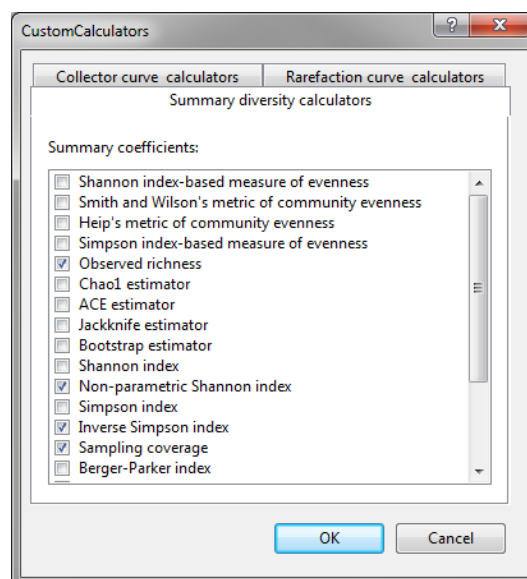


Figure 9.5.32: The *Custom Calculators* dialog box.

See 19.4.2.6 for more details on the parameter settings for the single-sample diversity analysis.

9.5.6 Identification against a taxonomic database

In this analysis, all sample reads are identified against a taxonomic database. The operational taxonomic units (further called OTUs) can be defined in two ways: either directly from the phylotypic levels defined in the taxonomic database used for the automated identification of the reads, or from the sequence data itself by performing a sequence clustering first, and then determining the consensus taxonomy per cluster. Finally, for each sample the OTU abundances are stored as characters in the database (see 19.4.1).

For this analysis, BioNumerics makes use of the *mothur* [35] project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). BioNumerics uses the flexibility of the algorithms incorporated in *mothur* and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results.

When starting the analysis from the *Main* window, one additional dialog page is displayed where you can select the sequence read set experiment type that should be used for the analysis. All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select <Next> to proceed.

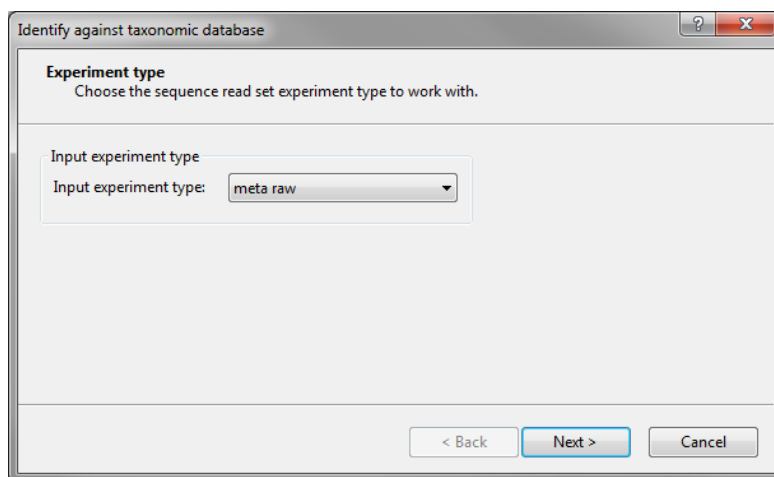


Figure 9.5.33: The *Identify against taxonomic database* wizard: Experiment type settings.

Similarly, when starting the analysis from within the *Metagenomics* window, another additional dialog page is displayed (see below). In this dialog, the entry information and the input sequence read set experiment type need to be specified. Press <Add entry> to select the entry to analyze. This opens the *Select entry* dialog box where the entry can be highlighted in the database list, press <OK> to return to the *Single sample diversity analysis* wizard where the sequence read set experiment type to be used for the analysis can be selected from the drop down list with available sequence read set experiment types present in the database. Select <Next> to proceed.

When starting the metagenomics analysis from the *Sequence read set experiment* window, the first page of the *Identify against taxonomic database* wizard asks for the OTU determination settings (see below). When starting the analysis from the *Main* window or the *Metagenomics* window, this will be the second page of the wizard. The OTUs can be determined two ways.

- The option **Determine OTUs by taxonomic identification** will determine the unique sequences in the sequence read set, identify the sequences against the reference taxonomy and create OTUs from the taxonomic identification results.

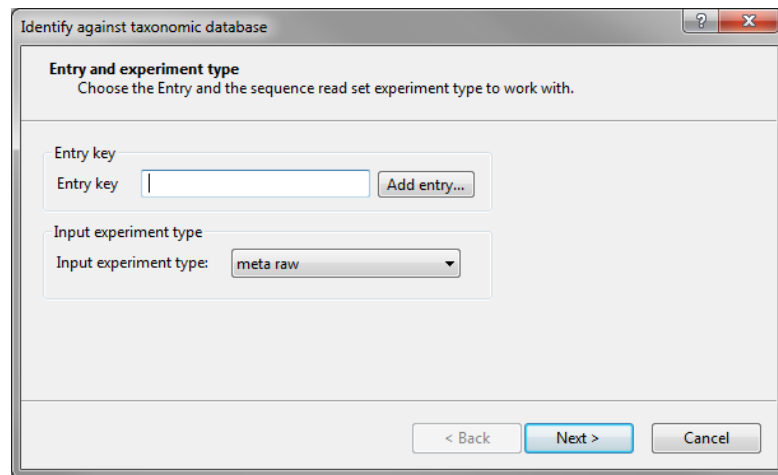


Figure 9.5.34: The *Identify against taxonomic database* wizard: Entry and Experiment type settings.

- The option ***Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs*** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. For each of the reads in the cluster the taxonomic identification is calculated and based on these results, the cluster consensus taxonomy is defined.

Select one of the two options and press **<Next>** to proceed.

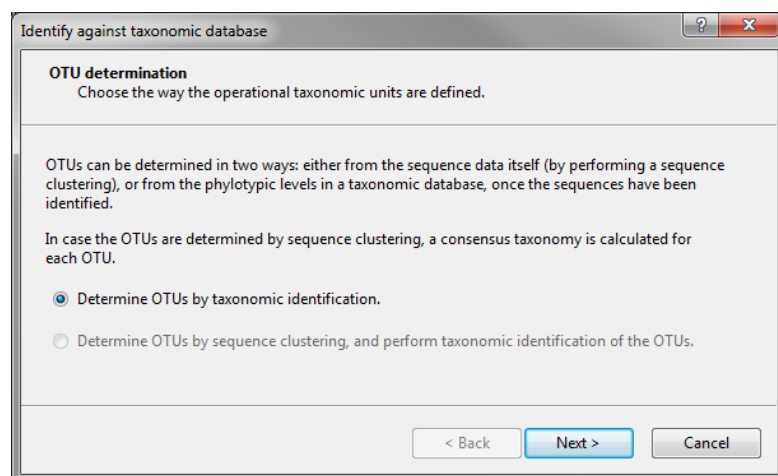


Figure 9.5.35: The *Identify against taxonomic database* wizard: OTU determination settings.

The taxonomic identification settings only require the taxonomic reference database to be selected from the drop down list and to specify the start and end level of the taxonomic identification for the current analysis. Select **<Next>** to proceed.



See [19.1.2](#) for more information on the taxonomic reference database.

The output experiment type settings define to which character experiment type the OTU abundance results will be saved to the database. You can leave the default settings “Choose automatically”. Doing so, a character set is created for each unique combination of a taxonomic database and a taxonomic level. If desired, a custom prefix can be added to be used in front of the unique character experiment type names that will be created. Leaving all settings default enables you to save the analysis results for multiple entries based on the same reference taxonomy to the same character experiment types, allowing a follow-up analysis on

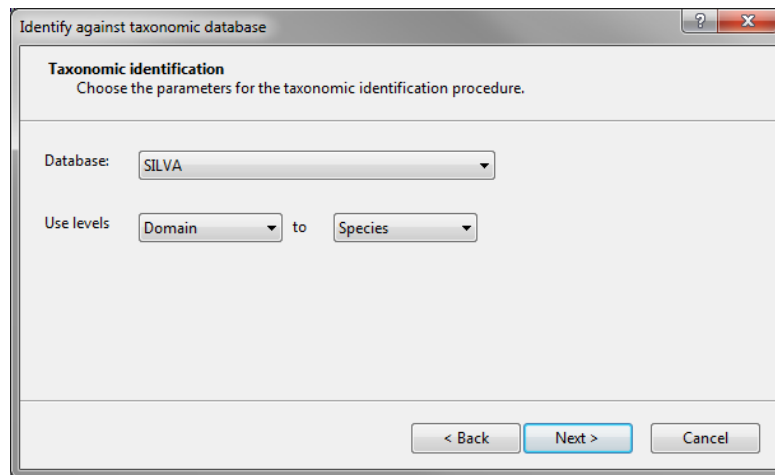


Figure 9.5.36: The *Identify against taxonomic database* wizard: Taxonomic identification settings.

these character values in e.g. the *Comparison* window. Select <**Finish**> to complete the wizard, and to start the analysis in the *Metagenomics* window.

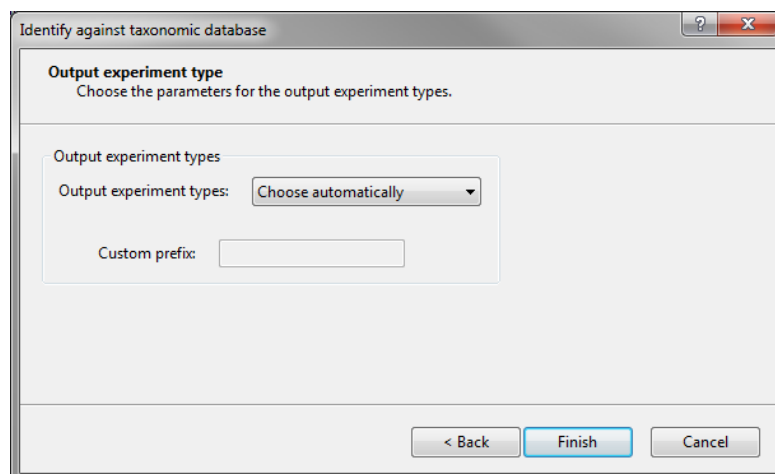


Figure 9.5.37: The *Identify against taxonomic database* wizard: Output experiment type settings.

See [19.4.1.5](#) for more details on the parameter settings for the identification against a taxonomic database.

Chapter 9.6

Exporting sequence read sets

With the *Export sequence read sets* option, listed under the topic *Sequence read sets data* in the *Export* dialog box (see Figure 9.6.1), sequence read sets can be exported to FASTQ or FASTA formatted files. Paired-end data will be split over two files.

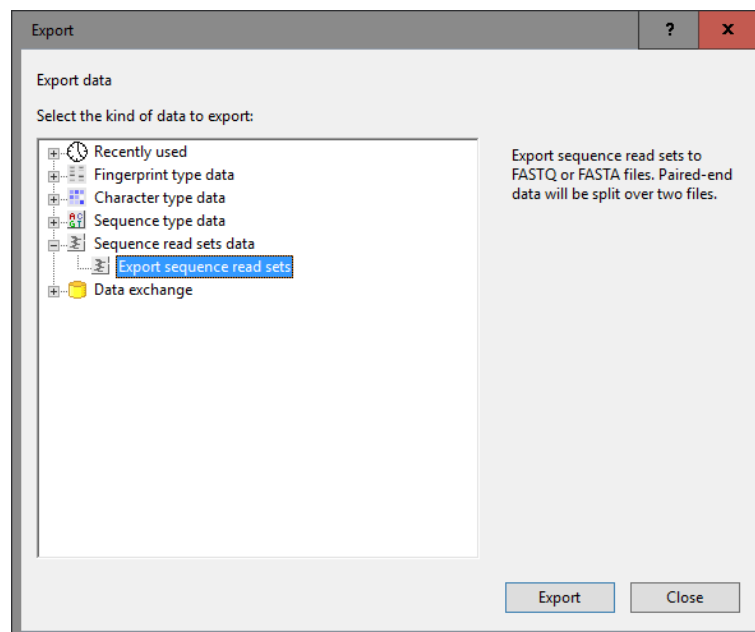


Figure 9.6.1: The *Export sequence read sets* option in the *Export* dialog box.

In the *Database entries* panel of the *Main* window, select the entries to export. A single entry can be selected by holding the **Ctrl**-key and left-clicking (**CTRL+click**). Check boxes for selected entries are indicated as ☒. In order to select a group of entries, hold the **Shift**-key and click on another entry. All the entries in the database can be selected using *Edit* > *Select all* (**Ctrl+A**).

Selecting *Export sequence read sets* under *Sequence read sets data* in the *Export* dialog box and pressing <*Export*> calls the *Export sequence read sets* dialog box (see Figure 9.6.2).

Browse for a new or existing export location with the <*Browse*> button.

All sequence read set types defined in the database are displayed. Select the *Sequence read set type(s)* to export. To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.

Select the *Export format*. The choice is offered between *FASTQ* and *FASTA*.

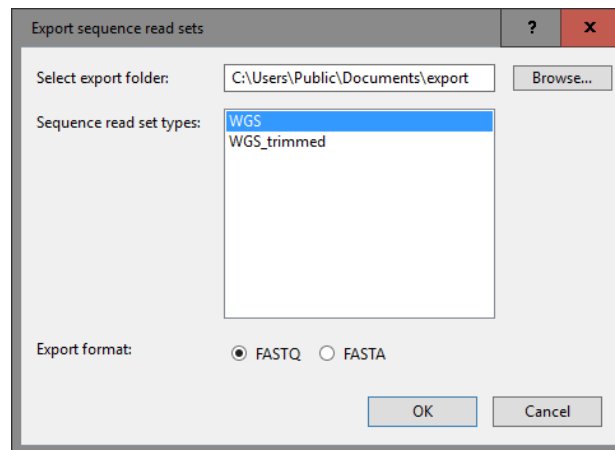


Figure 9.6.2: The *Export sequence read sets* dialog box.

Pressing <**OK**> exports the data to the selected location.

Each single-end sequence will be exported to one gzipped fastq or fasta file depending on the selected format. Each file name is composed of the **Key** followed by the name of the sequence read sets experiment.

Paired-end sequences will be exported to two gzipped fastq or fasta files which contain each one end of the paired-end reads. The file names are composed of the **Key** followed by the name of the sequence read sets experiment and the suffix _1 or _2.

Part 10

Matrix types

Chapter 10.1

Setting up matrix type experiments

10.1.1 Defining a new matrix type

To create a new matrix type, highlight the *Experiment types* panel in the *Main* window and select **Edit > Create new object...** (+). In the *Create a new experiment type* dialog box, click on **Matrix type** and press <OK>. This will display the *Create new matrix type* dialog box (see Figure 10.1.1).

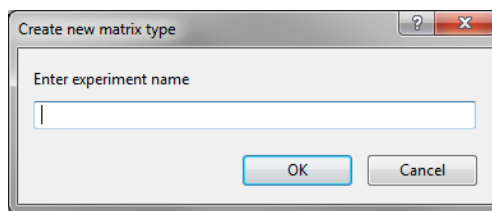


Figure 10.1.1: The *Create new matrix type* dialog box.

The dialog prompts you to enter a **Matrix type name**. Enter a name for the new matrix type and press <OK>. The new matrix type will appear in the *Experiment types* panel.

10.1.2 Matrix comparison settings

In the *Matrix type* window, the comparison settings defined for the matrix type are shown in *Comparison settings* panel. These settings can be accessed with **Settings > Comparison settings...** (⚙️) in the *Matrix type* window, but also in the *Comparison* window. See 10.2 for a detailed explanation.

10.1.3 Importing matrix data

10.1.3.1 Introduction

Using the **Import similarity matrix** option, listed under the topic **Matrix type data** in the *Import* dialog box (see Figure 10.1.2), similarity and distance matrices present in text files can be imported in a BioNumerics database and linked to new or existing database entries.

Each text file should contain entry information in the first column and similarity or distance values in the other columns. The similarity or distance values should be separated by tabs (see Figure 10.1.3). Partial matrices (e.g. DNA homology matrices) are accepted by the plugin.

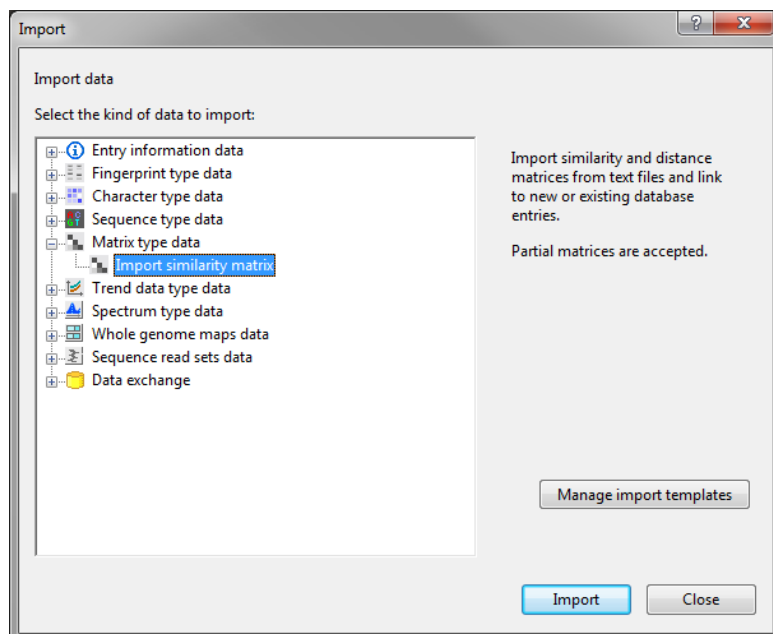


Figure 10.1.2: Import similarity matrix.

Similarity_matrix.txt - Notepad

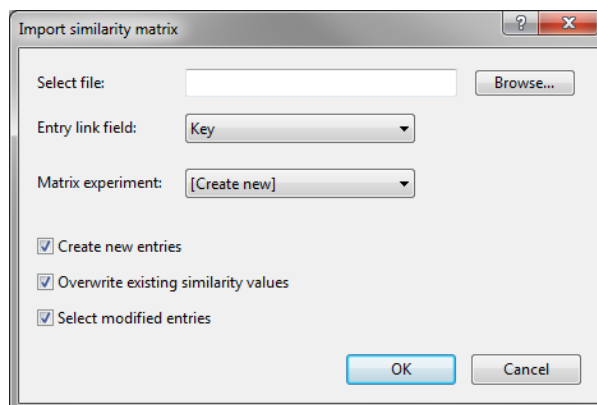
File Edit Format View Help

```
G@Ge107@008      100.00
G@Ge108@013      92.57 100.00
G@Ge109@010      90.91 89.13 100.00
G@Ge109@011      90.65 88.83 99.53 100.00
G@Ge109@004      88.25 88.18 92.62 93.18 100.00
G@Ge107@013      89.08 78.33 87.41 88.32 88.17 100.00
G@Ge111@002      88.77 87.03 83.77 84.40 86.22 82.23 100.00
G@Ge108@010      81.81 77.22 88.29 89.10 89.43 89.20 80.94 100.00
G@Ge111@003      79.09 78.68 84.79 85.68 85.01 81.85 81.57 95.51 100.00
G@Ge107@012      89.81 83.57 85.22 84.69 84.16 83.49 85.41 72.63 69.28 100.00
G@Ge108@014      89.63 86.25 89.16 89.48 88.88 89.47 83.37 80.93 74.71 95.21 100.00
G@Ge107@007      88.49 80.37 80.97 80.13 79.62 75.60 83.42 66.50 65.30 92.63 86.35 100.00
G@Ge107@011      83.75 76.47 80.54 79.85 79.16 72.49 81.96 65.03 63.91 93.13 86.42 96.99 100.00
G@Ge107@016      75.17 78.35 70.39 71.15 75.33 72.83 80.99 70.55 68.11 71.55 66.86 71.21 68.10 100.00
```

Figure 10.1.3: Matrix format.

10.1.3.2 The Import wizard

Selecting **Import similarity matrix** under **Matrix type data** in the **Import** dialog box and pressing **<Import>** brings up the **Select similarity matrix file** dialog box (see Figure 10.1.4).

Figure 10.1.4: The *Select similarity matrix file* dialog box.

Pressing the **<Browse>** button allows you to select the text file, located on your computer, external drive or on a network location. Note that only one text file can be imported at once.

The **Key** field or an existing non-default information field in the database can be selected from the **Entry link field** list. If existing entries are present in the database with this linked information, the import tool will link the imported data to these entries. If the entries are not yet present in the database, the import tool will create new entries (if the option **Create new entries** is checked) and link the data to these entries. New keys are automatically generated during import.

The similarity values can be linked to an existing matrix type experiment or to a new matrix type experiment (**Create New**).

If a new matrix type needs to be created in the database, a dialog box pops up when pressing the **<OK>** button (see Figure 10.1.5).

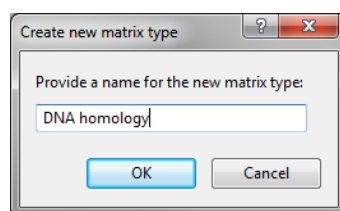


Figure 10.1.5: Provide a new matrix type name.

The dialog that is displayed prompts for the matrix type name.

When **Create new entries** is checked, the import tool is allowed to create new entries in the database.

Check the option **Overwrite existing similarity values** if you want the software to be able to overwrite matrix type information for existing entries.

If the option **Select modified entries** is checked, entries in the database that were modified during the import routine will be selected after import.

Pressing **<OK>** will start the import.



When mapped **Key** information exceeds the maximum number of allowed characters (i.e. 60 characters), an error message pops up. The data will not be imported into the database.



When mapped entry field information exceeds the maximum number of allowed characters (i.e. 79 characters), an error message pops up. The data will not be imported into the database.

In order to view the imported information in the database, the database needs to be restarted.

Chapter 10.2

Cluster analysis of matrix data

10.2.1 Comparison settings

Please note that cluster analysis of matrix values requires the Tree and network inference module (TN) to be present in your BioNumerics configuration.

To calculate a cluster analysis based on imported similarity/distance values, select the matrix type to analyze in the *Experiments* panel of the *Comparison* window, and choose **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**.... The *Comparison settings* wizard appears. The *Similarity coefficient* wizard page deals with the similarity coefficient (Figure 10.2.1).

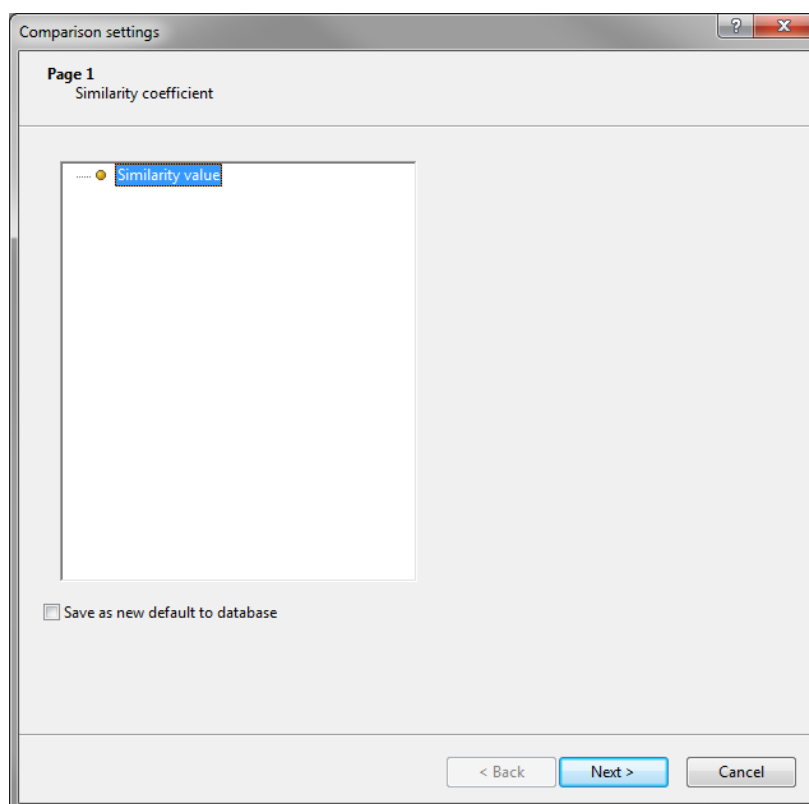


Figure 10.2.1: The *Similarity coefficient* wizard page.

Since the similarity/distance values are already calculated and saved with the matrix type experiment, there is only one option present: **Similarity value**.

Pressing <**Next**> opens the *Cluster analysis* wizard page, which deals with the calculation of a dendrogram from the similarity matrix and is discussed in [13.2.6](#).

Pressing <**Next**> again in the *Cluster analysis* wizard page starts the cluster analysis. When finished, a dendrogram and similarity matrix are shown for the matrix type.

10.2.2 Matrix display functions

Display options for matrix types in the *Comparison* window are discussed in [13.3.2](#).

Part 11


Composite data sets

Chapter 11.1


Setting up composite data sets

11.1.1 Introduction

A composite data set is a character table that contains all the characters of one or more experiment types. It is a "container" of experiment types in BioNumerics, i.e. it holds the data coming from one or several other experiments, but it does not necessarily correspond to an actual physical experiment.

In addition to the obvious reason of creating a clustering based on multiple data sets, a composite data set also offers some additional interesting features compared to single character types. These include a function to discriminate groups based upon *differential characters* in the *Comparison* window (**Composite** > **Discriminative characters**) and a function to perform *transversal clustering*, i.e. based on the characters (**Composite** > **Calculate clustering of characters...** ). These functions will be discussed in 11.2. Lastly, composite data sets are used for creating *band matching tables*, a feature that is discussed in 4.3.

11.1.2 Defining a new composite data set

To create a new composite data type, highlight the *Experiment types* panel in the *Main* window and select **Edit** > **Create new object...** . In the *Create a new experiment type* dialog box, click on **Composite data set** and press <OK>. This will display the *Add new composite data set* dialog box (see Figure 11.1.1).

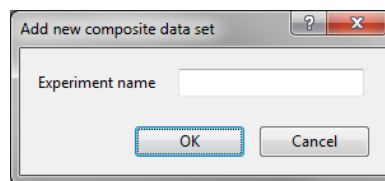



Figure 11.1.1: The *Add new composite data set* dialog box to add a new composite data set to the database.

The dialog prompts for the **Composite data set name**. Enter a name for the experiment type and press <OK> to confirm the creation of the experiment. The new experiment will appear in the *Experiment types* panel.

To determine which experiment types should be included in the composite data set, click on the composite data set in the *Experiment types* panel of the *Main* window and select **Edit** > **Open highlighted object...** , **Enter**). Alternatively, simply double-click on the composite data set. The *Composite data type* window will open (see Figure 11.1.2).

The *Composite data type* window lists all experiment types defined in the database. Initially, no experiment

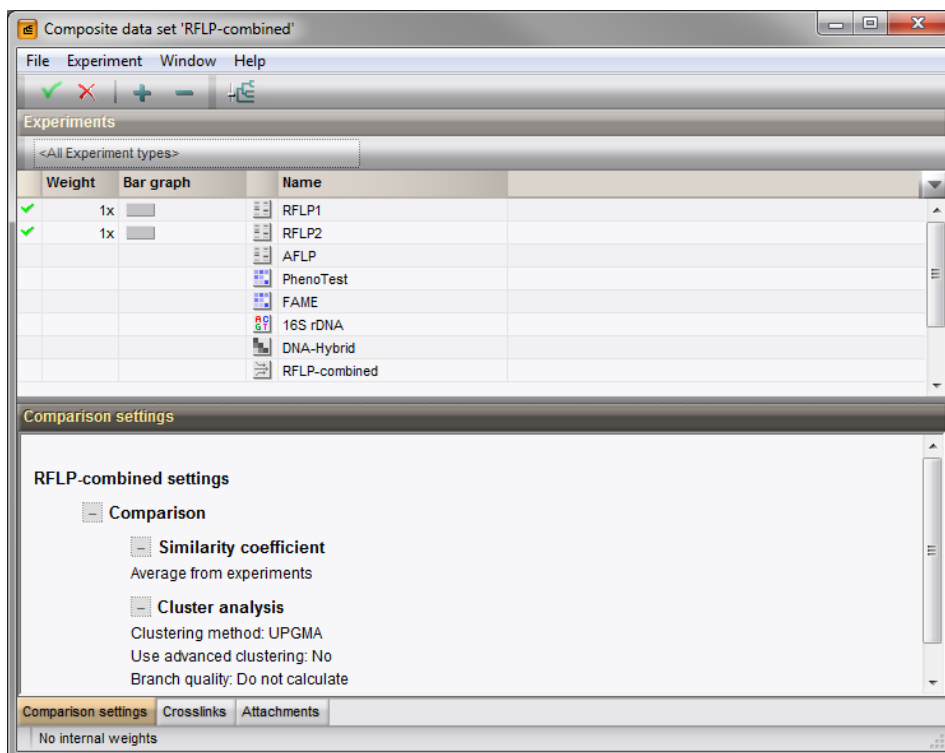


Figure 11.1.2: The *Composite data type* window.

types are included in the composite data set.

Selected experiment types can be included in the composite data set with *Experiment* > *Include experiments* (✓). Multiple experiment types can be selected using the **Ctrl**- and **Shift**-key.

Experiments can be excluded again from the data set with *Experiment* > *Exclude experiments* (✗).

An experiment type that is included in the composite data set is marked with a green V-sign and has a default **weight** of 1×, as shown in the **Weight** column and graphically indicated in the **Bar graph** column.

If the individual matrices of the experiments are averaged to obtain a combined matrix, the similarity values will be multiplied by the weights the user has specified for each experiment (see step 3 described in 11.2.1).

The weight can be increased with *Experiment* > *Increase weight* (+) or decreased with *Experiment* > *Decrease weight* (-).

If a very high weight is to be entered, a more convenient option is *Experiment* > *Set weights*. This will display the *Set weight* dialog box (see Figure 11.1.3).

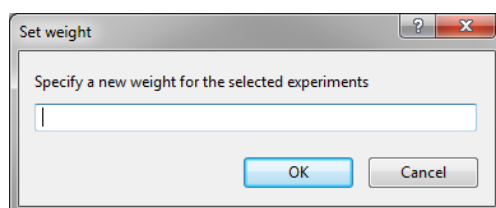


Figure 11.1.3: The *Set weight* dialog box.

Enter the weight that should be applied for the highlighted experiment types.

In order to treat individual characters on an equal basis while averaging matrices, the program can automatically use weights proportional to the number of tests each experiment contains. This correction is

achieved with **Experiment** > **Correct for internal weights**. When enabling this correction the status bar shows "Correct for internal weights".



The correction for internal weights also applies to banding patterns: if technique RFLP1 reveals 10 bands between entries A and B, whereas RFLP2 only reveals 5 bands, the similarity value resulting from RFLP1 will be twofold more important in averaging similarity between entries A and B.




The correction for internal weights and the manual weight assignment can be combined. The program will then multiply the weights obtained after correction by the weights assigned by the user.



In case step 4 described in [11.2.1](#) is chosen further in the analysis, i.e. the character sets are merged to a combined character set to which a similarity coefficient is applied, the user defined weights also have their function: in this case, the program multiplies each character of a given experiment with the weight assigned to that experiment. This feature is useful in case the ranges of combined experiments are different; for example when one experiment has a character value range between 0 and 1 and another experiment has a range between 0 and 100, a numerical coefficient (for more information on the coefficients, see [11.2](#)) would in practice only rely on the second experiment. Assigning a weight of 100× to the first experiment makes them equally important for quantitative coefficients.

11.1.3 Composite data set comparison settings

In the *Composite data type* window, the comparison settings defined for the composite data set are shown in *Comparison settings* panel (see Figure [11.1.2](#)). These settings can be accessed with **Experiment** > **Comparison settings...** () in the *Composite data type* window, but also in the *Comparison* window. See [11.2](#) for a detailed explanation.

Chapter 11.2

Cluster analysis of composite data sets

11.2.1 Principles

A clustering based upon a similarity matrix can be performed on an individual experiment type or on a combination of experiment types. The methods that BioNumerics uses to arrive at dendrograms representing combined techniques are represented schematically in Figure 11.2.1.

Please note that cluster analysis of composite data sets requires the Tree and network inference module (TN) to be present in your BioNumerics configuration.

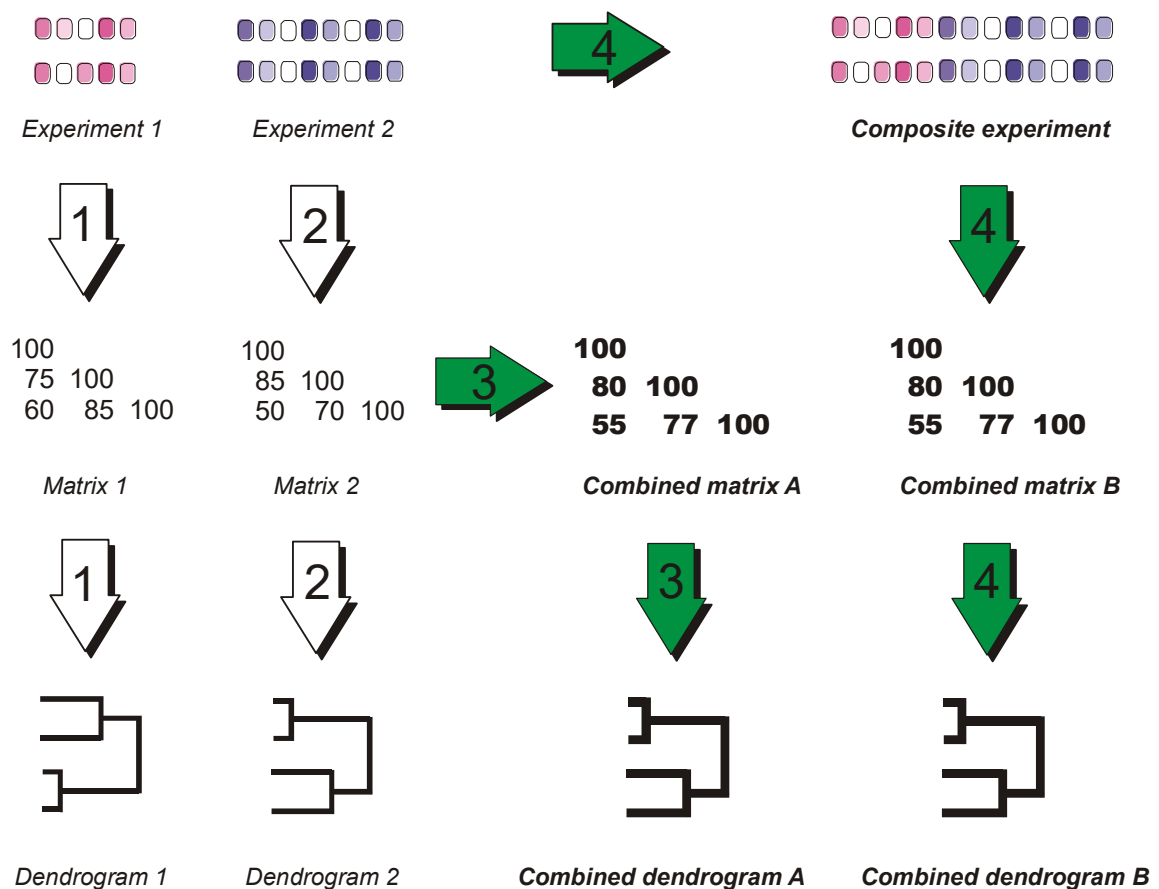


Figure 11.2.1: Scheme of possibilities in BioNumerics to obtain combined dendrograms from multiple experiments.

- Flows 1 and 2 represent the steps to obtain dendrograms for two single experiments, experiment 1 and experiment 2, respectively. The steps involve the creation of a similarity matrix and the calculation of a dendrogram based on this matrix.
- Flow 3 is the first method to calculate a combined dendrogram from multiple experiments: the individual similarity matrices are first calculated and from these matrices, a combined matrix (A) is calculated by averaging the values. The averaging can happen in two ways: each value can be considered equally important, or the program can assign a weight proportional to the number of tests in an experiment. In addition, the user can define an extra weight for each experiment manually.
- Flow 4 starts directly from the character tables, and merges all characters from different experiment types to obtain a *composite data set*. From this composite data set, a similarity matrix is calculated (combined matrix B), resulting in combined dendrogram B.

Both steps 3 and 4 require a *composite data set* to be generated.

11.2.2 Calculating a dendrogram from a composite data set

Calculating a dendrogram from a composite data set is almost the same as for a single experiment. To calculate a cluster analysis on a composite data set experiment, select the composite data set to analyze in the *Experiments* panel of the *Comparison* window, and choose **Clustering > Calculate > Cluster analysis (similarity matrix)...** The *Comparison settings* wizard for composite data sets pops up (Figure 11.2.2).

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the way the similarity matrix is calculated.

If the composite data set comprises one or more fingerprint types or character types, different *character aspects* of those experiment types can be loaded in the composite data set by specifying the aspect from the drop-down list in the *Experiments* panel (see 13.2.5).

The hierarchical representation on the left allows you to select **Average from experiments** (corresponding to step 3 in Figure 11.2.1, averaging the matrices of the experiments according to the defined weights) or to treat the data as **Character data** and select a coefficient from the **Numerical**, **Binary** or **Multi-state** category (corresponding to step 4 in Figure 11.2.1, merging the experiments to a composite character table). Depending on the selected coefficient, the relevant settings are displayed on the right. Each of the categories can be collapsed by clicking on the small ”-” (minus) sign that precedes the category name.



Fingerprint types that are included in the composite data set are only available as character data after a band matching is performed (see 4.3.2).

Numerical coefficients treat the character values from the composite data set as numbers.

The **Pearson correlation** (or Pearson product-moment correlation) is calculated as:

$$r_{j,k} = \frac{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)(x_{i,k} - \bar{x}_k)}{\sqrt{\sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2 \sum_{i=1}^n (x_{i,k} - \bar{x}_k)^2}}$$

with

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j}$$

and

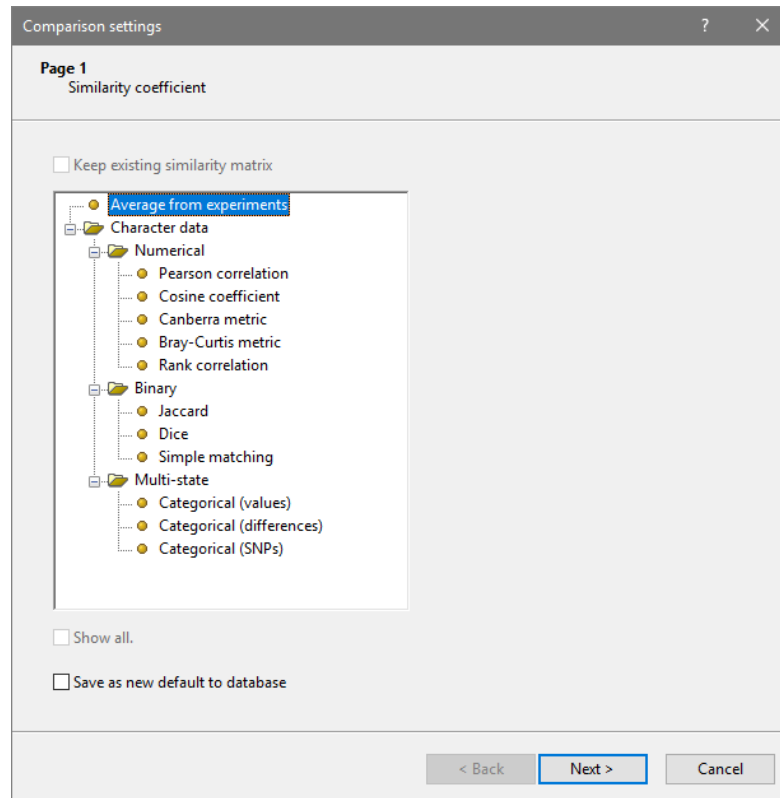


Figure 11.2.2: The *Similarity coefficient* wizard page for composite data sets, grouping the settings that determine how the similarity matrix is calculated.

$$\bar{x}_k = \frac{1}{n} \sum_{i=1}^n x_{i,k}$$

Hereby, n denotes the number of characters in the composite data set and $x_{i,j}$ and $x_{i,k}$ the i^{th} character value of entries j and k , respectively.

The related ***Cosine coefficient*** is calculated as:

$$C_{j,k} = \frac{\sum_{i=1}^n x_{i,j} x_{i,k}}{\sqrt{\sum_{i=1}^n x_{i,j}^2 \sum_{i=1}^n x_{i,k}^2}}$$

The ***Canberra metric*** is calculated as:

$$D_{CANB(i,j)} = \frac{1}{n} \sum_{i=1}^n \frac{|x_{i,j} - x_{i,k}|}{|x_{i,j} + x_{i,k}|}$$

The ***Bray-Curtis*** metric is calculated as:

$$D_{BC(j,k)} = \frac{\sum_{i=1}^n |x_{i,j} - x_{i,k}|}{\sum_{i=1}^n |x_{i,j} + x_{i,k}|}$$

The ***Rank correlation*** (or Spearman rank-order correlation) coefficient first transforms an array of characters into an array of ranks according to the magnitude (intensity) of the character values. The rank arrays are then compared using the Pearson product-moment correlation coefficient. The ***Rank correlation*** is known to be a very robust coefficient, but with low sensitivity.

Following parameters are available for numerical coefficients:

- Checking *Use square root conversion* can be particularly useful when comparing highly related organisms. This has the effect that narrow branches on a dendrogram are stretched out relatively more than distant links.
- The feature *Standardize characters* standardizes each character by subtracting its mean value and dividing by its standard deviation. The result is that all characters have equal influences on the similarity. It may be meaningful to enable *Standardize characters* in the following cases. (1) For some techniques, e.g. fatty acid methyl ester analysis, it is common that some fatty acids occur in high amounts, whereas other fatty acids occur only in very small amounts. It is likely that the major fatty acids will account for most of the discrimination between the organisms studied, whereas the minor fatty acids, which may be as valuable from a taxonomic point of view, are masked. (2) When creating composite character sets from different experiments, the ranges of the experiment may be different. When using a coefficient such as the Pearson correlation coefficient, characters with a higher range will have more influence on the similarity and the dendrogram. With fingerprints, the above situation occurs when the optical densities are different.

For a *Binary* coefficient, a character can only have two states: positive or negative (0 or 1).

The *Jaccard* coefficient is calculated as:

$$S_J = \frac{N_{AB}}{N_A + N_B - N_{AB}}$$

with N the total number of characters. N_A , N_B and N_{AB} are the number of characters that are positive for entry A , entry B and both A and B , respectively. N_{ab} is the number of characters that is negative for both A and B .

The *Dice* is calculated as:

$$S_D = \frac{2N_{AB}}{N_A + N_B}$$

The *Simple matching* is calculated as:

$$S_{SM} = \frac{N_{AB} + N_{ab}}{N}$$

Jaccard and *Dice* correlation are very related to each other whereas *Simple matching* is more fundamentally different. The Jaccard and Dice coefficients only consider "scoring characters" being two positive characters in both data sets, whereas the simple matching coefficient also considers two negative characters as scoring.

If the combined experiments are comparable in terms of biological meaning, reaction type and numerical range, it is possible to use one of the *Binary* or *Numerical* coefficients. For example, if both experiments involve substrate utilization tests and are recorded either positive or negative, the best option is to select under *Binary* coefficients (*Jaccard*, *Dice* or *Simple matching*). If both character tests are registered quantitatively as numerical values between 0 and 100, a suitable option is to select under *Numerical* coefficient (*Pearson correlation*, *Cosine correlation*, *Canberra metric*, *Bray-Curtis metric*, or *Rank correlation*).



It can be proven that in case of binary data sets the option *Average from experiments* offers exactly the same results when *Correct for internal weights* is enabled in the composite data set settings (see 11.1.2).

In the case however, that the *ranges* of the combined experiments are different, e.g. a range between 0 and 10 for one character experiment and between 0 and 100 for another character experiment or a range between

0 and 255 (8-bit) for one fingerprint experiment and between 0 and 65,535 (16-bit) for another fingerprint experiment, **Numerical** coefficients are not suitable, as they would assign much more weight to the second experiment than to the first. In such cases, you should either take the similarity values from the individual experiments (option **Average from experiments**) and average them into a new matrix, or specify user-defined weights for the experiments, so that their final weights are comparable (see Notes in 11.1.2).

The **Categorical** coefficient can be chosen in case all the characters of the individual experiment types are *multi state* characters. As opposed to *binary*, where only two states are known, multi state characters are defined as characters that can take more than two states. However, as opposed to *numerical* characters, the different states represent discrete categories, which cannot be ranked somehow. Examples are phage types, Multi Locus Sequence Types (MLST), colors, etc..

The **Categorical (values)** coefficient works directly on the character values. It has following parameters:

- With **Calculate as distance** checked, the distance between two entries is reported. With this option unchecked, the software calculates similarity values. Note that there is a maximum distance of 200; distance values that exceed this maximum will be clipped to 200. The distance D is calculated with this coefficient as:

$$D = \frac{N\Delta_{AB}}{N_{AB}}$$

with N the total number of characters in the comparison, Δ_{AB} the number of characters different between entry A and B , and N_{AB} the number of characters present in both entries.

- **Ignore zero values** will leave any character out of the pairwise comparison for which at least one entry in the pair has a zero (0) value. With other words, zeros in a character set will be treated the same way as missing values if this option is checked.
- With **Fuzzy logic** checked, the coefficient will score each character match decreasingly with increasing distance between the values, between full match (zero distance) and no match (distance = tolerance).
- A certain **Tolerance** can be specified for values to be considered as belonging to the same category. This makes it possible to treat non-discrete (non-integer) values as categorical. By default, the **Tolerance** value is set to zero, which means that no tolerance is allowed, i.e. the values must be identical to be considered the same category.

The **Categorical (differences)** coefficient also works on the character values and calculates by default a distance matrix. Hence, the corresponding option is grayed out. The **Categorical (differences)** coefficient differs from **Categorical (values)** in that it uses the absolute number of differences instead of a normalized distance, i.e. $D = \Delta_{AB}$.

The coefficient has an additional parameter called **Scaling factor**, to deal with the hard-coded maximum of 200 that can be calculated for a distance value. Sensible values for the **Scaling factor** are “1”, “10” and “100”, allowing the correct visualization of maximally 200, 2000 and 20000 different character values, respectively, in a cluster analysis. To trace back the number of different character values from the dendrogram branches or distance matrix, the displayed values needs to be multiplied with the **Scaling factor** used.

The **Categorical (SNPs)** coefficient is designed for comparing Single Nucleotide Polymorphism (SNP) matrices, as obtained from a wgSNP analysis (see 8.10). Being a multi-state or categorical coefficient, it matches A against A, C against C, G against G, and T against T with a 100% score. It specifically takes IUPAC degenerated bases into account by matching two-fold ambiguous bases against the bases they represent with 50% of a perfect match score (e.g. Y matches for 50% against either C or T), three-fold ambiguous bases with a 33.3% score against the bases they represent (e.g. H matches for 33.3% against A, C or T) and N with a 25% score against any base A, C, T or G.

Similar to the **Categorical (differences)** coefficient, the **Categorical (SNPs)** coefficient calculates by default an absolute distance matrix, which can be scaled using the **Scaling factor** parameter.

If a similarity matrix already exists for the selected experiment, an option **Keep existing similarity matrix** appears. When checked, the previously calculated similarity matrix will be used and all coefficient options will appear gray (disabled).

Check **Save as new default to database** if you want the specified comparison settings to be saved in the database as default settings.

It is obvious that the possibility of approach 4 described in 11.2.1, i.e. merging two character sets into a combined character set, is only applicable to comparable character sets. It makes no sense, and is even impossible to combine a phenotypic test panel with a sequencing experiment in this way. When experiments of different nature are to be used for consensus groupings, the only remaining approach is to combine the obtained individual similarity matrices (approach 3 in 11.2.1). However, the option to create an average matrix from individual experiment matrices only works well in case two conditions are fulfilled: (i) the expected similarity range for both experiments is comparable, and (ii) the matrices are complete, i.e. for each experiment there is a similarity value present for each pair of entries. Suppose that two experiment types are to be combined which generate strongly different similarity levels, e.g. DNA homology values on the one hand and 16S rDNA sequence similarity on the other hand. In many cases, DNA homology values will range from 100% to 40% or less, whereas 16S rDNA sequence similarity will range between 100% and 90% or even higher. It is clear that the small but very significant differences in 16S rDNA sequence similarity will be masked by the much larger differences (including experimental error) of DNA hybridization, and will have no contribution to the clustering based upon averaging of matrices. In such cases, other methods are needed to compose a consensus matrix, that "takes the best of it all".

The principle of averaging matrices is even worse when one or more matrices are incomplete. Suppose three entries in BioNumerics A, B, and C. Consider the following matrices for these three entries, generated from 16S rDNA aligned sequence similarity and DNA hybridization (Figure 11.2.3). The DNA hybridization matrix is incomplete, a situation which may happen frequently.

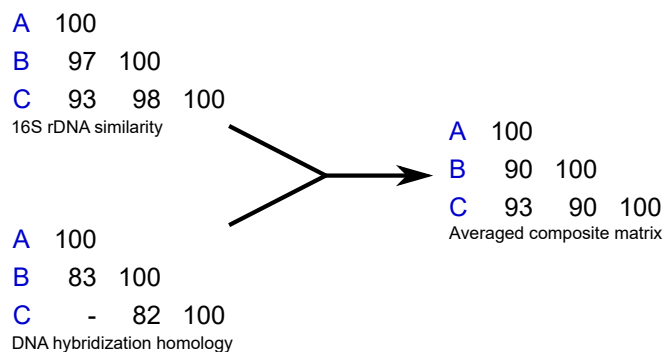


Figure 11.2.3: Illustration of problems occurring when incomplete similarity matrices are averaged.

The averaged matrix created in the composite data set from these two experiments shows averaged values for (AB) and (BC) but for (AC) it has taken the only available value, 93%. The resulting matrix provides a completely distorted view of the relationships between these three organisms, as it suggests A and C to be closest related. In reality however, one can predict, based upon the lower 16S rDNA similarity, that (AC) will be much less related than (AB) and (BC).

The above is an obvious example where averaging similarity matrices is not a good approach, and therefore, another algorithm has been incorporated in BioNumerics, based upon linearization of the consensus matrix with respect to the individual experiment matrices. The consensus matrix is composed in such a way that it constitutes a third degree function of each individual experiment matrix, and the result is that it reflects each of the constituent matrices as closely as possible.


Pressing **<Next>** opens the *Cluster analysis* wizard page, which deals with the calculation of a dendrogram


from the similarity matrix and is discussed in 13.2.6.


Pressing <Next> again in the *Cluster analysis* wizard page starts the cluster analysis. When finished, a dendrogram and similarity matrix are shown for the composite data set.

When similarity matrices (see 13.3.2) are calculated for each experiment type that is included in the composite data set, a consensus matrix can be calculated with **Composite > Calculate consensus matrix...**



11.2.3 Composite data set display functions


Composite data sets are visualized in the *Comparison* window as *character tables*, for which several display functions are available. For these options to apply, make sure that the composite data set is shown in the *Experiment data* panel by pressing the eye button () next to the experiment name in the *Experiments* panel.

Initially, the composite data set is displayed as a binary character set. In case the display is changed using any of the commands discussed below, it is possible to revert to the binary representation with **Composite > Show presence/absence** ()

To show the quantitation values as a color scale select **Composite > Show quantification (colors)** () . This color scale can be set in the preferences (see 2.3.3).

Select **Composite > Show quantification (values)** () to show the corresponding character values for all entries in the comparison.

In case the composite data set includes a character type experiment for which a mapping is defined (see 6.1.2.7), the character mappings can be shown with **Composite > Show categorical data (if available)** () . Similarly, **Composite > Show categorical data and colors (if available)** () shows the character mappings against a quantitative color scale.



A convenient option to quickly check the behavior of an individual character in the composite data set is to list the entries according to the values of this character. Thereto, click on a character in the header of the *Experiment data* panel and select **Composite > Sort by character** () . The entries are now sorted by increasing value of the selected character. A dendrogram is not displayed any more, since it would impose a different order on the entries. If a dendrogram was calculated previously, it can be called again from the *Analyses* panel.



The order in which experiment data are displayed in a composite data set is the same as the experiment order in the *Experiments* panel of the *Main* window (see 2.3.2): re-sorting the *Experiments* panel will result in an updated display order of the experiment data in all composite data sets when the comparison is opened again.

11.2.4 Finding discriminative characters between entries

BioNumerics offers the possibility to rearrange the characters in a composite data set according to their discriminatory power, i.e. their ability to discriminate between the selected entries on the one hand side and the remaining (unselected) entries in the comparison on the other hand side. This can be achieved with the option **Composite > Discriminative characters**.

Before choosing this command, make a selection of entries in the *Comparison* window that you wish to discriminate from the other entries. This can be done manually or using the available search functions (see 3.3.8 and 3.3.9). With **Edit > Arrange entries > Bring selected entries to top** (, **Ctrl+T**) the selected entries are grouped on top of the list. Pressing the eye button () next to the experiment name in the *Experiments* panel shows the image of the composite data set in the *Experiment data* panel.

When selecting **Composite > Discriminative characters**, the characters are reorganized in such a way that those characters positive for the selected entries and negative for the other entries occur left, and those characters negative for the selected entries and positive for the other entries occur right.

11.2.5 Transversal clustering


The input for a cluster analysis in a composite data set is a *data matrix*. A data matrix of n entries having p characters looks like in Figure 11.2.4: the entries are presented as *rows* and the characters as *columns*. In BioNumerics, the data matrix should not necessarily be complete: some missing character values are allowed, for example if test results are ambiguous or not available.


	Char 1	Char 2	...	Char p
Entry 1	Val 11	Val 12	...	Val 1 p
Entry 2	Val 21	Val 22	...	Val 2 p
...
Entry n	Val $n1$	Val $n2$...	Val np

Figure 11.2.4: Data matrix of n entries and p characters.

A simple and efficient way to visualize associated groups of characters (columns) with groups of entries (rows) in a data matrix is to construct a two-way clustering of the data matrix, i.e. in which the entries are clustered by means of their character values (the conventional clustering as described in 13.1; also called *Q*-clustering), and the characters are clustered by means of their values per entry (*R*-clustering).

The result is a data matrix in which both the entries and the characters are ordered according to their relatedness (Figure 11.2.5), which we will call *transversal clustering*. This representation makes it easy to visually associate clusters of characters with clusters of entries. For example, the first group of entries (E1, E9, and E5) is separated from the others by a cluster of characters (C5, C16, C3, and C11) which are all more positive in the first cluster than in the other clusters. Another group of three characters (C14, C17, and C20) separates the second group of entries (E3, E6, and E2) from the other clusters because they are less positive.

In BioNumerics, a transversal clustering can be calculated from a composite data set. This feature is illustrated best when the composite data set is displayed in the *Experiment data* panel (click on the composite data set and display its character table by pressing the eye button  next to the experiment name in the *Experiments* panel.)

A cluster analysis of the entries is calculated as described in 13.2.6. Selecting **Composite > Calculate clustering of characters...**  calls the *Cluster analysis of characters* dialog box (Figure 11.2.6).

The dialog box offers a choice between the **Pearson correlation** for numerical characters, the **Jaccard**, **Dice**, and **Simple matching** coefficients for binary data, and the **Categorical** coefficient for multi-state or categorical characters. For a description of the coefficients, see 11.2.2.

The character dendrogram appears horizontally in the caption of the data matrix display of the composite data set.

It may be useful to drag the separator bar between the image panel and its caption down to obtain more

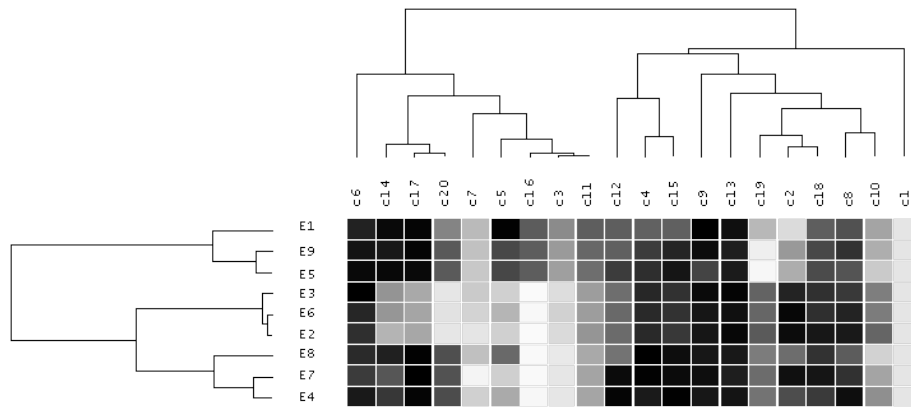


Figure 11.2.5: Transversal clustering of entries (horizontal) and characters (vertical).

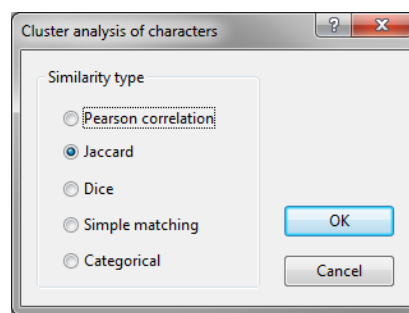


Figure 11.2.6: The *Cluster analysis of characters* dialog box.

space for the character dendrogram and the character names.

Part 12

Whole Genome Maps

Chapter 12.1

Background

The Whole Genome Map experiment type is designed to import data from the ArgusTMWhole Genome Mapping System (OpGen[®]; <http://www.OpGen.com>). The ArgusTMWhole Genome Mapping System generates high resolution, **ordered** whole genome restriction maps from single (microbial) DNA molecules. As whole genome mapping provides highly detailed strain information, it is very well suited for discriminating very closely related strains (e.g. outbreaks). The analysis of whole genome map data in BioNumerics is hence mainly focused on strain typing and characterization. In brief, map-based similarity clustering can accurately distinguish strains and describe relatedness between isolates. Aligning and directly comparing maps may aid to discover insertions, deletions and other genetic elements. The various display options allow to highlight genomic differences for an at-a-glance discovery.

The following chapters will describe the set-up and analysis of a whole genome map type experiment.



To be able to work with whole genome map type experiments, the Whole Genome Map data module (**GM**) needs to be present in your BioNumerics configuration. (To learn more, see [2.1.4](#)).

Chapter 12.2

Setting up Whole Genome Map experiment types

12.2.1 Defining a new Whole Genome Map type

To create a new whole genome map experiment type, click on the *Experiment types* panel to activate it and click *Edit > Create new object...* (+).

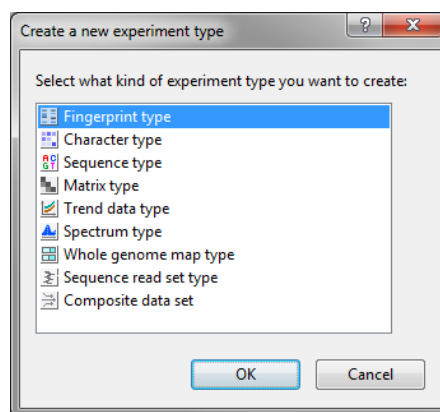


Figure 12.2.1: The first step in the creation of a new experiment type is making a selection in the *Create a new experiment type* dialog box.

In the *Create a new experiment type* dialog box that popped up, select **Whole genome map type** and press <OK> to continue (see Figure 12.2.1). This will display the *New whole genome map type* dialog box (see Figure 12.2.2).

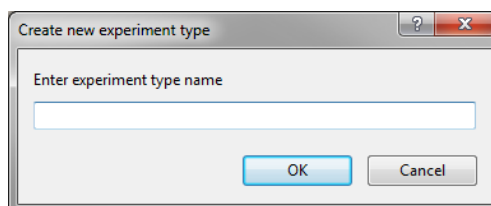


Figure 12.2.2: The second step in the creation of a new whole genome map experiment type.

The *New whole genome map type* dialog box prompts you to enter a name for the new experiment type. Enter a descriptive name and press <OK> to finalize the creation of a new whole genome map experiment

type.

The *Experiment types* panel now lists the newly created experiment type. The general and comparison settings of the experiment type can be adjusted in the *Whole genome map type* window (see 12.2.3). An experiment type can be deleted from the database by selecting **Edit > Delete selected objects...** (✖). After confirming twice, the experiment type will be marked for deletion. Upon reopening the database for a new session, the experiment type will no longer be listed in the *Experiment types* panel.

12.2.2 Importing Whole Genome Map data

12.2.2.1 Introduction

The import of whole genome map data in BioNumerics is based on XML files obtained from the OpGen[®] Argus[™] Optical Mapping System. Each set of fragments in the XML file has a header containing additional information, such as the strain/isolate ID, the restriction enzyme used, etc.. An example of an XML file with a partial fragment list is given in Figure 12.2.3.

```
<?xml version="1.0" encoding="UTF-8"?>
- <RESTRICTION_MAPS_DOCUMENT version="0.2">
  <DISPLAY_INFO GRIDSnap="false" GRID="false" RULERORIGIN="0" RULER="false" Y="0" X="0"
    SCALE="0.000147970"/>
  - <RESTRICTION_MAP INSILICO="false" ENZYME="NcoI" ID="Salmonella enterica (subsp.enterica
    serovar SaintPaul CDC B1605 SARA22)">
    <MAP_DISPLAY Y="100" X="10000" GROUPID="-1" CIRCULAR="true" ORIENTATION="1"
      ORDER="10" TRANS="255" STICK="false" EDITABLE="false" DBID="108358"/>
    - <FRAGMENTS OFFSET="1" SHIFT="0">
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="10901" I="0"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="4416" I="1"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="4504" I="2"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="9439" I="3"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="2081" I="4"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="7497" I="5"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="6151" I="6"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="3803" I="7"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="10459" I="8"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="5289" I="9"/>
      <F GAP="false" HIDE="false" HIGHLIGHT="false" STDDEV="0.000" S="4353" I="10"/>
    </FRAGMENTS>
  </RESTRICTION_MAP>
</RESTRICTION_MAPS_DOCUMENT>
```

Figure 12.2.3: Example of an XML file for import of whole genome map data.

Imported whole genome maps can be linked to new or existing database entries. Optionally, additional information, e.g. the file ID or applied enzyme, can be stored in entry information fields.

12.2.2.2 The Import wizard

12.2.2.2.1 Getting started



Before starting the import of whole genome maps, a new whole genome map experiment type can already be created. See 12.2.1 for more information on how to define a new whole genome map experiment type. If no whole genome map type is available upon import or if the imported data need to be saved in a new experiment type, the new whole genome map experiment type can also be created during the import process.

To start with the import of whole genome maps, select **File > Import...** (📁, Ctrl+I) to open the *Import* dialog box (see Figure 12.2.4). Selecting the option **Import whole genome maps** under **Whole genome maps data**, will start the *Import whole genome maps* wizard (see Figure 12.2.4).

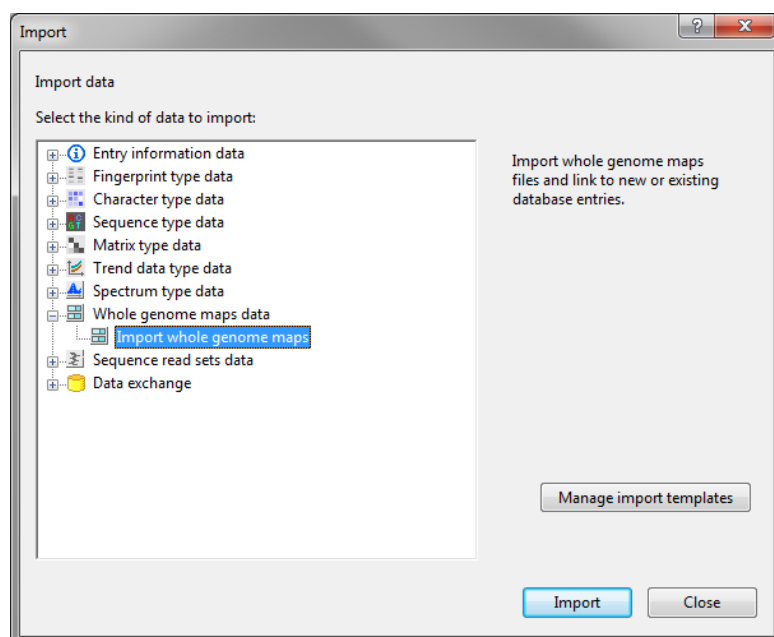


Figure 12.2.4: Import whole genome maps.

12.2.2.2.2 Input

The *Import whole genome maps* wizard consists of multiple steps: (1) selecting the files to import (*Input*, this section), (2) specifying how to import data into the database (*Import template*, 12.2.2.2.3) and (3) defining how records should be linked to database entries (*Database links*, 12.2.2.2.4).

In the *Input* wizard page (see Figure 12.2.4), selecting the **<Browse>** button will allow to select the file(s) to be imported. These files can be located on your computer, external drive or a network location. It is possible to select multiple files at once. Deleting one or multiple files from the list of files to be imported can be achieved by selecting the item(s) from the list and pressing the **<Delete>** button. By clicking the **<Delete All>** button, all files present in the import list will be removed at once.

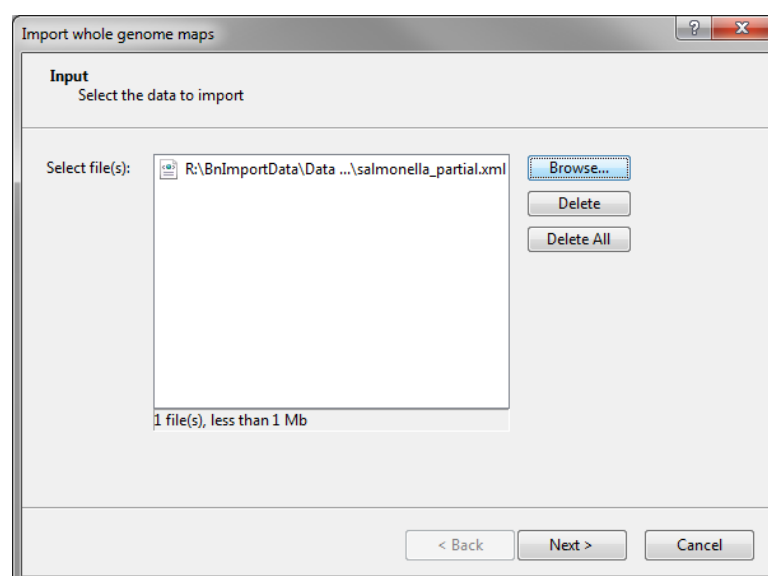


Figure 12.2.5: The first step of the whole genome map import involves selecting the XML files in the *Input* wizard page.

For the import as illustrated in Figure 12.2.5, one .xml file was selected, containing whole genome map data for three entries. Once the appropriate files are in the import list, press **<Next>** to proceed to the *Import template* wizard page.

12.2.2.2.3 Import template

To specify which data is saved to the database, an import template needs to be created. Once created, this import template remains available in the database for repeated import of similar data at later stages. Import templates can be edited, and even exported to and imported from other databases, or shared by colleagues. More details on the management of import templates can be found in 3.3.5.4.

In the *Import template* wizard page (Figure 12.2.6), BioNumerics asks to specify how the data should be imported into the database. More specifically, it needs to know to what experiment type the imported data should be linked, and which import template should be used to identify entries and automatically link XML file information to information fields in the database.

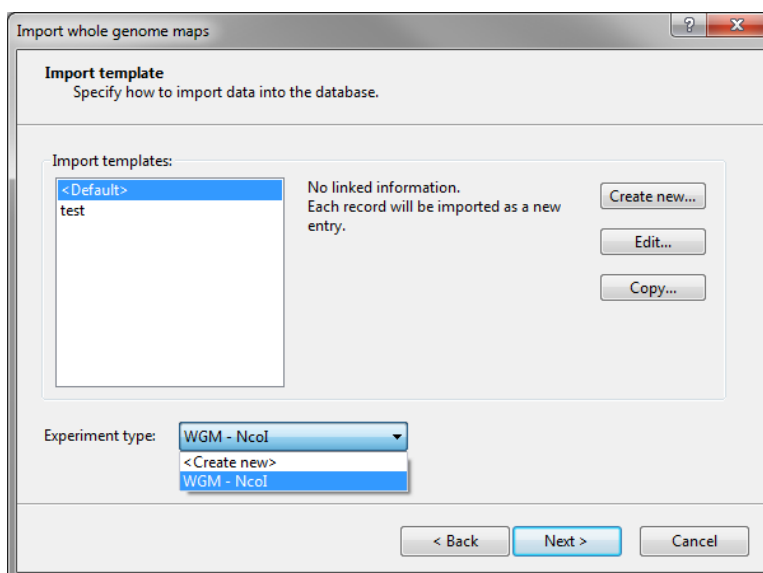


Figure 12.2.6: In the second step of the whole genome map import, the import template and experiment type need to be specified in the *Import template* wizard page.

For the *Import templates*, essentially three options are available: (1) using the default import template, (2) creating a new import template or (3) selecting an existing template from the list.

(1). When using the basic *<Default>* import template, the whole genome map data from XML file (in this example for three strains), will be directly imported, without any linked information. Each record will be imported as a new entry in the database, with an automatically generated key. Hence for the example file, three new entries would be created in the database, with a newly generated key and with no information in the information fields of the entries.

(2). An import template can be newly created to tailor the import rules to your database structure, by selecting **Create new...**. Typically all rows in the grid of the popped up window can be associated with a new or existing entry information fields. Initially, the rows are not linked to any information in the database: the *Destination type* and *Destination* for all rows is set to *<None>* (see Figure 12.2.7). Specifying a *Destination* for one or multiple selected rows can be done by pressing the **<Edit destination>** button or by double-clicking the *Source type*. This action pops up a new dialog box prompting for the new destination for the selected row(s) (see Figure 12.2.8).



To select multiple rows, hold the **Ctrl**-key on the keyboard and click the rows to be selected. To select a complete range of rows at once, select the first row, hold the **Shift**-key and click the last row.



With more than one row selected, the last-clicked row should be double-clicked to edit the destinations.

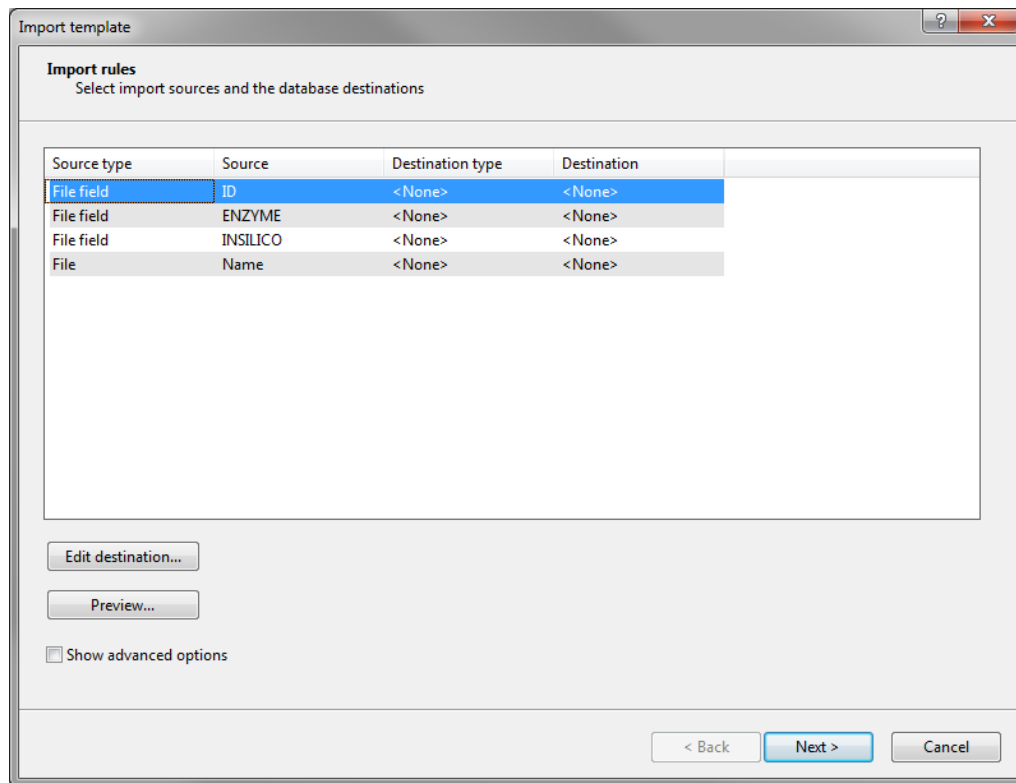


Figure 12.2.7: Editing the import template rules during whole genome map import.

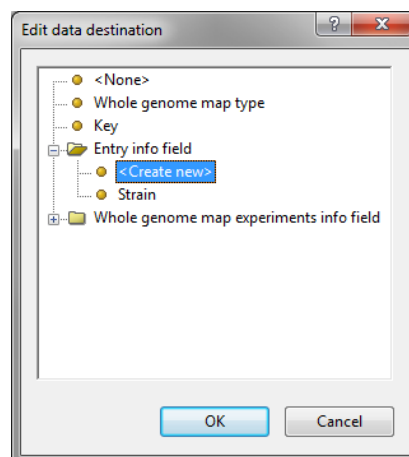


Figure 12.2.8: Specifying the data destination for the import template.

The information of the selected row(s) can be linked to (see Figure 12.2.8):

- A *Whole genome map type*.
- The default information field *Key*.

- A new or existing, non-default entry information field, via **Entry info field**: select an existing field from the list, or use **<Create new>** to add a new information field to your database. In the latter case, where a row will be linked to a new entry information field by pressing **<OK>**, a new dialog box will pop-up and prompt for a name for the newly to be created entry information field. A default name is suggested by BioNumerics, but can be edited as desired.

Pressing **<OK>** will create the new entry information field in the database, and update the information in the *Destination type* and *Destination* columns in the grid.

By using the **<Preview>** button, an example of the results of the import rules can be checked in the popped-up window. If desired, the import rules can be further tailored, e.g. parsing specific information from the information provided under a given source, by checking the **Show advanced options** box. More information on the use of these advanced options is detailed in 3.3.5.5.

Press **<Next>** to proceed. If the defined import template rules will not parse information to the default information field *Key*, the next step is to define if other linked information fields are to be used to look up the database entries. In Figure 12.2.9, for instance, the *Strain* information field is checked for this purpose.

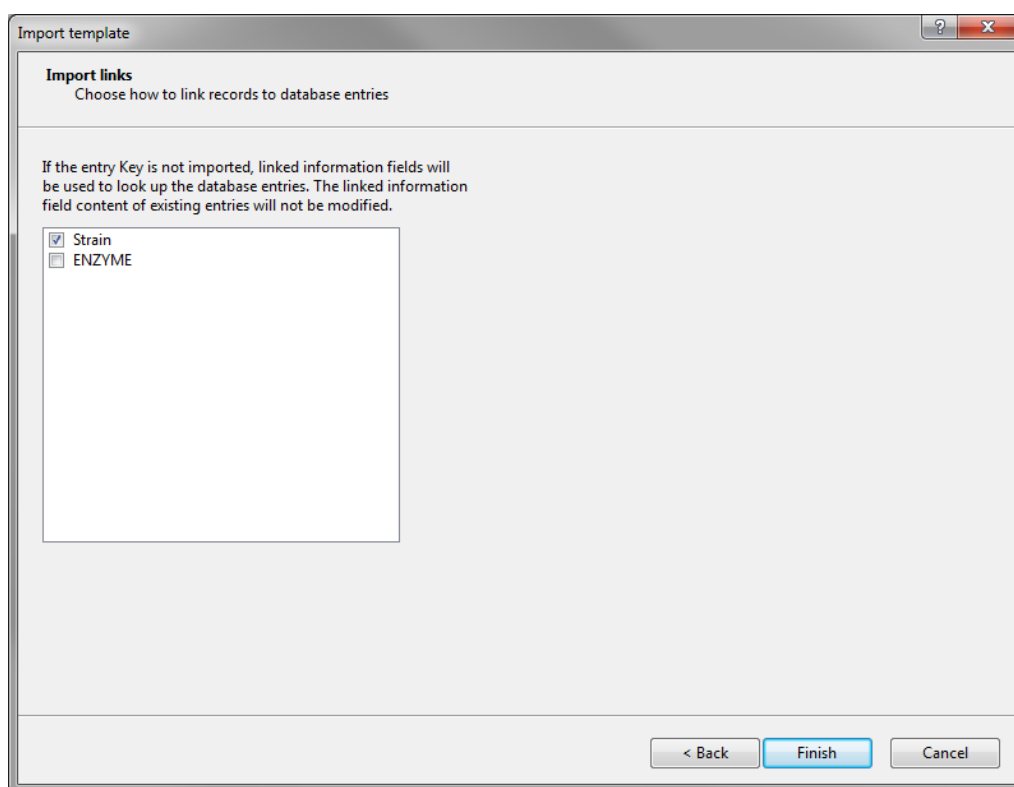


Figure 12.2.9: Defining import links for the import template.

This concludes the creation of the new import template and by pressing **<Finish>**, you will be prompted to give a name and (optionally) a description for the newly defined import template. The option **Share this import template** is available for further management of the import templates (see 3.3.5.4 for details).

(3). Finally, a third option is to use an existing template, that was previously created or shared by another user. This can be achieved by selecting the name of the import template from the list in the *Import template* wizard page. If desired, the existing template can be edited as described above by selecting **<Edit...>**.

Once the import rules are defined in the import template, the experiment type where the data are to be saved also needs to be specified. For the **Experiment type**, either an existing whole genome map experiment type can be selected, or the option **<Create new>** can be used to create a new experiment type, as shown in Figure 12.2.6. The latter option can be chosen if, for instance, another restriction enzyme is used for generating

the whole genome maps or if no whole genome map experiment type is present yet in the database. In this case, the wizard will prompt you to enter a name for the experiment type to be newly created and will ask your confirmation, prior to proceed. Using the **<Back>** button, it is always possible to return and modify the selection of the *Experiment type* or the *Import templates* as desired.

Once the *Experiment type* and the *Import template* are specified, press **<Next>** to proceed to the *Database links* wizard page of the *Import whole genome maps* wizard wizard.

12.2.2.2.4 Database links

The last step in the import wizard is to define how the records should be linked to the database entries in the *Database links* wizard page.

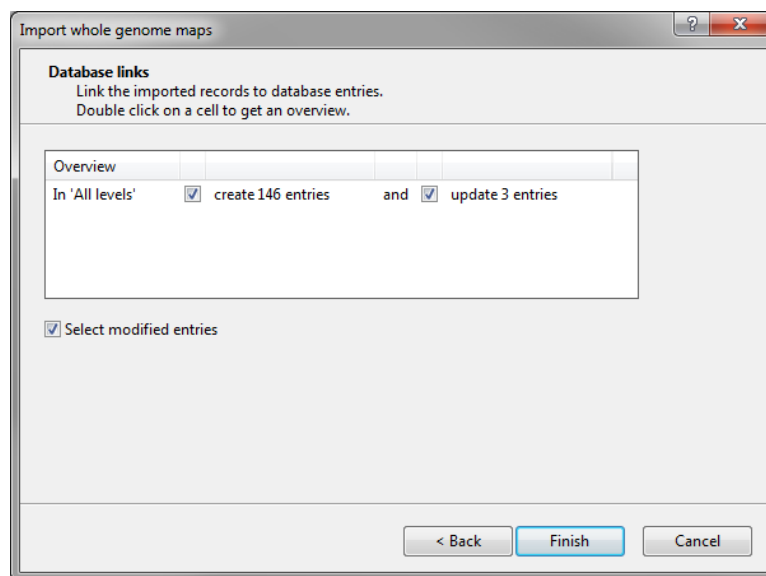


Figure 12.2.10: The last step of the whole genome map import comprises the link of the records to the database entries in the *Database links* wizard page.

At this point, the user can still define if he/she only wants to create new entries, and not altering anything on data already present in the database or vice versa, by (un)checking the respective boxes (see Figure 12.2.10). When **Create *x* entries** is checked, the import tool is allowed to create the new entries in the database. When **Update *x* entries** is checked, the import tool is allowed to update the information for existing entries. When in doubt, double-clicking on **Create *x* entries** or **Update *x* entries**, will show an entry list of the entries that will be created or updated, respectively. Double-clicking on one of the entry keys opens the corresponding *Entry* window. By default, the check box **Select modified entries** is checked, which implies that, after import, entries that were created or updated will be selected in the BioNumerics *Main* window. By pressing **<Finish>**, the actual import of the data into the whole genome map experiments will start.

12.2.3 Editing Whole Genome Map experiment type settings

The *Whole genome map type* window (see Figure 12.2.11) can be opened by clicking on the whole genome map type in the *Experiment types* panel and selecting **Edit > Open highlighted object...** (🔗, Enter). Alternatively, simply double-click on the whole genome map type.

The *Whole genome map type* window allows reviewing and adjustment of the whole genome map experiment type settings and consists of three panels: the *Comparison settings* panel, *Crosslinks* panel and *Attachments* panel. The latter two panels are organized in tabs in the lower panel in the default configuration.

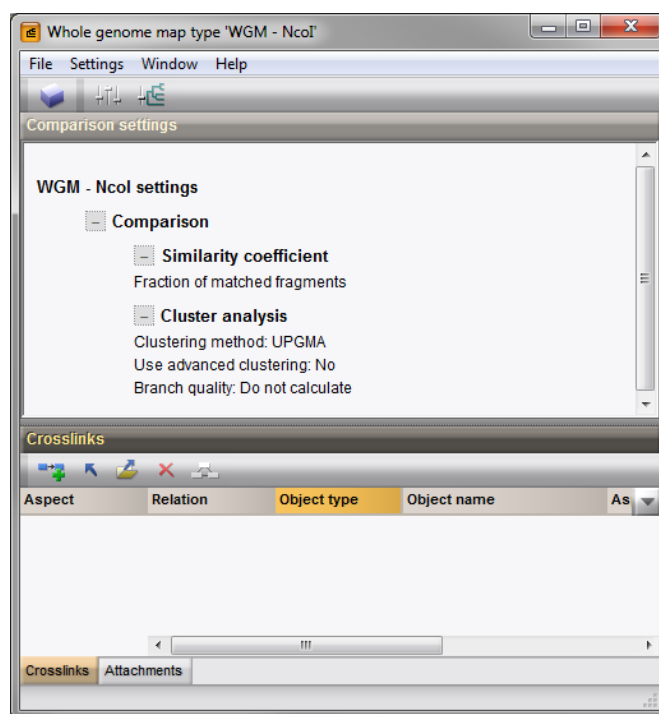


Figure 12.2.11: The *Whole genome map type* window allows reviewing and adjustment of the whole genome map experiment type settings

- The *Comparison settings* panel lists the currently active similarity coefficient and clustering parameters, adjustable by selecting **Settings > Comparison settings...** (⚙️) (see further in this section). Next to the comparison settings, the general settings can furthermore be adjusted via **Settings > General settings...** (⚙️), as detailed further in this section.
- The *Crosslinks* panel allows to establish relations between objects. To learn more about cross-links, see 3.2.15.
- The *Attachments* panel allows to add and manage attachments. For more details on working with attachments, see 3.2.13.

In the *Whole genome map type* window the comparison settings defined for the whole genome map experiment type are shown in the *Comparison settings* panel. The similarity coefficient and cluster analysis settings can be accessed with **Settings > Comparison settings...** (⚙️), but equally from the *Comparison* window. See 12.4 for detailed explanation. By ticking the option **Save as new default to database** in the *Comparison settings* wizard all adjusted settings are saved in the database for the particular whole genome map experiment type, and will henceforth be recalled when calculating a new cluster analysis.

The *Whole genome map type* window allows to change the general settings defined for the whole genome map type by selecting **Settings > General settings...** (⚙️).

In the *Whole genome maps experiment type settings* dialog box (see Figure 12.2.12), all general settings are listed and adjustable.

By pressing <OK>, all settings are saved in the database and will be applied by default for a new to start analysis. All these general settings can be individually overruled in the *Comparison* window by ticking the **Save as default to database** box in the respective dialogs *Filter fragments* dialog box, *Match whole genome maps (alignment)* dialog box, *Match whole genome maps (pattern)* dialog box and *Align whole genome maps* dialog box. More information on the individual settings can be found in the chapters on clustering (12.4) and aligning (12.5) whole genome maps.

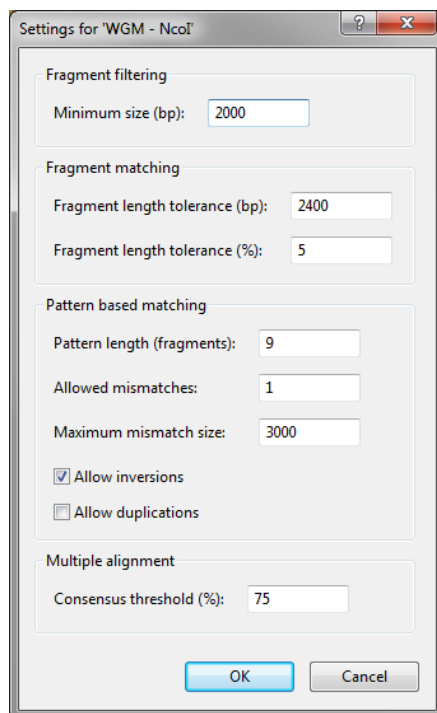


Figure 12.2.12: The *Whole genome maps experiment type settings* dialog box allows saving preferred general settings as default for the experiment type.

12.2.4 The Whole Genome Map experiment card

Clicking on the green colored dot of a whole genome map data type experiment in the *Database entries* panel of the *Main* window pops up the individual *Whole genome map* window. Alternatively, open the *Entry* window of the entry and click on the flask next to the experiment name in the *Experiments* panel. The *Whole genome map* window states the names of the entry and the experiment type in its heading.

The *Whole genome map* window consists of two dockable panels: the *Whole genome map* panel and the *Fragment list* panel (see Figure 12.2.13). The *Whole genome map* panel is further divided into the *Whole genome map name* panel, displaying the entry key name and the *Whole genome map display* panel with a customizable graphical representation of the individual whole genome map. The *Fragment list* panel lists the size of all fragments (bp) of the whole genome map in order of appearance.

In the *Whole genome map* panel the whole genome map can either be displayed in a single or multi line view. To switch between the views, select **View > Multiline view** (☰). An example of a multi line view, adjusting the fragments to the width of the *Whole genome map* window by spreading them over multiple lines, is given in Figure 12.2.13. In case of a multi line view, the *Whole genome map name* panel will additionally list the start and end positions of the displayed lines (bp). Further zooming is possible using the yellow scroll bar.

Initially, the *Whole genome map display* panel uses the active display settings, defined in the *Whole genome maps display settings* dialog box, but the display of the whole genome map fragments can be customized by selecting **View > Display settings...** (⚙️). This will pop up the *Whole genome maps display settings* dialog box, where both the *Fragment text* and the *Fragment coloring* settings can be adjusted, as detailed in 12.3.1. (The options under **Matching** and **Visible range** in the *Whole genome maps display settings* dialog box are disabled for the *Whole genome map* panel, because they are inapplicable here.)

In the *Fragment list* panel, all fragments of the whole genome map of the entry are listed in order of appearance. The size of the fragments is indicated in base pairs and a visual length indication is given by means of a bar graph. Additionally, the two columns *Color* and *Comment* will show the color and/or label text of

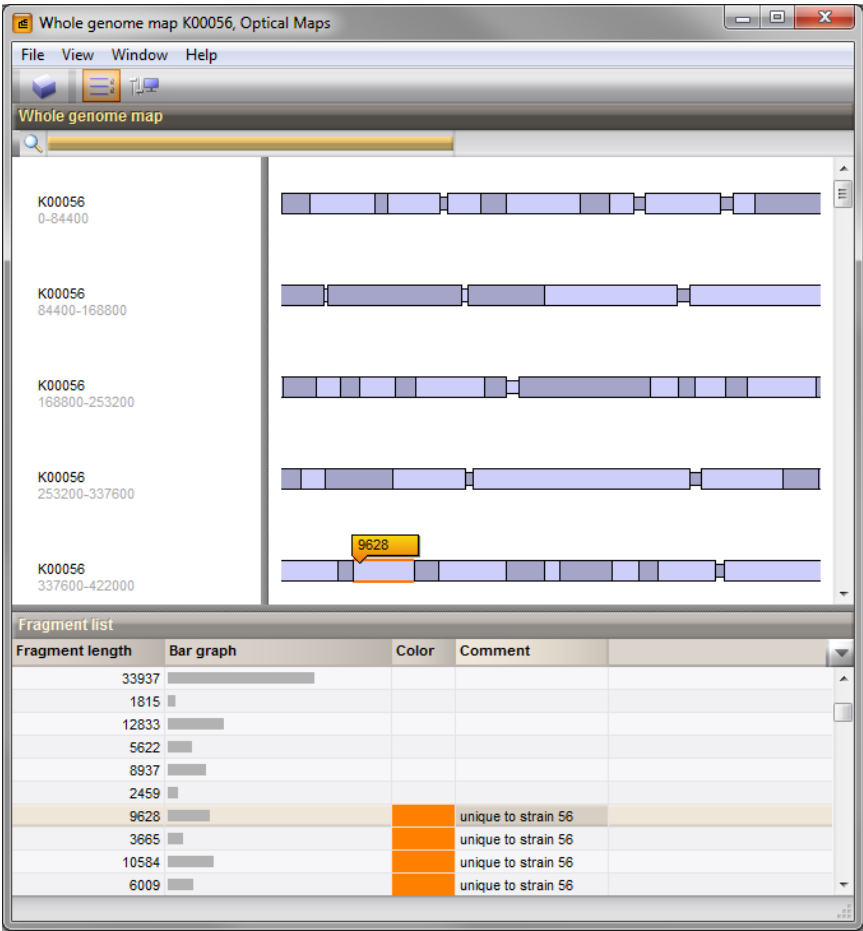


Figure 12.2.13: Whole genome map *Experiment card* window.

annotated fragments, if present (see 12.3.2 and Figure 12.2.13). If desired, the layout of this table can be altered by pressing the column properties button (⌵) in the upper right corner of the panel and selecting **Set active fields** to hide or display columns by (un)checking the respective boxes. The position of the columns can be shifted by selecting the column (e.g. color), going to the column properties button (⌵) and selecting **Move 'Color' to left** or **Move 'Color' to right**.

Selecting a whole genome map fragment in the fragment list by a single click, will simultaneously select this fragment in the *Whole genome map display* panel (see Figure 12.2.13), for easy localization of a single fragment on the whole genome map. Also vice versa, the counterpart of a fragment selected in the *Whole genome map display* panel, will automatically be selected in the fragment list.


A map fragment can be searched for in the list by pressing the column properties button (⌵) and selecting **Find in table....** This will pop-up the *Find* dialog box (see 3.2.11 for details).

If desired, the complete fragment list, including the fragment comment (if present), can be exported by copying the content to the clipboard and pasting it in a program of choice. This can be achieved by pressing the column properties button (⌵) and selecting **Copy content to clipboard**.

Chapter 12.3


Whole Genome Maps: display and search options

12.3.1 Display settings

Different options are available to visualize Whole Genome Map data in a *Comparison* window (see [13.2](#)) based on a selection of entries in the *Main* window. Before using these display options, make sure that the whole genome map data are shown in the *Experiment data* panel by clicking the  button in the *Experiments* panel. This action will enable the buttons of the Whole Genome Map toolbar in the *Experiment data* panel and the commands listed under the menu **GenomeMaps**.



We recommend to maximize the *Comparison* window, and to use maximal space for the *Experiment data* panel by minimizing the *Dendrogram* panel and *Similarities* panel.

To adjust the Whole Genome Map display settings, select **GenomeMaps > Display settings...** () to open the *Whole genome maps display settings* dialog box. Adjustable settings include visualization options for the fragments, the matching information and the stretch of the whole genome map to be displayed (see [Figure 12.3.1](#)).

To display or remove text within an individual fragment, following options are available under **Fragment text**:

- **None:** All text will be removed from the individual map fragments.
- **Fragment length (bp):** The size of the individual fragments, expressed in base pairs, will be shown on the fragments.
- **Fragment length (kb):** This option will display the size of the individual fragments in kilo bases.
- **Start position (bp):** The starting point of the given fragment on the whole genome map will be displayed in bp.
- **Start position (kb):** The starting point of the given fragment on the whole genome map will be displayed in kb.
- **Index:** All fragments on the whole genome map will be numbered individually.

To confirm your selection, press **<OK>** to view the selected display options in the *Experiment data* panel.

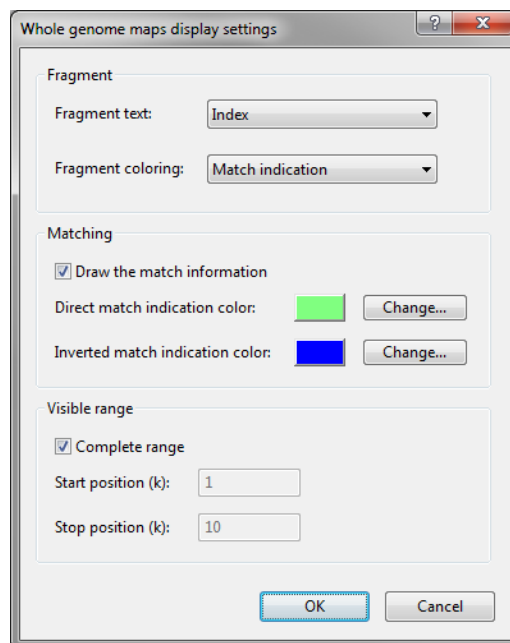


Figure 12.3.1: Adjusting display settings in the *Whole genome maps display settings* dialog box



If the selected text is not displayed in the fragment, you might need to enlarge the size of the size of the fragments by using the yellow zoom sliders (see also 2.3.7). Zooming in or out can also be done via the menu commands, toolbar buttons or shortcut keys **Layout > Zoom in** (🔍, **Ctrl+Page Up**) and **Layout > Zoom out** (🔍, **Ctrl+Page Down**), respectively. If desired, a further zoom into a specific range within a whole genome map can be accomplished by setting a **Visible range** (see further on).



The size of an individual fragment on a map (in bp) is also indicated when hovering over a given fragment in the *Experiment data* panel in the *Comparison* window.

Next to the adding text information to the individual fragments of a whole genome map, the coloring of the fragments can also be modified to facilitate its viewing by the user. The following options are available under **Fragment coloring** in the *Whole genome maps display settings* dialog box (see Figure 12.3.1):

- **Label:** Annotated fragments (see 12.3.2) will be colored according to the label given and hence accentuated for easy recognition.
- **Alternating:** All fragments will be colored alternating using two shades of grey, to better identify the individual fragments on a map.
- **Fragment length:** All fragments on a map will be colored in different shades of blue according to the fragment size. The darker the color, the bigger the fragment size.
- **Match indication:** When *Pattern Match Classes* were calculated (see 12.6.3), this option will multi-color the fragments accordingly.

Press **<OK>** to confirm the option and to view the resulting display in the *Comparison* window.

Besides the various display options for the individual whole genome map fragments, there are also viewing options on the matching information. In the *Whole genome maps display settings* dialog box (see Figure 12.3.1) you can select or unselect if you want to see the matching information drawn between the whole genome maps in a comparison. The *direct match indication* shows the corresponding matching fragment

(using the chosen settings, see 12.5) on the whole genome map on top or beneath a given whole genome map of an entry in a comparison in a sense direction. The *inverted match indication* will highlight possible inversions by indicating matching fragments in an anti-sense direction. The display color of both the direct and inverted match indication can be customized by pressing the according **<Change...>** button. This pops up the *Color* dialog box.

An example of the match indication display is given in Figure 12.3.2. In this example direct matches are colored green and inverted matches blue. Fragments in a whole genome map not matching with a fragment from the whole genome map above or below in the comparison are designated with a red line.

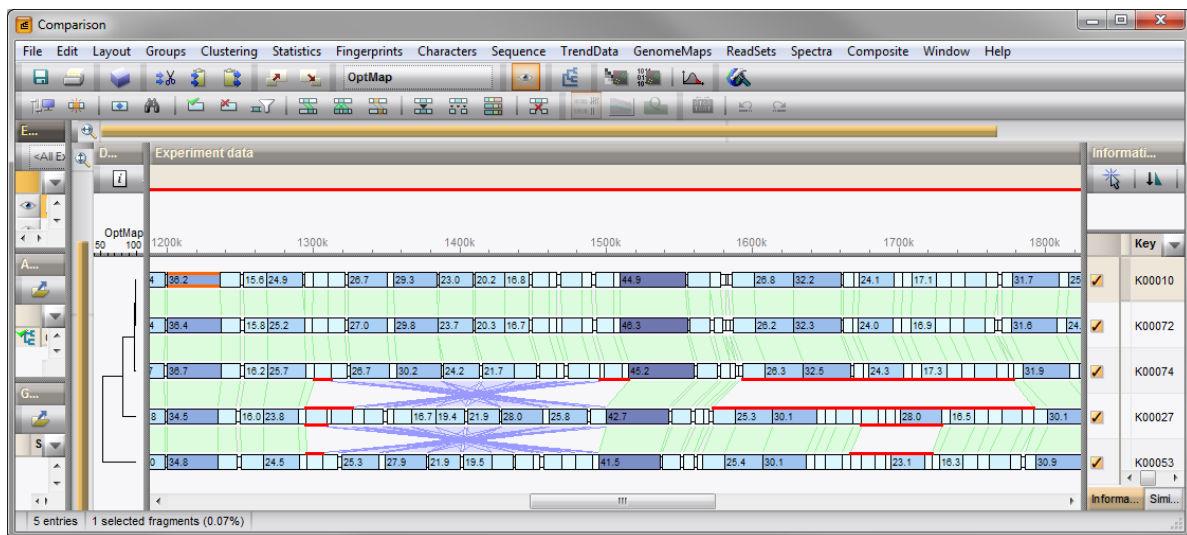


Figure 12.3.2: An example of the match indication display in the *Comparison* window, highlighting genomic differences



The matching information will only be displayed if once an alignment (see 12.5) of whole genome maps is calculated. Inverted match indications will only be shown if a pattern based calculation was performed (see 12.5.2.4) and inversions were detected.

In some cases it can be interesting to further zoom into a specific region of a whole genome map to look at differences at the individual fragment level. If zooming in by the zoom sliders, the menu commands, toolbar buttons or shortcut keys **Layout > Zoom in** (🔍, **Ctrl+Page Up**) and **Layout > Zoom out** (🔍, **Ctrl+Page Down**), remains insufficient, a further so-called **Visible range** can be set. Hereto, in the *Whole genome maps display settings* dialog box (see Figure 12.3.1), first inactivate the *Complete range* selection in order to manually enter the desired start and end positions of the **Visible range** frame, expressed in kilo bases. Confirm your selection by pressing **<OK>**, to view the selected **Visible range** in the *Experiment data* panel. Further zooming can now be accomplished by using the classic zoom tools mentioned above for a highly detailed view.

12.3.2 Fragment annotation: customizable labels

This is an interesting option to label selected fragments. For instance, when genomic differences are detected after calculating an alignment of whole genome maps in the *Comparison* window (see 12.5), this will allow to annotate these fragments for further reference. In order to add a label to a single or to multiple fragments, these fragments first need to be selected. This can be accomplished in various ways, either manually or via an automated search. The latter option is further detailed in 12.3.3.

A single individual fragment can easily be selected by hovering over the whole genome map and clicking the left mouse button on the fragment of interest. Multiple fragments can be simultaneously selected by

holding the **Shift** or **Ctrl** key and clicking on the desired fragments on the whole genome map. A range of fragments can be selected by pressing the left mouse button and dragging it over the fragments of interest, whether this be on individual whole genome maps or across different whole genome maps in a *Comparison* window. Selected fragment(s) are indicated by an orange line, as shown in the middle image of Figure 12.3.3. This manual selection of fragments has only a temporal status; henceforth by clicking outside of the displayed whole genome maps all selected fragments will become unselected.



Figure 12.3.3: Visualization of different steps for the annotation of whole genome map fragments with customizable labels.



Selected fragments can easily be highlighted to facilitate their viewing, by selecting **GenomeMaps > Highlight selection...** (🖋️, **Ctrl+H**) (see 12.3.3.5).

Once the appropriate fragments are selected, labels can be appended using the **GenomeMaps > Edit fragment label...** (📄) command. In the *Fragment comment* dialog box that popped up the label can be customized by adding a text comment of your choice in the **Fragment label** box (see Figure 12.3.4).

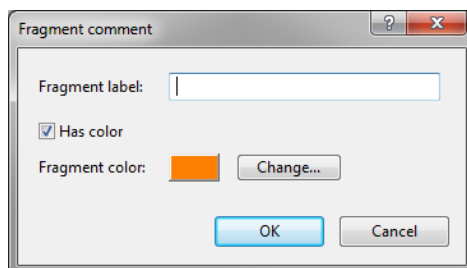


Figure 12.3.4: The *Fragment comment* dialog box allows to annotate fragments with customizable labels.


By selecting **Has color**, the **Fragment color** can additionally be customized for visualization purposes by pressing the **<Change...>**. In the popped up *Color* dialog box, colors can be adjusted. Pressing **<OK>** will complete the fragment annotation.

An example of customized labels, colored with orange diamonds, can be seen in the rightmost image of Figure 12.3.3. Hereto, the fragment visualization was set to **Fragment coloring** according to the label in the *Whole genome maps display settings* dialog box (see 12.3.1 and Figure 12.3.1).


While hovering over an annotated fragment on a whole genome map, the text of the customized label will be indicated, as shown in the rightmost image of Figure 12.3.3). One of the major advantages of annotating fragments, apart from the at-a-glance viewing options, is that the added text is searchable and henceforth a useful tool to readily find and/or select these fragments at a later stage (for details see 12.3.3.2 and Figure 12.3.5).

12.3.3 Fragment search and selection

12.3.3.1 Introduction

Different options are available to search and select Whole Genome Map data in a *Comparison* window (see 13.2). Before using any of the options explained in this section, make sure that the Whole Genome Map data are shown in the *Experiment data* panel by clicking the  button in the *Experiments* panel. This action will enable the buttons of the Whole Genome Map toolbar in the *Experiment data* panel and the commands listed under the menu **GenomeMaps**. In this section different options for automated searches of particular fragments are discussed. Manual selection of fragments is equally possible, as described under 12.3.2.

12.3.3.2 Fragment search

A fragment search can be launched by selecting **GenomeMaps > Find fragments...** (, **Ctrl+F**). In the *Find fragments* dialog box that pops up (see Figure 12.3.5), fragments can be searched for according to their size, label or activity status. Annotated fragments (see 12.3.2), can be quickly pursued by entering (parts of) the label text next to **Text in label** in the *Find fragments* dialog box (see Figure 12.3.5). In this way, previous discoveries of genomic differences can be promptly picked-up again. By pressing **<OK>**, the resulting fragments are selected and indicated with an orange border in the *Experiment data* panel, as shown in the middle image of Figure 12.3.3.

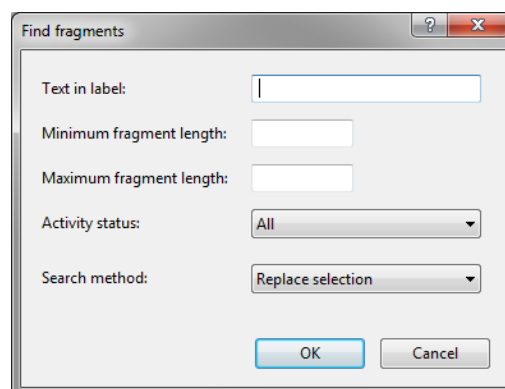


Figure 12.3.5: Fragment search options in the *Find fragments* dialog box.



The resulting fragments from a search action are only temporarily selected; henceforth by clicking outside of the displayed whole genome map fragments, these will become unselected.



The resulting fragments can easily be highlighted to facilitate their viewing, by selecting **GenomeMaps > Highlight selection...** (, **Ctrl+H**) (see 12.3.3.5).

Whole genome map fragments can also be searched for according to their size. When entering a **Minimum fragment length** in the *Find fragments* dialog box, expressed in base pairs, all fragments with a size equal to or higher than the indicated size will be obtained. Contrariwise, by entering a **Maximum fragment length**, all fragments with a size equal to or lower than the indicated size will be retained. A size range for delineating fragments can consequently be specified by entering both a **Minimum fragment length** and **Maximum fragment length** in bp (see Figure 12.3.5). Pressing **<OK>** to confirm the search option will select the corresponding fragments in the *Experiment data* panel of the *Comparison* window.

The resulting fragment selection is now available for further actions, such as annotation (see 12.3.2), changing the activity status (see 12.3.3.4), visualization (see 12.3.3.5), finding matching fragments (see 12.3.3.6) or charting (see 14).

Parallel to a size or label based quest, there is an option to take into account the so-called **Activity status** of the fragments. In brief, inactive fragments are always excluded from all further calculations (for more details, see 12.3.3.4). In the *Find fragments* dialog box, following options are available under **Activity status** (see Figure 12.3.5):

- *All*: All fragments compliant with the specified size and/or label based search parameters will be retrieved, regardless of their activity status.
- *Active only*: Only fragments with an active status and compliant with the other specified search options will be recovered by the search action. For instance, when a further refinement of a set of active fragments is desired, the resulting fragments can easily be inactivated (see 12.3.3.4).
- *Inactive only*: This option will limit the search results to fragments with an inactive status. This is, for instance, a convenient way to quickly reactivate a particular group of fragments.

Alongside the size, label and activity status based search options, the **Search method** can be specified in the *Find fragments* dialog box (see Figure 12.3.5). Following options are available:

- *Replace selection*: Using this option, any previously selected fragments (either manual or from an automated search) will become unselected and replaced by the fragments compliant with the search parameters.
- *Add to selection*: The fragments from this search action will be added to all currently selected map fragments. In this way, different search actions can be combined. For example, if one wishes to extend a series of manually selected fragments with a specific search criterion.
- *Search in selection*: This option will narrow down the search results of the current selection of map fragments using the specified search criteria, e.g. for refinement of a particular search action.

Press <OK> to confirm and view the resulting fragments.

12.3.3.3 Fragment filtering

The purpose of this tool is to quickly filter out fragments below a given size. For instance, since very small fragments of DNA immobilized to optical glass surface can become detached and washed away during wet chemistry processing, the resolution of the whole genome mapping technique using a single restriction enzyme is approximately 2 kb. For this reason, it could be desirable to exclude smaller fragments from further calculations. Whereas a fragments resulting from a search action (see 12.3.3.2) only remain temporarily selected, the fragments compliant with the filter size will be permanently retained and have an active status, since the **Fragment filtering** settings are saved in the experiment type settings (see also 12.2.3 and Figure 12.2.12). Hence, the small, filtered out fragments will be recalled when a comparison is opened and remain inactive. For more details on the activity status, see 12.3.3.4.

To define a fragment filter, select **GenomeMaps > Filter fragments...**  to open the *Filter fragments* dialog box (see Figure 12.3.6).

The fragment filter, as defined by the size in bp entered next to **Minimum size (bp)** in the *Filter fragments* dialog box (see Figure 12.3.6), will inactivate all fragments with a size smaller than the indicated size, and keep all fragments with a size equal to or higher than the minimum size active for further analyses. Pressing <OK> will apply the filtering and save the **Fragment filtering** settings with the actual whole genome map experiment type settings (see 12.2.3), for the current database session. By ticking the **Save as default to database** option, the specified **Fragment filtering** settings are saved within the database and will be default applied when a new BioNumerics session is started.

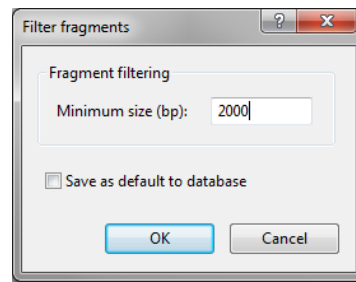


Figure 12.3.6: The *Filter fragments* dialog box enables size-based fragment filtering.

The filtered out fragments not compliant with the minimum filter size, are displayed consistent with all inactive fragments, i.e. by a smaller height of the fragments at issue compared to the normal active fragments, as shown in Figure 12.3.7 (see also 12.3.3.4).



Filtered out fragments can only be reactivated by reapplying the fragment filter with no **Minimum size** (or a smaller size in case of partial reactivation), and **NOT** by the command *GenomeMaps > Activate selected fragments...* (📁), discussed in 12.3.3.4.

12.3.3.4 Activate / Inactivate selected fragments

The activity status of whole genome map fragments divides them into two possible states: active or inactive individual map fragments. Fragments with an inactive status are always ignored in further calculations (similarity based clustering, alignments, etc.). Upon import, all whole genome map fragments have an active state by default. In certain cases, for instance, it can be interesting to exclude small or fragments considered unreliable from further analysis. Ergo BioNumerics provides the options to activate or inactivate individual map fragments.

In order to change the activity status, the fragment(s) of interest first need to be selected. This can either be accomplished manually or via an automated search, as described in 12.3.2 and 12.3.3.2, respectively. By selecting *GenomeMaps > Inactivate selected fragments...* (🗑️), the fragment selection is rendered inactive. Contrariwise, using the *GenomeMaps > Activate selected fragments...* (📁) command will change the activity status of the selected fragments from inactive to active, with one exception as detailed below.

Inactive whole genome map fragments are always displayed distinctively from active ones, by means of a smaller height for the fragments with an inactive state, as shown in Figure 12.3.7.

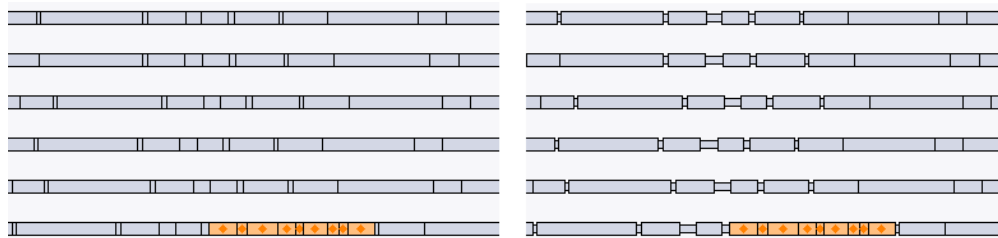


Figure 12.3.7: Visualization example of inactive fragments.

Whereas fragments resulting from a search action are only temporarily selected, the **Activity status** is saved in the database. Reactivation can be accomplished via a simple search action through the option **Activity status** in the *Find fragments* dialog box, with the exception of filtered out fragments (see 12.3.3.3).



Whole genome map fragments, inactivated by using the **GenomeMaps > Filter fragments...** (🔍), **cannot** be reactivated using the **GenomeMaps > Activate selected fragments...** (🔍). These filtered out fragments can only be reactivated by reapplying the fragment filter, as discussed in 12.3.3.3.

12.3.3.5 Visualization of selected fragments

This tool is very convenient for a quick at-a-glance discovery of selected fragments resulting from various search actions. To name a few examples: following a fragment search in the *Find fragments* dialog box (see also 12.3.3.2), finding discriminating fragments in the *Find discriminating fragments* dialog box (see also 12.6.2) or finding matching fragments using the **GenomeMaps > Select matching fragments** (🔍, **Ctrl+M**) command (see also 12.3.3.6).

A selection of whole genome map fragments can be readily highlighted by selecting **GenomeMaps > Highlight selection...** (🔍, **Ctrl+H**). The highlighting can quickly be undone by reselecting **GenomeMaps > Highlight selection...** (🔍, **Ctrl+H**) once more.

Selected fragments, marked by a double orange border, are highlighted by dimming the other map fragments, as shown in Figure 12.3.8.



Figure 12.3.8: Visualization of highlighted map fragments for at-a-glance viewing of selected fragments.

12.3.3.6 Selecting matching fragments

Rather than finding discriminating fragments, it can even be interesting to find matching fragments for a specific map fragment in a series of whole genome maps. This option was designed to quickly find corresponding fragments, matching to the specified settings, on whole genome maps across entries in a comparison. Together with the highlighting tools, (see 12.3.3.5), this allows an at-a-glance discovery of matching fragments across all entries in the *Experiment data* panel of a *Comparison* window.

Upon manual or automated selection of an individual or multiple fragment(s) of interest, matching fragments, i.e. fragments meeting the active **Fragment matching** and **Pattern based matching** criteria specified in the *Whole genome maps experiment type settings* dialog box (see Figure 12.2.12), can be quickly picked up by selecting **GenomeMaps > Select matching fragments** (🔍, **Ctrl+M**).

As a result, the matching fragments will be selected, and hence marked by an orange border. Further highlighting is possible by selecting **GenomeMaps > Highlight selection...** (🔍, **Ctrl+H**), as shown in Figure 12.3.8.




Chapter 12.4

Clustering of Whole Genome Map data

12.4.1 Starting up a cluster analysis

Cluster calculation of whole genome maps (this chapter), as well as alignment (see [12.5](#)), display of experiment images of selected entries (see [12.3](#)), etc. is directed from the *Comparison* window. More details on the general functionalities of the *Comparison* window are discussed in [13.2](#).

Please note that cluster analysis of whole genome maps requires both the Whole Genome Maps (**GM**) and the Tree and Network Inference (**TN**) modules to be present in your BioNumerics configuration (see [2.1.4](#)).

After making a selection of database entries in the *Main* window, a comparison can be created and displayed in a new *Comparison* window by highlighting the *Comparisons* panel and selecting **Edit > Create new object...** (). To calculate a cluster analysis, first select in the *Experiments* panel the whole genome map type on which the cluster analysis should be based. Optionally, the whole genome maps can be displayed by pressing the  button next to the experiment name. Secondly, select **Clustering > Calculate > Cluster analysis (similarity matrix)...** to proceed. Alternatively, the  button can be used to select *Calculate cluster analysis* from the pop-up menu shown in Figure [13.2.4](#). As a result, the *Comparison settings* wizard will appear (see Figure [12.4.1](#)).

12.4.2 Whole Genome Map comparison settings

BioNumerics offers two main approaches for comparing whole genome maps: *Alignment based* and *Pattern based* comparison. The *Alignment based* approach uses a size tolerance algorithm for pairwise comparing whole genome map fragments, taking into account the order of the map fragments in a sense direction only. The *Pattern based* approach is based on a newly developed algorithm, which takes into account a neighborhood of fragments matching within a given size tolerance (all or not allowing mismatches) (see also [12.5.2.5](#)). In contrast with the alignment based comparison, the pattern match concept allows to detect inversions and duplications.

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix (page 1) and the clustering method to be applied (page 2). The *Similarity coefficient* wizard page deals with the similarity coefficient (see Figure [12.4.1](#)).

The hierarchical representation on the left hand side provides an overview of the available coefficients. Depending on the selected coefficient, the relevant settings are displayed on the right. The coefficients are subdivided in two categories: *Alignment based* and *Pattern based*. Each of the categories can be collapsed by clicking on the small ”-” (minus) sign that precedes the category name.

The *Alignment based* approach essentially makes use of a size tolerance algorithm for pairwise comparing whole genome map fragments, taking into account the order of map fragments in a sense direction.

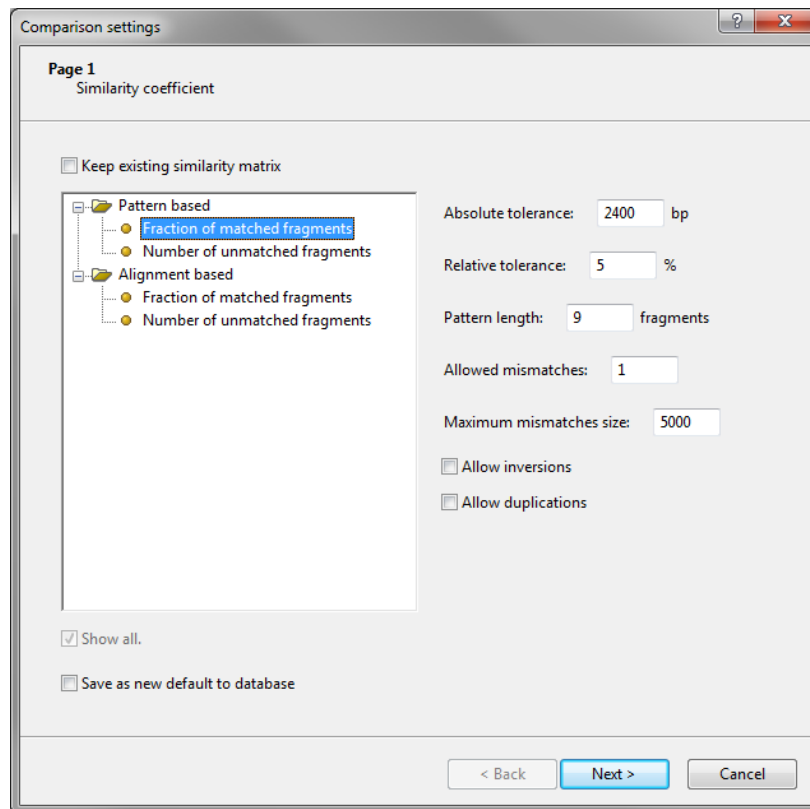


Figure 12.4.1: The *Similarity coefficient* wizard page, which deals with the choice of the similarity coefficient.

The following parameters can be specified for the *Alignment based* approach:

- **Absolute tolerance:** An absolute tolerance (indicated in bp) will be considered for matching fragments: if the size difference is lower than the specified tolerance size, the whole genome map fragments will be regarded as matching.
- **Relative tolerance:** A relative tolerance in relation to the average length of two fragments (indicated in % of the fragment size) is considered for matching: if the size difference of the two fragments is lower than this relative tolerance, the fragments will be indicated as matching.

Both the **Absolute tolerance** and **Relative tolerance** settings can be used on an individual basis (by specifying solely one option), but they can also be combined. In this case, BioNumerics will always apply the highest tolerance for considering fragments to match. For instance, an absolute tolerance of 1000 bp will match two fragments with a length of 8000 and 8600 bp, but will not match two fragments with sizes of 60000 and 62000 bp. By additionally allowing a relative tolerance of 5 %, the fragments of 8000 and 8600 bp will still match, according to the highest tolerance threshold (in this case the absolute tolerance of 1000 bp; relative tolerance here is only 5 % or 415 bp) and the fragments with a size of 60000 and 62000 bp will also match. Here the relative tolerance of 5 % (or 3050 bp) will be applied, as it exceeds the absolute tolerance of 1000 bp. In this way, a higher tolerance can be allowed for larger fragments.

The *Pattern based* matching takes into account a neighborhood of N fragments matching within a given size tolerance, with the possibility of allowing a number of mismatches.

The following parameters can be specified for the *Pattern based* approach:

- **Absolute tolerance:** An absolute tolerance (indicated in bp) will be considered for matching frag-

ments: if the size difference is lower than the specified tolerance size, the whole genome map fragments will be regarded as matching.

- **Relative tolerance:** A relative tolerance in relation to the average length of two fragments (indicated in % of the fragment size) is considered for matching: if the size difference of the two fragments is lower than this relative tolerance, the fragments will be indicated as matching.
- **Pattern length:** The pattern length defines the number N of neighboring fragments taken into account when recognizing a pattern of N fragments that is consistent with the size tolerance criteria.
- **Allowed mismatches:** This option specifies the number of mismatches allowed in the pattern, i.e. number of fragments in the pattern that do not meet the specified tolerance settings.
- **Maximum mismatches size:** This parameter limits the maximum size (bp) of the allowed mismatches in the pattern. For instance, if one mismatch is allowed (one fragment in the pattern not meeting the size tolerance criteria) and its size is lower than the indicated mismatch size, the pattern of fragments will be considered as matching. If two or more mismatches are allowed, the sum of the mismatch sizes should be lower than the specified maximum mismatches size (bp) in order for the pattern of fragments to be considered as matching.
- **Allow inversions:** If checked, pattern recognition will be performed in both a sense and anti-sense direction, allowing to detect inversions between whole genome maps.
- **Allow duplications:** If checked, fragments will be allowed to match more than one corresponding fragment in an other whole genome map.

For both the *Pattern based* and *Alignment based* approach, two similarity coefficients can be chosen to calculate the pairwise similarities:

- **Fraction of matched fragments:** This option calculates the relative fraction (expressed in %) of matching fragments, considering the specified parameters.
- **Number of unmatched fragments:** This is essentially a distance coefficient, and reports the absolute number of fragments that are not matching between two entries.

If a similarity matrix already exists for the selected experiment, the option **Keep existing similarity matrix** appears. When selected, the previously calculated similarity matrix will be used for calculating the dendrogram and all coefficient parameters (for both *Pattern based* and *Alignment based* coefficients) will appear gray (disabled).

Checking the option **Save as new default to database** will save the specified comparison settings as default settings in the database. If not checked, the last comparison settings will only apply during the current BioNumerics session; if the software is closed, these settings will not be saved. The settings as defined in the *Comparison settings* wizard are stored along with the whole genome map experiment type. A dialog box with the same settings can be called from the *Whole genome map type* window (see 12.2.3).



In 12.4.3 is discussed how to have BioNumerics automatically calculate the optimal pattern and alignment based parameter values for a whole genome map experiment type.



The pattern match settings can be evaluated via **GenomeMaps > Pattern match statistics...**, as discussed in 12.5.2.5.

Once the similarity coefficients are specified, press <Next> to proceed to the *Cluster analysis* wizard page, which deals with the calculation of a dendrogram from the similarity matrix. This is comprehensively discussed in 13.2.6. After selecting an appropriate clustering algorithm, press <Next> again to start the cluster analysis. When finished, a dendrogram and corresponding similarity matrix are shown for the experiment.

12.4.3 Optimization of similarity coefficient parameters

BioNumerics offers an interesting option to automatically calculate the optimal settings for similarity coefficient settings for a given whole genome map experiment type. The principle is as follows: the user selects a number of entries which he or she wants to cluster into a comparison. The program will calculate similarity matrices with different values for a selected parameter (e.g. absolute tolerance,...). Within a limited range, the optimal setting for a similarity coefficient parameter yields the matrix with the highest group contrast: scores as high as possible within groups and as low as possible between groups. This translates in the highest standard deviation on the matrix of similarity values. The same process can be launched to find the best range for the next parameter (e.g. relative tolerance), etc.. Given the principle of the method, it is important to select entries belonging to different groups or showing a maximum of heterogeneity.

The best way to proceed is with comparison groups (see 13.3.4) already defined, e.g. based upon cluster analysis or partitioning (see 13.3.4). The program will then optimize the intergroup separation based upon these groups. If no groups are defined, the standard deviation of the whole matrix is optimized, which also works in case the comparison contains some groups of more related maps.

Click on the whole genome map type in the *Experiments* panel and select **Clustering > Optimize similarity coefficient parameter...**. The *Optimization parameter* wizard page appears (see Figure 12.4.2).

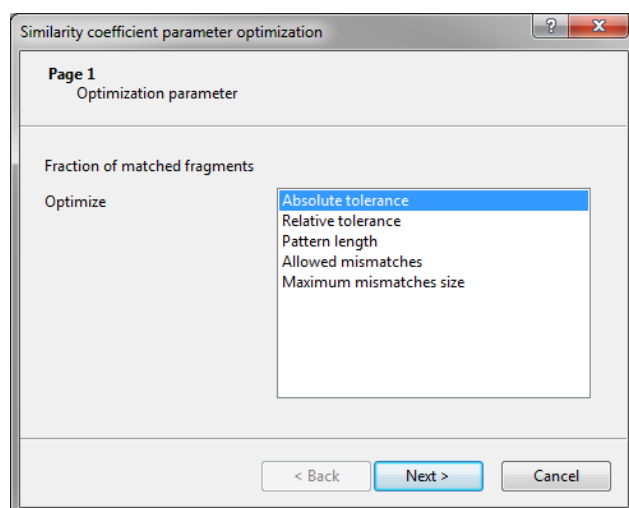


Figure 12.4.2: The *Optimization parameter* wizard page allows selection of the parameter to be optimized.

On the first page, the parameter to be optimized can be selected. Depending on whether an *Alignment based* or *Pattern based* matching is used, different parameters are listed: **Absolute tolerance** and **Relative tolerance** for both alignment and pattern based coefficients; for pattern based coefficients the parameters **Pattern length**, **Allowed mismatches** and **Maximum mismatches size** can also be selected for optimization.

The currently used coefficient (e.g. fraction of matched fragments) is indicated.

Select the parameter to optimize, e.g. **Allowed mismatches**, and press **<Next>**. The second page of the *Optimization range* wizard page is now displayed (see Figure 12.4.3).

With **From** and **To**, a range of optimization values can be specified as the lowest and highest value that the program will evaluate. **Step** is the interval at which optimization values will be evaluated. It is important to keep in mind that a wide range and a small **Step** will result in a large number of optimization values to be tested and therefore in a long calculation time.

Enter a range and a step size and press **<Next>**. BioNumerics now tries to calculate the best **Allowed mismatches** value. When an optimal value was found within the range specified, this value is reported. The program asks "Copy this value to the similarity coefficient?".

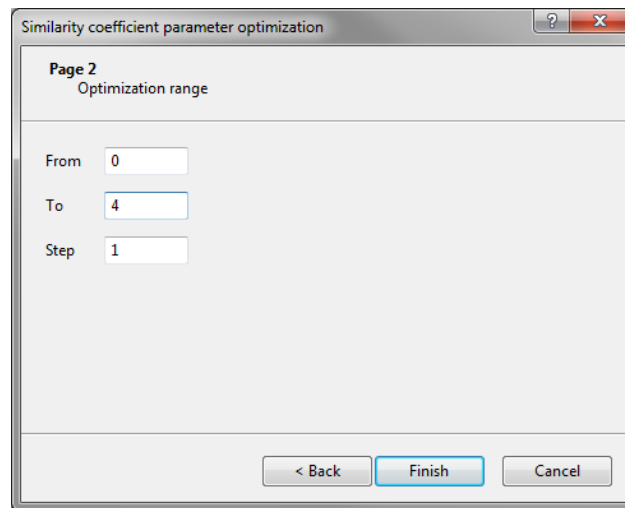


Figure 12.4.3: The *Optimization range* wizard page, where the optimization range can be specified.

If you press <Yes>, the comparison settings will be updated with the optimal *Allowed mismatches* value. Next, the *Parameter optimization* window appears (see Figure 12.4.4).



If the message "No optimal value was found within the given range." appears, this means that no maximum was detected in the parameter optimization curve: the curve is either flat or continuously increasing or decreasing over the examined range. A solution could be to base the parameter optimization on a more heterogeneous set of samples and/or to define comparison groups prior to launching the calculations, or to adjust the range.

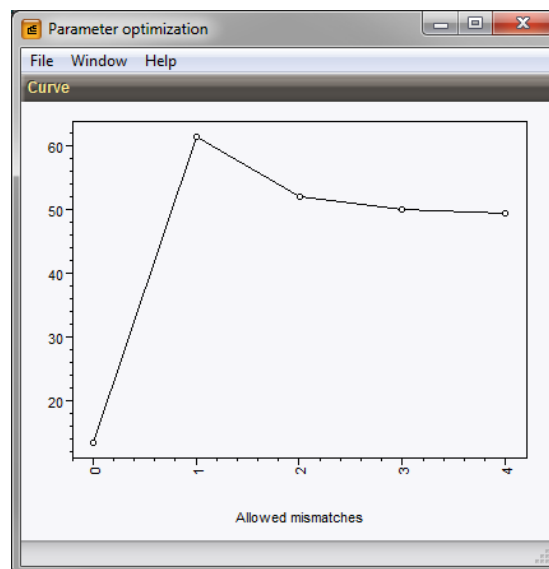


Figure 12.4.4: The *Parameter optimization* window.


The *Parameter optimization* window shows a diagram with the group separation (Y-axis) in function of the allowed mismatches (X-axis). Obviously, the maximum in the curve corresponds to the optimal optimization value. The window can be closed with **File > Exit**.

The procedure can be repeated for the other similarity coefficient parameters *Absolute tolerance*, *Relative tolerance*, *Pattern length* and *Maximum mismatches size*.

12.4.4 Exporting Whole Genome Map information from a comparison

The information, contained in the database information fields for the entries in the comparison, can be exported with **File > Export > Export database fields....** The export file, popped up as `export.csv` in Microsoft Excel, contains all information from the *Information fields* panel.

The similarity matrix can be exported with **File > Export > Export similarity matrix....** The export file, popped up as `export.txt` in Notepad, is a tab-delimited text file which contains the similarity values with the entry keys as descriptors.

Information on how a cluster analysis was calculated can be displayed in a report by selecting **Clustering > Show information** (). This *Report* window can be saved in HTML or TXT format.

Chapter 12.5

Alignment of Whole Genome Maps

12.5.1 Background

Like cluster analysis (see [12.4](#)) and the display of experiment images of selected entries (see [12.3](#)), the alignment of whole genome maps is done in the *Comparison* window. More details on the general functionalities in the *Comparison* window are discussed in [13.2](#).

BioNumerics offers the same two approaches for aligning whole genome map fragments as for cluster analysis: *Alignment based* and *Pattern based*. The *Alignment based* approach uses a size tolerance algorithm for pairwise comparing whole genome map fragments, taking into account the order of map fragments in a sense direction only. The *Pattern based* approach is based on a newly developed algorithm, which takes into account a neighborhood of a number of fragments matching within a given size tolerance (all or not allowing mismatches). In contrast to the size tolerance matching of map fragments, the pattern match concept allows to detect possible inversions and duplications.

The following paragraphs will discuss the pairwise ([12.5.2](#)) and global ([12.5.3](#)) alignment of whole genome maps.

Please note that for aligning whole genome maps both the Whole Genome Map data module (GM) and the Tree and Network Inference module (TN) need to be present in your BioNumerics configuration (see [2.1.4](#)).

12.5.2 Pairwise alignment

12.5.2.1 Introduction

After making a selection of database entries in the *Main* window, a comparison can be created and displayed in a new *Comparison* window by highlighting the *Comparisons* panel and selecting *Edit > Create new object...* (🛠️). In the *Experiments* panel, select the whole genome map experiment type and display the maps by pressing the 🖼️ button next to the experiment name or by selecting *Layout > Show image* (🖼️).

12.5.2.2 Fragment match based alignment of whole genome maps

To start up an alignment based purely based on size tolerance matching of fragments, select *GenomeMaps > Alignment match...* (🖼️) to show the *Match whole genome maps (alignment)* dialog box (see [Figure 12.5.1](#)).

The *Alignment match* approach makes use of a size tolerance algorithm for pairwise comparing whole genome map fragments, taking into account the order of map fragments in a sense direction only.

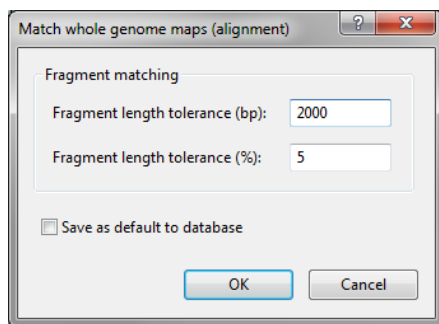


Figure 12.5.1: Adjusting the size threshold for fragment matching in the *Match whole genome maps (alignment)* dialog box.

The following parameters can be specified for fragment matching:

- **Absolute tolerance:** An absolute tolerance (indicated in bp) will be considered for matching fragments: if the size difference of the two fragments is lower than the specified tolerance size, the whole genome map fragments will be regarded as matching.
- **Relative tolerance:** A relative tolerance in relation to the average length of two fragments (indicated in % of the fragment size) is considered for matching: if the size difference of the two fragments is lower than this relative tolerance, the fragments will be regarded as matching.

Both the **Absolute tolerance** and **Relative tolerance** settings can be used on an individual basis (by solely specifying one option), but they can also be combined. In this case, BioNumerics will always apply the highest tolerance for considering fragments to match (see also 12.4.2).

Checking the option **Save as new default to database** will save the specified fragment matching settings as default settings in the database. If not checked, the last used settings will only apply during the current BioNumerics session; if the software is closed, these settings will not be saved. The settings as defined in the *Match whole genome maps (alignment)* dialog box are stored along with the whole genome map experiment type. A dialog box with the same settings can be called from the *Whole genome map type* window (see 12.2.3).

An example of a pairwise alignment based on a fragment match is given in Figure 12.5.2. In this example the direct matches between fragments on the whole genome map above and/or below are indicated green. Map fragments not matching, according to the specified size tolerance parameters, with a fragment from the whole genome map above or below in the comparison are designated with a red line. The order of the fragments displayed in the *Experiments* panel can be adjusted by using **Edit > Move entry up** (↑, Shift+Up) and **Edit > Move entry down** (↓, Shift+Down) when an entry is highlighted in the *Information fields* panel. Alternatively, selected entries can be brought to the top by selecting **Edit > Arrange entries > Bring selected entries to top** (⚡, Ctrl+T). (see also 13.2).



If no match indication is displayed in the *Experiments* panel, check the option **Draw the match information** in the *Whole genome maps display settings* dialog box, accessible via **GenomeMaps > Display settings...** (⚙).

12.5.2.3 Alignment tools

Any alignment of whole genome maps can be easily removed by selecting **GenomeMaps > Remove alignment** (✖). Apart from the zoom functionalities and the visible range (see 12.3.1), the view of the displayed alignment can be adjusted for a more detailed look on a region around a certain map fragment in a pairwise alignment by using **GenomeMaps > Line up** (📏). This is a handy tool, for instance, if (a) genomic

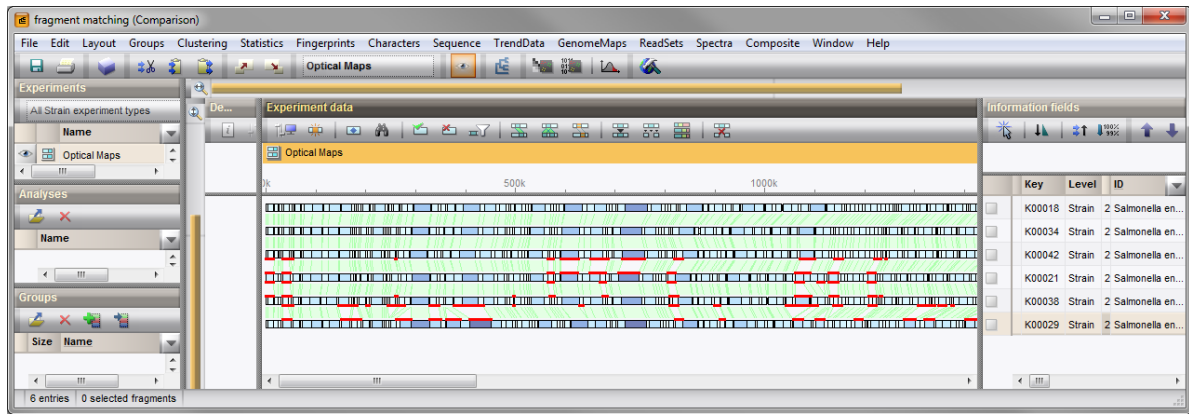


Figure 12.5.2: Size tolerance based pairwise alignment of whole genome maps in the *Comparison* window.

event(s) on one of the neighboring maps in the alignment has locally shifted the pairwise match indication. The line-up tool is demonstrated in Figure 12.5.3, for adjusting the view around the indicated fragment of 54686 bp. Suppose you want to see the behavior of this selected fragment (bordered with two orange lines) in the other whole genome map entries in the comparison. From the left hand side image in Figure 12.5.3 this is not straightforward given the shifted match indication. After using the line-up tool by first selecting the fragment of interest, followed by **GenomeMaps > Line up** (🔍), it is much easier to evaluate, for instance, the presence of corresponding fragments among the other entries, as demonstrated in the right hand side image of Figure 12.5.3. By simply directly selecting another fragment, followed by **GenomeMaps > Line up** (🔍), another whole genome map region can be adjusted for detailed viewing. The line-up can be undone by removing the alignment via **GenomeMaps > Remove alignment** (🗑️).



Figure 12.5.3: The line-up tool allows adjusting the view of the alignment around a given whole genome map fragment.

The alignment as displayed in the *Comparison* window, can be printed and exported from the *Comparison print preview* window after selecting **File > Print preview...** (🖨️, **Ctrl+P**), as detailed in 13.3.9.

12.5.2.4 Pattern match based alignment of whole genome maps

To start up an alignment based on pattern matching of fragments, select **GenomeMaps > Pattern match...** (🔍) to open the *Match whole genome maps (pattern)* dialog box (see Figure 12.5.4).

The *Pattern based* matching takes into account a neighborhood of N fragments matching within a given size tolerance, possibly allowing a number of mismatches.

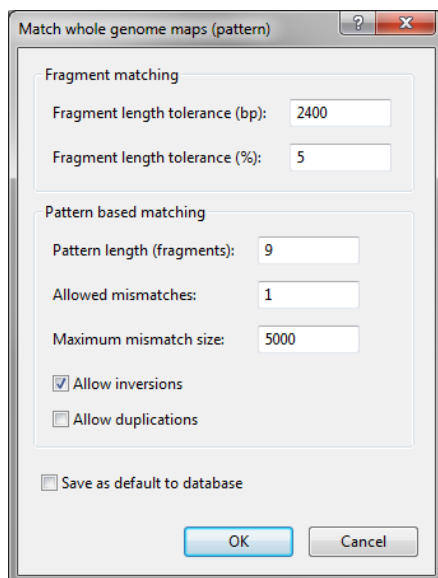


Figure 12.5.4: Adjusting the pattern match settings in the *Match whole genome maps (pattern)* dialog box.

The following parameters can be specified for pattern matching:

- **Absolute tolerance:** An absolute tolerance (indicated in bp) will be considered for matching fragments: if the size difference of the two fragments is lower than the specified tolerance size, the whole genome map fragments will be regarded as matching.
- **Relative tolerance:** A relative tolerance in relation to the average length of two fragments (indicated in % of the fragment size) is considered for matching: if the size difference of the two fragments is lower than this relative tolerance, the fragments will be regarded as matching.
- **Pattern length:** The pattern length defines the number of neighboring fragments taken into account when recognizing a pattern of N fragments that are consistent with the size tolerance criteria.
- **Allowed mismatches:** This option specifies the number of mismatches allowed in the pattern, i.e. number of fragments in the pattern that do not meet the specified tolerance settings.
- **Maximum mismatches size:** This parameter limits the maximum size (bp) of the allowed mismatches in the pattern. For instance, if one mismatch is allowed (one fragment in the pattern not meeting the size tolerance criteria) and its size is lower than the indicated mismatch size, the pattern of fragments will be considered as matching. If two or more mismatches are allowed, the sum of the mismatch sizes should be lower than the specified maximum mismatches size (bp) in order for the pattern of fragments to be considered as matching.
- **Allow inversions:** If checked, pattern recognition will be performed in both a sense and anti-sense direction, allowing to detect inversions between whole genome maps.
- **Allow duplications:** If checked, fragments will be allowed to match more than one corresponding fragment in another whole genome map.

Checking the option *Save as new default to database* will save the specified pattern matching settings as default settings in the database. If not checked, the last used settings will only apply during the current BioNumerics session; if the software is closed, these settings will not be saved. The settings as defined in the *Match whole genome maps (pattern)* dialog box are stored along with the whole genome map experiment

type. A dialog box with the same settings can be called from the *Whole genome map type* window (see 12.2.3).

An example of a pairwise alignment based on a pattern match of fragments is shown in Figure 12.3.2. The match indication will also show inversions, if this option was checked in the *Match whole genome maps (pattern)* dialog box and inversions were detected among the entries. Map fragments not matching with fragments from a whole genome map of an entry on top or below, according to the specified pattern match parameters, are indicated by a red line. The order of the fragments displayed in the *Experiments* panel can be adjusted by using *Edit > Move entry up* (↑, Shift+Up) and *Edit > Move entry down* (↓, Shift+Down) when an entry is highlighted in the *Information fields* panel. Alternatively, selected entries can be brought to the top by selecting *Edit > Arrange entries > Bring selected entries to top* (⬆, Ctrl+T). (see also 13.2). The alignment tools are discussed in 12.5.2.3.

The reliability of the pattern match settings can be evaluated, and is further discussed in paragraph 12.5.2.5. It is possible to compare both the fragment and pattern match results for two entries in the *Pairwise comparison* window. This can be achieved by selecting the entries in the *Main* window prior to selecting *Analysis > Compare two entries* (Ctrl+2). For more details on this pairwise comparison, see 13.3.3.

12.5.2.5 Pattern match statistics

The pattern match criteria take into account a neighborhood of fragments to match within a given size tolerance. The significance of this pattern match is related to the likeliness of such a match occurring 'by accident'. Therefore, BioNumerics offers the so-called *Pattern match statistics* option to evaluate reliability of the chosen settings on a set of whole genome maps in a *Comparison* window. This is simulated by creating maps by randomly drawing fragments from the set. These simulated maps should be unrelated. The % of matches in this simulated set is the *p*-value of a match occurring by accident, and should hence be as low as possible to consider the given pattern match criteria as trustworthy.

To start the pattern match statistics analysis, select *GenomeMaps > Pattern match statistics...* to open the *Pattern match statistics* dialog box (see Figure 12.5.5).

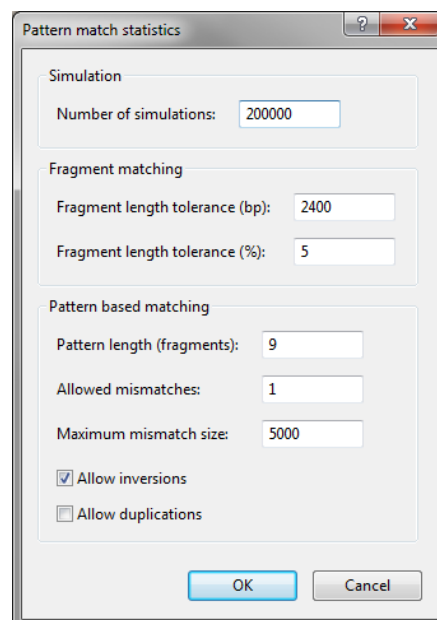


Figure 12.5.5: Evaluating the reliability of the pattern match criteria through the *Pattern match statistics* dialog box.

In the *Pattern match statistics* dialog box, one can specify the *Number of simulations* to run (see 12.5.2.5) in the respective box. The pattern match criteria to be evaluated can be entered under *Fragment matching* for

the size tolerance settings and under **Pattern based matching** for the pattern match settings. These matching criteria are detailed in the *Match whole genome maps (pattern)* dialog box and under 12.5.2.4.

Pressing <OK>, will start the calculation and pop-up the result window, indicating the **Random match fraction** (%) of the set.

This *p*-value should be as low as possible, and allows understanding the trustworthiness of the used pattern match parameters.

12.5.3 Multiple alignment

12.5.3.1 Introduction

The calculation of a multiple alignment of whole genome maps will involve the calculation of a pairwise similarity matrix, based on the specified size tolerance criteria for matching fragments, prior to calculating the multiple alignment. BioNumerics will automatically create this pairwise similarity matrix according to the specified settings in the course of the process. In this way, the whole genome maps will be ordered according to their similarity, supporting the visualization of the multiple alignment.

Please note that for aligning whole genome maps both the Whole Genome Map data module (GM) and the Tree and Network Inference module (TN) need to be present in your BioNumerics configuration (see 2.1.4).

12.5.3.2 Calculating a multiple alignment

After making a selection of database entries in the *Main* window, a comparison can be created and displayed in a new *Comparison* window by highlighting the *Comparisons* panel and selecting **Edit > Create new object...** (+). In the *Experiments* panel, select the whole genome map experiment type and display the maps by pressing the button next to the experiment name or by selecting **Layout > Show image** (image icon).

To start up a multiple alignment of whole genome maps, select **GenomeMaps > Align...** (image icon) to show the *Align whole genome maps* dialog box (see Figure 12.5.6).

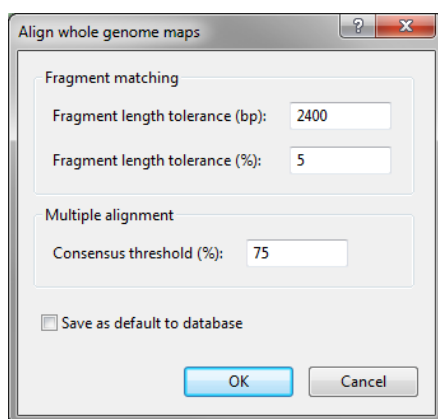


Figure 12.5.6: Defining the multiple alignment settings in the *Align whole genome maps* dialog box.

In the *Align whole genome maps* dialog box the following parameters regarding the *Fragment matching* and the *Multiple alignment* can be specified for calculating the multiple alignment:

- **Fragment length tolerance (bp):** An absolute tolerance (indicated in bp) will be considered for matching fragments: if the size difference of the two fragments is lower than the specified tolerance size, the whole genome map fragments will be regarded as matching (see also 12.5.2.2).

- **Fragment length tolerance (%):** A relative tolerance in relation to the average length of two fragments (indicated in % of the fragment size) is considered for matching: if the size difference of the two fragments is lower than this relative tolerance, the fragments will be regarded as matching (see also 12.5.2.2).
- **Consensus threshold (%):** This setting relates to the visual aspect of the multiple alignment and the consensus whole genome map that will be displayed on top in the *Experiment data* panel. A consensus map is defined from the whole genome maps of all entries in the comparison and the user-defined percentage of whole genome maps having the same fragment at a given position will determine if a fragment appears in the consensus. For instance a value of 75 % determines that matching fragments need to occur in at least 75 % of the entries before they will be placed underneath each other and included in the consensus map.

Checking the option *Save as new default to database* will save the specified fragment matching settings as default settings in the database. If not checked, the last used settings will only apply during the current BioNumerics session; if the software is closed, these settings will not be saved. The settings as defined in the *Match whole genome maps (alignment)* dialog box are stored along with the whole genome map experiment type. A dialog box with the same settings can be called from the *Whole genome map type* window (see 12.2.3).

Pressing <OK> will start the calculations and as a result all whole genome maps will be aligned in the *Experiment data* panel, as shown in Figure 12.5.7.

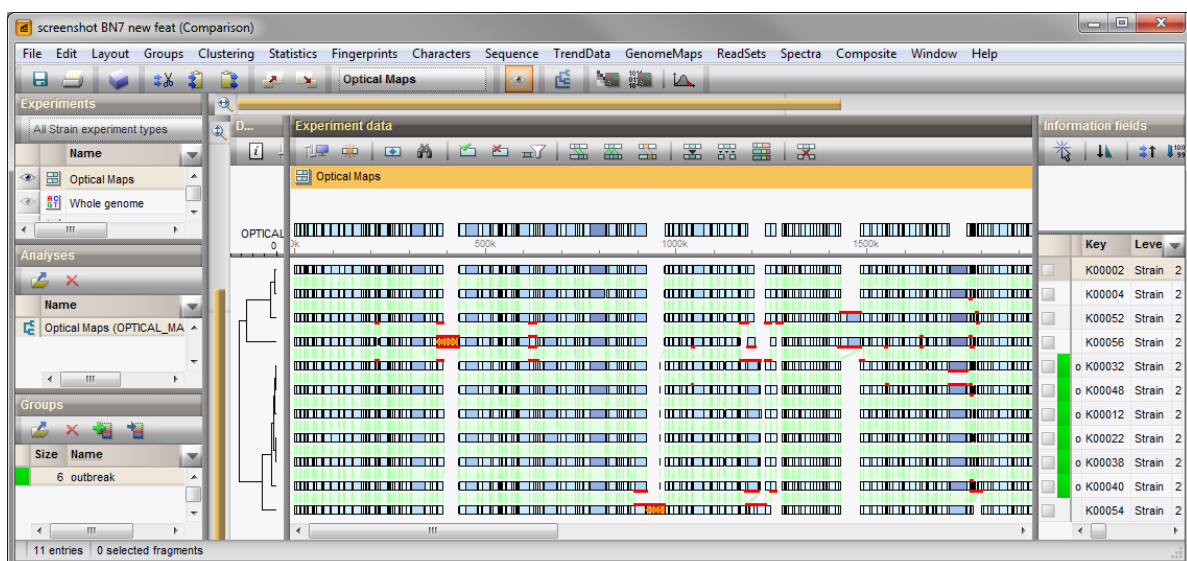


Figure 12.5.7: Size tolerance based multiple alignment of whole genome maps in the *Comparison* window.

12.5.3.3 Alignment tools

The alignment of whole genome maps can be easily removed by selecting **GenomeMaps > Remove alignment** (🗑️). By selecting an individual fragment in a whole genome map, and selecting **GenomeMaps > Fragment to class selection (Ctrl+Q)** all corresponding fragments will be selected, a so-called class of fragments. In combination with the highlighting tool (**GenomeMaps > Highlight selection...** (🔍), Ctrl+H), see 12.3.3.5), this allows a fast overview of the presence of this particular fragment across the whole genome maps in a comparison. If then, **GenomeMaps > Class to fragment selection (Ctrl+Shift+Q)** is selected, all fragments will be individually selected, as indicated at the bottom of the *Comparison* window. As with

every temporary selection of fragments, clicking outside the whole genome maps in the *Experiment data* panel, will remove the selection (see 12.3.3).

The command **GenomeMaps > Fragment to class selection (Ctrl+Q)**, not only serves visualization purposes, but is also a selection that allows further analysis (see 12.5.3.4).

12.5.3.4 Further analysis

Every so-called class of fragments corresponds to a character in a binary character matrix. This character matrix can be easily viewed by creating a composite data set based on the whole genome map experiment type (see 11.1), followed by selecting the composite experiment type in the *Experiments* panel and **Layout > Show image** (🖼️). The class of fragments obtained by **GenomeMaps > Fragment to class selection (Ctrl+Q)** (see 12.5.3.3), will automatically be indicated in the character matrix of the composite data experiment by a yellow arrow. In this way, it can easily be verified if a particular class of fragments is present in a particular strain/isolate. Also, hovering over a whole genome map fragment in the *Experiments* panel, will now not only show the size of the fragment, but also include the index of the fragment class as a second number. Moreover, the use of the character data in a composite data set, allows, for instance, to look which classes of fragments behave similarly for the entries in the comparison, through a transversal clustering. Another option is to arrange the characters in the composite data set according to their discriminatory power for a selection of entries. This and more options are detailed in chapter 11. The same selection obtained by **GenomeMaps > Fragment to class selection (Ctrl+Q)** (see 12.5.3.3), is also concurrently selected for matrix mining (see 20) and charts (see 14).

Chapter 12.6

Finding discriminating fragments

12.6.1 Background

As finding discriminating fragments is important to characterize (a) group(s) of strains/isolates and possibly relate them to biological markers, two ways for finding such fragments are implemented in BioNumerics, starting from a comparison. The first method is designed to find discriminating fragments for a single, user defined (group of) strain(s)/isolate(s), and is further detailed in paragraph 12.6.2. A second approach, called *Pattern Match Classes*, is designed to find fragments that discriminate between two or more groups of strain-s/isolates and involves the creation of classes of fragments, that are present or absent across strains/isolates in a comparison, as is further detailed in paragraph 12.6.3.

12.6.2 Discriminating fragments for a selection of entries

Firstly, a selection of one or multiple entries, for which one wants to search the discriminating fragments for, needs to be made by checking the boxes of these entries in the *Information fields* panel of the *Comparison* window. Secondly, selecting **GenomeMaps > Find discriminating fragments...**, will open the *Find discriminating fragments* dialog box (see Figure 12.6.1).

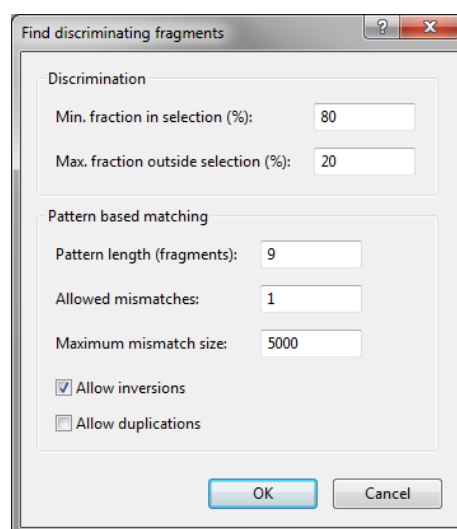


Figure 12.6.1: The *Find discriminating fragments* dialog box specifies the settings for finding discriminating fragments for a selection of entries.

In the *Find discriminating fragments* dialog box, both the criteria for the fragment discrimination and the

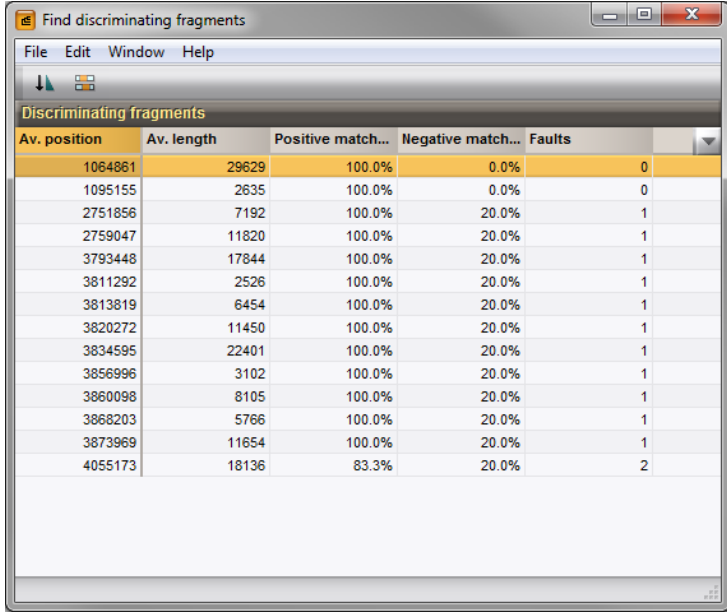
settings for fragments to be considered as matching can be specified:

- **Min. fraction in selection (%)**: This setting specifies the minimal fraction of the selected entries that should contain the considered fragment. For instance, a value of 80 % would mean that at least 8 out of 10 selected entries need to contain the fragment in their whole genome map to be regarded as a discriminating fragment candidate.
- **Max. fraction outside selection (%)**: In order for a fragment to be discriminating, it should not only be common to the entries in the selection, but also be absent in the other entries in the comparison. Therefore, this setting specifies the maximum percentage of the entries outside the selection, that contain the considered fragment.
- **Pattern length**: The pattern length defines the number N of neighboring fragments taken into account when recognizing a pattern of N fragments that is consistent with the active size tolerance criteria.
- **Allowed mismatches**: This option specifies the number of mismatches allowed in the pattern, i.e. number of fragments in the pattern that do not meet the specified tolerance settings.
- **Maximum mismatches size**: This parameter limits the maximum size (bp) of the allowed mismatches in the pattern. For instance, if one mismatch is allowed (one fragment in the pattern not meeting the size tolerance criteria) and its size is lower than the indicated mismatch size, the pattern of fragments will be considered as matching. If two or more mismatches are allowed, the sum of the mismatch sizes should be lower than the specified maximum mismatches size (bp) in order for the pattern of fragments to be considered as matching.
- **Allow inversions**: If checked, pattern recognition will be performed in both a sense and anti-sense direction, allowing to detect inversions between whole genome maps.
- **Allow duplications**: If checked, fragments will be allowed to match more than one corresponding fragment in another whole genome map.

Pressing <OK> will start the calculation and pop-up the results in the *Discriminating fragments* window (see Figure 12.6.2).

The *Discriminating fragments* window lists the discriminating fragment candidates (i.e. fragments that are compliant with the settings specified in the *Find discriminating fragments* dialog box) for the selection of entries in a table format in the *Discriminating fragments* panel. By default, this table contains the following columns:

- **Av. position**: This column lists the average start position (bp) of the discriminating fragments present in the selection of entries.
- **Av. length**: This column lists the average length (bp) of the discriminating fragments present in the selection of entries.
- **Positive matches (%)**: This number indicates the percentage of the selected entries that contain this fragment in their whole genome map. A value of 100 % indicates that all entries where one wanted to find discriminating fragments effectively contain the listed fragment.
- **Negative matches (%)**: This number indicates the percentage of matches outside the entry selection (i.e. unwanted matching with other entries from the comparison) for this particular fragment. In other words, a value of 20 % means that 20 % of the entries outside the selection also contain this particular fragment.
- **Faults**: This columns lists the exact number of violations, i.e. fragments not occurring in all entries of the selection, or also occurring in entries outside the selection.



The screenshot shows a window titled "Find discriminating fragments" with a menu bar (File, Edit, Window, Help) and a toolbar. Below the toolbar is a table titled "Discriminating fragments". The table has five columns: "Av. position", "Av. length", "Positive match...", "Negative match...", and "Faults". The table contains 15 rows of data. The first row is highlighted in yellow.

Av. position	Av. length	Positive match...	Negative match...	Faults
1064861	29629	100.0%	0.0%	0
1095155	2635	100.0%	0.0%	0
2751856	7192	100.0%	20.0%	1
2759047	11820	100.0%	20.0%	1
3793448	17844	100.0%	20.0%	1
3811292	2526	100.0%	20.0%	1
3813819	6454	100.0%	20.0%	1
3820272	11450	100.0%	20.0%	1
3834595	22401	100.0%	20.0%	1
3856996	3102	100.0%	20.0%	1
3860098	8105	100.0%	20.0%	1
3868203	5766	100.0%	20.0%	1
3873969	11654	100.0%	20.0%	1
4055173	18136	83.3%	20.0%	2

Figure 12.6.2: A list of discriminating fragment candidates in the *Discriminating fragments* window.

The discriminating fragment information can be sorted in ascending order for a highlighted column, by clicking on the header of the column (e.g. Av. length), prior to selecting **Edit > Sort according to highlighted column** (📉).

Selecting a discriminating fragment from the list, either by double-clicking it or by selecting **Edit > Select fragments** (📋), will concurrently select this fragment in the *Experiment data* panel, allowing a fast and at-a-glance localization of the discriminating fragment for the entry selection. This is demonstrated, in combination with the highlighting tool (*GenomeMaps > Highlight selection...* (📌, **Ctrl+H**), see 12.3.3.5), in Figure 12.6.3. Multiple discriminating fragment from the list can be selected at the same time by holding the Shift or CTRL key while selecting from the table, followed by **Edit > Select fragments** (📋). Highlighting will then allow to easily view those fragments on the whole genome maps in the *Comparison* window.



To quickly locate the negative matches for a selected discriminating fragment, select **GenomeMaps > Select matching fragments** (📋, **Ctrl+M**) (see also 12.3.3.6). These fragments and the corresponding entries will then be marked in the *Comparison* window. Using **GenomeMaps > Highlight selection...** (📌, **Ctrl+H**) will further facilitate the viewing.

In the *Discriminating fragments* window, additional features, such as table layout, search function, exporting, etc... are available by pressing the column properties button (⚙️) in the upper right corner of the *Discriminating fragments* panel. For example, columns in the table can be shown or hidden by selecting **Set active fields** and (un)checking the respective boxes. The position of the columns in the table can be adjusted by selecting the column (e.g. Av. length) and selecting **Move 'Av. length' to left** or **Move 'Av. length' to right** via the column properties button.

A fragment can be searched for in the in the list by selecting **Find in table...** via the column properties button.

Also, the complete discriminating fragment list can be exported by copying the content to the clipboard and pasting it into a program of choice via **Copy content to clipboard**. Likewise, there is also the option to save it to a file via **Save content to file** (see 3.2.12).

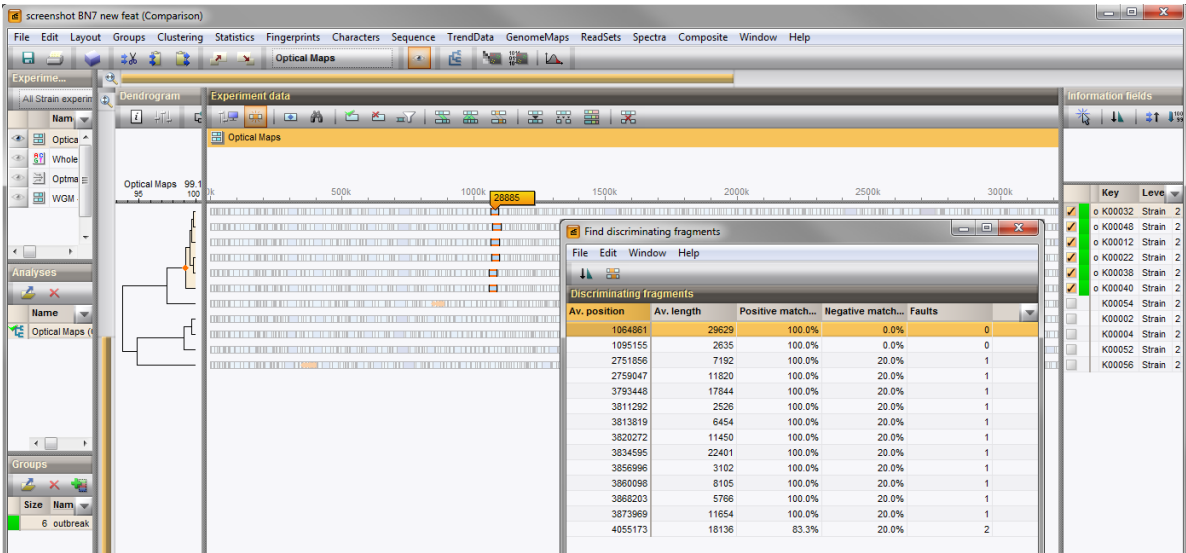



Figure 12.6.3: Concurrent selection of discriminating fragments in the *Discriminating fragments* window and the *Comparison* window.

12.6.3 Pattern Match Classes

To start up the creation of pattern match classes, select **GenomeMaps > Create pattern match classes...** () to open the *Create pattern match classes* dialog box.

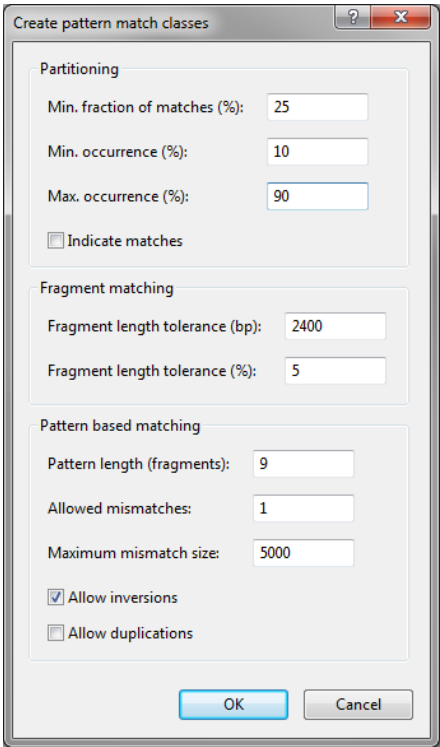


Figure 12.6.4: The *Create pattern match classes* dialog box

In the *Create pattern match classes* dialog box both the partitioning criteria and the criteria for considering fragments as matching need to be specified for the creation of pattern match classes (see also 12.6.3).

For the **Partitioning**, the following parameters can be specified:

- **Min. fraction of matches (%)**: .
- **Min. occurrence (%)**: .
- **Max. occurrence (%)**: .

If the **Indicate matches** box is checked, the matching information will also be drawn in the *Experiment data* panel, using the active color settings for the match indication specified in the *Whole genome maps display settings* dialog box (see also Figure [12.3.2](#)).

The other parameters in this dialog box, relating to the fragment matching criteria, are identical to the ones in the *Match whole genome maps (pattern)* dialog box (see also [12.5.2.4](#)).

Pressing <**OK**> will start the calculations. As a result, the number of pattern match classes found will pop-up in a separate window and the pattern match classes will be displayed in the *Experiment data* panel.

Part 13

Basic cluster analysis

Chapter 13.1

Cluster analysis: an introduction

13.1.1 Similarity-based cluster analysis

Cluster analysis is one of the most popular ways of revealing and visualizing hierarchical structure in complex data sets. Cluster analysis is a collective noun for a variety of algorithms that have the common feature of visualizing the hierarchical relatedness between samples by grouping them in a dendrogram or tree. The most universally applied methods are pairwise clustering algorithms that use a distance or similarity matrix as input (Figure 13.1.1). We refer to this as *similarity-based cluster analysis* or cluster analysis *sensu stricto*. UPGMA (Unweighted Pair Group Method using Arithmetic Averages), Complete Linkage, Single Linkage, and Ward's method are examples of such methods. The advantage of these methods is that they can be applied to any type of data, as long as there exists a suitable similarity or distance coefficient that can generate a similarity (distance) matrix from the data. As such, similarity-based clustering can be applied to incomplete data sets or data that is not presented in the form of a data matrix (e.g. electrophoresis band sizes).

In the analysis steps outlined in Figure 13.1.1, one should consider the matrix of pairwise similarities (or distances) as the complete comparative information between all the samples analyzed. Obviously, for larger numbers of samples, interpreting a similarity matrix becomes hardly simpler than looking at the original data. This is why a similarity matrix is not usually calculated as a final result, but as an intermediate step for grouping algorithms such as cluster analysis or multi-dimensional scaling.

The real simplification of the data is obtained by cluster analysis. Both the power and the weakness of a dendrogram lie in its ability to present an easy to interpret, well-structured, hierarchical grouping of the samples. Indeed, simplification means loss of information, and there is no way to present the data in a simple and easily interpretable way, yet holding all the information. As a consequence, every dendrogram resulting from a non-artificial data set will contain errors, the amount of error being proportional to the complexity of the similarity matrix. A second source of error results from the fact that hierarchical clustering always imposes hierarchical structure, even if the data does not support it. The fact that even a perfectly random data set results in a dendrogram with branches, is a clear example of the danger that hierarchical clustering holds. Various statistical methods allow the error associated with dendrogram branches or their uncertainty to be estimated, e.g. standard deviation values and the cophenetic correlation (see 13.3.7). Other methods, such as bootstrap, allow the probability of dendrogram branches, as a result of the data set, to be estimated.

Since for each of the experiment types available in BioNumerics, different algorithms are available to obtain a similarity matrix from the actual data matrix, the steps involved in performing a similarity-based cluster analysis (or cluster analysis *sensu stricto*) are discussed in the corresponding sections (4.2 to 11.2).

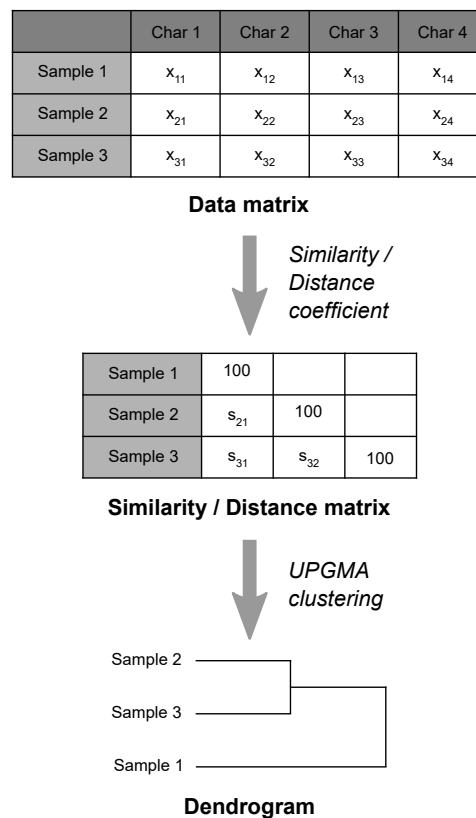


Figure 13.1.1: Steps in similarity-based cluster analysis.

13.1.2 Degeneracy of dendrograms

Another problem with pairwise hierarchical clustering methods such as UPGMA is the degeneracy of the solution. Whereas UPGMA results in just one tree, in many cases there exist a number of equally good alternative solutions. Such degeneracies are very likely to occur in cases where the similarity matrix contains multiple identical values. In practice, binary and categorical data sets and banding patterns treated as absent/present states result in frequent occurrence of identical similarity values, whereas quantitative measurements registered as decimal numbers almost never yield identical similarity values. To understand how the occurrence of identical similarity values can result in multiple possible trees, we consider the example of three banding patterns (Figure 13.1.2). As can be seen from this simple example, $s[A,B]$ and $s[B,C]$ are both 0.75, whereas $s[A,C]$ is 0.50. The way how UPGMA constructs a dendrogram is by first searching for the highest similarity value in the matrix, and linking the two samples from which it results. In the present example, $[A,B]$ and $[B,C]$ are equivalent solutions, two partial dendrograms can be constructed: one with $[A,B]$ linked at 75% (solution 1) and the other with $[B,C]$ linked at 75% (solution 2). In the next step of UPGMA, the remaining sample is linked at the average of its similarity with the samples already grouped. In solution 1, this leads to C being linked at 62.5% to $[A,B]$, whereas in solution 2, A is being linked at 62.5% to $[B,C]$. Both dendrograms suggest a quite different hierarchical relatedness but actually none of them truly reflects the relationships suggested by the data set and the similarity matrix (Figure 13.1.2 a).

Another inconsistency in pairwise clustering results from the inability to deal with infringements upon the transitivity rule of identity. When sample A is identical to sample B, and sample B is identical to sample C, the transitivity rule predicts that A will be identical to C as well. Infringements upon this rule are particularly found in the comparison of banding patterns, where the identity of bands is judged based upon their distance, using a position tolerance value that specifies a maximum distance between bands to be considered identical. The example in Figure 13.1.2 b illustrates the result of a UPGMA clustering of three banding patterns for which one band is slightly shifted. With a position tolerance as indicated on the figure, the pairs of patterns

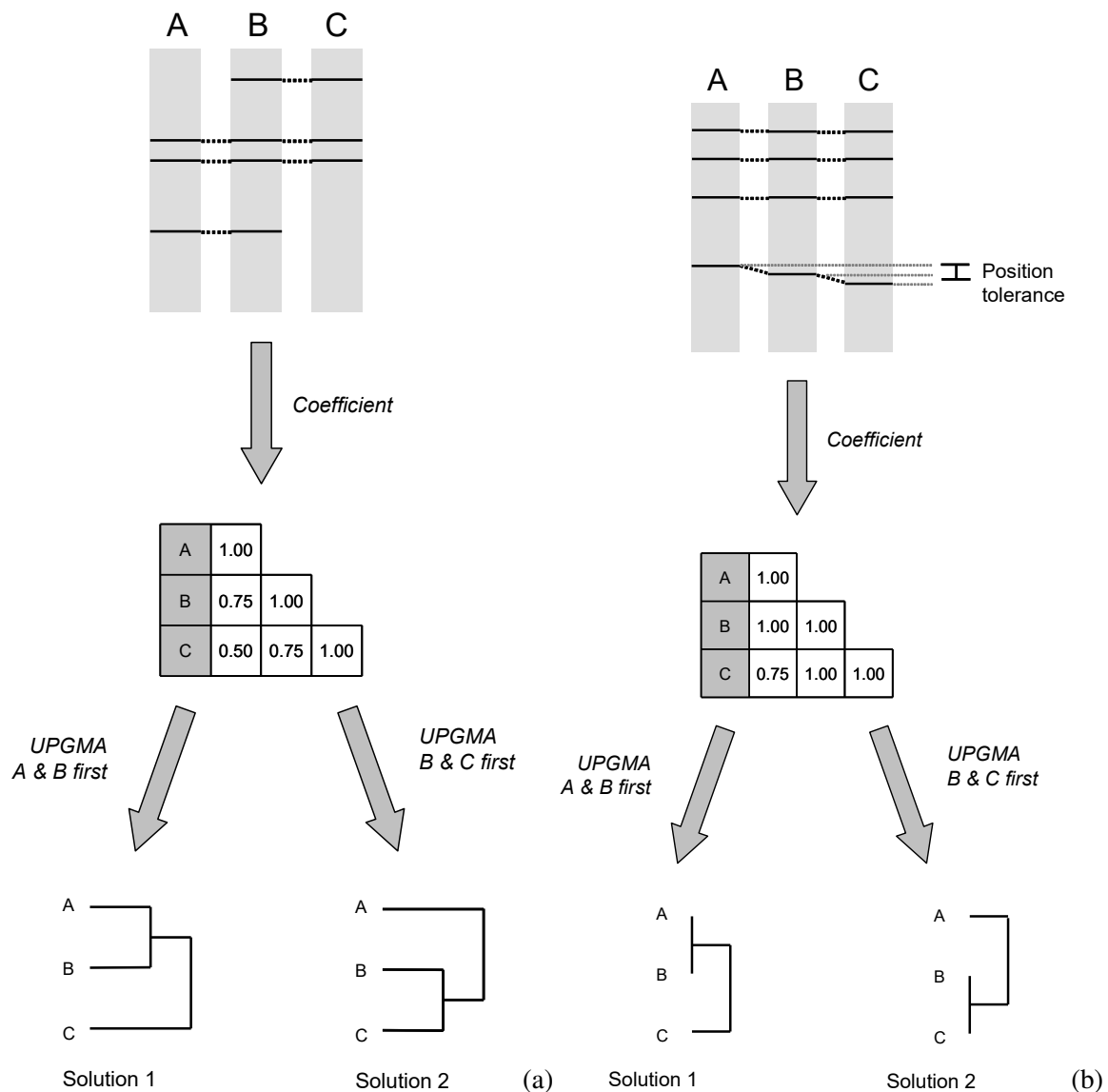


Figure 13.1.2: A scenario of three banding patterns resulting in two possible UPGMA solutions (a) and infringement upon the transitivity rule for sample identity and resulting dendrograms (b).

[A,B] and [B,C] will have a 100% score, whereas [A,C] will have only 75% similarity, as the distance between their lower bands is greater than the position tolerance specified. Similarly as explained above, the UPGMA algorithm has two choices to perform the first linkage, and the results are displayed as solution 1 and solution 2. Neither of the two dendrograms reflects the discrepancy indicated by the similarity values, but instead, each dendrogram falsely suggests a hierarchical structure that is not supported by the data.

The problem of degeneracy is discussed in the present chapter (see 13.3.6) and is even more extensively covered in 16.

13.1.3 Cluster analysis *sensu lato*

In addition to cluster analysis *sensu stricto*, which is based upon a matrix of similarities between database entries and a subsequent algorithm for calculating bifurcating hierarchical dendrograms representing the clusters of entries, a variety of other techniques exist that have the common feature of producing a hierar-

chical tree-like structure (*dendrogram*) from the set of sample data provided. These algorithms are referred to as cluster analysis *sensu lato* in this manual and comprise e.g.:

- Phylogenetic clustering methods: Methods which attempt to create trees that optimize a specific phylogenetic criterion. These methods start from the data set directly rather than from a similarity matrix.
- Minimum spanning trees: These trees are calculated from a distance matrix, and possess the property of having a total branch length that is as small as possible.

These and other tree and network inferring methods are dealt with in [16](#).

Chapter 13.2


Comparisons in BioNumerics

13.2.1 The Comparison window

A *Comparison* in BioNumerics includes every function which allows to compare database entries. This involves the display of experiment images of selected entries, the calculation and display of cluster analyses, alignment of sequences, and calculation of principal component analysis (PCA) and multi-dimensional scaling (MDS) projects, etc..

The *Comparison* window in BioNumerics presents a comprehensive overview of all available experiments for a selection of entries and enables the user to show and compare any combination of images of experiments. A comparison is always created from a selection of database entries (see [3.3.8](#)). The maximum number of entries in a comparison is 20,000.

A number of settings, such as calculation priority, default location and maximum size of a saved similarity matrix, apply specifically to comparisons and the *Comparison* window. These preferences can be specified in the *Preferences* window (see [2.3.3](#)).

A comparison is created from a selection of database entries and displayed in a new *Comparison* window by highlighting the *Comparisons* panel in the *Main* window and selecting **Edit > Create new object...** () (see [Figure 13.2.1](#) for an example).

The *Comparison* window is divided in seven main panels:

- *Dendrogram* panel: shows the dendrogram (if calculated)
- *Experiment data* panel: showing the images of the experiments
- *Information fields* panel: shows the database fields in the same layout as in the database (see [3.3.7](#))
- *Similarities* panel: shows the similarity values
- *Experiments* panel: shows the available experiment types
- *Groups* panel: shows the comparison groups (if defined)
- *Analyses* panel, which shows all analyses done in the comparison

Initially, the *Dendrogram* panel, the *Experiment data* panel, the *Similarities* panel, the *Groups* panel and the *Analyses* panel are empty. You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally.

All panels in the *Comparison* window are dockable and their position can therefore be changed according to your own preferences. For more information on the display of dockable panels, see [2.3.4](#).

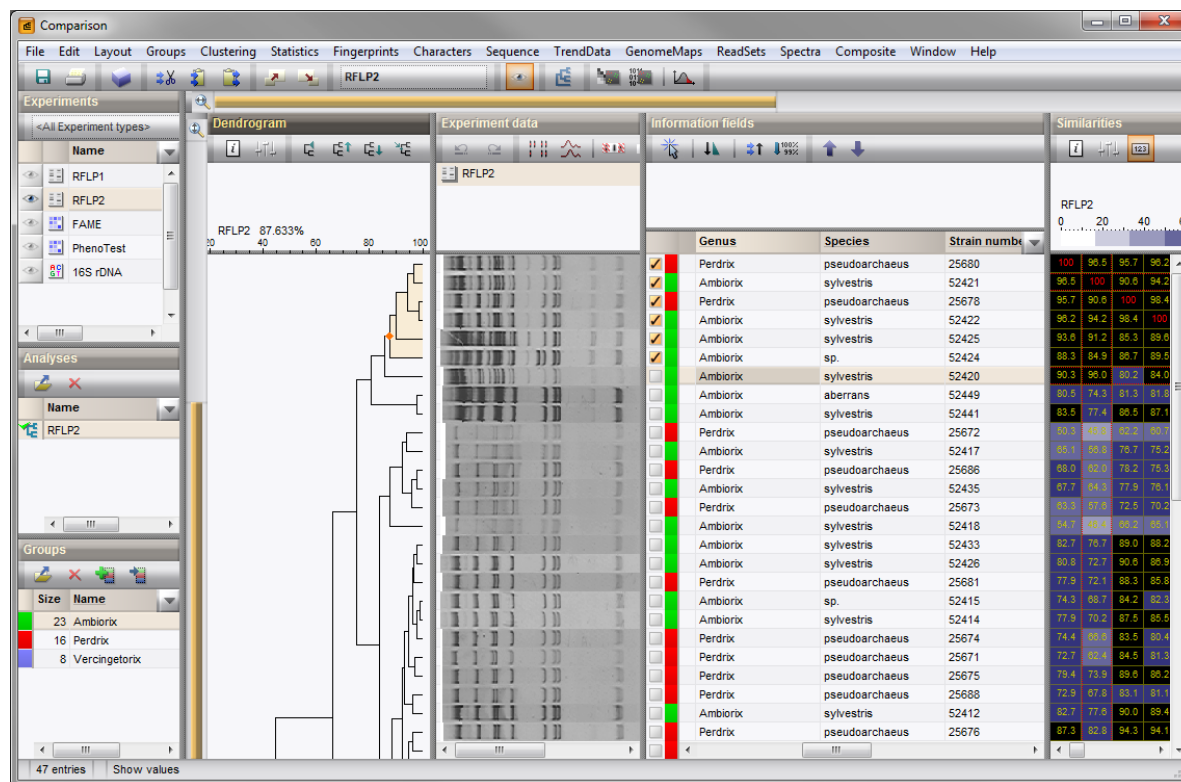








Figure 13.2.1: The *Comparison* window in **DemoBase Connected**, displaying a dendrogram and similarity matrix based on **RFLP2** and groups defined based on the 'Genus' information field.



The *Dendrogram* panel, *Experiment data* panel, *Information fields* panel and *Similarities* panel behave as a group, i.e. these panels cannot be docked outside this group and they cannot be displayed in a window of their own (undocked).



The *Information fields* panel in the *Comparison* window is similar to the *Database entries* panel in the *Main* window and contains the database information in tabular format (grid panel). In fact, the display options as set for the *Database entries* panel will be taken over in the *Information fields* panel. You can drag the separator lines between the information field columns to the left or to the right, in order to divide the space among the information fields optimally. Clicking the column properties button (⌵), which is located on the right hand side in the information fields header in the *Information fields* panel, gives access to functions allowing information fields to be displayed or hidden, frozen, or moved to the left or to the right. For detailed information on the display options of object grid panels, see 3.2.7.

From the *Experiments* panel, you can select one of the available experiment types, to show an image, calculate a dendrogram, or show a similarity matrix. Each experiment type in the *Experiments* panel contains three objects: a button and the experiment type name, and an eye icon on the left hand side of the button.

In case of a fingerprint type, the button is shown as ; character experiments as ; sequence types as ; matrix types as ; trend data types as  and composite data sets as .







The experiments in the *Experiments* panel of the *Comparison* window are listed in the same order as they are listed in the *Experiments* panel of the *Main* window. This feature also allows one to control the order in which experiment data are displayed in a composite data set (see 11.2.3).

To display the data for an experiment type in the *Experiment data* panel, click on the eye button  next to the experiment name in the *Experiments* panel. When the image of an experiment type is displayed, the button is shown as . Data from a second experiment type will be shown on the right from data that are


already displayed.



To display more than one image at a time, we recommend to maximize the *Comparison* window, and to use maximal space for the *Experiment data* panel by minimizing the *Dendrogram* panel and *Similarities* panel.


If insufficient space is available to show several experiment images at the same time, you can scroll through the *Experiment data* panel, or use the zoom functions **Layout > Zoom in** (, **Ctrl+Page Up**) and **Layout > Zoom out** (, **Ctrl+Page Down**). The zoom sliders indicated with  and  can be used to zoom selectively in the horizontal or vertical direction, respectively. See 2.3.7 for a detailed description of zoom slider functions.

In the caption of the *Experiment data* panel, you can drag the separator line between the images to the left or to the right, in order to reserve more or less horizontal space for a particular experiment image. The original aspect ratio (proportion height to width) of the image will not be maintained by this action.

To select an experiment type in the *Comparison* window, you can either click on the experiment type name in the *Experiments* panel, on the image itself or select the experiment type from the drop-down menu  in the toolbar.

When an experiment type is selected, both the image caption in the *Experiment data* panel (if the image is shown) and its name in the *Experiments* panel are highlighted. All functions listed under **Clustering**, **Statistics**, **Fingerprints**, **Characters**, **Sequence**, **Trend Data**, and **Composite** as well as some **Layout** functions, apply to the selected experiment type.

13.2.2 Adding and removing entries


Selections of entries made in the *Database entries* panel of the *Main* window are also shown in the *Information fields* panel of the *Comparison* window and vice versa. The entries in a newly created comparison are automatically selected () since they were all selected in the database.


You can manually select and unselect entries in the *Information fields* panel (see Figure 13.2.1), using the **Ctrl** and **Shift**-keys as described in 3.3.8.

To invert the selection, i.e. to select all entries except the currently selected ones, use **Edit > Invert selection**.


Selections can be added or removed from an existing comparison:



With **Edit > Delete selection**, the selected entries are removed from the comparison. The program will ask for confirmation to remove the selection from the comparison. You will not be able to undo this operation.

With **Edit > Cut selection** (, **Ctrl+X**), the selected entries are removed from the comparison and are copied to the clipboard.

With **Edit > Paste selection** (, **Ctrl+V**), selected entries are added to the comparison. If no dendrogram is present, they are placed at the position of the selection bar. This tool can be used to rearrange entries in the *Comparison* window without calculating a dendrogram (see also 13.2.3).

Entries can be added to an existing comparison at any time. The entries first need to be copied to the clipboard from the *Main* window or from another comparison.

To copy entries to the clipboard, select the entries (e.g. in the *Main* window) first and use **Edit > Copy selection** (, **Ctrl+C**).

To cut entries from one comparison into another, use **Edit > Cut selection** (, **Ctrl+X**) in one comparison and **Edit > Paste selection** (, **Ctrl+V**) in the other comparison.

New database entries can be added to an existing *dendrogram* (see 13.2.6 on how to calculate a dendrogram) in this way: select the new entries in the database, open an existing comparison with dendrogram, and paste

the selection into the comparison. Both the similarity matrix and the dendrogram will be updated, which uses considerably less time than recalculating the whole cluster analysis.




Entries can also be selected from the *Dendrogram* panel: hold the **Ctrl**-key and left-click (**Ctrl+click**) on a branch node to select/unselect a cluster on the dendrogram at once (see also 4.2).

For adding and removing entries from a global sequence alignment, see 8.3.11.

13.2.3 Rearranging entries in a comparison

The cut and paste functions can be used to rearrange entries in the *Comparison* window (see 13.2.2). Some other convenient functions are available for rearranging entries in a comparison, as explained below.


Select **Edit > Arrange entries > Arrange entries by field** () to sort the entries alphabetically according to the highlighted database field.


Select **Edit > Arrange entries > Arrange entries by field (inverted)** to sort the entries in reverse alphabetical order (z-a) according to the highlighted database field.





When two or more entries have identical strings in a field used to rearrange the order, the existing order of the entries is preserved. As such it is possible to categorize entries according to fields that contain information of different hierarchical rank, for example genus and species. In this case, first arrange the entries based upon the field with the lowest hierarchical rank, i.e. species, and then upon the higher rank, i.e. genus.

When a field contains numerical values, which you want to sort according to increasing number, use **Edit > Arrange entries > Arrange entries by field (numerical)**.

Use **Edit > Arrange entries > Arrange entries by field (inverted+numerical)** to sort according to decreasing number.


In case numbers are combined numerically and alphabetically, for example entry numbers [213, 126c, 126a, 126c], you can first arrange the entries alphabetically using **Edit > Arrange entries > Arrange entries by field** () (or **Edit > Arrange entries > Arrange entries by field (inverted)**), and then numerically using **Edit > Arrange entries > Arrange entries by field (numerical)** (or **Edit > Arrange entries > Arrange entries by field (inverted+numerical)**). The result will be [126a, 126b, 126c, 213].

Selected entries can be placed at the position of the cursor (the entry you last clicked on) with **Edit > Arrange entries > Bring selected entries to cursor** or moved to the top of the comparison with **Edit > Arrange entries > Bring selected entries to top** (, **Ctrl+T**).

An individual entry can be moved up or down by clicking on it and then selecting **Edit > Move entry up** (, **Shift+Up**) or **Edit > Move entry down** (, **Shift+Down**), respectively. When using the  and  buttons, you can move an entry to the top or the bottom of a comparison at once by holding the **Ctrl**-key.

13.2.4 Saving and loading comparisons

A comparison can be saved and all calculations done on the data it contains, will be stored along. This includes the last calculated similarity matrix, any dendrogram that has been calculated (see 13.2.6), and band matching analyses (see 4.3). Comparisons can be shared with other users of the same database.

Use **File > Save** (, **Ctrl+S**) to save a comparison. When the comparison does not exist yet, the *Save comparison as* dialog box pops up (see Figure 13.2.2).

A name for the comparison needs to be specified in the text box on top.

The comparison can be saved as a **File** or inside the relational **Database**. In the latter case, the comparison

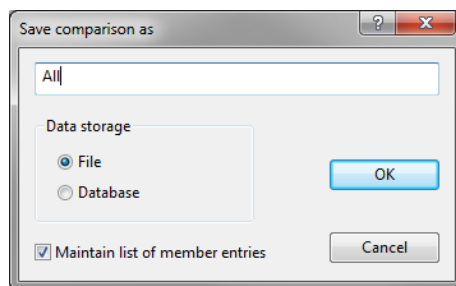


Figure 13.2.2: The *Save comparison as* dialog box.

will be visible by other users who are connected to the same database.

If *Maintain list of member entries* is checked, the software will keep a list of entries that are included in the comparison in a separate table of the relational database (see 21.1). This makes it possible e.g. to indicate in the *Comparisons* panel how many entries the comparison contains and to show all the comparisons a given entry is member of in the *Comparisons* panel of the *Entry* window. This feature has a small performance penalty, which is why it can be switched off.

With **File > Save as...**, an existing comparison can be saved under a different name.

Saved comparisons are listed in the *Comparisons* panel of the *Main* window.

To open an existing comparison, select it from the list in the *Comparisons* panel of the *Main* window and use **Edit > Open highlighted object...** (🖱️, **Enter**). Alternatively, just double-click on the comparison name.

13.2.5 Experiment type aspects

Aspects of experiment types allow to perform e.g. cluster analyses or statistical tests on all or predefined subsets of experimental data, linked to the entries in the comparison.

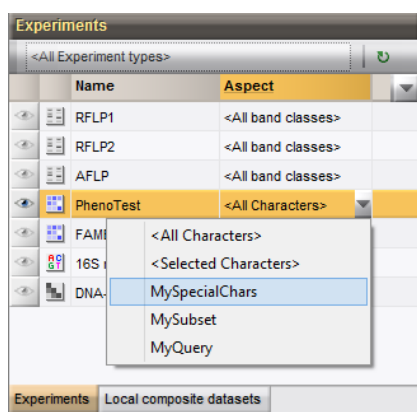


Figure 13.2.3: The *Experiments* panel in **DemoBase Connected**, illustrating *aspects* for a character type experiment.

To perform an analysis on a certain aspect, simply select that aspect from the *Aspect* column in the *Experiments* panel in the *Comparison* window. The display of the data *Experiment data* panel will be updated automatically, but any analysis should be calculated again to reflect the data contained in the selected aspect.

For **Character type experiments**, following aspects are available:

- **<All characters>**: all characters in the character type. This aspect is used by default when a compar-

ison is first created.

- **<Selected characters>**: all currently selected characters (indicated with colored triangles) in the character type.
- Any character view that is defined for the character type (see 6.1.2.11).

For **Fingerprint type experiments**, following aspects are available:

- **<All band classes>**: all band classes that are stored in the fingerprint type (see 4.1.5.10). This aspect is used by default when a comparison is first created.
- Any band class view that is defined for the fingerprint type (see 4.1.5.10).

For **Spectrum type experiments**, following aspects are available:

- **<All peak classes>**: all peak classes that are stored in the spectrum type (see 5.5). This aspect is used by default when a comparison is first created.
- Any peak class view that is defined for the spectrum type (see 5.5.5).

Other experiment types do not offer character aspects.



Selecting an aspect of a fingerprint type or spectrum type only has an effect on the band matching table or peak matching table, respectively. This corresponds to how the data would be used in a composite data set (see 4.3). Changing the aspect has *no effect* when a cluster analysis is calculated directly on the fingerprint type or spectrum type.




The last-used aspect of an experiment type is saved along with the comparison.

13.2.6 Calculating a dendrogram

In cluster analysis *sensu stricto*, calculating a dendrogram is a two-step process. First, entries are compared two by two and a *pairwise similarity matrix* is constructed. Next, a *dendrogram* is constructed based on the similarity matrix. Please note that the Tree and network inference module (TN) needs to be present in your BioNumerics configuration in order to calculate a cluster analysis.

In the *Comparison* window, select an experiment type in the *Experiments* panel for which you want to calculate a dendrogram.

Select **Clustering > Calculate > Cluster analysis (similarity matrix)**.... Alternatively, press the  button, in which case a menu pops up as depicted in Figure 13.2.4.

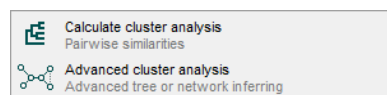


Figure 13.2.4: Cluster analysis menu popped up from the dendrogram button.

The first choice (**Calculate cluster analysis**) is the matrix-based cluster analysis discussed in this chapter, whereas the second option (**Advanced cluster analysis**) is dealt with in 16.

When selecting the option **Calculate cluster analysis** from the cluster analysis menu the *Comparison settings* wizard is called (see Figure 13.2.5).

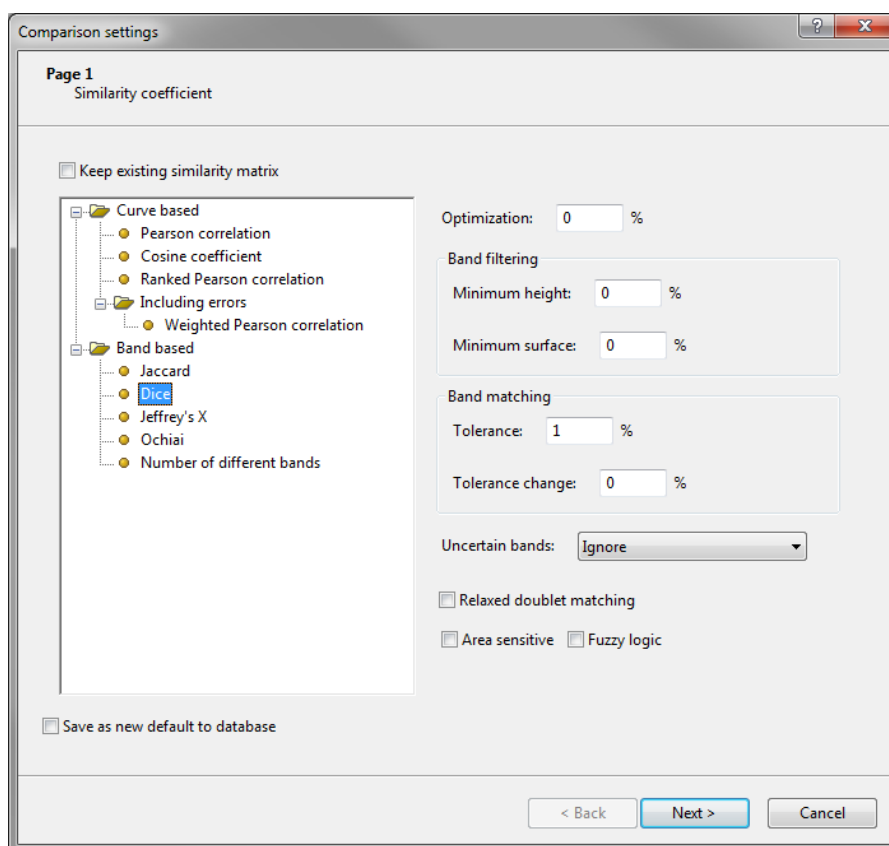


Figure 13.2.5: The *Similarity coefficient* wizard page, which deals with the choice of the similarity coefficient. In this example, the settings for a fingerprint type experiment are shown.

The *Comparison settings* wizard allows you to specify the settings related to the similarity coefficient for calculation of the similarity matrix and the clustering method to be applied. The *Similarity coefficient* wizard page deals with the similarity coefficient. Depending on the experiment type, different settings are listed on this page. More information about these settings can be found in the cluster analysis sections of each experiment type (see 4.2 for fingerprints, 6.2 for characters, 8 for sequences, 7.2 for trend data and 11.2 for composite data sets).

Pressing <Next> calls the *Cluster analysis* wizard page (see Figure 13.2.6).

In the *Cluster analysis* wizard page, the options related to the clustering algorithms are grouped.

Under **Method**, the clustering algorithm to be applied to the similarity matrix can be selected. The program offers choice between the Unweighted Pair Group Method using Arithmetic averages (**UPGMA**), the **Ward** algorithm, the **Neighbor Joining** method, and two variants of UPGMA, namely **Single linkage** and **Complete linkage**. The option **Create graph** is explained in 16.

In **UPGMA**, the similarity between two clusters is calculated as the *average* of the similarity values between the individual elements in the clusters. In **Single linkage**, the similarity between two clusters is calculated as the similarity between the two most similar elements (*highest* similarity value) in the two clusters. In **Complete Linkage**, the similarity between two clusters is calculated as the similarity between the two most dissimilar elements (*lowest* similarity value) in the two clusters.

Under **Degeneracy handling**, one can check **Enable degeneracy handling** to have the program calculate all equivalent trees and apply a **Secondary criterion** to solve the indeterminacy left by the primary criterion (i.e., the standard clustering algorithm). With **Degeneracy**, one can specify how to deal with degenerated trees and with **Cut off above**, the maximum allowed percentage of degenerate entries relative to a cluster can be specified. For more information about degenerate trees, see 13.3.6.

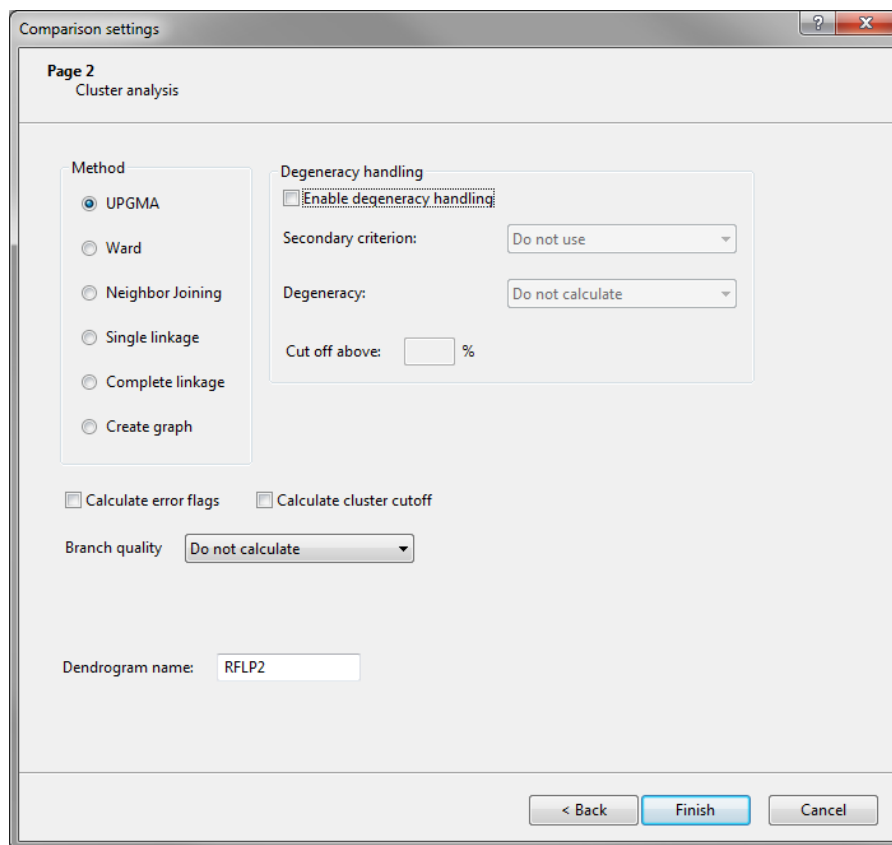


Figure 13.2.6: The *Cluster analysis* wizard page, which deals with the cluster analysis.

If **Calculate error flags** is checked, the program will calculate the standard deviations associated with each cluster. Check **Calculate cluster cutoff** to display a dendrogram with less significant branches in dashed lines. A **Branch quality** parameter can be selected that will be shown on each branch in the dendrogram. For more information regarding these cluster significance tools, see 13.3.7.

A **Dendrogram name** can be entered in the corresponding text box. By default, the name of the experiment type will be used. Only for character types, the name will be appended with the aspect (see 13.2.5) used. The cluster analysis will be listed under this name in the *Analyses* panel of the *Comparison* window. If an analysis with the same name already exists, it will be overwritten.

Pressing **<Finish>** starts the calculations. During the calculations, the program shows a green progress bar in the bottom left corner of the *Comparison* window and indicates the progress as a percentage.

When finished, the dendrogram and the similarity matrix are displayed in their corresponding panels (see Figure 13.2.1). The name entered for the calculated dendrogram (by default the experiment name) is shown on top of the similarity scale in the *Dendrogram* panel. In the *Analyses* panel, the dendrogram name is shown as well, preceded by an "active analysis" icon (🟢). If more than one analysis is listed in the *Analyses* panel, a previous analysis can be called again by clicking on it and selecting **File > Analysis components > Open** (📄). An analysis can be removed with **File > Analysis components > Delete** (🗑️).

The parameters and settings of the cluster analysis can be reviewed with **Clustering > Show information** (📖). This displays a report containing all comparison settings (see 13.3.1).

When the comparison is saved (see 13.2.4), the dendrogram will be saved along.



Although a comparison can contain many dendrograms, there is only one similarity matrix: the displayed matrix will always correspond to the dendrogram that was last calculated.

Chapter 13.3

General comparison functions

13.3.1 Dendrogram display functions

A number of dendrogram display options are available to facilitate the interpretation of a dendrogram.

In the *Comparison* window, with a dendrogram calculated (see 13.2.6), select **Clustering > Dendrogram display settings...** (🔧) to pop up the *Dendrogram display settings* dialog box (Figure 13.3.1)

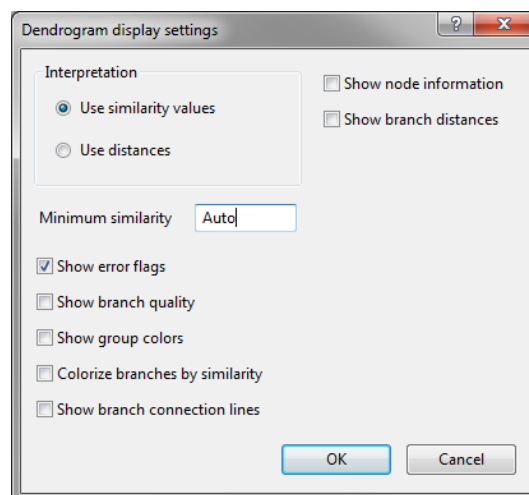


Figure 13.3.1: The *Dendrogram display settings* dialog box.

Under *Interpretation*, the choice is offered between *Use similarity values* (the default option) and *Use distances* to interpret the dendrogram in terms of similarity values or distances, respectively.

A *Minimum similarity* for display on the dendrogram (corresponding with the left edge of the *Dendrogram* panel) can be entered as a number. In case "Auto" is displayed or when the text box is left blank, the software automatically limits the displayed similarity range to the depth of the dendrogram.

The options *Show error flags* and *Show branch quality* can be checked to display the error flags and branch quality, respectively, if these cluster significance values were calculated (see 13.3.7).

With *Show group colors*, dendrogram branches are colored according to the Group colors. See 13.3.2 on how to define Groups.

When *Colorize branches by similarity* is checked, dendrogram branches are shaded according to their average similarity value.

The option *Show branch connection lines* only applies to unrooted trees, e.g. neighbor joining dendrograms.

The connection lines that appear after checking this option link the dendrogram tips with the corresponding entries.

When **Show node information** is checked, the similarity or distance value (depending on the option selected under **Interpretation**) of each dendrogram node is displayed. Checking **Show branch distances** will display the branch length underneath each branch.

Entries can be selected from within the *Dendrogram* panel of the *Comparison* window:

Pressing **F4** unselects any previous selection of database entries.

To select an individual entry, **Ctrl+click** a dendrogram tip (where a branch ends in an individual entry). Repeat this action to unselect the entry.


To select a cluster on the dendrogram at once, **Ctrl+click** a branch node. Repeat this action to unselect a branch.


When a dendrogram node or tip is clicked on, a diamond-shaped cursor appears on that position. In case of a dendrogram node, the branch is highlighted. The average similarity or distance at the cursor's place is shown in the upper left corner of the *Dendrogram* panel. You can also move the cursor with the arrow keys.



In some cases, it may be necessary to select the root of a dendrogram, for example if you want to (un)select all the entries of the dendrogram. In case of large dendrograms, selecting the root may be difficult using the mouse.

With **Clustering > Select root**, the cursor is placed on the root of the dendrogram and the complete dendrogram becomes highlighted.

A branch can be moved up or down to improve the layout of a dendrogram or make its description easier:


Click the branch which you want to move up in the dendrogram and select **Clustering > Move branch up** ()


Click the branch which you want to move down in the dendrogram and select **Clustering > Move branch down** ()

When the **Clustering > Move branch up** () or **Clustering > Move branch down** () command is executed and the branch is already at the top (resp. bottom) level of its parent branch, it will take the parent branch up (resp. down) next time the button is clicked.



To simplify the representation of large and complex dendrograms, it is possible to simplify branches by abridging them as a triangle.

Select a cluster of closely related entries and use **Clustering > Collapse/expand branch** () . Repeat this action to undo the abridge operation.

Another function, **Clustering > Reroot tree** () , only applies to so-called *unrooted trees*, i.e. neighbor joining, maximum parsimony and maximum likelihood trees. These clustering methods produce trees without any specification as to the position of the root or origin. Since users will want to display such trees in the familiar dendrogram representation, the tree is to be rooted artificially. "Re-rooting" is usually done by adding one or more unrelated entries (so-called *outgroup*) to the clustering, and using the outgroup as root. The result is a *pseudo-rooted tree*.

Information on how a dendrogram was calculated can be displayed in a report, which is displayed using **Clustering > Show information** () (see Figure 13.3.2).

This window displays the comparison settings, which were applied to calculate the current dendrogram, in a hierarchical structure.

The report can be exported as a HTML or text file with **File > Save as html** () or **File > Save as text** () , respectively.

The *Report* window is closed with **File > Exit**.

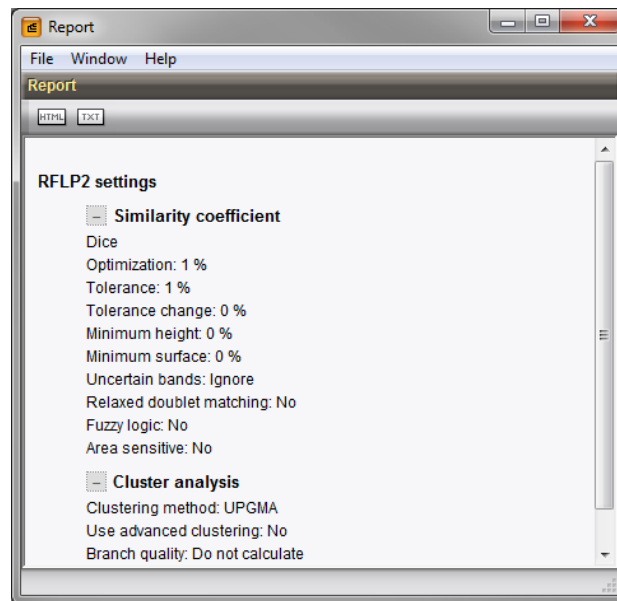


Figure 13.3.2: *Report* window with the comparison settings used to calculate dendrogram RFLP1 in **DemoBase Connected**.

13.3.2 Matrix display functions

The similarity matrix is displayed in the *Similarities* panel, in default configuration located at the right hand side of the *Comparison* window.

If the similarity matrix is not shown for the selected experiment, you can display it with **Layout** > **Show matrix**. This option is only available when a dendrogram was calculated for the selected experiment.

It may be necessary to reduce the space allocated for the image and for the information fields, in order to increase the space for the matrix panel, by dragging the separator lines between the panels.

Initially, the matrix is displayed as differentially shaded blocks representing the similarity values. The interval settings for the shadings is graphically represented in the caption of the *Similarities* panel (Figure 13.3.3).



Figure 13.3.3: Adjustable similarity shading scale.

To show the similarity values in the matrix, select **Clustering** > **Similarity matrix** > **Show values** (123). The matrix display settings can be accessed by selecting **Clustering** > **Similarity matrix** > **Display settings...** (124). This pops up the *Similarity matrix display settings* dialog box (Figure 13.3.4).

Under **Shades limits**, the maximum/minimum values for each interval can be entered as numbers.

With the option **Show rulers** enabled (the default setting), a set of horizontal and vertical rulers appear on the similarity cell where clicked. These rulers connect the two entries from which the similarity value is derived.

As an alternative to entering the **Shades limits** as numbers, you can drag the interval bars on the similarity shading scale (Figure 13.3.3); the matrix is updated instantly.

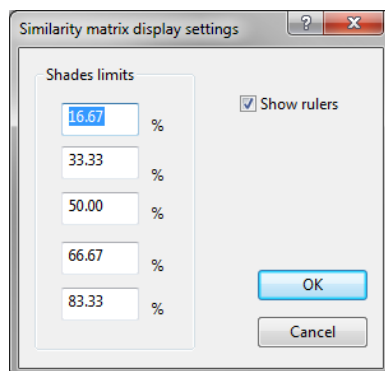


Figure 13.3.4: The *Similarity matrix display settings* dialog box.



If you find it difficult to read the similarity values against the shaded background, you can remove the shades with **Clustering > Similarity matrix > Display settings...** (⚙️) and entering 100% for each interval. Alternatively, a different color scale and text color could be specified via the preferences (see 2.3.3.6).

If you want to find the similarity value on the matrix between two entries in the comparison, click first on the point on the diagonal of the matrix corresponding to the first entry, and then on the second entry inside the *Information fields* panel (Figure 13.3.5). The similarity value is the intersection between the horizontal and the vertical rulers.

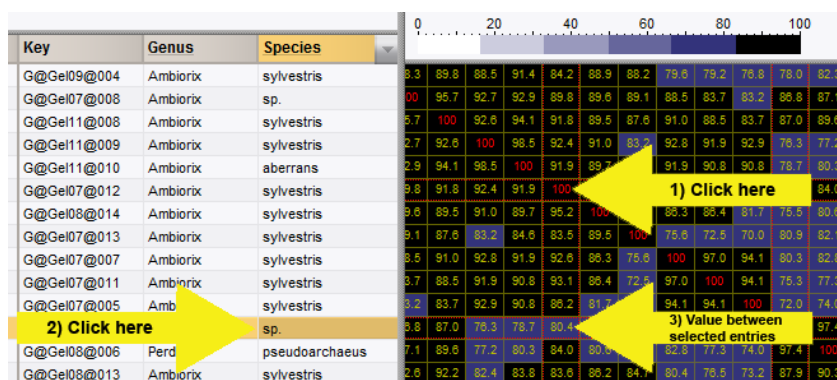


Figure 13.3.5: Work flow for finding a similarity value between two entries.

By double-clicking on a similarity block or value, you can pop up the detailed comparison between the two entries (13.2.1).

The similarity matrix can be exported as a text file using **File > Export > Export similarity matrix....** The export file, popped up as `export.txt` in Notepad, is a tab-delimited text file which contains the similarity values with entry keys as descriptors. You can export a text file which contains the same descriptors with the corresponding information fields using **File > Export > Export database fields....**

Information on how a similarity matrix was calculated can be displayed in a report by selecting **Clustering > Show information** (i). This pops up a *Report* window with the applied comparison settings in a hierarchical structure, similar as for a dendrogram (see 13.3.1).

13.3.3 Pairwise comparisons

From within any window where you can select entries, you can display a detailed comparison between two entries. This pairwise comparison shows all the images of the experiment types as well as the similarities obtained using the specified coefficients.

First select any two entries you want to compare (see 3.3.8). In the *Main* window, you can then use **Analysis > Compare two entries (Ctrl+2)**. The **Ctrl+2** shortcut works from within any window. The *Pairwise comparison* window appears (Figure 13.3.6).

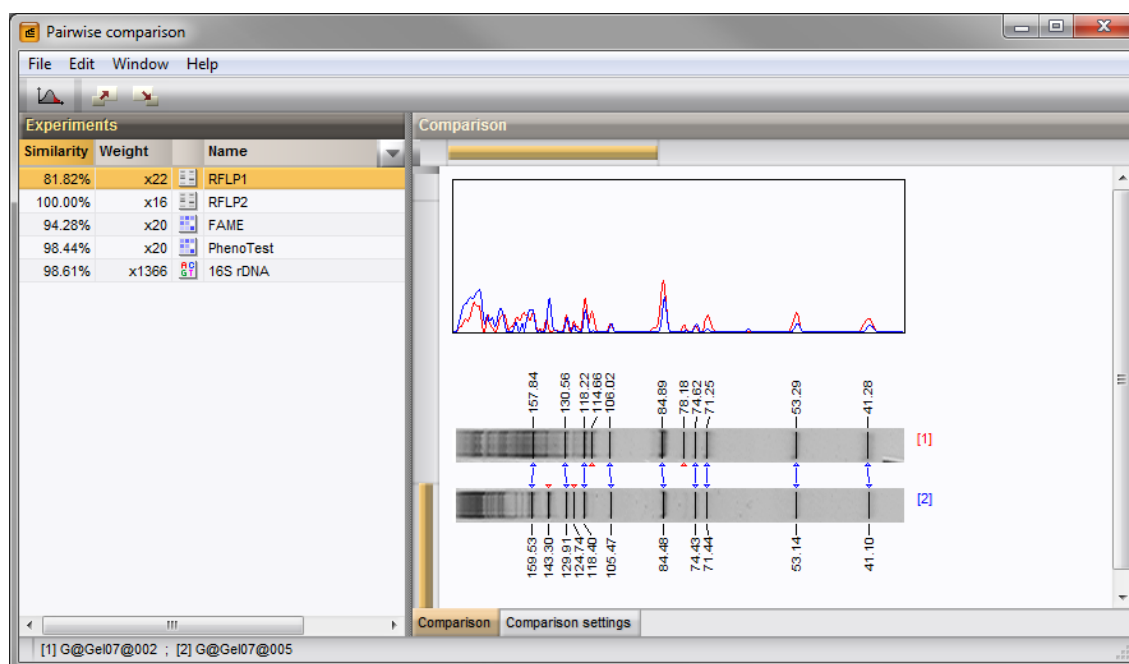


Figure 13.3.6: The *Pairwise comparison* window.

The *Pairwise comparison* window consists of three dockable panels: the *Experiments* panel, the *Comparison* panel and the *Comparison settings* panel. For detailed information about the display of dockable panels, see 2.3.4.

The *Experiments* panel (left panel in default configuration) displays the names of all experiment types present in the database in the "Name" column.

When an experiment type is present for both entries, the similarity value for this experiment type is shown in the information field "Similarity".


The "Weight" column displays the weight that would be assigned to the experiment if included in a composite data set (see 11.1.2).

The information fields can be displayed or hidden by pressing the column properties button (☷) and selecting **Set active fields** (see 3.2.7 for more information).

When clicking on an experiment type in the *Experiments* panel, the corresponding image is displayed in the *Comparison* panel (right panel in default configuration).

The *Comparison settings* panel lists the comparison settings used to calculate the similarity value of the selected experiment type.



The comparison settings are defined in the *Comparison settings* wizard. This dialog box can be accessed from the corresponding experiment type window (via **Settings > Comparison settings...** ) and from the *Comparison* window (via **Clustering > Calculate > Cluster analysis (similarity matrix)...**).

In case of fingerprint types, the detailed comparison of the band matching is shown if a band matching coefficient was chosen in the experiment settings (e.g. Dice coefficient in Figure 13.3.6).

In case of character types, all characters present in the character type are listed ("Character"), together with the character values and their corresponding colors. When a character mapping is defined, the categories are shown in the "Mapping" column.

In case of sequences, the aligned sequences are shown.

In case of trend data types, the trend curves for both entries are shown.


In case of whole genome maps, the match indication between the two whole genome maps is shown for both the specified alignment and pattern match settings, using the active display settings (see *dlg fragm display*).

In case of spectra, the visualization consist of three parts. At the top two tables with basic information about the spectra are shown. In the center a mirrored view of the two spectra is shown, containing both the curves and the peaks. At the bottom we have the band representation of the two spectra. These are each matched against the default peak class type, which is shown above the top band representation. The default peak class type can be changed from the *Spectrum type* window. A line connecting two peaks is shown if both peaks match with the same peak class.

In case of sequence read sets, a heat map displaying the keyword similarity between the two experiments is shown. Note that the current keyword length can be changed from the *Sequence read set type* window. The map is calculated using the local point density of keyword frequencies when these would be plotted against each-other. If the local points density is low enough, the individual points are shown and the corresponding keyword is added as point label. At a sufficiently high zoom level all individual points are shown, labels are displayed where the local points density is low.

13.3.4 Working with comparison groups

An important display function in the *Comparison* window is the creation of *comparison groups*. Comparison groups can be defined from clusters, from database fields, or just from any subdivision the user desires. They are normally displayed using rectangles of different colors next to the entries, each group having its own color. Alternatively, they can be displayed using different symbols or using alphanumerical codes. In the first place, comparison groups facilitate the comparison between a dendrogram or a dimensioning and a certain characteristic (database information). Comparison groups also make the homology display between dendrograms obtained from different experiments easier. In addition, comparison groups are necessary in a number of derived statistical analysis functions, such as Partitioning, Group separation, Discriminant Analysis, and MANOVA (13.3.8). Finally, comparison groups form an easy link between dimensional representations such as PCA, SOM or graphs and scatter plots on the one hand, and database field information on the other hand.

A selection of entries can be assigned to a group with **Groups > Create new group from selection** , **Ctrl+G**. A new dialog pops up (see Figure 13.3.7).

The *Create new group* dialog box prompts for the group name.

The group name is displayed in the **Name** column in the *Groups* panel, together with the number of selected entries (**Size**). A color is randomly chosen.

In the *Information fields* panel the name of the group is displayed in the **Group** column; the colors are displayed next to the check boxes.

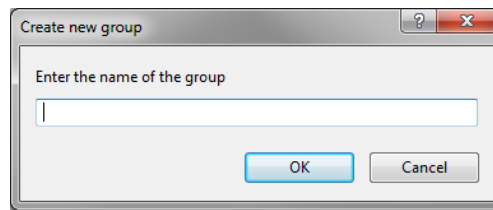


Figure 13.3.7: Specify a group name.

The text in the *Name* column in the *Groups* panel can be edited by clicking twice in the *Name* field. The cell will appear highlighted and information can be added/edited.

An alternative method to define comparison groups is by selecting a database field and having the program automatically create groups based upon the different names that exist in this database field. One should be aware, however, that any misspelled name or typographic error will result in a different group. The method works as follows:

Highlight a database field by clicking on its field name and select **Groups > Create groups from database field**. The *Group creation preferences* dialog box appears (Figure 13.3.8).

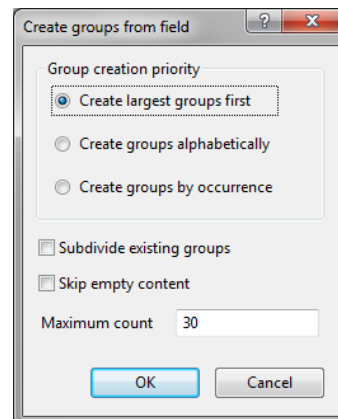


Figure 13.3.8: The *Group creation preferences* dialog box.

The *Group creation priority* determines the order in which comparison groups are assigned:

- **Create largest group first:** The group containing the largest number of entries will be group 1, the second largest group will be group 2, etc..
- **Create groups alphabetically:** Groups will be created according to the alphabetical order of the information field.
- **Create groups by occurrence:** Assigns groups in the order in which they are listed in the comparison; the first occurrence of a group member determines its group number.

When **Subdivide existing groups** is disabled (default setting), any previously defined comparison groups will first be removed and the program will assign the new groups based upon the selected database field only. If you check **Subdivide existing groups**, the program will keep the groups that are already defined, and split existing groups into more groups if differences in the selected database field are found.

With **Skip empty content** checked, no group will be created for the information fields that are empty.

A maximum number of comparison groups needs to be defined, with a default value of 30 (*Maximum count*).

If the **<OK>** button is pressed, the program will create comparison groups according to the highlighted information fields (see Figure 13.3.9).

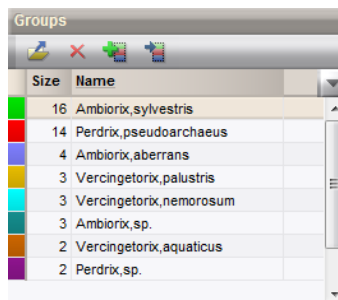


Figure 13.3.9: The *Groups* panel after executing the *Subdivide existing groups* command.

Since different colors are not equally distinguishable by different persons it may be useful to customize the comparison group colors in a user-defined scheme. To define an own group color scheme, select **Groups > Edit pre-defined group colors....** This brings up the *Group color* dialog box (Figure 13.3.10).

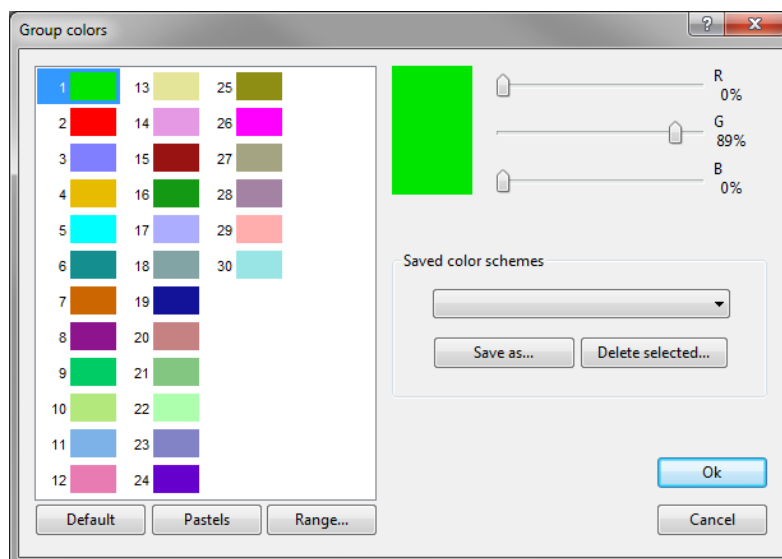


Figure 13.3.10: The *Group color* dialog box.

For each comparison group, three slider bars (red, green and blue, respectively) can be adjusted to produce any desired color.

A user defined color scheme can be selected from the drop-down list of *Saved color schemes*.

To delete a saved color scheme, first select it, and then press the **<Delete selected>** button.

To bring up the default color scheme, press **<Default>**. Another predefined scheme, using pastel colors, can be loaded by pressing **<Pastels>**.

It is also possible to generate a scheme of transition colors by pressing **<Range>**. This calls the *Create color range* dialog box (see Figure 13.3.11).

BioNumerics asks to enter the number of colors to include in the range. A number between 2 and 30 should be specified.

A thus obtained color scheme can be saved by pressing the **<Save as>** button. This calls the *Save color scheme* dialog box (see Figure 13.3.12).

The dialog prompts for a color scheme name.

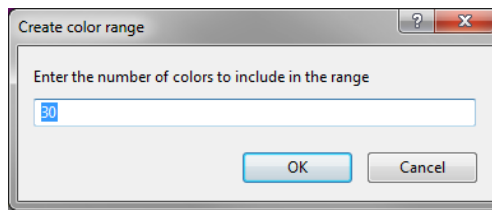


Figure 13.3.11: The *Create color range* dialog box.

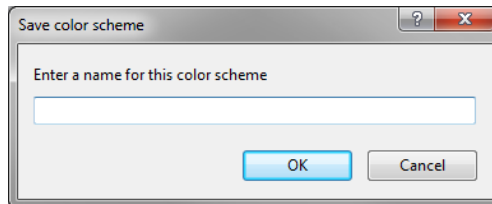


Figure 13.3.12: The *Save color scheme* dialog box, to specify a name for the new color scheme.

The name and color of a selected group can be edited with **Groups > Edit group...** (🖱️). This action calls the *Edit group* dialog box (see Figure 13.3.13).

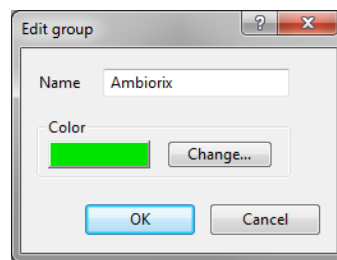


Figure 13.3.13: Edit a group.

In the *Edit group* dialog box the group **Name** and group **Color** can be changed.

Group assignments can be changed with **Groups > Assign selection to highlighted group** (🖱️, **Ctrl+Shift+G**). The selected entries are assigned to the selected group in the *Groups* panel.

All members of a selected group can be selected at once with **Groups > Select highlighted group members**.

All group assignments are removed from the *Comparison* window with **Groups > Reset groups**.

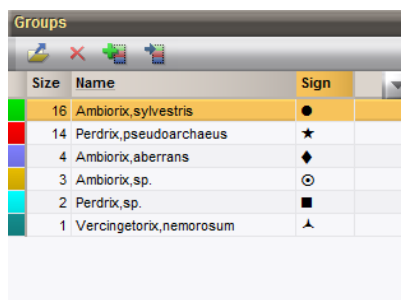
A single group can be removed from the *Comparison* window with **Groups > Delete highlighted group** (🖱️).

Selected entries can be removed from their assigned group with **Groups > Assign selection to no group**.

As an alternative for using group colors, it is possible to assign a symbol to each group by unchecking **Groups > Show groups using colors** in the menu of the *Comparison* window. The colors in the *Information fields* panel are replaced by symbols. To view the list of symbols in the *Groups* panel, select the 🖱️ button in the *Groups* panel and select **Sign** from the pull-down menu (see Figure 13.3.14).


In many cases, the entry keys or particular information fields may be too long to be displayed in particular comparison types, e.g. phylogenetic trees, PCA plots, MANOVAs, and rendered trees. In such cases, the entry keys can be replaced by a group code. The program assigns a letter to each defined comparison group, and within a group, each entry receives a number.

The group codes are shown by selecting **Layout > Use group numbers as key**. The keys in the *Information*



Size	Name	Sign
16	Ambiorix, sylvestris	●
14	Perdrix, pseudoarchaeus	★
4	Ambiorix, aberrans	◆
3	Ambiorix, sp.	⊙
2	Perdrix, sp.	■
1	Vercingetorix, nemorosum	⤵

Figure 13.3.14: Group symbols.

fields panel are replaced by a letter followed by a number. The letter corresponds to the Group letter. To view the list of letters in the *Groups* panel, select the  button in the *Groups* panel and select **Letter** from the pull-down menu. A legend to the Group numbers can be obtained with **File > Export > Export database fields...** in the *Comparison* window.

Alternatively, an information field can be displayed instead of the key by clicking on the information field and selecting **Layout > Use field as key**. The content of the information field is now displayed in the **Key** field and in e.g. a PCA plot, rendered tree, etc.

The group assignments are saved along with the comparison.

13.3.5 Local composite data sets

Local composite data sets share many features with regular composite data sets (see 11). The major difference is that local composite data sets are saved along with the comparison in which they were created and not available outside this comparison. Basically, they offer a convenient way to create composite data sets on-the-fly, without having to close and re-open the *Comparison* window.

Local composite data sets are managed in the *Local composite datasets* panel. By default, this panel appears as a tab below the *Experiments* panel (top left in the *Comparison* window).

To create a new local composite data set, select **Composite > Local composite dataset > Create...** (). This action will display the *Edit local composite dataset* dialog box (see Figure 13.3.15).

This dialog box allows you to create a new local composite data set or to edit an existing one.

The **Name** of the local composite data set, as will be used in the *Local composite datasets* panel, can be specified in the top part of the dialog.

All aspects (see 13.2.5) of all experiment types defined in the database are listed in the central part of the dialog. The check boxes on the left allow to include or exclude certain aspects. Values in the Weight column can be edited to set the weight of individual aspects.

Since a database can potentially contain a large list of experiment types and their aspects, checking the **Show selected only** option provides a more convenient overview of the included aspects when editing a local composite data set.

Pressing <OK> in the *Edit local composite dataset* dialog box will add the local composite data set to the *Local composite datasets* panel.

Double-clicking on a local composite data set in the *Local composite datasets* panel will open the *Edit local composite dataset* dialog box again.

Similar to regular composite data sets, local composite data sets are visualized in the *Comparison* window as *character tables*, for which several display functions are available. See 11.2.3 for more information.

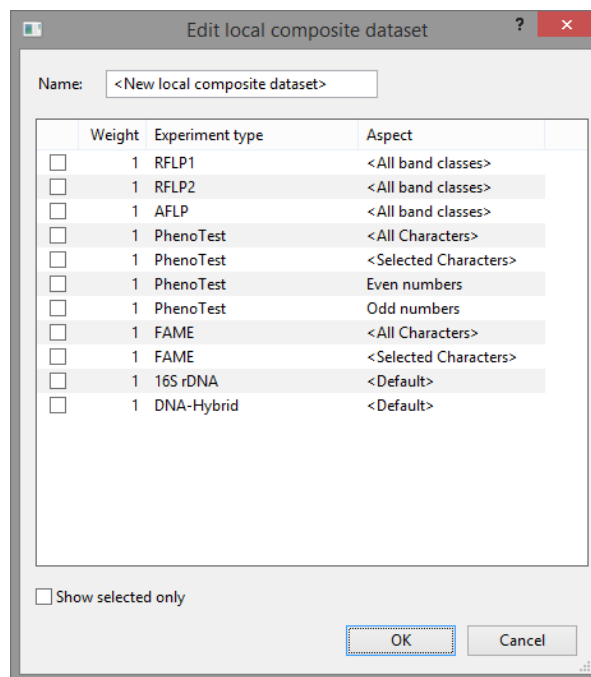


Figure 13.3.15: The *Edit local composite dataset* dialog box.

A local composite data set can be deleted by highlighting it in the *Local composite datasets* panel and selecting **Composite** > **Local composite dataset** > **Remove...** (✖). The software will ask for confirmation before actually removing the local composite data set.

13.3.6 Tree degeneracy

The outcome of a pairwise, hierarchical clustering algorithm is a single dendrogram. However, it is important to realize that in many practical cases multiple, equivalent solutions are possible. This problem arises if equal values occur in the similarity matrix (also called *ties*). This situation is inherent to the nature of the data and especially occurs if one is dealing with similarity coefficients acting on binary data. The comparison settings in BioNumerics contain a number of advanced options to deal with degeneracies. The principles are best illustrated with a data type that can potentially result in multiple tree solutions, e.g. a fingerprint type in case a band-based coefficient is used or a binary character type.

In the *Comparison* window, select an experiment type in the *Experiments* panel for which you want to calculate a dendrogram.

Select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)...**

Select a coefficient in the *Similarity coefficient* wizard page and press <Next> to go to the second page of the wizard, where the settings for **Degeneracy handling** can be specified (Figure 13.3.16).

The option **Enable degeneracy handling** becomes available when **UPGMA**, **Single Linkage** or **Complete Linkage** is checked. All three methods are pairwise clustering algorithms, which will construct dendrograms by grouping branches and/or entries pair by pair, using the highest similarity as criterion.

The **Secondary criterion** applies to those cases where two clusters have the same (highest) similarity with a third, in which case two different tree solutions exist. The program will then apply one of the following criteria to solve the indeterminacy left by the standard clustering algorithm (the primary criterion):

1. **Highest overall similarity**: the two clusters will be joined that result in the cluster with the highest

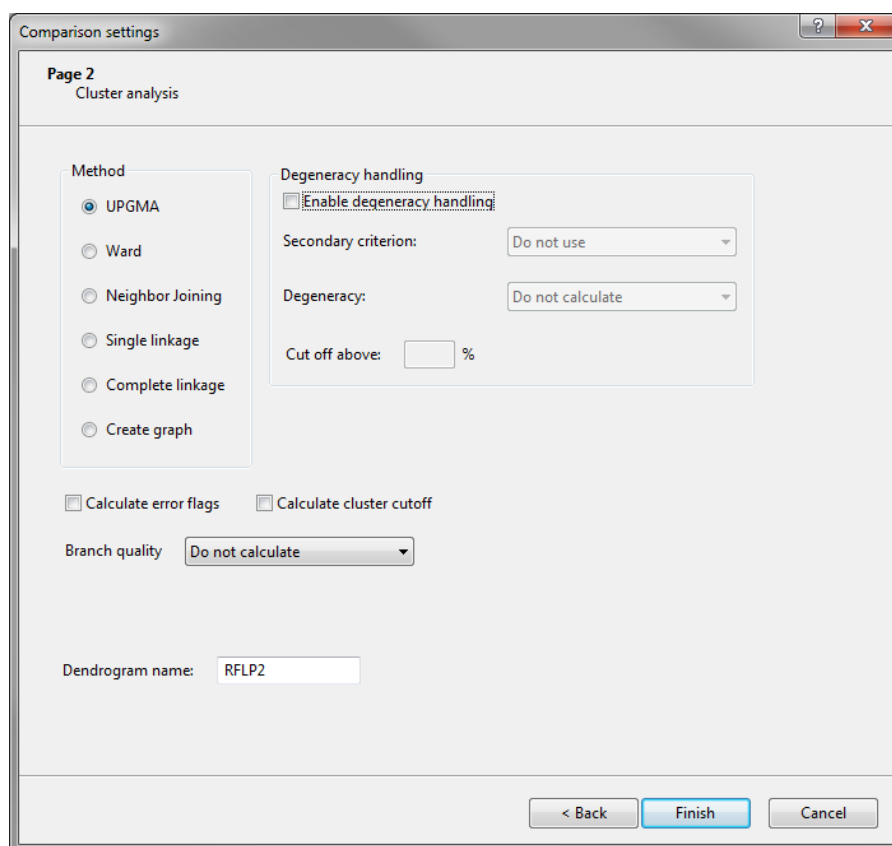


Figure 13.3.16: The *Cluster analysis* wizard page, where the settings for degeneracy handling can be specified.

overall similarity with all other members of the comparison.

2. **Largest number of entries:** the two clusters will be joined that result in the cluster with the largest number of entries.
3. **Most homogeneous clusters:** the two clusters will be joined that result in a cluster that has the highest internal homogeneity.
4. **Most identical matches:** the two clusters will be joined that result in a cluster with the most identical matches.

When **Do not use** is selected, no secondary criterion is taken into account. Note that criteria 1 and 3 are complementary to each other as 1 will only consider the external similarity values of the resulting clusters whereas 3 will only consider their internal similarity values.

Under **Degeneracy**, three options allow one to deal with degenerated trees:

1. **Do not calculate** will not look for degeneracies and will display just one solution. The differences with a conventional cluster analysis is that the best solution according to a secondary criterion (if specified) can be calculated.
2. The option **Clustering only** will calculate all degeneracies resulting from the primary criterion only and will not consider any secondary criterion specified.
3. **Clustering + secondary criterion** will use the specified secondary criterion to solve the degeneracies resulting from the pairwise clustering algorithm and will only display the degeneracies that remain

after the secondary criterion. It is unlikely that there will remain any degeneracies with this option checked.

The **Cut off above** parameter specifies the maximum allowed percentage of degenerate entries relative to a cluster. A *degenerate entry* is an entry that does not belong to a given cluster in the present tree, but that does belong to the cluster in at least one alternative solution. If zero is entered as cutoff value, no degenerate entries are allowed and as a consequence, a consensus tree is generated that includes all possible solutions. If 100 is entered or when the field is left blank, the degeneracy of the tree will not be reduced at all. If a percentage is entered, for example 20, all clusters for which there are more than 20% (relative to the number of entries in the cluster) degenerate entries will be displayed as consensus clusters with the degenerate entries included.

Each cluster that has degenerate entries relative to it, will have an indication of the number of degenerate entries (see Figure 13.3.17, which shows one degenerated entry for the selected cluster).

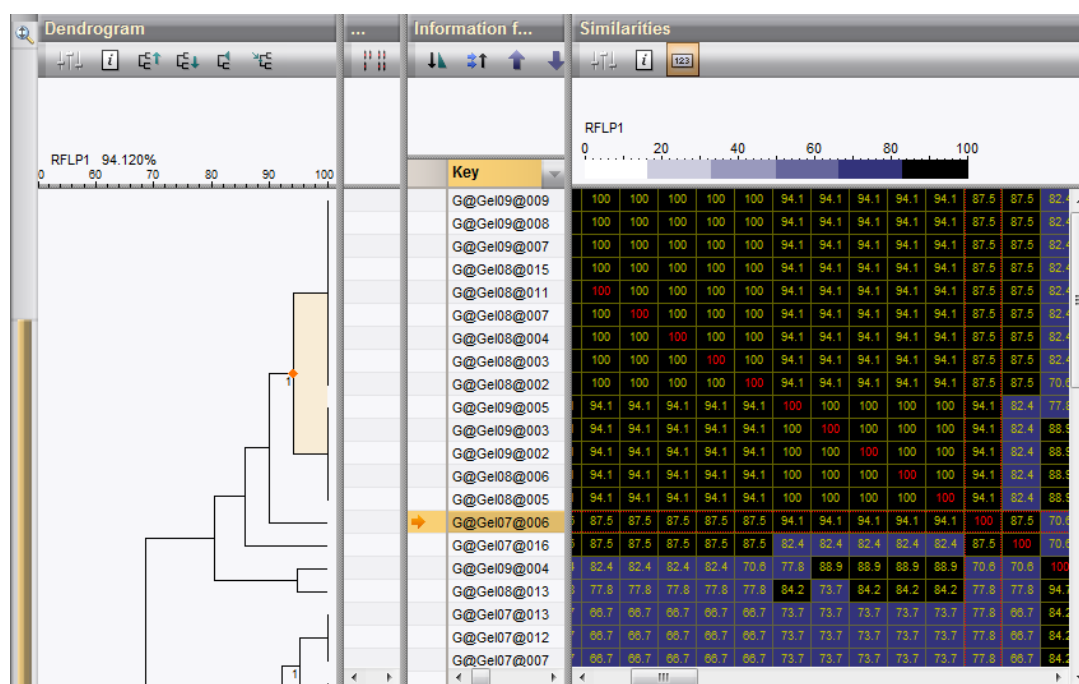


Figure 13.3.17: Advanced tree representation with a highlighted cluster, indication of the number of degenerated entries relative to the cluster, and the degenerated entry selected.

If there are degenerated entries relative to the highlighted cluster, you can find them by selecting **Clustering > Advanced tools > Select degenerate entries**. All degenerate entries relative to the cluster are now added to the selection.

The interpretation of degeneracies and tracking back their reason is sometimes difficult. The larger the tree and the deeper the branch, the more complex the degeneracies will be.

The example screen in Figure 13.3.17 is a capture taken from experiment **RFLP1 in DemoBase Connected**. The highlighted cluster has one degenerated entry, which is selected. The cluster consists of four sub-clusters which have an overall average similarity of 94.1%. The single degenerate entry, however, also has an average similarity of 94.1% with the second sub-cluster. The present solution has first linked sub-cluster 1 to sub-cluster 2 and then linked the single entry to the merged cluster. According to the criterion of UPGMA, however, an equivalent solution would be to first link the single entry to sub-cluster 2 and then link sub-cluster 1 to this new cluster. When the same clustering is done with zero as cutoff value, the cluster looks like in Figure 13.3.18. Note that the three sub-clusters are now linked together at the same level. The clusters that connect always at the displayed similarity level in the solution obtained using the secondary criterion

are represented by solid lines (in the present case, the single entry), whereas sub-clusters that cluster at higher levels using the secondary criterion are connected by an interrupted line.

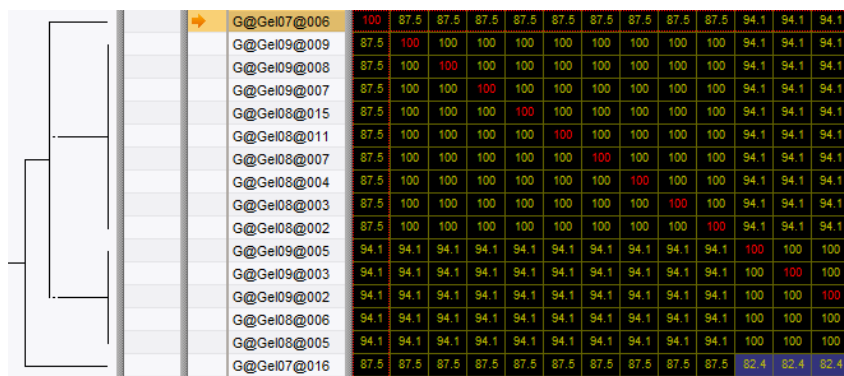


Figure 13.3.18: Detail of cluster highlighted in Figure 13.3.17, calculated with a cut off value of zero.

13.3.7 Cluster significance tools

13.3.7.1 Introduction

A dendrogram tells you something about the groups among a selection of entries, but nothing about the *significance*, i.e. the reliability or the trueness of these groups. Therefore, the software offers a range of methods that express the stability or the error at each branching level.

13.3.7.2 Error flags

The simplest indication of the significance of branches is showing the average similarities of the dendrogram branches (see 13.3.1).

The *standard deviation* of a branch is obtained by reconstructing the similarity values from the dendrogram branch and comparing the values with the original similarity values. The standard deviation of the derived values versus the original values is a measure of the reliability and internal consistence of the branch.

A dendrogram with error flags is calculated as follows:

Highlight the experiment type for which you want to calculate a dendrogram and select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**...

In the *Similarity coefficient* wizard page, specify a similarity coefficient and its parameters. Press <Next> to go to the second page of the wizard.

Select a clustering method and check **Calculate error flags**. Press <Next> to calculate the dendrogram.

A dendrogram appears with error flags drawn on each branch. The average similarity and the exact standard deviation is shown at the position of the cursor (see Figure 13.3.19). The smaller this error flag, the more consistent a group is. In the example, the *Perdrix* group has a small error flag, meaning that this group is very consistent. This group will for example not disappear by incidental changes such as tolerance settings, adding or deleting entries, etc..

The error flags can be removed from the dendrogram via the dendrogram display settings:

Select **Clustering** > **Dendrogram display settings**... (Figure 13.3.1), uncheck the option **Show error flags** in the *Dendrogram display settings* dialog box (Figure 13.3.1) and press <OK>.

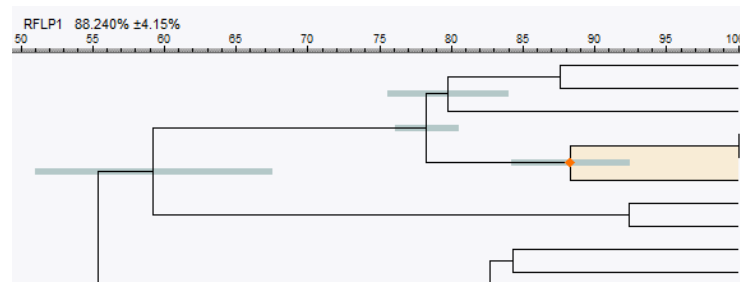


Figure 13.3.19: Dendrogram with error flags, detail. The average similarity and standard deviation is shown at the cursor's position (top).

13.3.7.3 Cophenetic correlation

The *Cophenetic Correlation* is another parameter to express the consistency of a cluster. This method calculates the correlation between the dendrogram-derived similarities and the matrix similarities. The value is usually calculated for a whole dendrogram, to have an estimation of the faithfulness of a cluster analysis. In BioNumerics, the value is calculated for each cluster (branch) thus estimating the faithfulness of each sub-cluster of the dendrogram. Obviously, you can obtain the cophenetic correlation for the whole dendrogram by looking at the cophenetic correlation at the root.

Highlight the experiment type for which you want to calculate a dendrogram and select **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)**...

In the *Similarity coefficient* wizard page, specify a similarity coefficient and its parameters. Press <Next> to go to the second page of the wizard.

Select a clustering method and select **Cophenetic correlation** as **Branch quality** parameter. Press <Next> to calculate the dendrogram.

A dendrogram appears with the cophenetic correlation shown at each branch (Figure 13.3.20), together with a colored dot, of which the color ranges between green-yellow-orange-red according to decreasing cophenetic correlation. This makes it is easy to detect reliable and unreliable clusters at a glance.

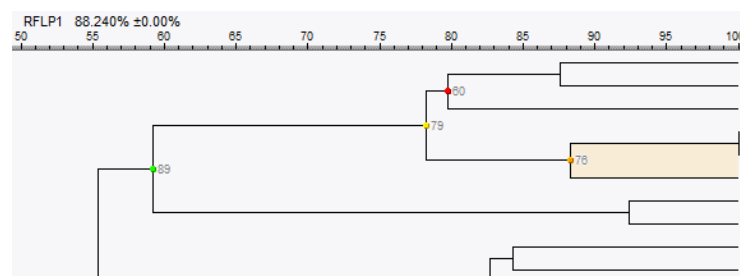


Figure 13.3.20: Dendrogram showing cophenetic correlation values, detail.

The colored dots and cophenetic correlation values can be removed from the dendrogram via the dendrogram display settings:

Select **Clustering** > **Dendrogram display settings...** (⚙️), uncheck the option **Show branch quality** in the *Dendrogram display settings* dialog box (Figure 13.3.1) and press <OK>.

13.3.7.4 Bootstrap analysis

Bootstrap analysis [14] measures cluster significance at a different level. Instead of comparing the dendrogram to its similarity matrix, it directly measures the influence of characters on the obtained dendrogram.

The concept is very simple: "sampling with replacement", i.e. characters are randomly left out from the character set and are replaced with others [16]. For each sampling case, the dendrogram is recalculated, and the relative number of dendrograms in which a given cluster occurs is a measure of its significance. This method requires the characters to be independent and equally important.

Since bootstrap analysis requires a closed character set, the method can only be performed on aligned sequences and character type data. In case of fingerprint type data, a band matching needs to be performed first (4.3).



To be able to perform bootstrap analysis on your data, do NOT choose *Average from experiments* in the *Comparison settings* wizard (see 11.2). The bootstrap analysis option is only accessible when using a similarity coefficient, i.e. when the data in the composite data set are treated as character data.



Bootstrap analysis of character data can only be performed on a composite data set (see 11.1).

To calculate bootstrap values, proceed as follows:


Highlight the experiment type for which you want to calculate a dendrogram and select **Clustering > Calculate > Cluster analysis (similarity matrix)...**

In the *Similarity coefficient* wizard page, specify a similarity coefficient and its parameters. Press <Next> to go to the second page of the wizard.

Select a clustering method and select **Bootstrap analysis** as **Branch quality** parameter.

Enter the number of **Bootstrap simulations** (samplings) to perform. For demonstration purposes, 100 samplings are sufficient.

Press <Next> and wait until the sampling and calculation process is finished.

The bootstrap values are shown on the resulting dendrogram in a similar way as the cophenetic correlation values (see Figure 13.3.20). They can be removed from the dendrogram via the dendrogram display settings: Select **Clustering > Dendrogram display settings...** , uncheck the option **Show branch quality** in the *Dendrogram display settings* dialog box (Figure 13.3.1) and press <OK>.

13.3.7.5 Cluster cutoff

Another way of looking at dendrograms is to try to delimit, by objective means, the relevant clusters from the non-relevant clusters. The simplest and most arbitrary method is to draw a vertical line through the dendrogram in a way that it cuts most homogeneous clusters from most heterogeneous clusters. However, there are more statistically founded methods to draw either straight lines, or to evaluate cluster by cluster and delimit relevant clusters at different similarity levels. The *Cluster Cutoff method* in BioNumerics is one of these statistical methods. The method draws a line through the dendrogram at a certain similarity level, and from the resulting number of clusters defined by that line, it creates a new, simplified, similarity matrix, in which all within-cluster values are 100%, and all between-cluster values are 0%. Then, the *Point-biserial correlation* is calculated, i.e. the correlation between this new matrix and the original similarity matrix. The same is done again for other cutoff similarity levels, and the level offering the highest Point-biserial correlation is the one offering the most relevant groups.

In BioNumerics, this standard method is even further refined, as the cutoff values can be different per cluster, allowing even more reliable clusters to be defined.

Highlight the experiment type for which you want to calculate a dendrogram and select **Clustering > Calculate > Cluster analysis (similarity matrix)...**

In the *Similarity coefficient* wizard page, specify a similarity coefficient and its parameters. Press <Next> to go to the second page of the wizard.

Select a clustering method and check **Calculate cluster cutoff**. Press <Next> to calculate the dendrogram. A dendrogram appears with those branches that were found to be below the cluster cutoff value shown in dashed lines.

13.3.7.6 Consensus tree

BioNumerics allows a *consensus tree* to be calculated from two or more individual dendrograms. These trees can be conventional clusterings or trees exported from the *Advanced cluster analysis* window, and can be generated from the same experiment type or from different experiment types.

BioNumerics will look for all branches that hold exactly the same entries in all trees and represent them as branches in the consensus tree.

Selecting **Clustering > Advanced tools > Create consensus tree...** calls the *Consensus tree* dialog box (see Figure 13.3.21).

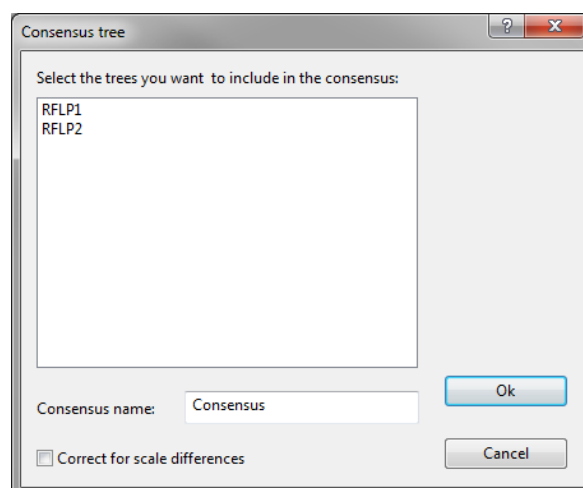


Figure 13.3.21: Create a consensus tree.

Select the trees from the list that you want to include in the consensus tree using the **Ctrl** and **Shift**-keys.

Specify a *Consensus name* for the consensus tree to be generated (the default name is **Consensus**).

With the option **Correct for scale differences**, the dendrograms will first be rescaled so that they have the same similarity ranges. The result is that dendrograms covering a narrow similarity range will have more impact on the consensus tree when this option is checked.

13.3.8 Group statistics

13.3.8.1 K-means partitioning

One function to let the software automatically determine comparison groups (see 13.3.4) is the mathematical function *K-means partitioning*. The user first creates groups based upon one or more strains (e.g. type strains) (see Figure 13.3.22 for an example). Then, the program automatically calculates for each entry of the cluster analysis in which group it fits best.

The partitioning method can be executed for an existing dendrogram in the *Comparison* window with **Groups > Partitioning of groups...** which pops up the *Partitioning* dialog box (Figure 13.3.23).

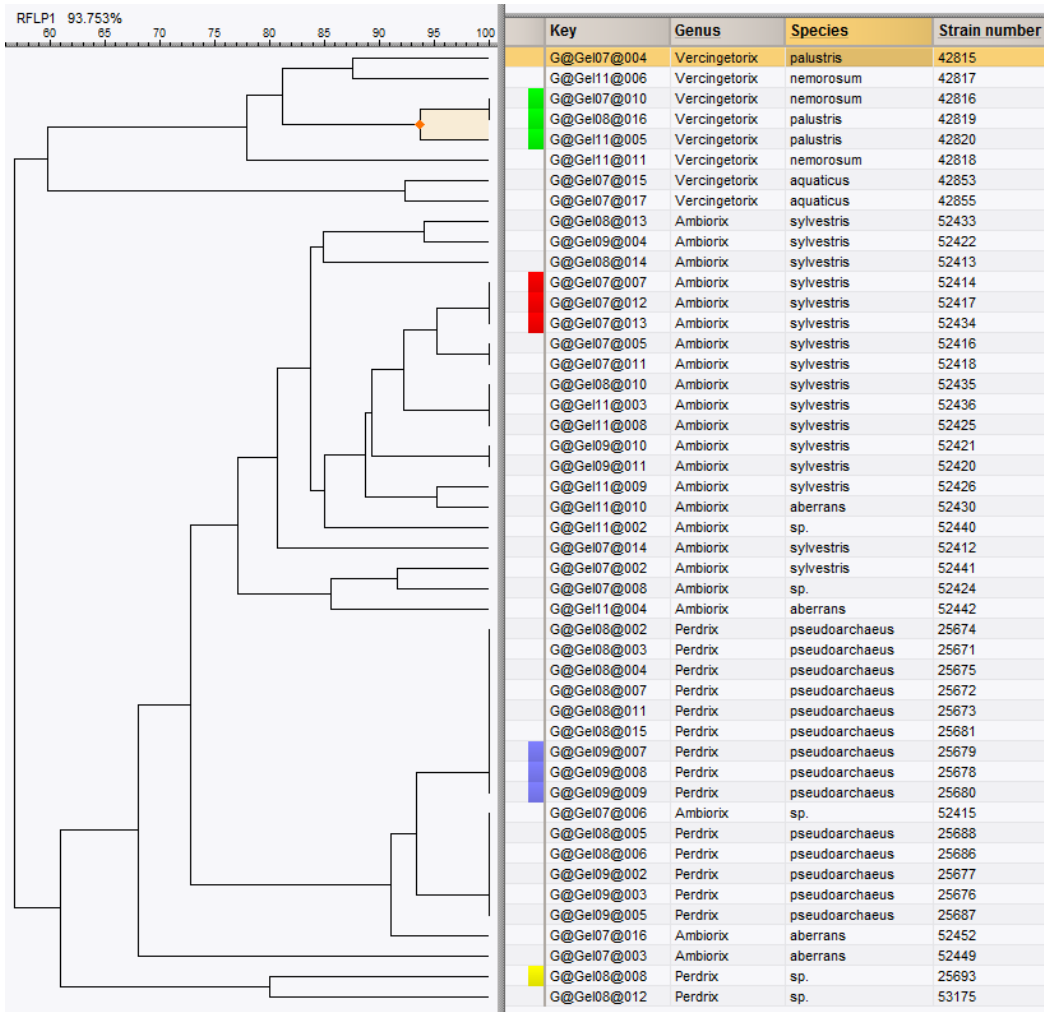


Figure 13.3.22: Example of manual group assignment in preparation of a partitioning process.

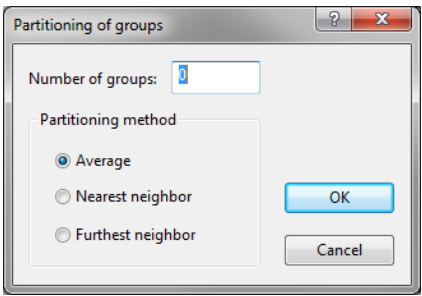


Figure 13.3.23: The *Partitioning* dialog box.

The *Number of groups* can be specified, so the algorithm automatically divides the entries into this predefined number of most relevant groups. If this option is left to zero (the default value), the program will only use the comparison groups that were defined manually.

The K-means partitioning can be based upon *Average* similarity with the group, upon the highest similarity (*Nearest neighbor*), or upon the lowest similarity (*Furthest neighbor*).

Obviously, the partitioning process must be iteratively executed, since by adding an entry to a group, the average similarity of the group as well as the highest and lowest similarities with entries may change.

After the calculations, all entries belong to one of the defined groups.

Note that these groups do not necessarily correspond exactly to the visual clusters on the dendrogram. This can be the case if the clusters on the dendrogram are not well-defined or inconsistent. A second cause of aberrations is the oversimplification of complex matrices by the UPGMA algorithm.

13.3.8.2 Group separation statistics

These statistical methods determine the stability of the defined comparison groups (see 13.3.4), whether they are defined manually, derived from clusters, using K-means partitioning, or created from an information field. They involve the *Jackknife* method and the *Group violations* measurement.

With **Groups > Group separation...** the separation between the defined groups are investigated. The *Group separation settings* dialog box is shown, allowing a number of choices to be made (Figure 13.3.24).

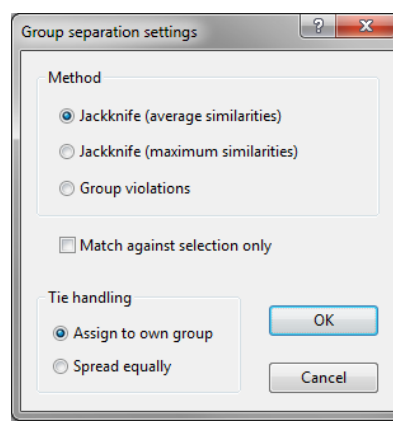


Figure 13.3.24: The *Group separation settings* dialog box.

The principle of the Jackknife method is to take out one entry from the list, and to identify this entry against the different groups. This can be done by calculating the average similarities with each group (*Jackknife (average similarities)*), or finding the highest similarities (*Jackknife (maximum similarities)*) with each group. This is done for all entries (unless *Match against selection only* is checked). The percentage of cases that entries are identified to the group they were originally assigned to, is a measure of the internal stability (significance) of that group. The percentage of cases that entries are identified to another group than originally assigned to, is indicative of lack of internal stability.

Using *Match against selection only*, you can let the program calculate the matches against a selection you made in the comparison, rather than against all entries of the groups.

In cases where an entry has an equal match with a member of its own group and a member of another group (a tie), there are two equally valid interpretations possible. The program can handle such ties in an "optimistic" way, i.e., by always assigning equal matches to their own group, or in a "realistic" way, by spreading ties equally between the own and the other groups.

The way ties are handled can be chosen in the *Group separation settings* dialog box under *Tie handling*. This includes two options, *Assign to own group* and *Spread equally*.

Clicking <OK> starts the calculations and displays the *Group separation* window (Figure 13.3.25).

The *Group separation* window displays a matrix of the groups that are present in the *Comparison* window. The group violations method compares all the similarity values within a group with those between a group and the other groups. All the values occurring in the overlap zones (see Figure 13.3.26) are considered "violations" of the integrity of the group.

The percentage of group violations for group A is the number of external entries scoring higher than the

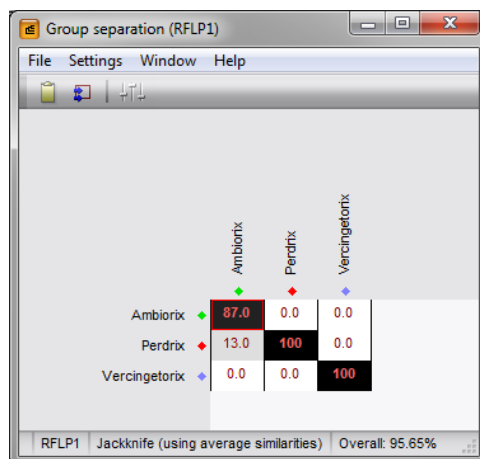


Figure 13.3.25: The *Group separation* window.

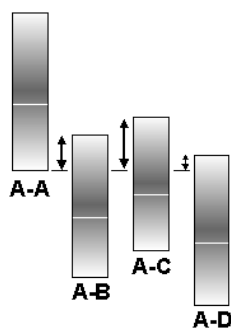



Figure 13.3.26: Schematic representation of internal similarity range of group A (A-A), and similarity ranges with other groups (A-B, A-C, and A-D). The overlapping values are group violations.

lowest internal values over the total number of similarity values considered. The percentages seen in the diagonal of the matrix are the percentages of non-violations. The number of misidentifications for members of group 1 are given in column 1, for members of group 2 in column 2, etc.. The overall quality of the group separation is indicated in the status bar of the window; it is the average of the diagonal, i.e. the total percentage of correct identifications.

Note that the values in the matrix are **not** reciprocal, i.e. the matrix is not symmetric!

In Figure 13.3.25 for example, 0% of group 1 members are identified as group 2, but 4.7% of group 2 members are identified as group 1.

When the Jackknife method is used, a value (or cell) in the group separation matrix can be selected, and with **File > Select cell members** () , the entries contributing to this cell will be selected in the *Comparison* window. The method is useful to identify entries that fit well or do not fit well in their assigned groups.



The interpretation of matching and non-matching entries is less easy when the **Spread equally** function has been chosen, since in that case, some entries may fall outside their group "unexpectedly" when they have an equally high score with another group.

Selecting  calls the *Group separation settings* dialog box again, to recalculate the group separation with different settings.

The group statistics can be copied to the clipboard using **File > Copy to clipboard** ().

13.3.9 Printing and exporting a cluster analysis

13.3.9.1 Introduction

When printing from the *Comparison* window, BioNumerics first shows a print preview. This print preview shows the same information as is shown in the panels of the *Comparison* window: for example, a dendrogram, one or more images from different experiments, metrics scale, etc.. One exception is the similarity matrix: the print preview does not print matrices unless you explicitly select it in the print preview. The preview looks exactly as it will look on printed pages. You can edit the layout of the print preview by adjusting the space allowed for the different items (dendrogram, image(s), information fields), by changing the size of the figure to fit on one or more pages, etc..



The print preview will always display dendrogram, experiment image and database information arranged from left to right, irrespective of the configuration of the *Comparison* window.

13.3.9.2 The Comparison print preview window

Selecting **File > Print preview...** (🖨️, **Ctrl+P**) in the *Comparison* window, opens the *Comparison print preview* window (Figure 13.3.27).

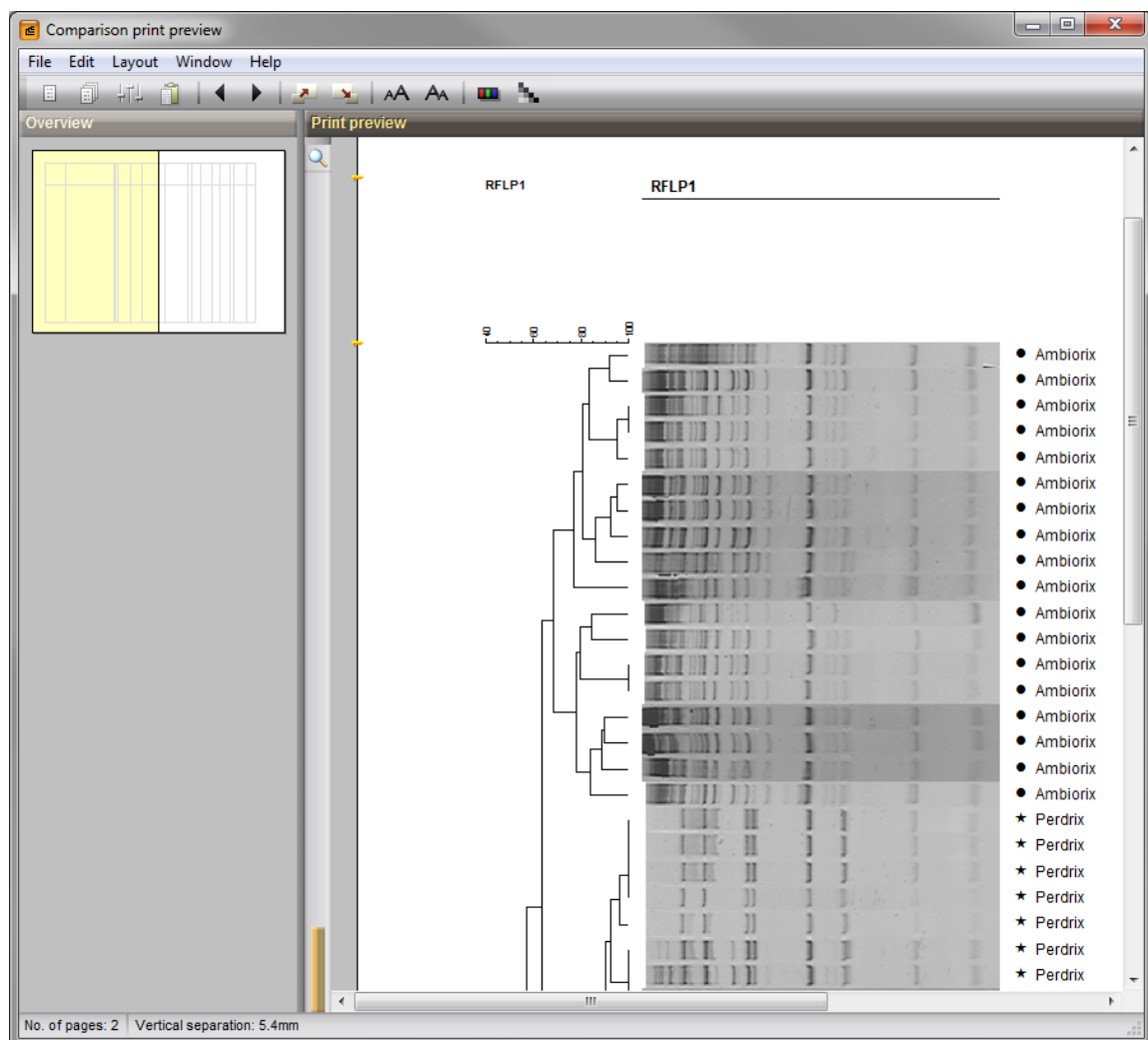




Figure 13.3.27: The *Comparison print preview* window.



The *Comparison print preview* window is divided in two panels, which are both dockable (see 2.3.4 for display options of dockable panels). The *Overview* panel shows an overview of the pages that will be printed, with the actual page in yellow. In the *Print preview* panel, the actual page is shown.

On top of the preview page, there are a number of small yellow slide bars. These slide bars represent the following margins, respectively:


- Left margin of the whole image;
- If dendrogram shown, right margin of dendrogram;
- If image shown, right margin of image;
- If groups are defined, right margin of groups;
- Right margin of entry keys or group codes (if not hidden);
- Right margins of different information fields (except the hidden fields);
- If the similarity matrix is shown, right margin of matrix.


Left on the first preview page, there are two slide bars: representing the top margin of the whole figure and the lower margin of the header, respectively. Left on the last page, there is one slide bar representing the bottom margin of the image. Each of these slide bars can be shifted individually to reserve the appropriate space for the mentioned items. The image is printed exactly as it looks on the preview.

With **Edit > Next page** (, **Page Down**) and **Edit > Previous page** (, **Page Up**), you can thumb through the pages that will be printed out.


It is possible to zoom in and out on a page using **Edit > Zoom in** (, **Ctrl+Page Up**) and **Edit > Zoom out** (, **Ctrl+Page Down**) or by using the zoom slider (see 2.3.7) in the *Print preview* panel. When zoomed in, the horizontal and vertical scroll bars allow you to scroll through the page.

The whole image can be enlarged or reduced with **Layout > Enlarge image size** () or **Layout > Reduce image size** ()


If a similarity matrix is available, it can be shown and printed with **Layout > Show similarity matrix** ()

The group information present in the *Groups* panel can be shown to the left with the option **Layout > Show group legend** ()


With **Layout > Show comparison information**, the name of the comparison (if already saved) and the number of entries are indicated on top of the first page. It is possible to display a header line with the database field names when **Layout > Show field names** is selected.


You can preview and print the image in full color with **Layout > Use colors** ()


13.3.9.3 Printing and exporting options

The preview can be printed using **File > Printer setup...** ()

The dialog box that appears is the standard Windows Print dialog box, allowing you to choose a printer and change the properties.

If the preview is covering more than one page, you can click on a specific page in the *Overview* panel to select a page from the range. With **File > Print this page** ()

the current page is printed. Use **File > Print all pages** () to print all pages at once.

If you want to export the image to another software package for further editing, use **File > Copy page to clipboard** ()

This pops up the *Copy comparison print to clipboard* dialog box (Figure 13.3.28).

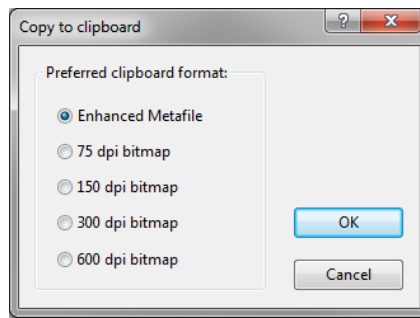


Figure 13.3.28: The *Copy comparison print to clipboard* dialog box.

The **Preferred clipboard format** can be selected. A choice is offered between the Windows **Enhanced Metafile** format, i.e. the standard clipboard exchange format between native Windows applications (default), or a **bitmap** file with **75 dpi**, **150 dpi**, **300 dpi** or **600 dpi** resolution. Many software applications, although supporting the enhanced metafile format, are unable to properly import some advanced BioNumerics clipboard files that make use of mixed vector, bitmap and (rotated) text components. If you experience such problems, you should select a bitmap file to be exported, or use another software application (or a more recent version of the same software) to import the graphical data.

With **File > Copy page to clipboard** (📄), only the current page is copied to the clipboard. If you want the whole image to be copied to the clipboard, first reduce the size of the image with **Layout > Reduce image size** (📐).



When preferred, the image of a fingerprint type can be shown and printed with a space between the gel strips. To do so, open the *Fingerprint type* window from the *Main* window and select **Layout > Show space between gelstrips**.

The print preview can be exported with **File > Export....** This calls the *Export image* dialog box.

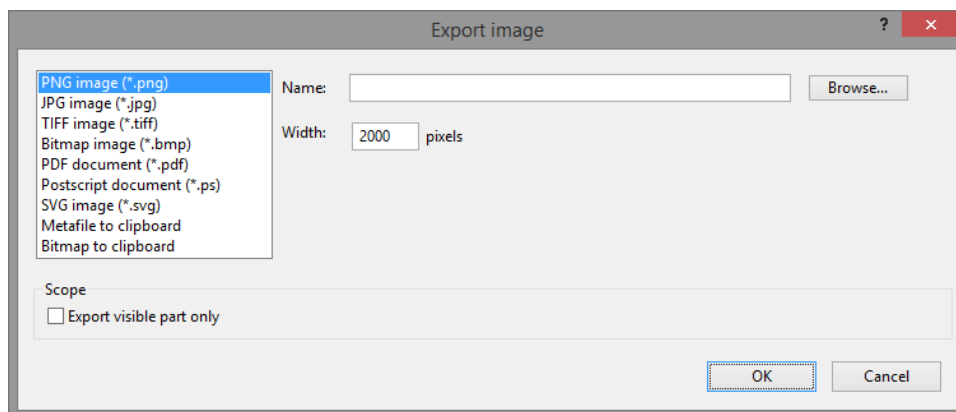


Figure 13.3.29: The *Export image* dialog box.

This dialog box allows you to export graphical information to a file or to the Windows clipboard in one of several available formats. In case a file is exported, a file **Name** should always be entered or browsed for via the **<Browse>** button. Exported files will open in their default editor. Information on the Windows clipboard can be pasted into other applications. Following export options are available:

- **PNG image (*.png)**: exports to a Portable Network Graphics (PNG) file. PNG is a raster graphics file format that supports lossless data compression. A **Name** and **Width** (in pixels) should be specified; the height will be determined automatically.
- **JPG image (*.jpg)**: exports to a Joint Photographic Experts Group (JPEG) file. JPEG or JPG is a

raster graphics file format that uses a lossy data compression. A **Name** and **Width** (in pixels) should be specified, as well as a **Quality** parameter. With the latter, a tradeoff can be obtained between storage size and image quality.

- **TIFF image (*.tiff)**: exports to a Tagged Image File Format (TIFF) file. TIFF is a raster graphics file format with optional lossless data compression. A **Name** and **Width** (in pixels) should be specified.
- **Bitmap image (*.bmp)**: exports to a BMP bitmap image or device independent bitmap (DIB) file. BMP is a raster graphics image file format used to store bitmap digital images, independently of the display device. A **Name** and **Width** (in pixels) should be specified.
- **PDF document (*.pdf)**: exports to a Portable Document Format (PDF) file. PDF is a file format used to present documents in a manner independent of application software, hardware, and operating systems. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **Postscript document (*.ps)**: exports to a PostScript (PS) file. PostScript is a computer language for creating vector graphics. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **SVG image (*.svg)**: exports to a Scalable Vector Graphics (SVG) file. SVG is an XML-based vector image format for two-dimensional graphics. A **Name** should be specified.
- **Metafile to clipboard**: copies the graphics as Windows enhanced metafile to the clipboard. Enhanced metafile is the standard clipboard exchange format between native Windows applications.
- **Bitmap to clipboard**: copies the graphics as a bitmap to the Windows clipboard. The **Width** (in pixels) should be specified.

13.3.9.4 Print templates

All the print layout settings specified in the *Comparison print preview* window can be saved to a print template. When selecting **File > Exit** in the *Comparison print preview* window a confirmation message appears, asking to save the current layout settings to a new print template (see Figure 13.3.30).

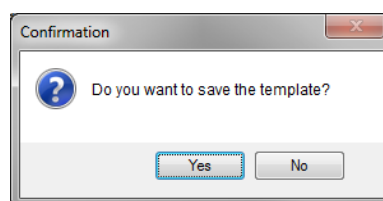


Figure 13.3.30: Confirmation message.

After confirmation, the *Save print template* dialog box appears (see Figure 13.3.31).

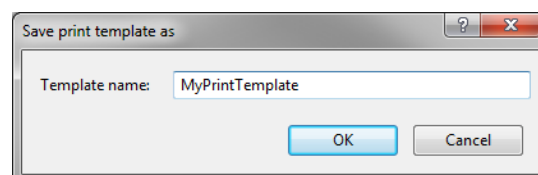


Figure 13.3.31: The *Save print template* dialog box.

This dialog prompts for the print **Template name**. Pressing <OK> saves the print template to the database.

All print templates defined in the current database are displayed in the Template drop-down list, available in the toolbar of the *Comparison print preview* window (see Figure 13.3.32). An existing print template can be loaded by selecting the template name from the list.

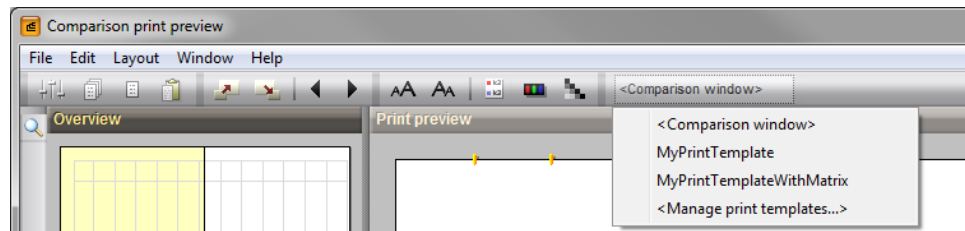


Figure 13.3.32: Template drop-down list.

To manage the print templates that are displayed from this drop-down list, select "<Manage print templates...>". This will display the *Print template* dialog box (see Figure 13.3.33).

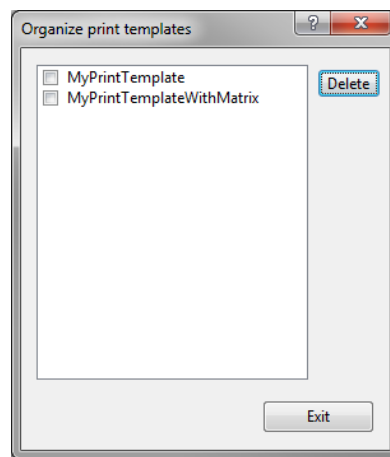


Figure 13.3.33: The *Print template* dialog box.

A selected print template can be deleted with *<Delete>*.

13.3.10 Displaying rendered trees

In publications and presentations, particularly in a phylogenetic context, a dendrogram is sometimes represented as a real tree with a stem and branches. Such representations can be achieved in BioNumerics using the *rendered tree* option in the *Comparison* window. This option should be used with care, as it will only produce acceptable pictures from a very limited number of entries and with fairly equidistant members.

Rendered trees can be created from a standard rooted tree in the *Comparison* window as well as from unrooted phylogenetic trees (Maximum Parsimony and Maximum Likelihood).

In the *Comparison* window, create a dendrogram containing a small and not too heterogeneous group of entries (e.g. 10 entries) and select **Clustering > Display rendered tree**. This pops up the *Rendered tree settings* dialog box (Figure 13.3.34).

The dialog box prompts for a number of settings:

Hide branches if shorter or equal to allows all entries that are very similar to be grouped together at one branch tip. This allows simpler trees to be produced and may avoid star-like branch tips to occur.

Hide distance labels if shorter or equal to sets a minimum for the distance values to be shown on the

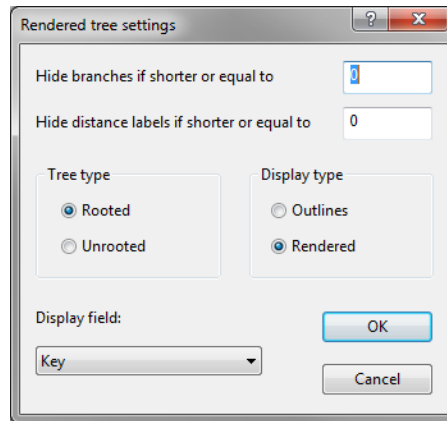


Figure 13.3.34: The *Rendered tree settings* dialog box.

branches. If many short distances occur, the labels may overlap, which can be avoided by only allowing larger distance to be shown. If the value is set to 100 (or more), no distance labels will be shown.

The **Tree type** can be **Rooted** or **Unrooted**. If the dendrogram is rooted by nature (e.g. UPGMA) it makes no sense to display an unrooted tree from it.

The **Display type** can be **Rendered**, i.e. with a thicker stem and smooth, gradually narrowing branches, or **Outlines**, where stem and branches are represented by straight lines.

From the pull-down list under **Display field**, one of the available database fields can be chosen for display on the rendered tree.



In case the information fields are too long, it is possible to replace them by a group code if comparison groups are defined. The use of group codes is explained in 13.3.4.

Pressing <OK> will display the current dendrogram in the *Comparison* window as a rendered tree in a separate *Rendered tree* window (Figure 13.3.35).

It is possible to zoom in and out on a page using **Layout > Zoom in** (🔍+) and **Layout > Zoom out** (🔍-). When zoomed in, the horizontal and vertical scroll bars allow you to scroll through the page.

The thickness of the tree branches can be adjusted with **Layout > Make tree thicker** (🌳) or **Layout > Make tree thinner** (🌳).

The size of the font, which is used for display of entry and distance labels, can be increased with **Layout > Increase font size** (A+) or decreased with **Layout > Decrease font size** (A-).

Branches of a rendered tree can be re-located by clicking on a branching point and dragging the branch to a different position.

The rendered tree can be exported to the clipboard as a Windows metafile with **File > Copy to clipboard (metafile)** or as a bitmap with **File > Copy to clipboard (bitmap)** and then pasted into another application. Alternatively, the rendered tree can be printed directly with **File > Print image...**

Rendered trees can also be displayed from parsimony and maximum likelihood trees (see 16). If a *rooted* rendered tree is exported, the highlighted branch of the unrooted tree will be used to create the root.

13.3.11 Analysis of the congruence between techniques

As soon as multiple techniques are used to study the relationships between organisms, the question arises how congruent the groupings obtained using the different techniques are. It is also interesting to compare the techniques by the level at which they discriminate the entries, in other words, the taxonomic depth of

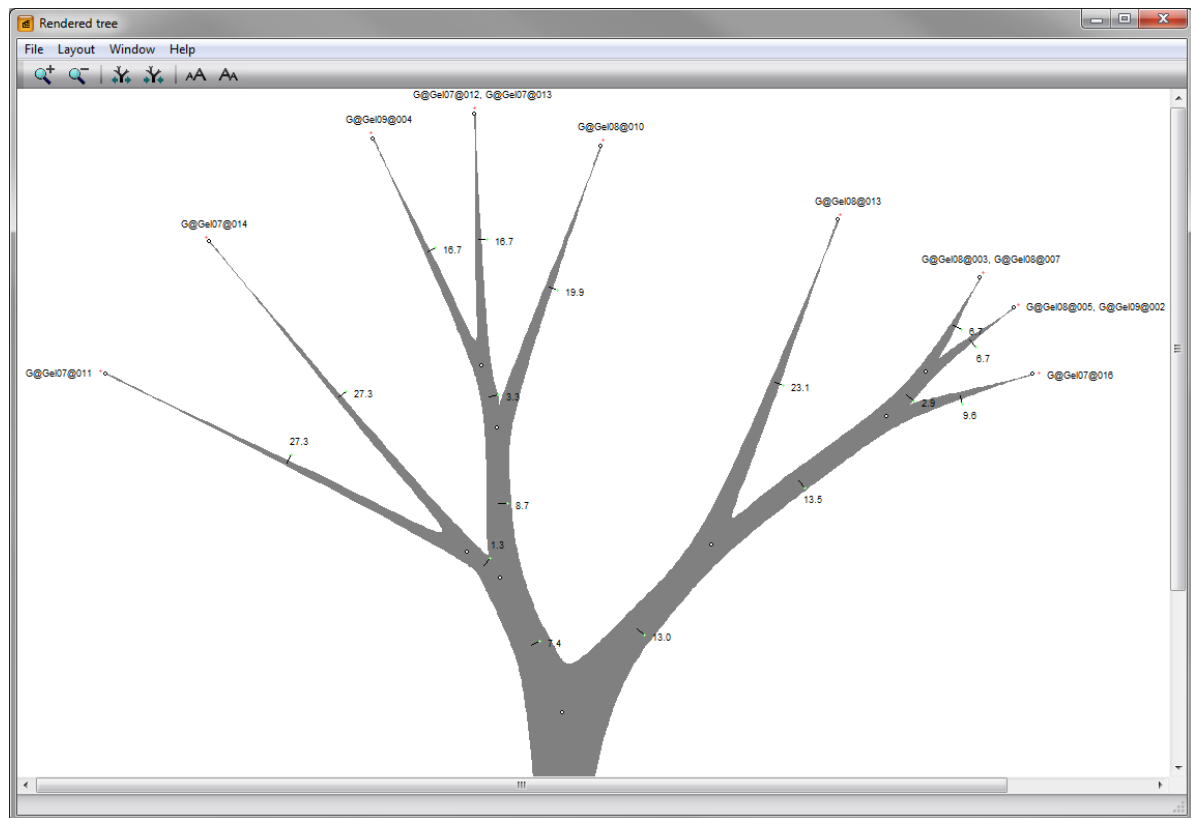


Figure 13.3.35: The *Rendered tree* window.

the techniques.

An evident way to perform such a study is by comparing the similarity matrices obtained from the different experiment types used. By plotting the corresponding similarity values in an X-Y coordinate system, one can easily observe the kind and degree of concordance at a glance. BioNumerics even calculates a regression curve through the plot.

In the *Comparison* window, with a cluster analysis present for each experiment type that you want to analyze the congruence for, select **Clustering > Congruence of experiments...** to display the *Experiment congruence* window (see Figure 13.3.36).

This window shows both a matrix of congruence values between the techniques (experiment types) and a dendrogram derived from that matrix. In addition to the correlation between techniques, the standard deviation on the values can be shown, as well as the significance of the correlation (green values).

The congruence matrix can be exported as an enhanced metafile using **File > Copy image to clipboard**, or printed directly with **File > Print image...**

A tab-delimited text export of the congruence matrix can be achieved with **File > Export similarity values**.

To edit the settings used to calculate the congruence between experiments, select **Calculate > Experiment correlations....** The *Correlation between experiments* dialog box pops up (Figure 13.3.37).

This dialog box shows the settings for the calculation of correlation between the experiment types.

Under **Correlation type**, two methods are available to calculate the congruence between two experiment types. The default method is by using the Pearson product-moment correlation coefficient (**Pearson correlation**). An alternative coefficient is **Kendall's tau**. The principle of **Kendall's tau** is as follows: if value A is higher than value B in experiment 1, then corresponding value A of experiment 2 should also be higher than corresponding value B of experiment 2. The less infringements on this statement, the more congruent

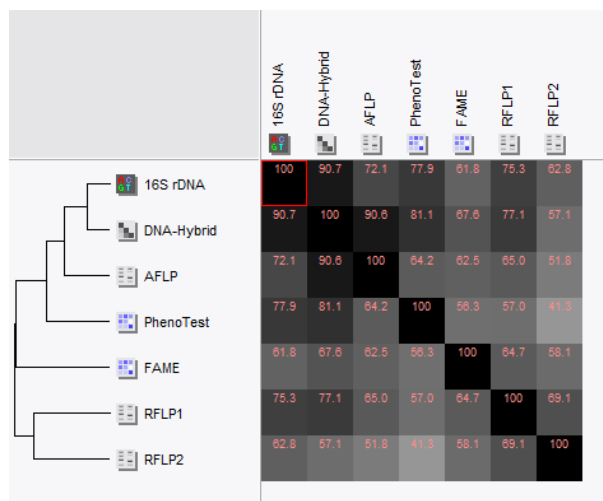


Figure 13.3.36: The *Experiment congruence* window. The congruence between experiments is calculated using the Pearson product moment correlation coefficient (default).

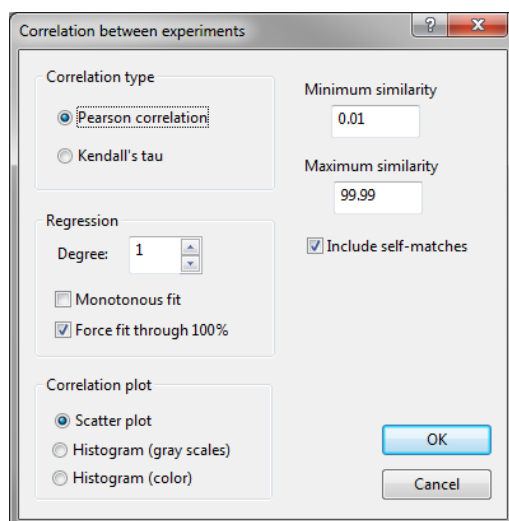


Figure 13.3.37: The *Correlation between experiments* dialog box.

the techniques are. **Kendall's tau** has the advantage over **Pearson correlation** that non-linear correlations still have significant scores.

The **Minimum similarity used** and **Maximum similarity used** allow a range of similarity values to be specified within which the analysis is done. Normally, one can enter 0% and 100%, respectively, for these values. **Include self-matches** is an option which gives the user the choice whether to include entries compared with themselves. Obviously, self-matches are always 100% and may thus influence the correlation obtained between two experiment types.

The **Regression** determines the kind of best-fitting curve that is calculated through a *Similarity plot* of two experiment types in the *Similarity plot* window. You can enter the **Degree** of the regression (first degree is linear, second degree is a quadratic function, etc.). If there is any concordance between techniques, one should expect that the function increases monotonously; with **Monotonous fit**, only such functions are allowed. With **Force through 100%**, the program will force the regression curve to pass through 100% for both techniques. In other words, if entries are seen identical in one technique, you would expect that they are seen identical in another technique as well. If you do not wish to see the regression in the similarity plot, enter 0 as **Degree**.

Under **Correlation plot**, you can choose **Scatter plot** to plot each pair of similarity values as one dot in a similarity plot between two experiment types. Especially for very large data sets resulting into dense scatter plots, it can be useful to average the number of dots in a given area and represent that average rather than the individual pairs. This can be achieved with **Histogram (gray scales)** and **Histogram (color)**. When color is chosen, a multi color scale is used that ranges continuously from white over blue, green, yellow, orange, and red to black.

When **<OK>** is pressed, the *Experiment congruence* window is updated to display the correlation values calculated using the new settings.

Click on a value in the similarity matrix and select **Calculate > Similarity plot**. Alternatively, just double-click the value. A *Similarity plot* window appears for the two selected experiments (Figure 13.3.38), with a regression curve fitted through the dots.

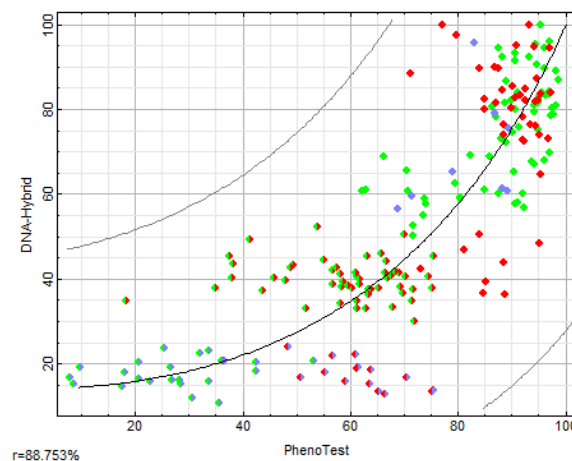


Figure 13.3.38: A *Similarity plot* window for two experiments with group colors shown.

This window displays an X-Y plot of the similarity values between each pair of entries in the *Comparison* window for the two experiment types.

Excluded values (due to **Minimum similarity used** and **Maximum similarity used** in the *Correlation between experiments* dialog box) are shown in gray.

While hovering the mouse pointer over the plot, it becomes a lasso selection tool. You can make a selection of dots by dragging the mouse over the plot. The corresponding database entries are selected. The selection status of the database entries is displayed as follows on the plot:

- Dots resulting from two selected entries are displayed in blue;
- Dots resulting from one selected entry and one non-selected entry are displayed in red;
- Dots resulting from two non-selected entries are displayed in black.



In a comparison of n entries, every entry forms a dot with $n - 1$ other entries. As such, if you select one dot, $n - 1$ other dots will appear in red. For the same reason, a restricted selection of dots on the plots can easily result in all dots becoming either red or blue (and all entries being selected).



You can click on any dot in the similarity plot to pop up a detailed pairwise comparison between the two entries (see 13.3.3).

When comparison groups (see 13.3.4) are present in the underlying *Comparison* window, they can be displayed on the similarity plot by selecting **File > Show group colors**. Since each dot is composed of two entries, the dots may be composed of two colors (see Figure 13.3.38).



The similarity plot can be exported as an enhanced metafile using **File > Copy image to clipboard**, or printed directly with **File > Print image...**

13.3.12 Identifying unknown entries

Comparisons can be used as a quick tool for identification, simply by pasting unknown entries in an existing comparison. More powerful and feature-rich identification methods available in BioNumerics are discussed in 15.

The unknown entries need to be selected first (see 3.2.4). Next, the selection can be copied to the clipboard with **Copy selection** ( from the *Main* window. The selection can then be pasted in an open comparison with **Edit > Paste selection** (, **Ctrl+V**).

If a dendrogram is present, it will be automatically recalculated. A tentative identification could be performed by observing the cluster(s) in which the unknown entries occur.

Additionally, entries in the comparison can be arranged according to their similarity with the highlighted entry by selecting **Edit > Arrange entries > Arrange entries by similarity** (). The highlighted entry now stands on top and all the other entries in the comparison are arranged by decreasing similarity with that entry. The similarity values are shown in the *Similarities* panel. The **Edit > Arrange entries > Arrange entries by similarity** () command can be repeated for each experiment type in the *Experiments* panel, in order to compare the different results. The program uses the default similarity coefficient, as specified in the corresponding experiment type window. A printout of the list of similarity values can be obtained with **File > Print database fields...** An export file of the similarity values is created with **File > Export > Export database fields...**



In case of a fingerprint type, you can also show the number of different bands between a highlighted entry and the other entries, by selecting **Number of different bands** as the default similarity coefficient (see 4.2.1).

13.3.13 Exporting data from a comparison

The command **File > Export > Export database fields...** exports all entry field information, as displayed in the *Information fields* panel.

Several experiment types can provide data as a character aspect. This includes character types and sequence types, but also e.g. fingerprint types after a band matching was performed (see 4.3).

This character data can be exported as text for the active experiment type with **File > Export > Export character data...** This action calls the *Export character data* dialog box (see Figure 13.3.39).

A text or symbol to indicate a missing character value can be provided in the **Export absent values as** text box. By default, this is a question mark (?).

Three options are available to specify how character data are exported:

- **Export values:** The quantitative character values are exported.
- **Export binary values:** Data are exported as presence/absence matrices (1 or 0).
- **Export mapped values:** The character mappings (if available) are exported.

With **Export character mapping (if present)** checked, the character mappings will be exported instead of the character values if a mapping is present. For example, nucleic sequence will be exported with their letter

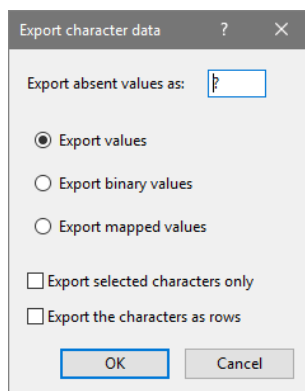


Figure 13.3.39: The *Export character data* dialog box.

notation (e.g. ATGC) instead of the integer numbers that BioNumerics uses internally. When no mapping is provided or when the option is unchecked, the character values will be exported.

By default, all characters from the active experiment / aspect are exported. Checking ***Export selected characters only*** exports only the selected characters, i.e. those indicated with a colored triangle in the header of the *Experiment data* panel.

The default export file format has the entries organized in rows and the characters in columns. By checking ***Export the characters as rows***, the data are transposed, i.e. the characters appear in rows and the entries in columns. This can be particularly useful for exporting large character sets, since most spreadsheet programs can display more rows than columns.

Pressing <OK> will export the characters and the display field for entries (most often the key) to a `export.csv` or `export.txt` file, depending on the ***Export table files in CSV format*** preference (see 2.3.3.2).

For the other export options, we refer to the Chapters that deal with the corresponding experiment types:

- For ***File > Export > Export bands...*** and ***File > Export > Export densitometric curves...***, see 4.2.5.
- For ***File > Export > Export sequences (tabular)...*** and ***File > Export > Export sequences (formatted)...***, see 8.3.14.
- For ***File > Export > Export similarity matrix...***, see 13.3.2.

Part 14

Charts

Chapter 14.1

Introduction

The *Charts and statistics* window provides a uniform charting interface that is accessible from various applications throughout the software. The *Charts and statistics* window needs a *data source* as input: a table of which the rows are the chart elements and the columns are the *properties* (see Figure 14.1.1). These properties can be values, strings, dates, colors, or booleans. Dates are technically treated as numerical values but some charts can represent them in true date format. The chart types that can be generated depend on the properties selected. For example, Figure 14.1.1 displays a data source table derived from the *Main* window, containing the entries as records and the database fields as properties. Some of these database fields are string-based (species, country), while others are numerical (virulence, size). The properties from the data source that are used within a specific chart are called the *components* of the chart. Figure 14.1.1 illustrates how different chart types are obtained by choosing or combining different field types as components:

- Choosing one string property as input (e.g. the species string) results in a **Frequency bar graph** where the categories are composed of entries that belong to the same species.
- When two string properties are combined, the result is a **Contingency table** with the number of entries (*frequencies*) displayed for every combination of categories defined by these strings (e.g., species and country in the figure).
- A combination of one string property and a numerical property (e.g., country and virulence) results in an **ANOVA chart** displaying the spread and standard deviation of the numbers for each of the categories present in the string property.
- When two numerical properties are combined, the result is a **Scatter chart** (e.g., virulence and size), plotting the entries in an X-Y chart with the two properties as axes.
- From one single numerical value (e.g., size), a **Profile chart** or **Bar graph** can be generated, displaying the size for the entries.

Although the examples illustrated in Figure 14.1.1 are among the most commonly used chart types, BioNumerics offers a number of additional chart types, which are discussed in detail further in this chapter.

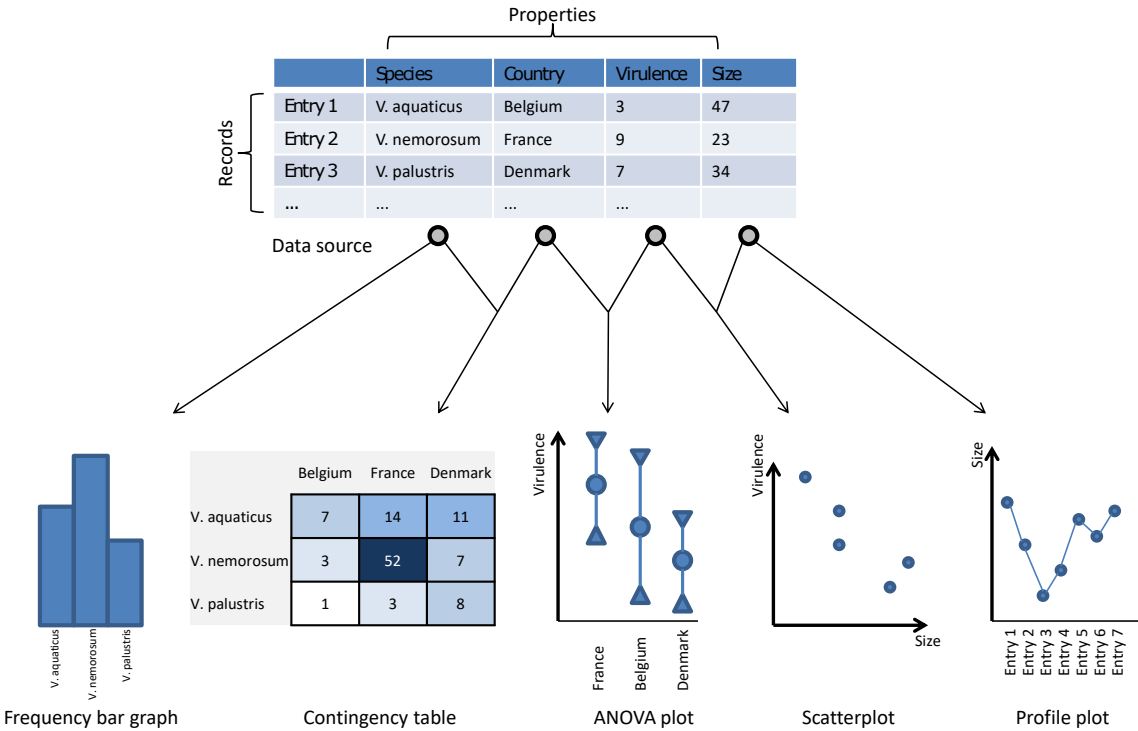


Figure 14.1.1: The data source and resulting charts.

Chapter 14.2

Chart components

As mentioned before, the chart components can be taken from string properties and numerical properties drawn from experiment data or entry information fields to populate the chart, but they can also be used for adding additional properties to the chart, for example the labeling of the axes, labeling of the chart elements, or colors, sizes and symbols of the chart elements. For such purposes, other properties such as strings, field states, group colors, can be used as well. In Figure 14.2.1 some examples of additional labeling are applied to the charts from Figure 14.1.1. Chart (A) is the profile chart of **size** for all entries with the species names shown as different colors. Chart (B) is a scatter chart of **virulence** versus **size**, with dot size determined by an additional numerical property. In addition, the entry labels were shown next to the dots. In (C), the ANOVA chart of virulence per country is shown with the individual entries displayed in color according to the species name.

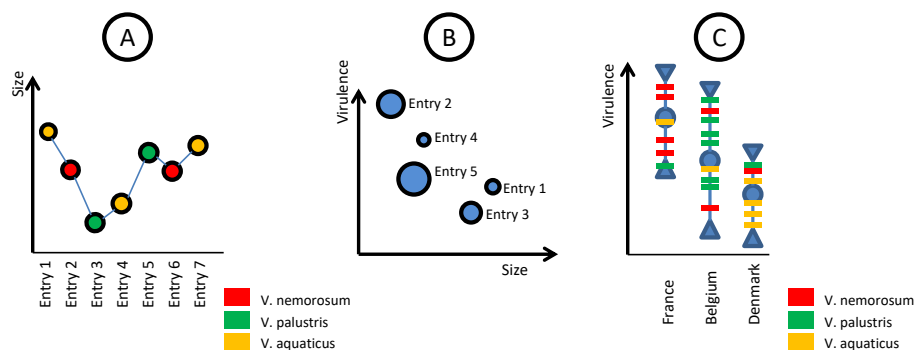


Figure 14.2.1: Examples of charts with additional components: (A) Profile chart with colored dots according to species name; (B) Scatter chart with dot size determined by an additional numerical property; (C) ANOVA chart with entries displayed in color according to species.

The number of chart types along with the number of combinations of additional components used to decorate the charts make it possible to generate a virtually endless number of chart visualizations, even for simple data sets as used in Figure 14.1.1.

Chapter 14.3

Creating a chart in BioNumerics


14.3.1 Introduction

A *Charts and statistics* window can be generated from different BioNumerics windows. The data sources available, and hence the chart types that can be generated, depend on the window from which the *Charts and statistics* window was launched. In this respect, it deserves attention that charts can look at the BioNumerics database information in two ways: depending on the data source, either the entries or the properties can form the elements of the chart. In the example of Figure 14.1.1, the chart elements (records) are database entries (samples, strains,...) and the chart components are made up of character values or information strings for these entries. Figure 14.3.1 shows an example of an inverse data matrix, where the characters form the data records, and the selected entries form the properties. The figure shows two chart types derived from this data source: a profile chart depicting all characters for a selected entry, and a scatter chart comparing all characters between two entries. In both examples, the chart elements are the characters.

The *Charts and statistics* window can be launched from the BioNumerics windows discussed below.


14.3.2 Creating charts from the BioNumerics main window

The data source is given by the set of entries and the associated information fields. The data source looks as in Figure 14.1.1: the individual elements (records) of the chart are database entries and the chart components are selected from the information fields. The chart is always created from the *selected entries* in the *Main* window.

In the *Main* window, the *Charts and statistics* window is launched with **Analysis > Chart and statistics...** (, **F7**), which opens the *Create chart* dialog box (see 14.3.6).

14.3.3 Creating charts from the Entry edit window

The data source is determined by one of the experiments that are attached to the entry and as such, it is character-based as in Figure 14.3.1. The elements of the resulting *Charts and statistics* window are experiment data for the selected entry. The most obvious resulting chart types are bar graphs or profile charts.

In the *Entry* window, the *Charts and statistics* window is launched with **File > Charts and statistics...** (, **F7**), which opens the *Create chart* dialog box (see 14.3.6).

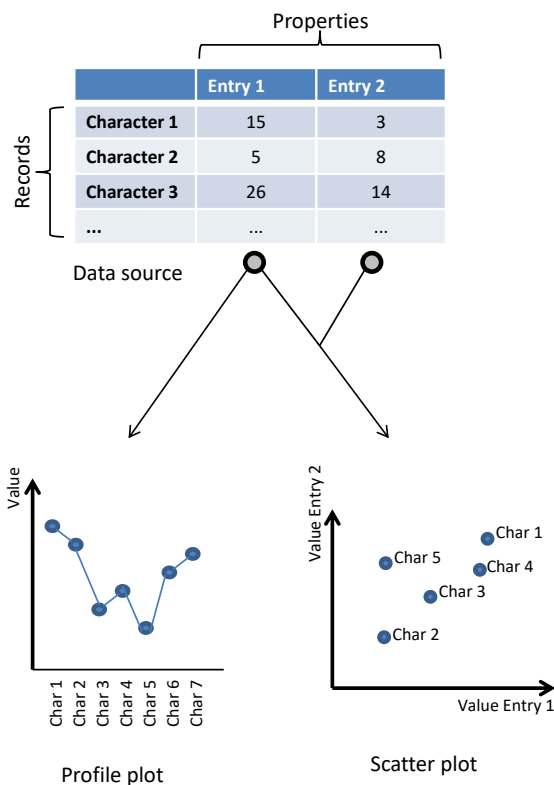


Figure 14.3.1: Data source with characters as records and two resulting charts: a profile chart for one entry and a scatter chart for two entries.

14.3.4 Creating charts from the Pairwise comparison window

Similar as in the *Entry* window, the data source is determined by one of the experiments that are attached to the entries and as such, it is character-based as in Figure 14.3.1. The elements of the resulting *Charts and statistics* window are experiment characters or curves for the two compared entries. The most obvious resulting chart types are scatter charts and profile charts combining the two entries.

In the *Pairwise comparison* window, the *Charts and statistics* window is launched with **File > Charts and statistics...** (📊, F7), which opens the *Create chart* dialog box (see 14.3.6).

14.3.5 Creating charts from the Comparison window

This window provides the richest data sources for the *Charts and statistics* window, including an entry-based source for all entries present in the comparison and character-based data sources for all experiments defined in the database. In the *Comparison* window, a third type of data sources is available: the **similarity data**. The records for this data source are not entries but **pairs of entries** and the properties are **similarities** calculated for those pairs using the available experiment data (see Figure 14.3.2). Similarity-based charts can be generated for those experiments for which a similarity matrix is available. Figure 14.3.2 shows an example of a profile chart of similarities for one selected experiment type and a scatter chart of similarities between two experiment types.



The source type "similarity data" can also hold dendrogram information such as branch length, distance from root, branch quality, member count etc..

In the *Comparison* window, the *Charts and statistics* window is launched with **Statistics > Chart and statistics...** (📊, F7), which opens the *Create chart* dialog box (see 14.3.6).

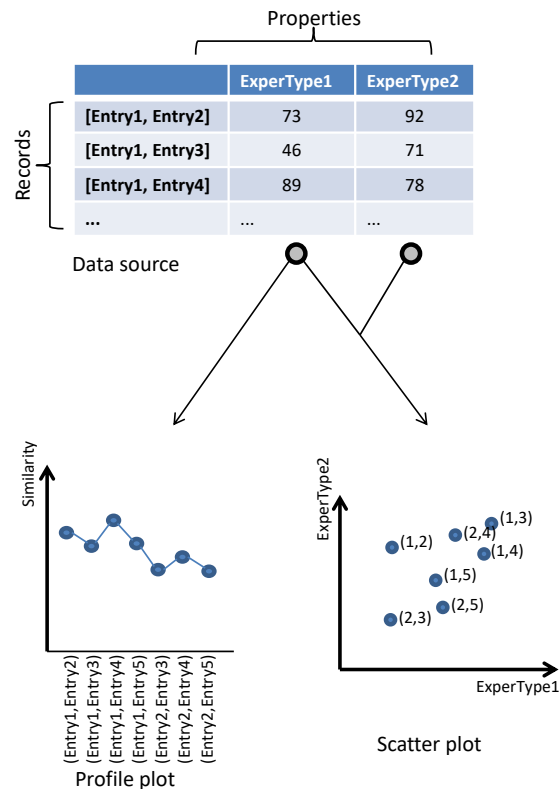


Figure 14.3.2: Data source with entry pairs as records and similarity values from different experiment types as properties. Two resulting charts are a profile chart displaying similarity values for a single experiment type and a scatter chart comparing similarity values between two experiment types.

14.3.6 Create chart dialog box

When the *Charts and statistics* window is launched from either of the above mentioned windows, the *Create chart* dialog box pops up (see Figure 14.3.3).

This dialog box pops up when the *Charts and statistics* window is launched in BioNumerics and allows the *data source* to be selected from a tree. Depending on the window it is launched from, different options are available.

- *Main* window: The *Create chart* dialog box contains only one choice: to create a chart from the currently selected entries.
- *Entry* window and *Pairwise comparison* window: The *Create chart* dialog box shows a tree of experiments categorized by type. The *Charts and statistics* window will have the selected experiment as data source.
- *Comparison* window: The *Create chart* dialog box shows a tree listing three types of data sources: **Comparison entries**, **Similarity data**, and **Experiment character values** (Figure 14.3.3). Under **Similarity data**, you can choose **Create from 'Similarity values'** and **Create from 'Current dendrogram'** (see 14.3.5). Under **Experiment character values**, the experiments are categorized by type. The *Charts and statistics* window will have the selected experiment as data source.

In case one or more chart templates have been saved (see 14.5.8), an additional item **Existing chart templates** appears in the data source tree. This item contains the saved chart templates.

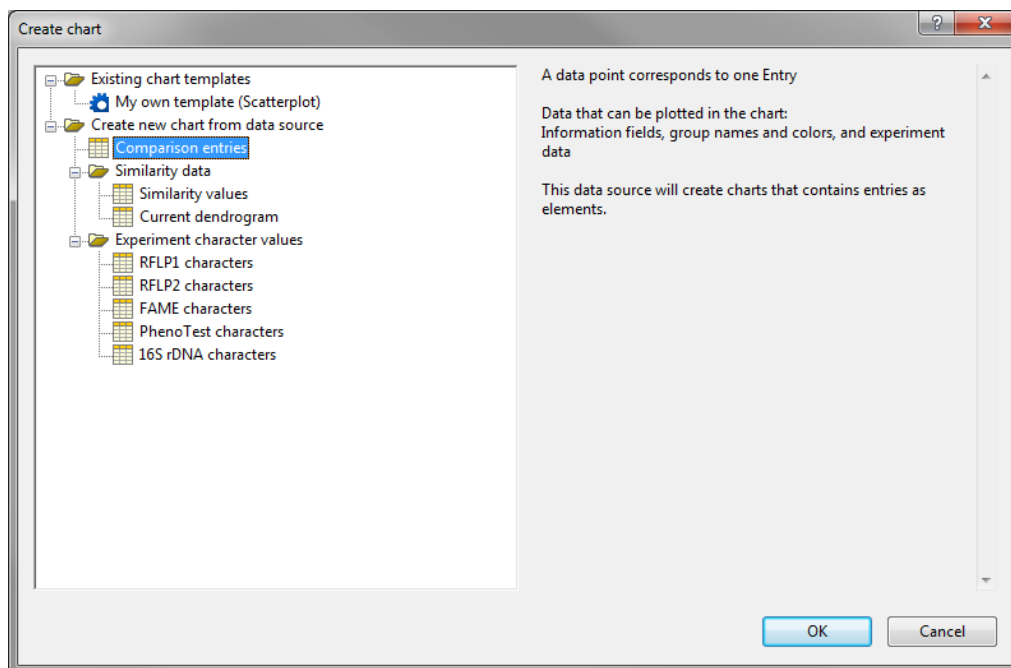


Figure 14.3.3: The *Create chart* dialog box launched from the *Comparison* window.

When pressing the **<OK>** button, the *Create chart* wizard will open (see 14.3.7) together with the *Charts and statistics* window (see 14.5).

14.3.7 Create chart wizard

The first step of the *Create chart* wizard (see Figure 14.3.4) is displayed after having selected the data source in the *Create chart* dialog box. The same wizard is called when selecting **Plot > Create plot wizard...** (🔧) in the *Charts and statistics* window.

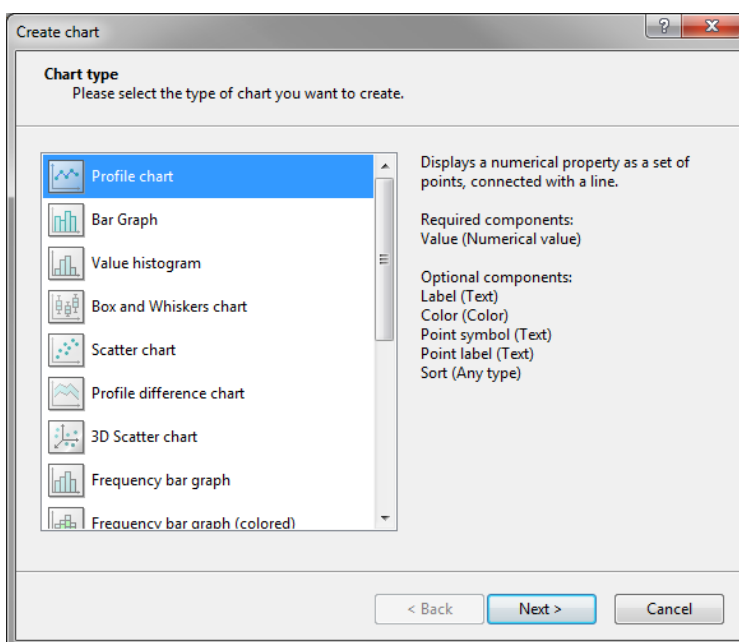
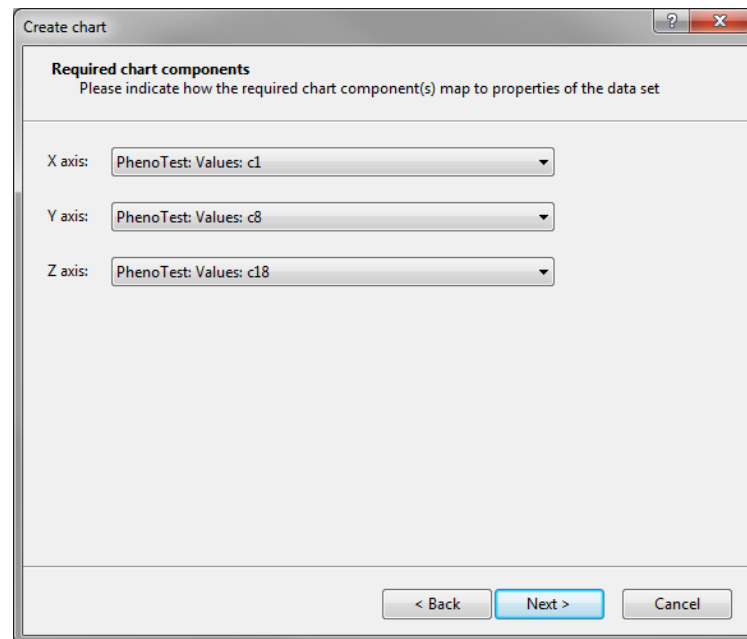


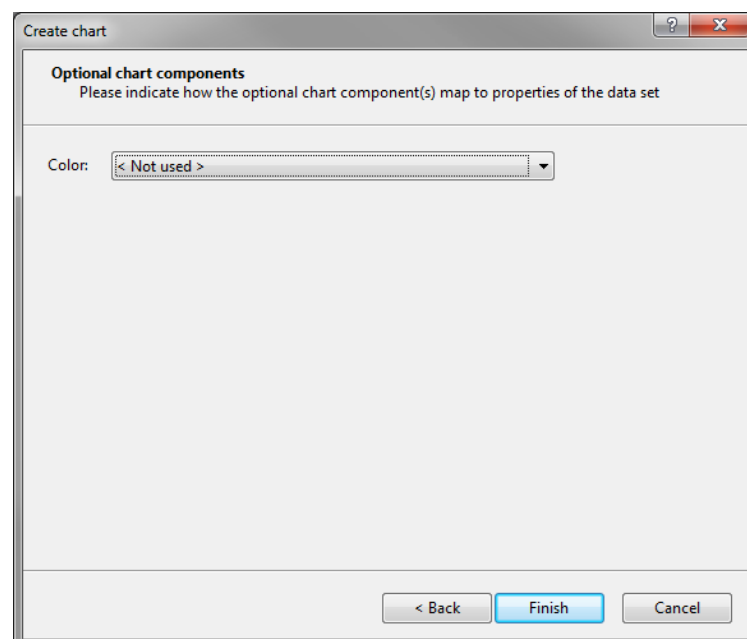
Figure 14.3.4: The *Chart type* wizard page.

The *Chart type* wizard page displays a list of all available chart types. For the selected chart type, a short description is shown on the right, along with the required components and the optional components, which are used to decorate the chart. For a detailed description of the chart types, see [14.4](#).



The screenshot shows a dialog box titled "Create chart" with a standard Windows window control bar (minimize, maximize, close). The main content area is titled "Required chart components" and includes a subtitle: "Please indicate how the required chart component(s) map to properties of the data set". Below this, there are three labeled dropdown menus: "X axis:" with the value "PhenoTest: Values: c1", "Y axis:" with the value "PhenoTest: Values: c8", and "Z axis:" with the value "PhenoTest: Values: c18". At the bottom of the dialog, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 14.3.5: Required Chart components.



The screenshot shows a dialog box titled "Create chart" with a standard Windows window control bar. The main content area is titled "Optional chart components" and includes a subtitle: "Please indicate how the optional chart component(s) map to properties of the data set". Below this, there is a single labeled dropdown menu: "Color:" with the value "< Not used >". At the bottom of the dialog, there are three buttons: "< Back", "Finish" (highlighted in blue), and "Cancel".

Figure 14.3.6: Optional Chart components.

In the second and third step of the wizard, the parameters for the selected chart need to be specified. For a detailed overview of all chart types and their components, see [14.4](#).

Most chart types require one or more *categorical data sets* and/or one or more *numerical data sets* (values) as components.

- *Categorical data sets* are based on strings, from which the categories are derived using string identity

as criterion. Any string-type information in the database can be used as categories, including string information fields, group names, character names, binary conversions for characters ("No" and "Yes") and character mappings.

- *Numerical data sets* are usually taken from numerical information fields and character values, but can also be similarity values and dendrogram properties.

Besides strings and values, a few other property types can be used in charts. With a few exceptions, these additional property types are used to define additional optional components, such as labels, colors, sizes, filtering, etc..

- *Colors* can be taken from field states, comparison groups and experiment characters. In the latter case, the colors defined in the character experiment are applied. Colors are used to add a dimension to a chart, see e.g. Figure 14.2.1.
- *Dates* can be taken from date-type information fields. They can be used as an alternative for a value in scatter charts and 3D scatter charts (see further).
- *Booleans* can only be obtained as a derived property (see 14.5.3). They can be used as a filter to restrict the chart to those items meeting a specific condition.

It is also possible to select the components in the *Data source overview* panel before creating a chart. This way BioNumerics will assist you in choosing a chart that is compatible with the selected data sources. The *Add new chart* wizard that will help you in this decision is called with **Plot > Add new plot from selected properties...** (+).

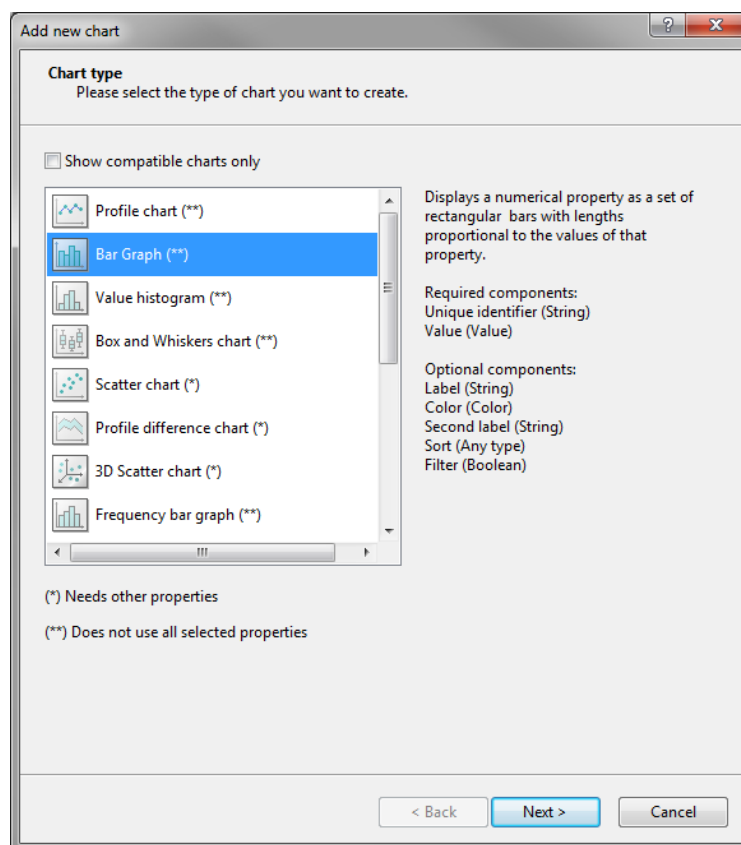


Figure 14.3.7: The *Chart type* wizard page.

The *Chart type* wizard page displays a list of all available chart types. For the selected chart type, a short description is shown on the right, along with the required components and the optional components, which are used to decorate the chart.

When the option **Show compatible charts only** is checked, only those charts that require no other properties as components and that can use all the selected properties as components will be displayed. If this option is not checked, the chart types that need other properties than the ones selected are marked with (*) while those chart types that do not use all selected properties are marked with (**).

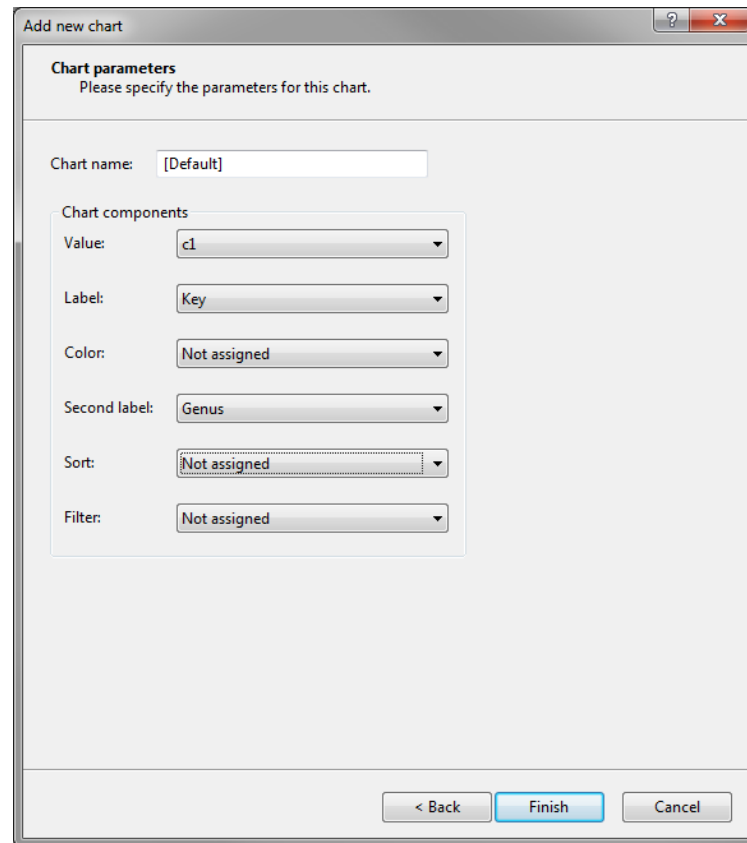


Figure 14.3.8: Required Chart components and optional Chart components.

In the second step of the wizard, the parameters for the selected chart need to be specified. For a detailed overview of all chart types and their components, see [14.4](#).

Chart types and components

14.4.1 Profile chart

14.4.1.1 Profile chart type

Displays a numerical property as a set of points, connected with a line (Figure 14.4.1).

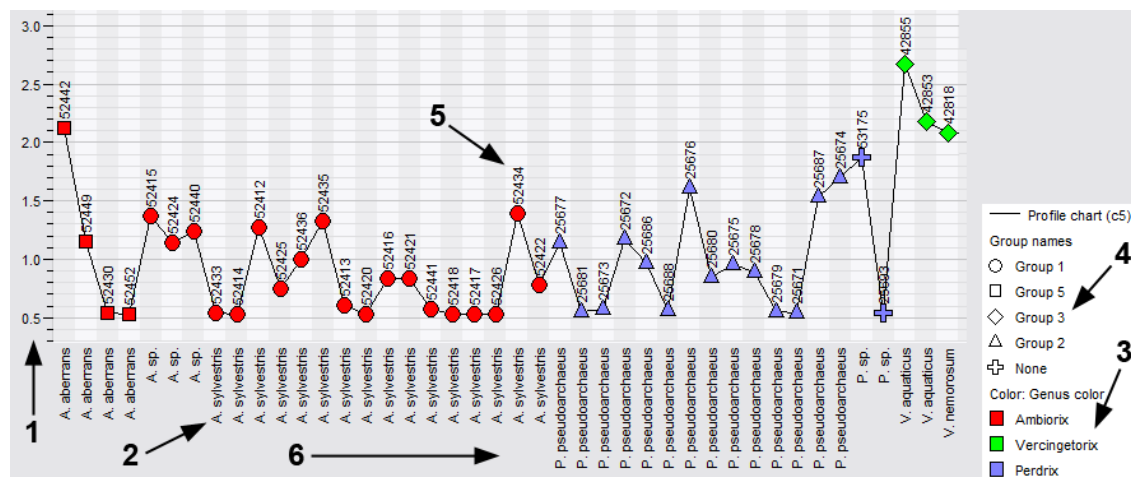


Figure 14.4.1: Profile chart illustrating component usage. 1: Value, 2: Label, 3: Color, 4: Point symbol, 5: Point label, 6: Sort.

14.4.1.2 Profile chart components

This chart type uses the following components (see also Figure 14.4.1):

Unique identifier A unique identifier determines the database components that will form the elements of the chart. The program sets the unique identifier automatically (usually keys or character names). This component should not be changed by the user.

Value A numerical property that is used to create the chart.

Label (optional): a string property to label each point in the profile chart.

Color (optional): a color property to color the points in the profile chart.

Point symbol (optional): a string property from which categories are derived that are represented as different symbols.

Point label (optional): a string property that can be used to place a second label on top of the points in the profile chart.

Sort (optional): a string or numerical property that can be used to sort the elements in the chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.1.3 Profile chart properties

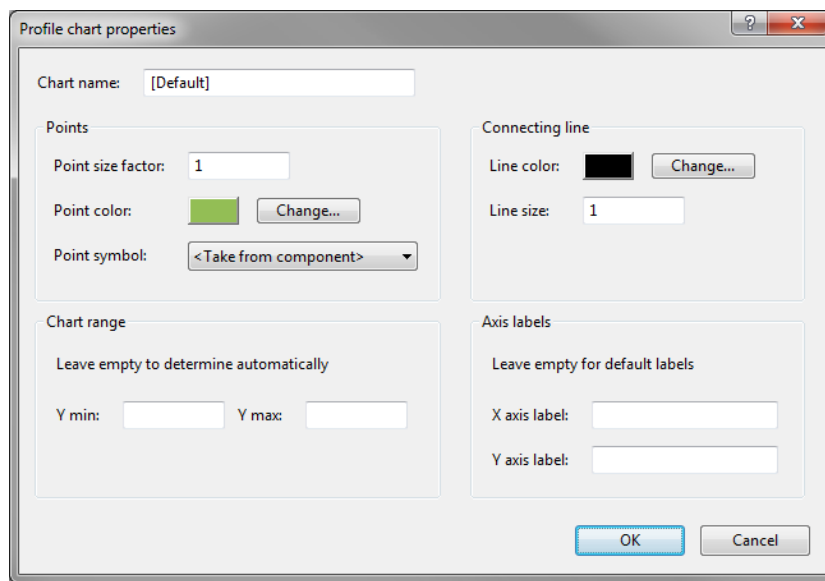


Figure 14.4.2: The *Profile chart properties* dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (⚙️). This opens the *Profile chart properties* dialog box (see Figure 14.4.2).

The following options can be changed:

Points To change the style of the points on the chart.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With **Change**, a point color can be picked from the RGB color table. If a color property is used as component, the point color specified here is shown as a bordering color around the symbols.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk. If string property from the data source is used as symbol component, this should be set to **Take from component**.

Connecting line To change the style of the line connecting the points on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With **Change**, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.2 Bar Graph

14.4.2.1 Bar Graph type

Displays a numerical property as a set of rectangular bars with lengths proportional to the values of that property (Figure 14.4.3).

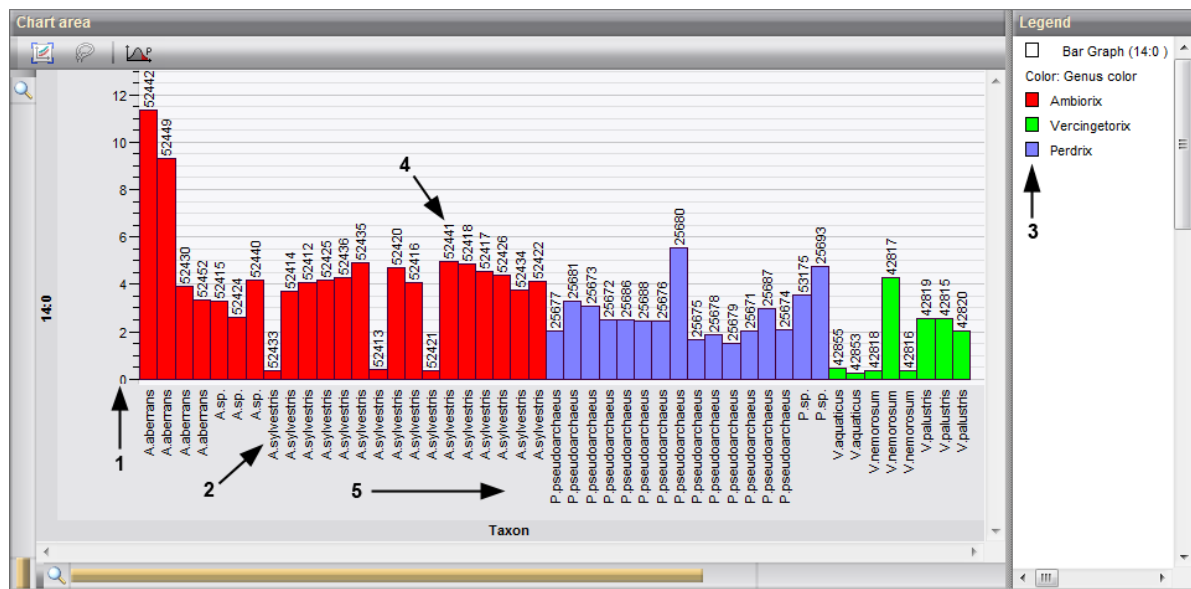


Figure 14.4.3: Bar graph illustrating component usage. 1: Value, 2: Label, 3: Color, 4: Second label, 5: Sort.

14.4.2.2 Bar Graph components

This chart type uses the following components:

Unique identifier A unique identifier determines the database components that will form the elements of the chart. The program sets the unique identifier automatically (usually keys or character names). This component should not be changed by the user.

Value A numerical property that is used to create the chart.

Label (optional): a string to label each bar in the bar graph.

Color (optional): a color property to color the bars in the bar graph.

Second label (optional): a string property that can be used to place a second label on top of the bars in the bar graph.

Sort (optional): a string or numerical property that can be used to sort the elements in the chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.2.3 Bar Graph properties

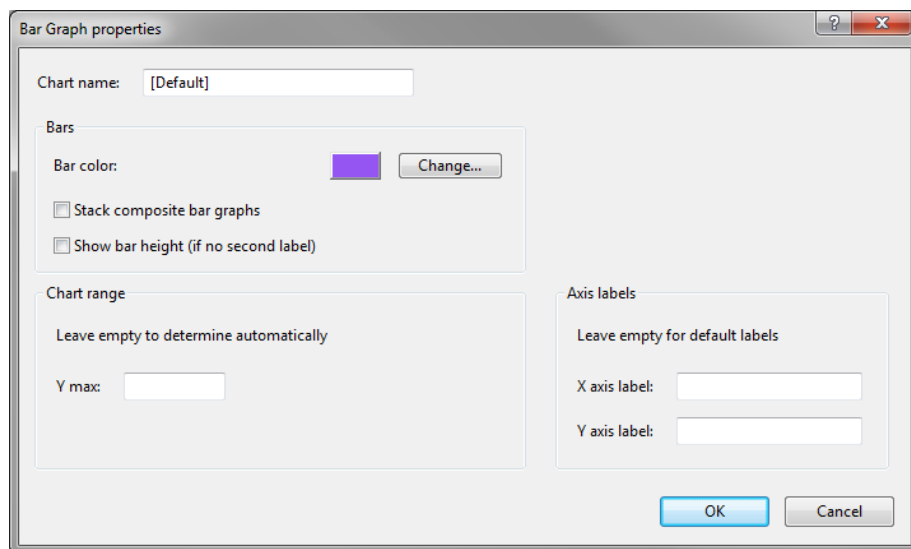


Figure 14.4.4: The *Bar Graph properties* dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (⌘+T). This opens the *Bar Graph properties* dialog box (see Figure 14.4.4).

The following options can be changed:

Bars To change the style of the bars on the chart.

Bar color Allows a color to be defined for the bars, e.g., to differentiate this chart from other charts. With **Change**, a bar color can be picked from the RGB color table. If a color property is used as component, the bar color specified here is shown as a bordering line.

Stack composite bar graphs When multiple bar graphs are created from the same unique identifiers, this option allows the bars to be stacked instead of being shown next to each other.

Show bar height (if no second label)

Chart range To define the range of the chart manually.

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.3 Value histogram

14.4.3.1 Value histogram type

Visualizes the distribution of a numerical property by plotting the frequencies over a set of binned values (Figure 14.4.5).

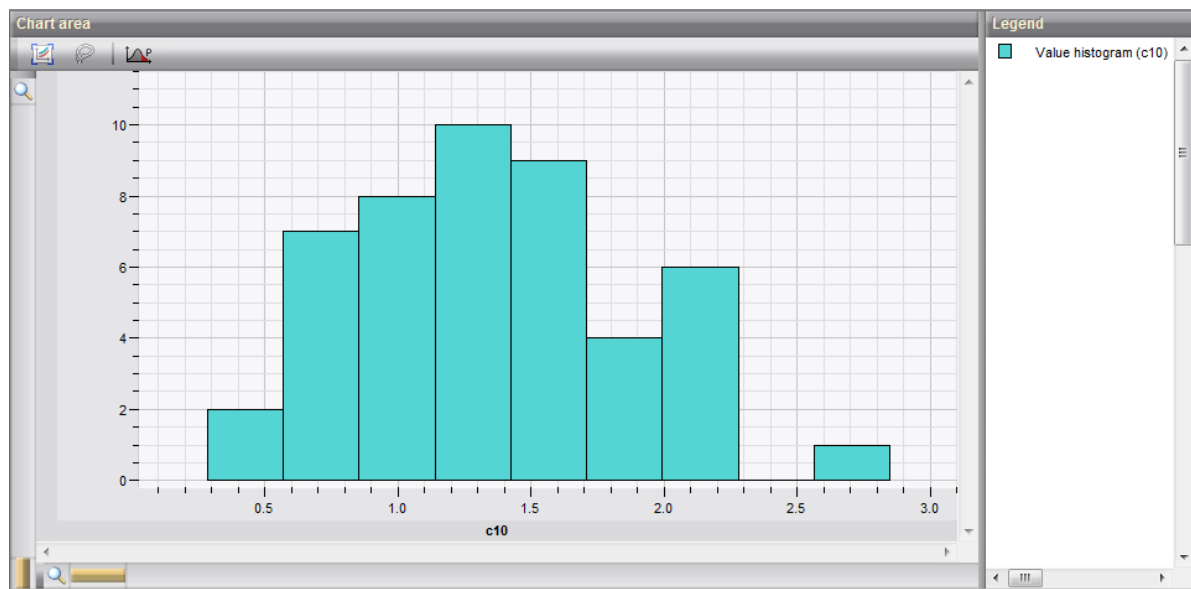


Figure 14.4.5: Value histogram.

14.4.3.2 Value histogram components

This chart type uses the following components:

Value A numerical property that is used to create the chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.3.3 Value histogram properties

The options for the chart can be changed with *Plot > Edit active plot properties...* (🔧). This opens the *Value histogram properties* dialog box (see Figure 14.4.6).

The following options can be changed:

Bin size The binning size for the calculation of the frequencies.

Determine automatically When this option is checked, the bin size is determined automatically based upon the range of the numerical property.

Appearance To change the visual appearance of the value histogram.

Color Allows a color to be defined for the chart, e.g., to differentiate this chart from other charts. With *Change*, a color can be picked from the RGB color table.

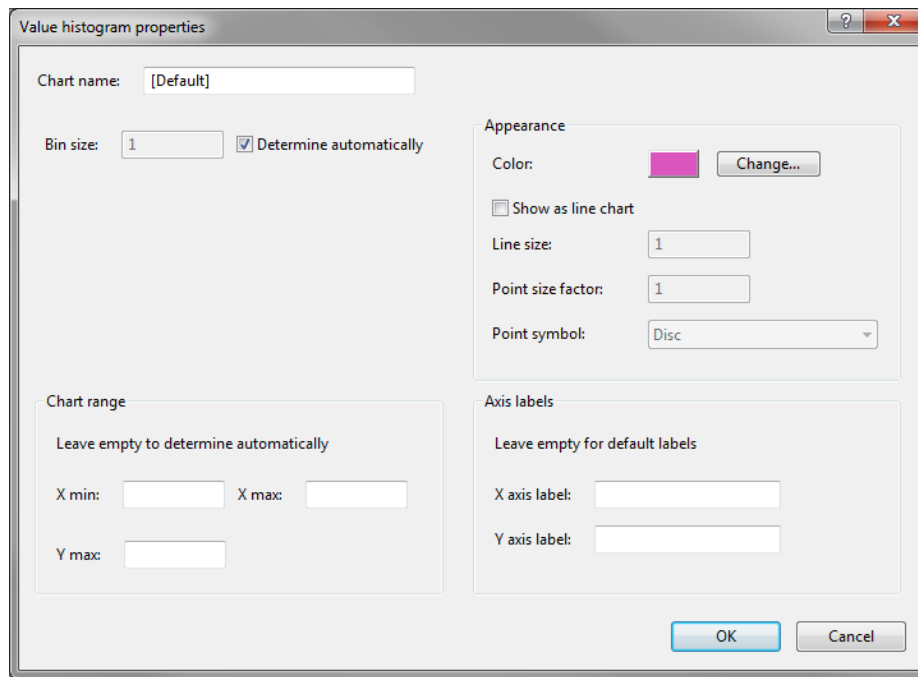


Figure 14.4.6: The *Value histogram properties* dialog box.

Show as line chart With this option enabled, the distribution is displayed as points connected with lines. By default (with the option disabled) the distribution is shown as a bar graph.

Line size With *Show as line chart* enabled, the line size for the connecting lines can be specified. The default value is 1.

Point size factor With *Show as line chart* enabled, a factor determining the size of the points can be specified. The default value is 1.

Point symbol With *Show as line chart* enabled, a point symbol can be selected. The default symbol is a disk.

Chart range To define the range of the chart manually.

X min The minimum value for the X axis (leave empty to allow automatic determination).

X max The maximum value for the X axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.4 Box and Whiskers chart

14.4.4.1 Box and Whiskers chart type

Provides a graphical summary of groups of numerical properties by plotting the median and lower and upper quartiles (Figure 14.4.7). The interpretation of a Box and Whiskers chart is illustrated in Figure 14.4.8.

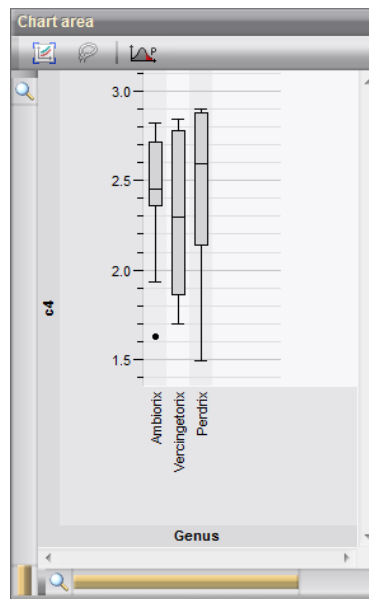


Figure 14.4.7: Box and Whiskers chart. In this example, the numerical property has been divided into categories based upon a categorical variable "Genus".

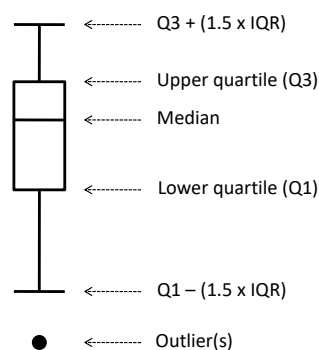


Figure 14.4.8: Explanation of Box and Whiskers chart.

14.4.4.2 Box and Whiskers chart components

This chart type uses the following components:

Value A numerical property that is used to create the chart.

Category (optional): a string property that is used to split the numerical property up in categories, so that a separate Box and Whiskers plot is obtained for each category. The plots are placed next to each other in the chart (see Figure 14.4.7).

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.4.3 Box and Whiskers chart properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Box and Whiskers chart properties* dialog box (see Figure 14.4.9).

The following options can be changed:

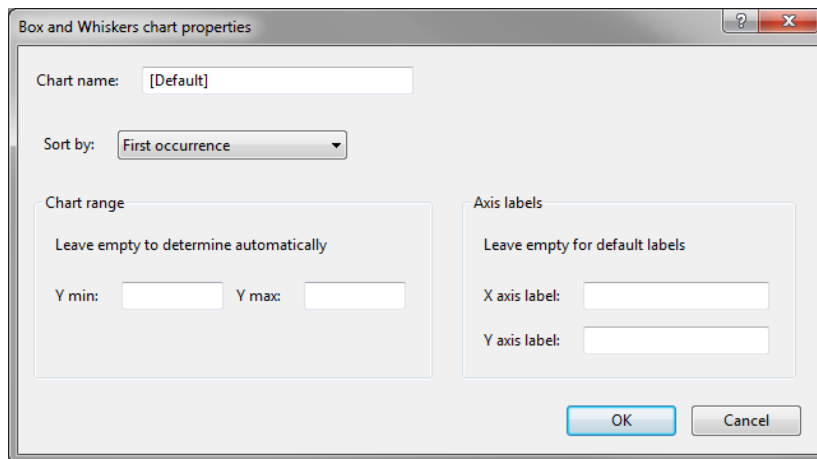


Figure 14.4.9: The *Box and Whiskers chart properties* dialog box

Sort by In case the Box and Whiskers chart has been divided into categories, the categories can be sorted according to different properties: *First occurrence*, *Alphabetical*, *Median* (increasing) and *Box size* (increasing).

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.5 Scatter chart

14.4.5.1 Scatter chart type

Displays the data as a collection of points on a surface, of which the position on the X and Y axis are determined by the values of two properties (Figure 14.4.10).

14.4.5.2 Scatter chart components

This chart type uses the following components:

X axis A numerical property used for the X-axis of the scatter chart.

Y axis A numerical property used for the Y-axis of the scatter chart.

Color (optional): a color property to color the points in the scatter chart.

Point size (optional): a numerical property that is used to assign a different size to the points in the scatter chart.

First Label (optional): a string property to label each point in the scatter chart.

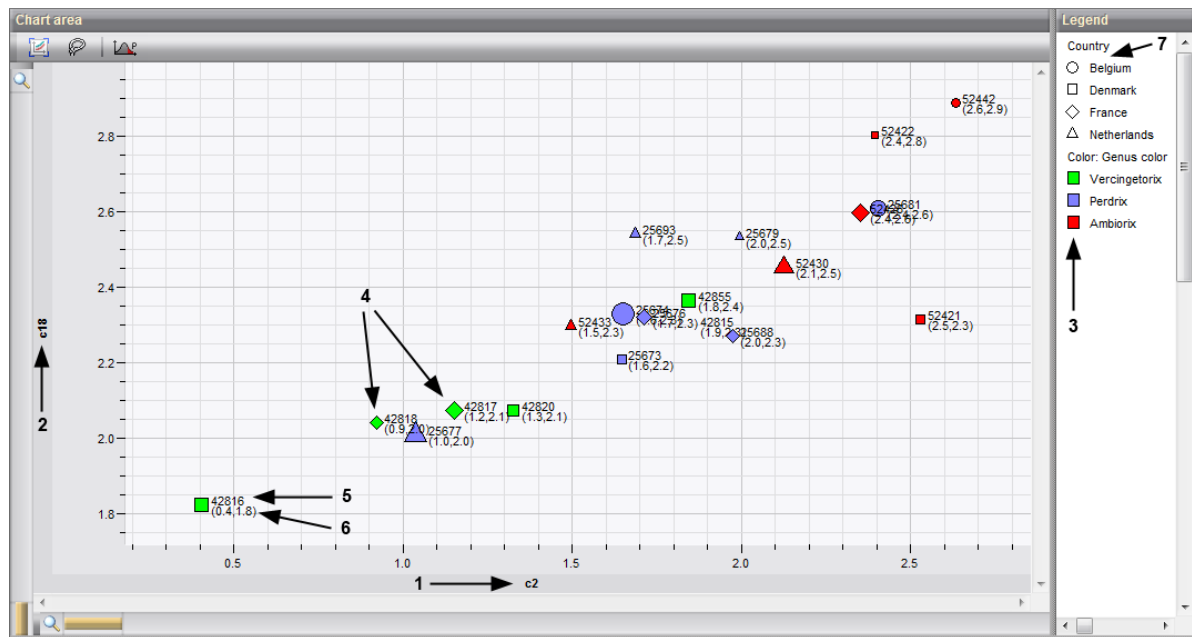


Figure 14.4.10: Scatter chart illustrating component usage. 1: X axis, 2: Y axis, 3: Color, 4: Point size, 5: First label, 6: Second label, 7: Point symbol.

Second Label (optional): an additional string property to label each point in the scatter chart. The second label appears underneath the first label.

Point symbol (optional): a string property from which categories are derived that are represented as different symbols.

Sort (optional): a string or numerical property that can be used to sort the elements in the chart. In a scatter chart, this option only has meaning in case a connecting line is drawn between the points.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.5.3 Scatter chart properties

The options for the chart can be changed with *Plot > Edit active plot properties...* (⚙️). This opens the *Scatter chart properties* dialog box (see Figure 14.4.11).

The following options can be changed:

Points To change the style of the points on the chart.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With *Change*, a point color can be picked from the RGB color table. If a color property is used as component, the point color specified here is shown as a bordering color around the symbols.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk. If string property from the data source is used as symbol component, this should be set to **Take from component**.

Connecting line To draw an optional line connecting the points on the chart and change the appearance of it.

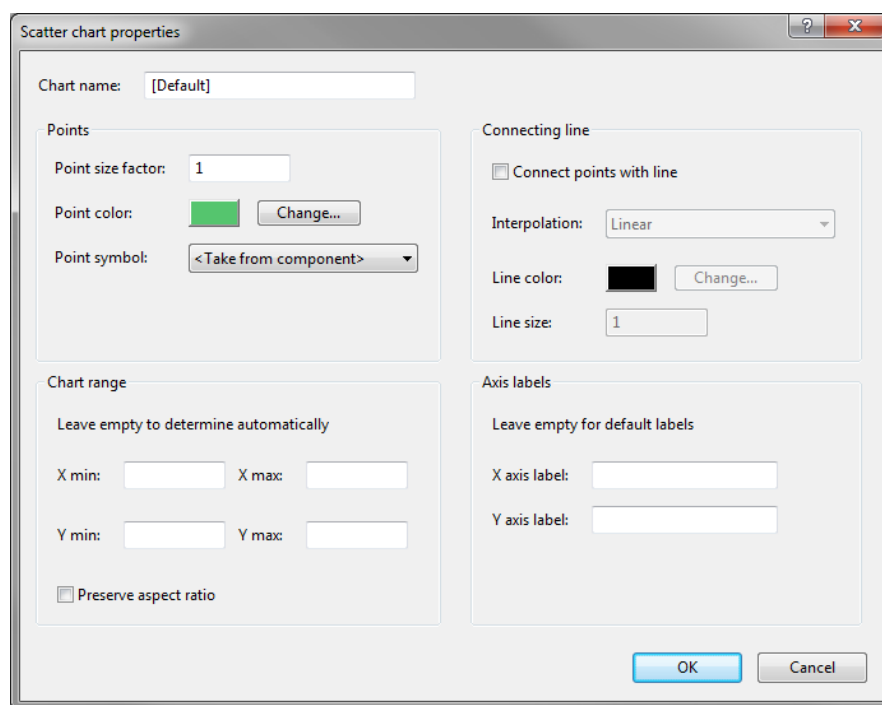


Figure 14.4.11: The *Scatter chart properties* dialog box.

Connect points with line Determines whether a line connecting the points of the scatter chart is drawn or not. Note that in most cases, a connecting line only makes sense in case the scatter chart is sorted according to the numerical property used for the X axis.

Interpolation This option determines the type of interpolation that should be used for the connecting lines:

- Linear: straight lines between the connected points.
- Splines (X+Y axis): cubic-spline interpolation both in X and Y direction.
- Splines (Y axis): cubic-spline interpolation on Y axis only.
- Jump (Y axis): a rectangular connection, first along the X axis and then along the Y axis.
- Jump (X axis): a rectangular connection, first along the Y axis and then along the X axis.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With **Change**, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

X min The minimum value for the X axis (leave empty to allow automatic determination).

X max The maximum value for the X axis (leave empty to allow automatic determination).

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Preserve aspect ratio Allows the scatter chart to be displayed so that the units on both axes have the same size. This option is only useful if the numerical properties for the X and Y axes are physically comparable, e.g. two enzymatic activities measured with the same instrument.

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.6 Profile difference chart

14.4.6.1 Profile difference chart type

Displays the difference between two numerical properties as a colored area (Figure 14.4.12).



Figure 14.4.12: Three combined profile difference charts, showing percentile differences for a character set over a number of entries. percentile differences shown are 40/60 (yellow), 30/70 (orange) and 10/90 (red).

14.4.6.2 Profile difference chart components

This chart type uses the following components:

Unique identifier A unique identifier determines the database components that will form the elements of the chart. The program sets the unique identifier automatically (usually keys or character names). This component should not be changed by the user.

Min Value A numerical property that is used to create the lower boundary of the chart.

Max Value A numerical property that is used to create the upper boundary of the chart.

Label (optional): a string property to label each point in the profile difference chart.

Sort (optional): a string or numerical property that can be used to sort the elements in the chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

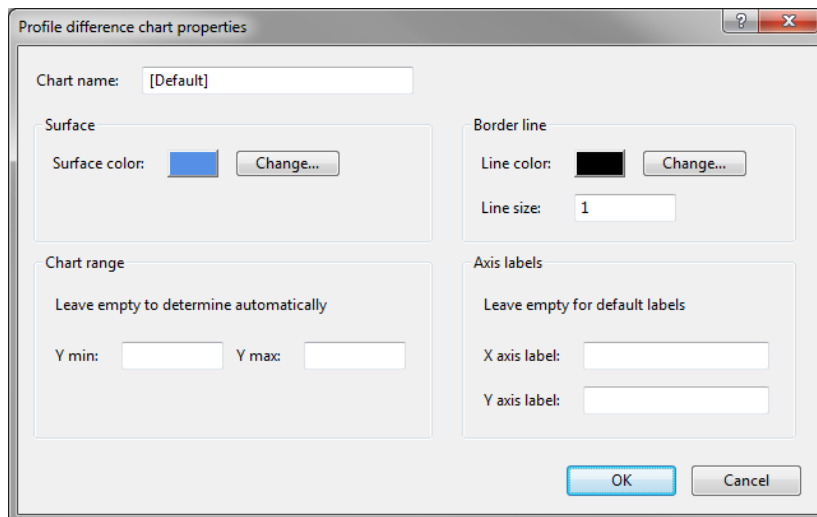


Figure 14.4.13: The *Profile difference chart properties* dialog box.

14.4.6.3 Profile difference chart properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (⚙️). This opens the *Profile difference chart properties* dialog box (see Figure 14.4.13).

The following options can be changed:

Surface To change the style of the surface on the chart.

Surface color Allows a color to be defined for the surface, e.g., to differentiate this chart from other charts. With **Change**, a surface color can be picked from the RGB color table.

Border line To change the style of the line bordering the surface on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With **Change**, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.7 3D Scatter chart

14.4.7.1 3D Scatter chart type

Displays the data as a collection of points in a 3-D space, with the values of three properties determining the position on the X, Y and Z axis. The 3-D view can be rotated in all directions by dragging the mouse over the image (Figure 14.4.14).

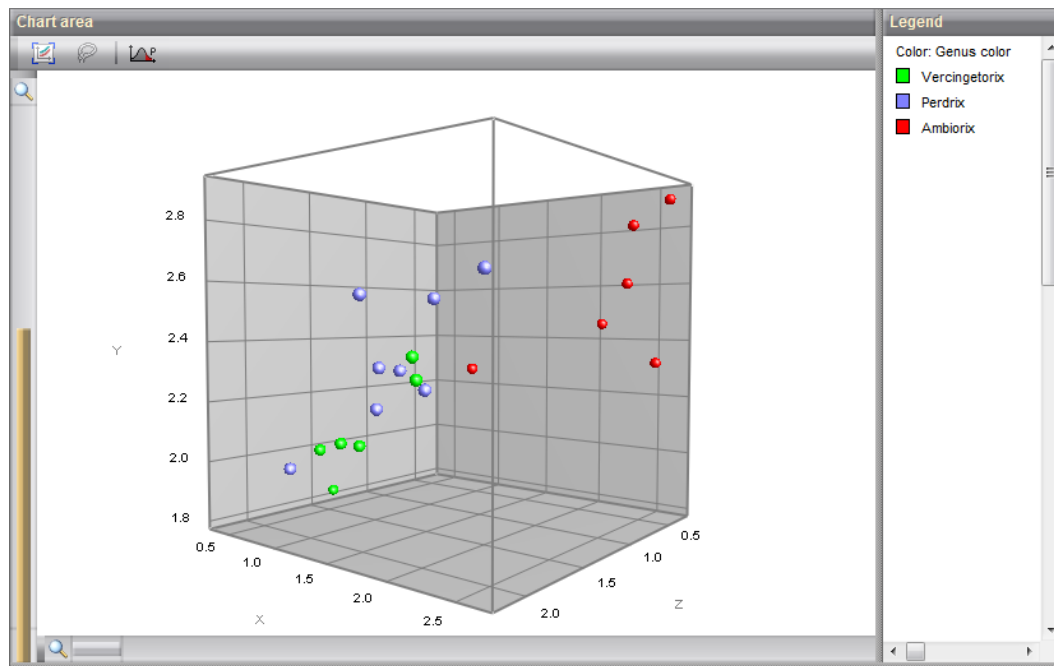


Figure 14.4.14: 3D scatter chart.

14.4.7.2 3D Scatter chart components

This chart type uses the following components:

X axis A numerical property used for the X-axis of the 3D scatter chart.

Y axis A numerical property used for the Y-axis of the 3D scatter chart.

Z axis A numerical property used for the Z-axis of the 3D scatter chart.

Color (optional): a color property to color the points in the 3D scatter chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.7.3 3D Scatter chart properties

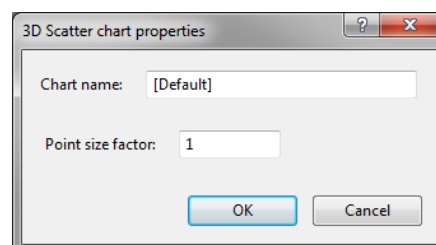


Figure 14.4.15: The 3D Scatter chart properties dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (⚙️). This opens the *3D Scatter chart properties* dialog box (see Figure 14.4.15).

The following options can be changed:

Point size factor A relative factor determining the size of the points. The default value is 1.

14.4.8 Frequency bar graph

14.4.8.1 Frequency bar graph type

A graphical display of tabular frequencies over a categorical property, shown as adjacent rectangles (Figure 14.4.16).

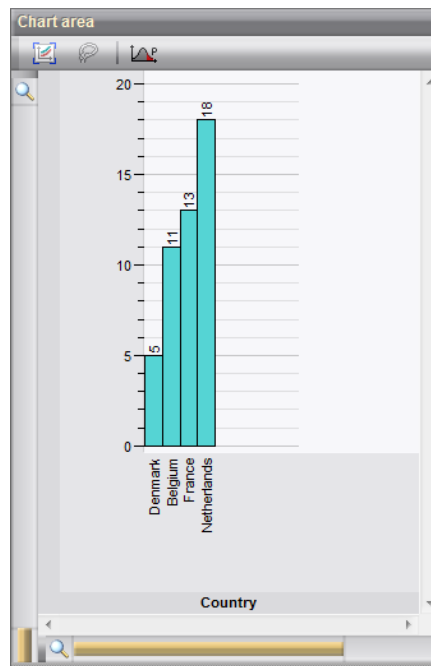


Figure 14.4.16: Frequency bar graph with "Country" as categorical property.

14.4.8.2 Frequency bar graph components

This chart type uses the following components:

Category A string property that is used to define the categories.

Label

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.8.3 Frequency bar graph properties

The options for the chart can be changed with *Plot > Edit active plot properties...* (🔧). This opens the *Frequency bar graph properties* dialog box (see Figure 14.4.17).

The following options can be changed:

Bars To change the style of the bars on the chart.

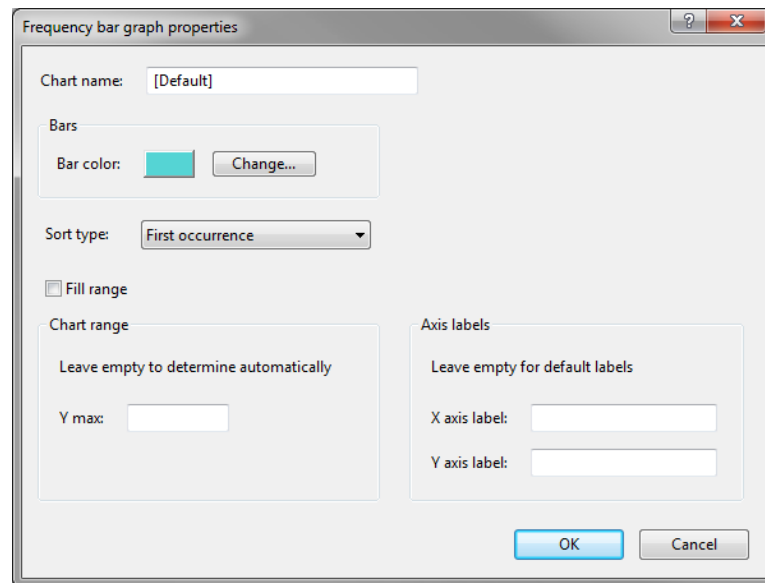


Figure 14.4.17: The *Frequency bar graph properties* dialog box.

Bar color Allows a color to be defined for the bars, e.g., to differentiate this chart from other charts. With *Change*, a bar color can be picked from the RGB color table. If a color property is used as component, the bar color specified here is shown as a bordering line.

Sort type Allows the categories to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Fill range

Chart range To define the range of the chart manually.

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.9 Frequency bar graph (colored)

14.4.9.1 Frequency bar graph (colored) type

A graphical display of tabular frequencies over a categorical property, shown as adjacent rectangles. A color coding represents a second categorical property (Figure 14.4.18).

14.4.9.2 Frequency bar graph (colored) components

This chart type uses the following components:

Category A string property that is used to define the categories.

Color A color property that is used to define the colored categories within the frequency bars.

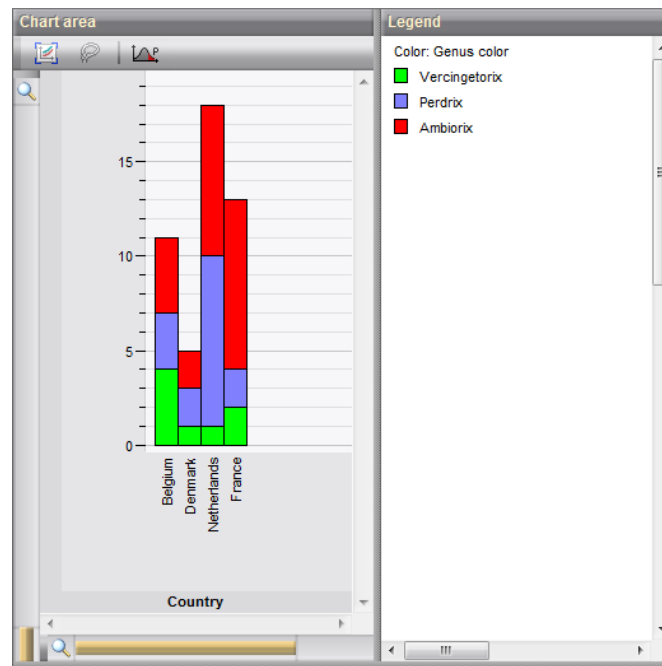


Figure 14.4.18: Frequency bar graph (colored) with string property "Country" used as primary categories and the field states from field "Genus" as color property.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.9.3 Frequency bar graph (colored) properties

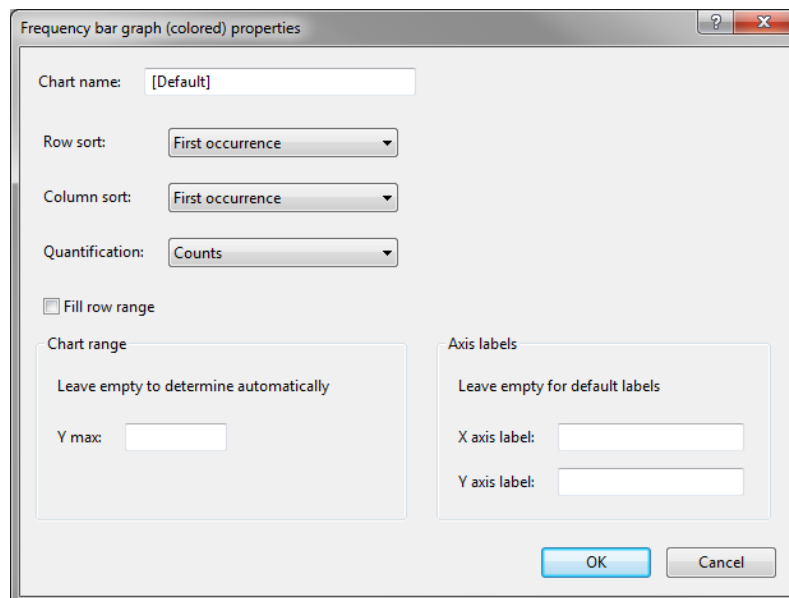


Figure 14.4.19: The *Frequency bar graph (colored) properties* dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Frequency bar graph (colored) properties* dialog box (see Figure 14.4.19).

The following options can be changed:

Row sort Allows the row categories (frequency bars) to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Column sort Allows the column categories (color property) to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Quantification Allows the quantification of the bars to be shown as absolute numbers (*Counts*) or *Percentages*.

Fill row range

Chart range To define the range of the chart manually.

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.10 Contingency chart

14.4.10.1 Contingency chart type

Highlights the relation between two categorical properties, by displaying the frequency distribution in a table format (Figure 14.4.20).

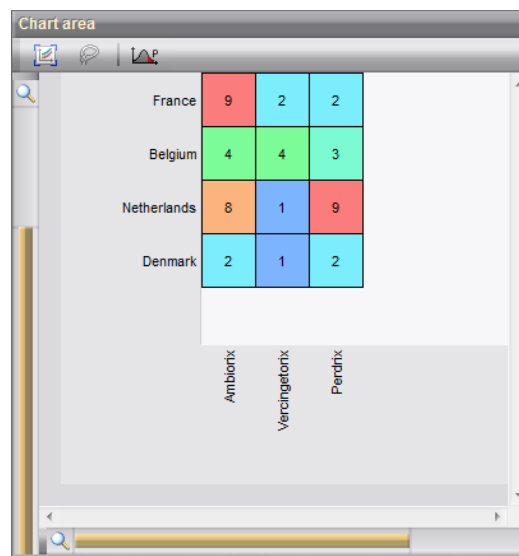


Figure 14.4.20: Contingency table with Genus and Country as categorical variables.

14.4.10.2 Contingency chart components

This chart type uses the following components:

Column The first categorical variable shown as columns in the table.

Row The second categorical variable shown as rows in the table.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.10.3 Contingency chart properties

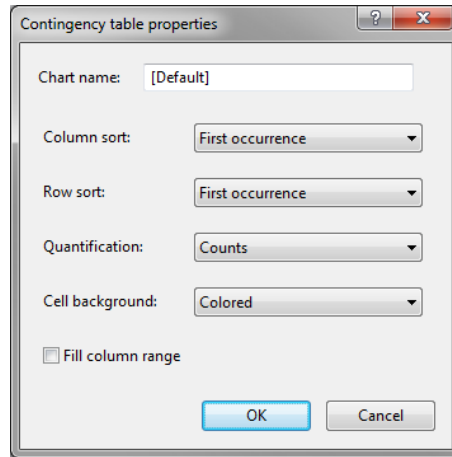


Figure 14.4.21: The *Contingency table properties* dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Contingency table properties* dialog box (see Figure 14.4.21).

The following options can be changed:

Column sort Allows the column categories to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Row sort Allows the row categories to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Quantification Allows the quantification of the cells to be shown as absolute numbers (*Counts*), percentages of the row totals *Row percentages*, percentages of the column totals *Column percentages*, or *Residuals*. The *residual* is a measure for the deviation from the estimated number of counts in that cell and is calculated as

$$\frac{[N_{oij} - n_{ij}]}{\sqrt{n_{ij}}}$$

with N_{oij} the observed cell count and n_{ij} the estimated cell count.

Cell background Allows the cell background to be colored based upon the member count, which can either be *Colored*, *Grayscale*, or *None*.

Fill column range

14.4.11 3D Contingency table

14.4.11.1 3D Contingency table type

Highlights the relation between two categorical properties, by displaying the frequency distribution in a 3D bar graph (Figure 14.4.22).

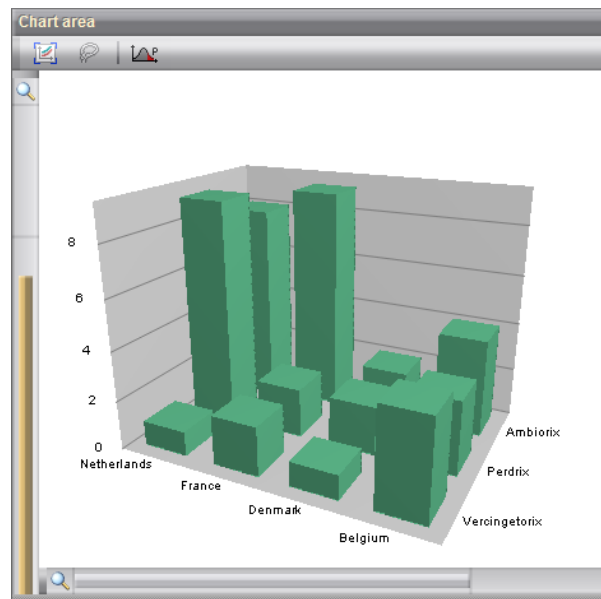


Figure 14.4.22: 3D contingency chart.

14.4.11.2 3D Contingency table components

This chart type uses the following components:

First category The first categorical variable shown as the X-axis in the 3D chart.

Second category The second categorical variable shown as the Y-axis in the 3D chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.11.3 3D Contingency table properties

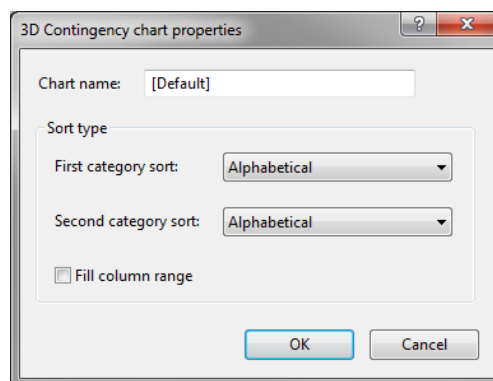


Figure 14.4.23: The 3D Contingency chart properties dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (⚙️). This opens the *3D Contingency chart properties* dialog box (see Figure 14.4.23).

The following options can be changed:

Sort type :

First category sort Allows the first category to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Second category sort Allows the second category to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Fill column range

14.4.12 ANOVA chart

14.4.12.1 ANOVA chart type

The ANOVA chart (ANalysis Of Variance) provides a graphical summary of a numerical property grouped over categories by plotting the averages and standard deviations for the categories (Figure 14.4.24).

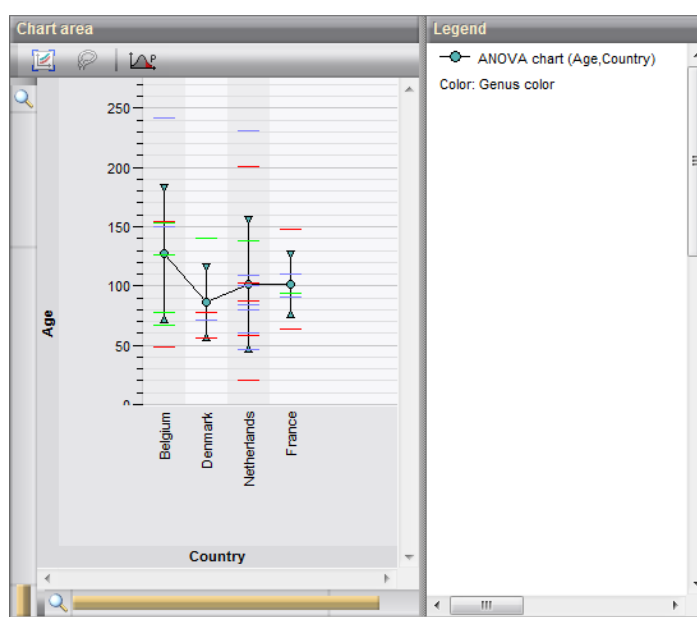


Figure 14.4.24: ANOVA chart summarizing a value property "Age" over the categories "Country". A color property "Genus" is used as an optional color component.

14.4.12.2 ANOVA chart components

This chart type uses the following components:

Value The numerical property used to calculate the ANOVA.

Category The categorical variable used to determine the ANOVA categories.

Color (optional): a color property used to color the elements in the ANOVA chart.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.12.3 ANOVA chart properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *ANOVA chart properties* dialog box (see Figure 14.4.25).

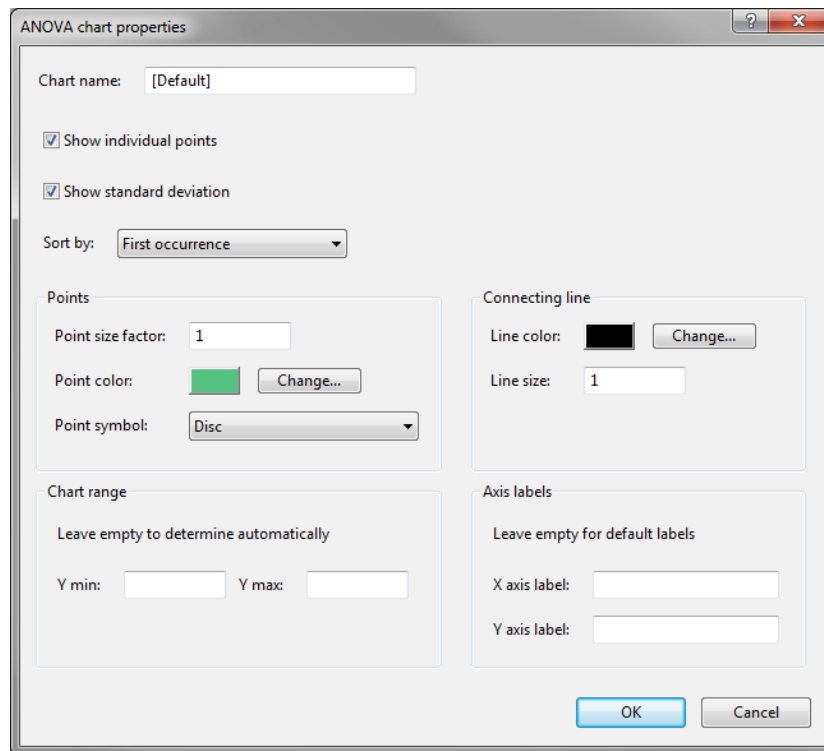


Figure 14.4.25: The ANOVA chart properties dialog box.

The following options can be changed:

Show individual points To display the data set elements as individual lines on the chart.

Show standard deviation To display the standard deviation bars on the ANOVA chart.

Sort by Allows the categories to be sorted according to the following properties: *First occurrence*, *Alphabetical*, *Average*, and *Standard deviation*.

Points To change the style of the points manually.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With *Change*, a point color can be picked from the RGB color table. If a color property is used as component, the point color specified here is shown as a bordering color around the symbols.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk.

Connecting line To change the style of the line connecting the points on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With *Change*, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.13 Component summary (mean)

14.4.13.1 Component summary (mean) type

Summarizes a set of numerical properties by displaying the average value for each property in a line chart (Figure 14.4.26). This chart type may have any number of numerical properties as input.

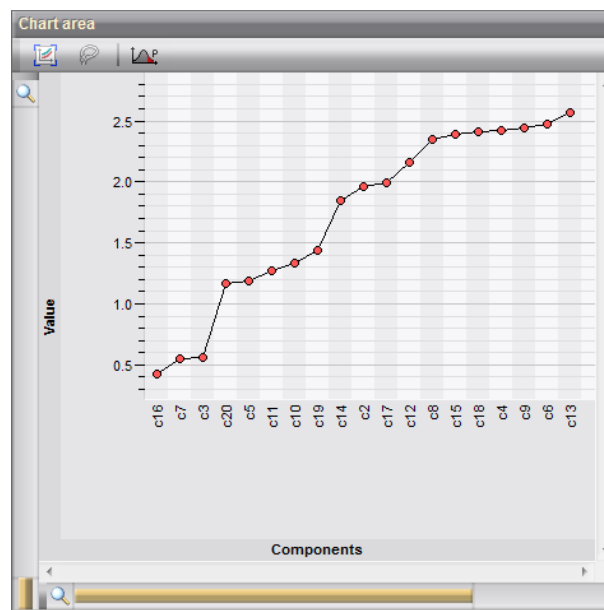


Figure 14.4.26: A Component summary (mean) chart generated from 19 numerical properties.

14.4.13.2 Component summary (mean) components

This chart type uses the following components:

Value (required): The numerical value used as summary component.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.13.3 Component summary (mean) properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Component summary (mean) properties* dialog box (see Figure 14.4.27).

The following options can be changed:

Sort by Allows the properties to be sorted according to the following parameters: *First occurrence*, *Alphabetical*, *Average*, and *Standard deviation*.

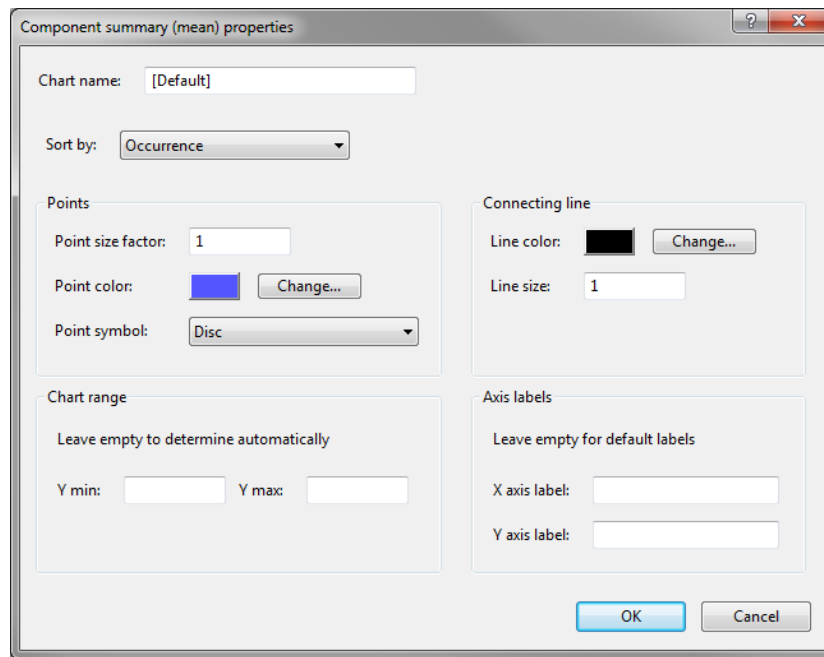


Figure 14.4.27: The *Component summary (mean) properties* dialog box.

Points To change the style of the points manually.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With *Change*, a point color can be picked from the RGB color table.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk.

Connecting line To change the style of the line connecting the points on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With *Change*, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

14.4.14 Component summary (quantile)

14.4.14.1 Component summary (quantile) type

Summarizes a numerical property by displaying a specific quantile (Figure 14.4.28). This chart type may have any number of numerical properties as input.

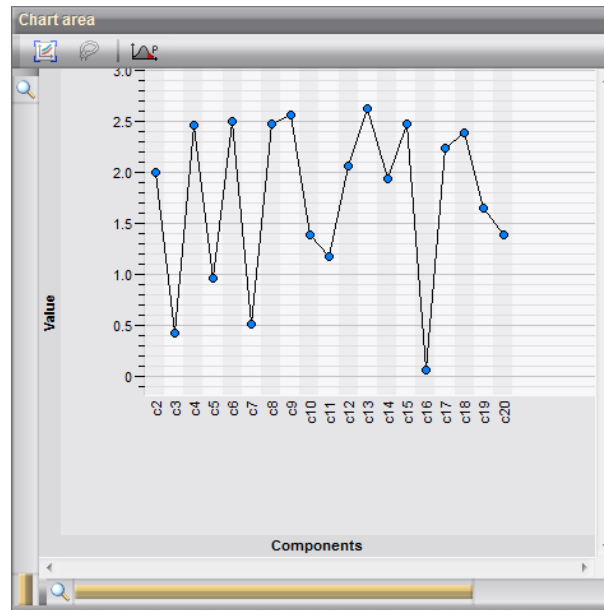


Figure 14.4.28: A Component summary (quantile) chart generated from 19 numerical properties and with a percentile of 50 (median).

14.4.14.2 Component summary (quantile) components

This chart type uses the following components:

Value (required): The numerical value used as summary component.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.14.3 Component summary (quantile) properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Component summary (quantile) properties* dialog box (see Figure 14.4.29).

The following options can be changed:

Sort by Sorts the categories according to one of the following properties: *First occurrence*, *Alphabetical*, *Increasing count*, and *Decreasing count*.

Points To change the style of the points manually.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With *Change*, a point color can be picked from the RGB color table.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk.

Connecting line To change the style of the line connecting the points on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With *Change*, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

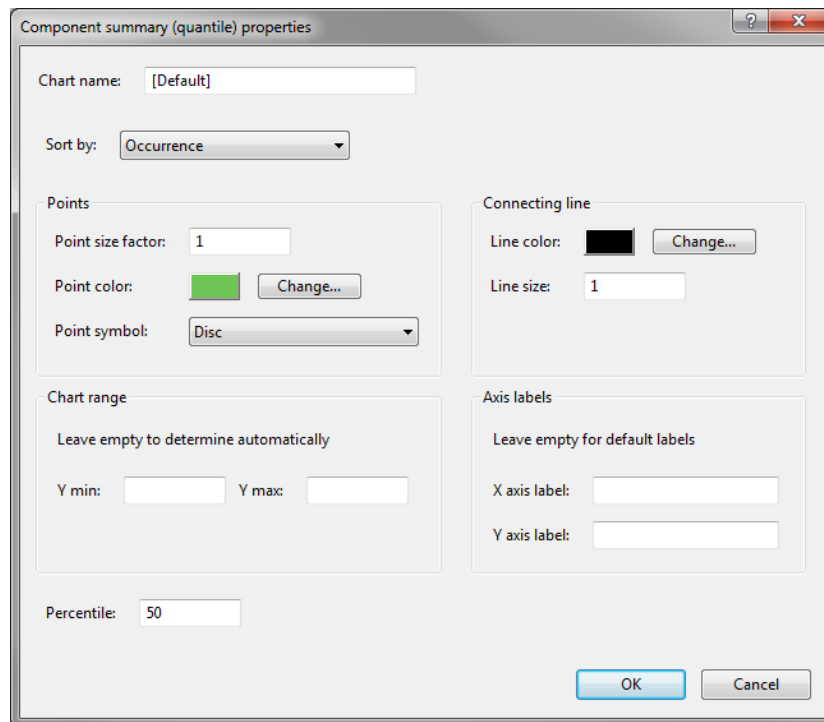


Figure 14.4.29: The *Component summary (quantile) properties* dialog box.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

Percentile The percentile value (0 - 100) used to calculate the quantiles. By default, the Median (50) is used.

14.4.15 Component summary (range count)

14.4.15.1 Component summary (range count) type

Summarizes a numerical property by counting the number of times it falls within a specified range;(Figure 14.4.30). This chart type may have any number of numerical properties as input.

14.4.15.2 Component summary (range count) components

This chart type uses the following components:

Value (required): The numerical value used as summary component.

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

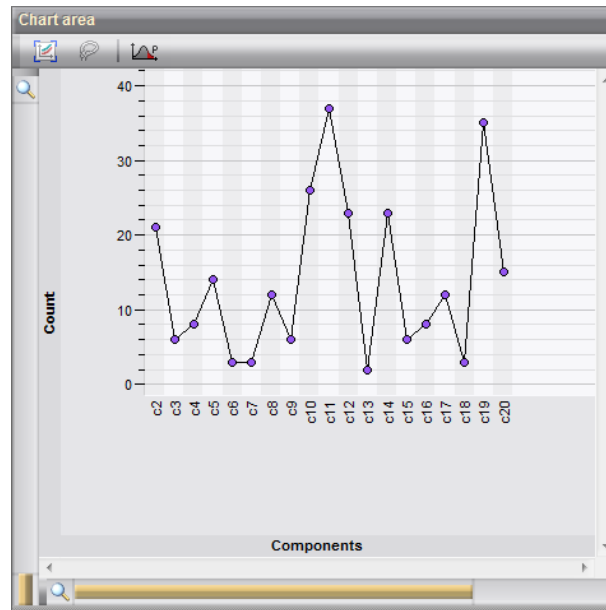


Figure 14.4.30: A Component summary (range count) chart generated from 19 numerical properties with range between 1 and 2.

14.4.15.3 Component summary (range count) properties

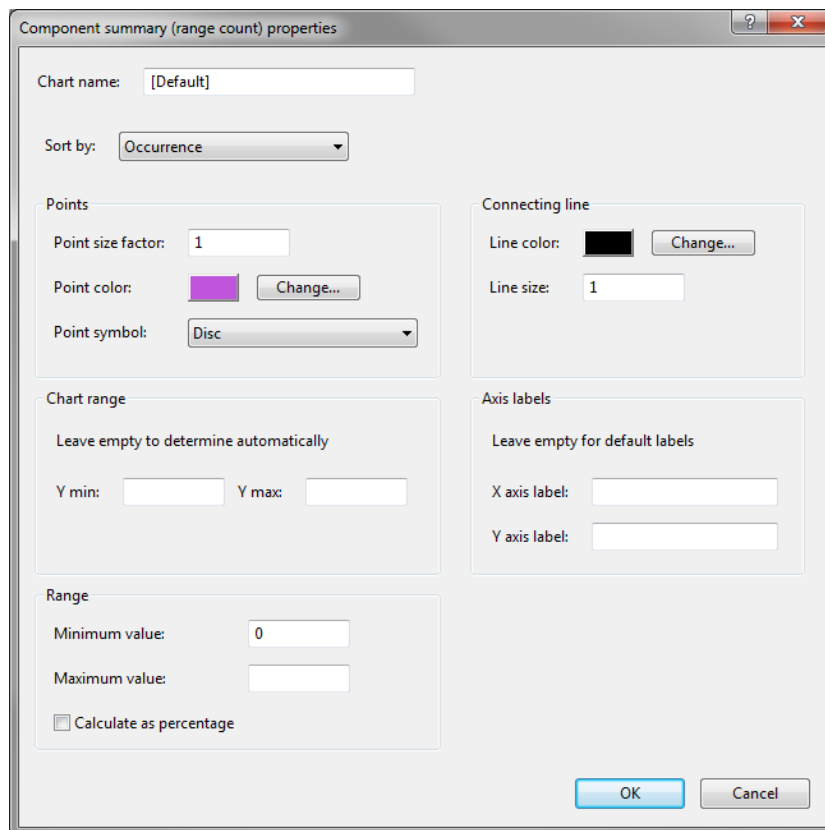


Figure 14.4.31: The *Component summary (range count) properties* dialog box.

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Component summary (range count) properties* dialog box (see Figure 14.4.31).

The following options can be changed:

Sort by Sorts the chart by a selected property (*Occurrence*, *Alphabetical*, *Increasing value*, *Decreasing value*).

Points To change the style of the points manually.

Point size factor A relative factor determining the size of the points. The default value is 1.

Point color Allows a color to be defined for the points, e.g., to differentiate this chart from other charts. With *Change*, a point color can be picked from the RGB color table.

Point symbol Allows a specific symbol to be defined for the points, e.g., to differentiate this chart from other charts. The default symbol is a disk.

Connecting line To change the style of the line connecting the points on the chart.

Line color Allows a color to be defined for the line, e.g., to differentiate this chart from other charts. With *Change*, a line color can be picked from the RGB color table.

Line size The thickness of the line can be entered in pixels.

Chart range To define the range of the chart manually.

Y min The minimum value for the Y axis (leave empty to allow automatic determination).

Y max The maximum value for the Y axis (leave empty to allow automatic determination).

Axis labels To name the axes of the chart manually.

X axis label The label for the X axis (leave empty to allow automatic naming).

Y axis label The label for the Y axis (leave empty to allow automatic naming).

Range To set the range for the value component.

Minimum value Defines the minimum value of the component summary range.

Maximum value Defines the maximum value of the component summary range.

Calculate as percentage Displays the range counts as a percentage of the total number of values.

14.4.16 Pie chart histogram table

14.4.16.1 Pie chart histogram table type

A pie chart histogram table displays the frequency distribution of up to three categorical properties. Two categories (string properties) are represented in a tabular format, one category as rows, the other as columns (cfr. contingency table). The third category (a color property) is represented in the pie charts displayed in each cell. The frequency over the row and column category is indicated by the size of the pie charts (Figure [14.4.32](#)).

Note that each chart component is optional:

- Leaving out the column category results in a row of pie charts;
- Leaving out the row category results in a column of pie charts;
- Leaving out the color category results in a contingency table with frequencies displayed as gray circles with sizes proportional to the frequencies.

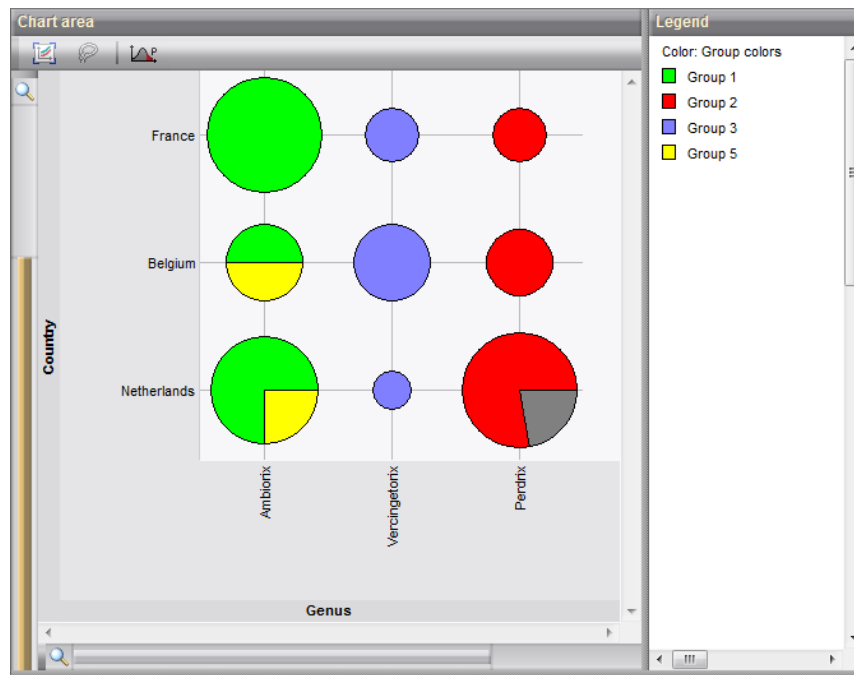


Figure 14.4.32: A pie chart histogram table .

14.4.16.2 Pie chart histogram table components

This chart type uses the following components:

Column category Specifies what string property will be used to create the different categories that determine the columns of the table. This component can be omitted, in which case a single column will be created.

Row category Specifies what string property will be used to create the different categories that determine the rows of the table. This component can be omitted, in which case a single row will be created.

Pie chart category Specifies what color property will be used to create the different categories that determine the pies in the pie charts. This component can be omitted, in which case each cell will contain a monotonous gray circle.

Pie chart abundance

Filter (optional): A Boolean property that specifies what records from the data source will be included in the chart. Only records that have a TRUE value for this property will be included.

14.4.16.3 Pie chart histogram table properties

The options for the chart can be changed with **Plot > Edit active plot properties...** (🔧). This opens the *Pie chart histogram table properties* dialog box (see Figure 14.4.33).

The following options can be changed:

Default size A size factor that determines the default size of the pie charts.

Row sort Determines how the rows of the plot should be sorted. The categories can be sorted according to one of the following properties: *First occurrence*, *Alphabetical*, *Reverse alphabetical*, *Increasing count*, and *Decreasing count*.

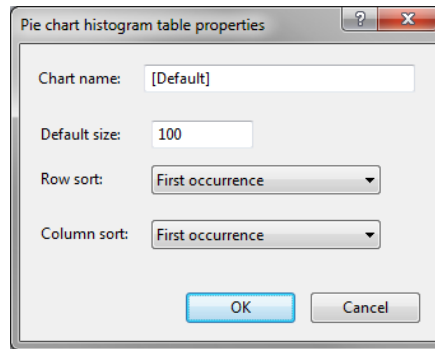


Figure 14.4.33: The *Pie chart histogram table properties* dialog box.

Column sort Determines how the columns of the plot should be sorted. The categories can be sorted according to one of the following properties: *First occurrence*, *Alphabetical*, *Reverse alphabetical*, *Increasing count*, and *Decreasing count*.

Chapter 14.5

Charts window

14.5.1 Window layout

The *Charts and statistics* window consists of 5 dockable panels (see Figure 14.5.1):

- *Chart list* panel lists all the charts and fits created in this window.
- *Data source overview* panel shows the tree with data source and the properties from which components can be selected to create or decorate a chart.
- *Chart area* panel is the panel where the charts are displayed.
- *Legend* panel shows the meaning of the different symbols and/or colors used.
- *Chart report* panel lists the data source, the components with their use and the chart display options for the selected chart.

A *Charts and statistics* window is always attached to data source from which it was created and provides a "live" presentation of the data: changing entry selections or group colors for example, is instantly synchronized in the *Charts and statistics* window. Once a chart is generated, the components can still be changed by selecting other properties from the data source as components, or new components from the data source can be added. The "Live" status is indicated in the *Data source overview* panel. For example, if a *Charts and statistics* window is created from a *Comparison* window with the comparison entries as data source, the data source tree is indicated as **Comparison entries (Live data)**. All properties from the data source can still be selected as components. If the window from which the *Charts and statistics* window was created is closed, the data source will disappear, and the data source will no longer be indicated as **Live data**. In that case, only those properties from the data source that are used in a chart will remain available, the others will become unavailable. As soon as a component is removed from a chart, the corresponding property in the data source will disappear as well.



Changes made in the database to information fields or character values are not automatically synchronized in the *Charts and statistics* window. To update such changed property information in the charts, the command **Dataset > Refresh** can be used.

Once a chart is created it is displayed in the *Charts and statistics* window, the chart is shown in the *Chart area* panel. The name of the chart is listed in the *Chart list* panel. By default, the name is composed by the chart type and the properties used (between brackets). Since the chart name can be changed by the user (see 14.5.4), the chart type is listed in the column 'Type' as well. The column 'Visible' indicates whether a chart is visible or not (see 14.5.6). The properties from the data source that are used as chart components are indicated with a green flag in the *Data source overview* panel. The *Legend* panel explains the symbol

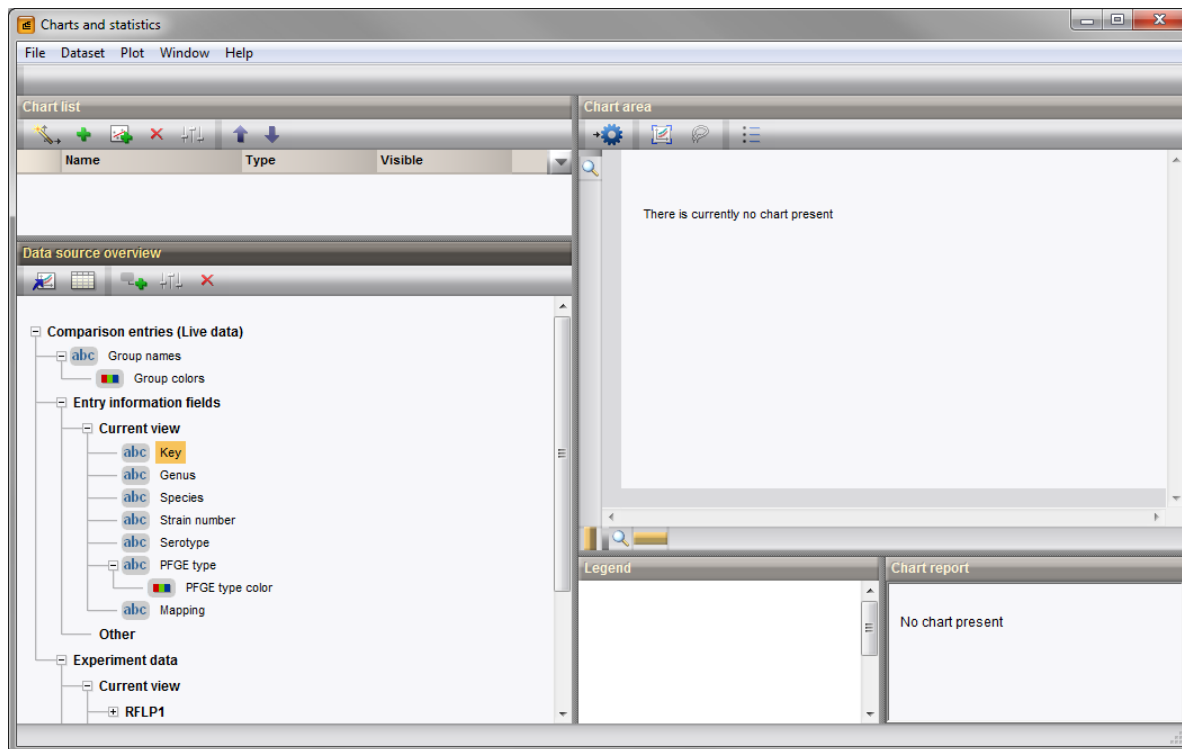


Figure 14.5.1: The *Charts and statistics* window.

shape and colors for those charts where symbols and/or colors are used. Note that in case a chart uses one unique symbol, this legend is also shown in the *Chart list* panel.

The information in the *Legend* panel can be shown next to the chart in the *Chart area* panel with the option **Plot > Toggle legend on chart**. The position of the chart borders can be modified by placing the mouse on a border and dragging the double arrow to the desired position in the *Chart area* panel with the mouse. Using the zoom sliders, it is possible to zoom in and out on the chart in the horizontal or vertical direction. Resetting the zoom sliders is done with **Plot > Reset zoom** (🔍).

14.5.2 Editing components of a chart

Once a chart has been created, it is still possible to change the selected components or add new components. In the *Data source overview* panel, the properties that are used as chart components are flagged with a green V-sign.

A property can be selected or unselected as component by right-clicking on the property and selecting **Use as Chart <component>** from the floating menu, where <component> can be *category*, *value*, *color*, *sort*, etc..

If the property is currently in use as chart component, the option is flagged in the floating menu. If the property can be used for different components, all choices are listed in the floating menu.

The selected property can also be assigned or unassigned as component with **Dataset > Use property as plot component...** (🔗). This calls the *Link property to chart component* dialog box (see Figure 14.5.2).

The *Link property to chart component* dialog box lists all the chart components to which the property can be assigned.

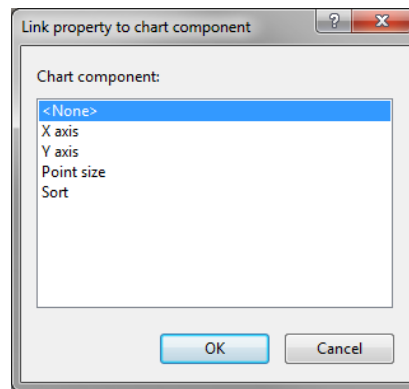


Figure 14.5.2: The *Link property to chart component* dialog box.

14.5.3 Derived properties

14.5.3.1 Introduction

The properties listed in the *Data source overview* panel form the basis for the components of a chart (see also 14.4). Each property belongs to a fixed type and can therefore only be used for chart components that require this property type (e.g. string, value, color, date, boolean). To enhance the flexibility and increase the possibilities beyond the existing properties, the software allows *derived properties* to be created. A derived property uses one or more data source properties from a given type as input argument(s) and produces a new property from a given type as output. The derived property may also require one or more options.

For example, a derived property ***Convert to string*** requires a *value property* as input argument and returns a *string property* containing the values as strings. The derived property has an *option* to enter a fixed number of digits and the minimum number of characters.

A derived property can be created by first highlighting in the *Data source overview* panel the property or properties required as argument(s). Multiple properties can be highlighted by holding down the **Ctrl**-key while selecting. Then, the *Derived property type* wizard page is called with **Dataset > New derived property...** (🔧) (see Figure 14.5.3).

14.5.3.2 The Create derived property wizard

In the first page of the *Create derived property* wizard, the derived property is chosen from a list. The derived properties listed are those that are compatible with the argument(s) chosen.

In the second and last page of the wizard, a property name, the argument assignment and the options can be entered. This page is specific for the selected derived property.

A derived property is shown in the *Data source overview* panel with three red dots on the right side of the icon (see Figure 14.5.5). The derived property branches from the property it uses as input argument. If the derived property uses more than one property as input arguments, it branches from the first property in the list of arguments.

Once created, the name and the options of a derived property can still be edited with **Dataset > Edit derived property...** (🔧). This action results in a dialog box similar as in Figure 14.5.4, however, the arguments cannot be altered anymore.

An overview of all derived properties can be found in 14.5.3.3.

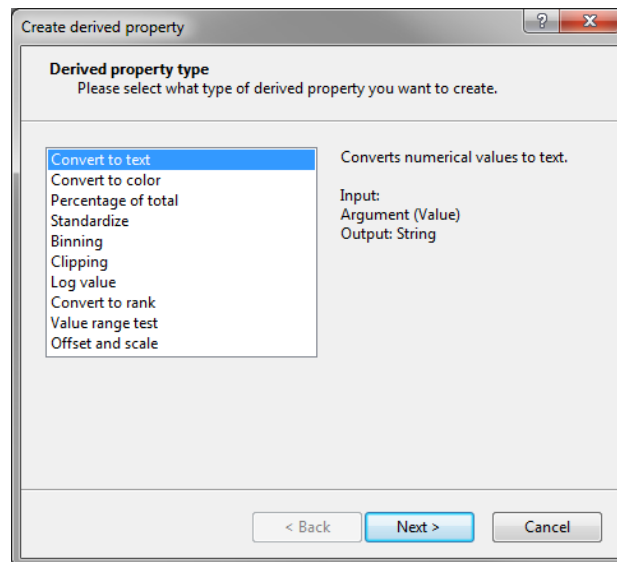


Figure 14.5.3: The *Derived property type* wizard page.

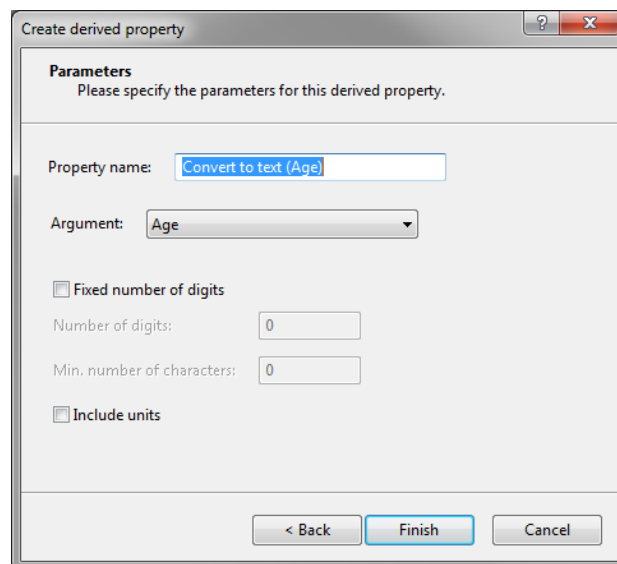


Figure 14.5.4: The second page is specific for the selected derived property.

14.5.3.3 Data Set Derived Properties

14.5.3.3.1 Convert to text

Converts numerical values to strings.

This derived property has the following features:

Input

Argument Value.

Output String.

Options

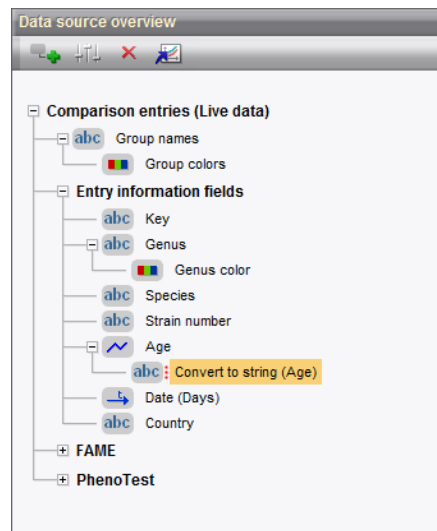


Figure 14.5.5: The *Data source overview* panel with a derived property **Convert to string** (**age**) derived from the value property **Age**.

Fixed number of digits When checked, the minimum length and number of digits in the output strings can be entered.

Number of digits Specifies a fixed number of decimal digits (default 0).

Min. number of characters Specifies a minimum number of characters (default 0).

Include units When checked, the units will be added to the strings (if available).

14.5.3.3.2 Convert to color

Converts values to a set of colors.

This derived property has the following features:

Input

Argument Value.

Output Color.

Options

Color range Specifies a predefined color scheme to be applied.

Auto range If checked, the range is determined automatically from the data source.

Minimum value With *Auto range* unchecked, a minimum value for the color range can be entered.

Maximum value With *Auto range* unchecked, a maximum value for the color range can be entered.

14.5.3.3.3 Percentage of total

Converts numerical values to their fractions of the sum of the set (expressed as percentage).

This derived property has the following features:

Input

Argument Value.

Output Value.

Options None.

14.5.3.3.4 Standardize

Subtracts the average and divides by the standard deviation.

This derived property has the following features:

Input

Argument Value.

Output Value.

Options None.

14.5.3.3.5 Binning

Converts numerical values to a set of pre-defined binning values.

This derived property has the following features:

Input

Argument Value.

Output Value.

Options

Bin size Specifies the size for binning (default 1).

14.5.3.3.6 Clipping

Clips a numerical value so that it falls within a specified range.

This derived property has the following features:

Input

Argument Value.

Output Value.

Options

Minimum value Specifies the minimum value of the clipping range.

Maximum value Specifies the maximum value of the clipping range.

Remove if outside range When checked, values that fall outside the clipping range will be removed from the chart.

14.5.3.3.7 Log value

Calculates the logarithm of numerical values.

This derived property has the following features:

Input

Argument Value.

Output Value.

Options None.

14.5.3.3.8 Convert to rank

Converts numerical values to their ranks in an ordered set (from low to high).

This derived property has the following features:

Input

Argument Value.

Output Value.

Options

Calculate as percentile When checked, the ranks will be calculated as percentiles rather than integer rank numbers.

Inverted When checked, the ranking will be done from high to low.

14.5.3.3.9 Value range test

Checks if values lie inside a pre-defined range. The result is of a boolean type, which can for instance be used as a filter component for the chart.

This derived property has the following features:

Input

Argument Value.

Output Boolean.

Options

Not smaller than Specifies the minimum of the range.

Not larger than Specifies the maximum of the range.

14.5.3.3.10 Offset and scale

Multiplies a value by a fixed factor and adds a fixed offset.

This derived property has the following features:

Input

Argument 1 Value.

Output Value.

Options

Offset The offset value to be added.

Scale The scaling factor to be multiplied by.

New unit Specifies a unit to be applied to the new values.

14.5.3.3.11 Linear combination

Applies a linear combination to two sets of values.

This derived property has the following features:

Input

Argument 1 Value.

Argument 2 Value.

Output Value.

Options

Factor 1 Specifies a factor to multiply with values of argument 1.

Factor 2 Specifies a factor to multiply with argument 2.

14.5.3.3.12 Product

Calculates the product to two sets of values.

This derived property has the following features:

Input

Argument 1 Value.

Argument 2 Value.

Output Value.

Options None.

14.5.3.3.13 Division

Divides two sets of values.

This derived property has the following features:

Input

Argument 1 Value.

Argument 2 Value.

Output Value.

Options None.

14.5.3.3.14 Average

Calculates the average of a set of values. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options None.

14.5.3.3.15 Sum

Calculates the sum of a set of values. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options None.

14.5.3.3.16 Minimum

Calculates the minimum of a set of values. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options None.

14.5.3.3.17 Maximum

Calculates the maximum of a set of values. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options None.

14.5.3.3.18 Percentile

Calculates the particular percentile of a set of values. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options

Percentile The percentile as a value between 0 and 100.

14.5.3.3.19 Count different values

Counts how many different values are present in a set of properties. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options

14.5.3.3.20 Count values in range

Counts how many different property values are inside a specific range. This derived property can have any number of numerical properties as input.

This derived property has the following features:

Input

ValueOpen Value (linked automatically).

Output Value.

Options

Minimum value The minimum value of the range.

Maximum value The maximum value of the range.

Calculate as percentage If checked, the number will be given as a percentage of the total number of values.

14.5.3.3.21 Convert to value

Converts strings to numerical values.

This derived property has the following features:

Input

Argument String.

Output Value.

Options None.

14.5.3.3.22 Compare text

Compares strings to a fixed argument string, and returns booleans. Booleans can for instance be used as a filter component for the chart.

This derived property has the following features:

Input

Argument String.

Output Boolean.

Options

Compare to text Specifies the string to be compared to the strings from the data source.

14.5.3.3.23 Convert to color

Converts strings to colors, so that each unique string has a different color.

This derived property has the following features:

Input

Argument String.

Output Color.

Options None.

14.5.3.3.24 Merge two strings

Concatenates string pairs from two sets of strings.

This derived property has the following features:

Input

String 1 String.

String 2 String.

Output String.

Options

Template Specifies how the strings should be concatenated. The first string is indicated as ^1, the second string as ^2. The default template is ^1,^2 which means that the two concatenated strings will appear as

String1,String2.

Any alphanumerical characters and spaces can be entered before, between and after the string indicators. Example:

^1 (^2) will appear as String1 (String2).

14.5.3.3.25 Compare two strings

Compares string pairs from two sets of strings and returns true or false if they are the same or different. The result is a set of booleans.

This derived property has the following features:

Input

String 1 String.

String 2 String.

Output String.

Options None.

14.5.3.3.26 Decorate text

Adds predefined text before and/or after a string.

This derived property has the following features:

Input

Argument String.

Output String.

Options

Template Specifies how the string should be decorated. The string is indicated as ^1. The default template is (^1) which means that the string will appear between brackets.

Any alphanumerical characters and spaces can be entered before and/or after the string indicators. Example:

LMG ^1 (T) will appear as LMG String (T).

14.5.3.3.27 Abbreviate

Abbreviates a string to a fixed number of characters.

This derived property has the following features:

Input

Argument String.

Output String.

Options

Maximum number of characters Specifies the number of characters the string should be abbreviated to.

Write dot Check this option to write a dot at the end of the abbreviated string.

14.5.3.3.28 Regular expression substring

Performs a regular expression match on a string and returns a sub-patterns (in round brackets).

This derived property has the following features:

Input

Argument String.

Output String.

Options

Pattern A regular expression (see [21.2](#)), containing a sub-pattern.

14.5.3.3.29 Count string occurrence

Counts how many times a specific string is found in a set of string properties. This derived property can have any number of string properties as input.

This derived property has the following features:

Input

Output

Options

14.5.3.3.30 Count unique strings

Counts how many different unique strings are found in a set of string properties. This derived property can have any number of string properties as input.

This derived property has the following features:

Input

StringOpen String (linked automatically).

Output Value.

Options

Calculate as percentage Display the string occurrence as a percentage rather than an absolute value.

14.5.3.3.31 Convert to value

Converts dates to numerical values, with days as unit.

This derived property has the following features:

Input

Argument Date.

Output Value.

Options

Start date Specifies a start date. Days will be counted from this start date onwards.

14.5.3.3.32 Convert to text

Converts dates to strings.

This derived property has the following features:

Input

Argument Date.

Output String.

Options

String type Specifies how the date should be calculated:

- **Date**: Full date as YYYY-MM-DD
- **Week**: Date binned into weeks and presented as YYYY-MM-D1...D7
- **Month**: Date binned into months and presented as YYYY-MM
- **Quarter**: Date binned into quarters and presented as YYYY-QX
- **Year**: Date binned into years and presented as YYYY

14.5.3.3.33 Date difference

Takes the difference between two dates, and returns it as a numerical value.

This derived property has the following features:

Input

Date 1 Date.

Date 2 Date.

Output Value.

Options None.

14.5.3.3.34 And

Performs a boolean AND operation.

This derived property has the following features:

Input

Value 1 Boolean.

Value 2 Boolean.

Output Boolean.

Options None.

14.5.3.3.35 Or

Performs a boolean OR operation.

This derived property has the following features:

Input

Value 1 Boolean.

Value 2 Boolean.

Output Boolean.

Options None.

14.5.3.3.36 Not

Performs a boolean NOT operation.

This derived property has the following features:

Input

Value Boolean.

Output Boolean.

Options None.

14.5.3.3.37 Convert to text

Converts booleans to strings.

This derived property has the following features:

Input

Value Boolean.

Output String.

Options

False status Specifies the string to appear in case of FALSE (default is "No").

True status Specifies the string to appear in case of TRUE (default is "Yes").

14.5.3.3.38 Convert to color

Converts booleans to a set of colors.

This derived property has the following features:

Input

Argument


Output Color.

Options

Color for 'true'

Color for 'false'

14.5.4 Changing properties of a chart


For each chart type, it is possible to change the name and a number of properties, such as colors, symbols, dot and line sizes, labels, parameters, etc. These properties can be changed with **Plot > Edit active plot properties...** (). Since most properties are specific for each chart type, they are described under [14.4](#).

14.5.5 Creating fits on charts

14.5.5.1 The Add new fit wizard

In the first page, the fit model can be chosen from a list.

In the second page of the wizard, the options for the selected model can be entered.

In case of a scatter chart, a model function can be fitted on the chart. The fit can be created by selecting the scatter chart in the *Chart list* panel and choosing **Plot > Create new fit for active plot...** (). The *Add new fit* wizard that pops up creates the fit in two pages (Figure [14.5.6](#)).

When a fit has been created it appears in the *Chart area* panel and in the *Chart list* panel as a fit type (Figure [14.5.8](#)).

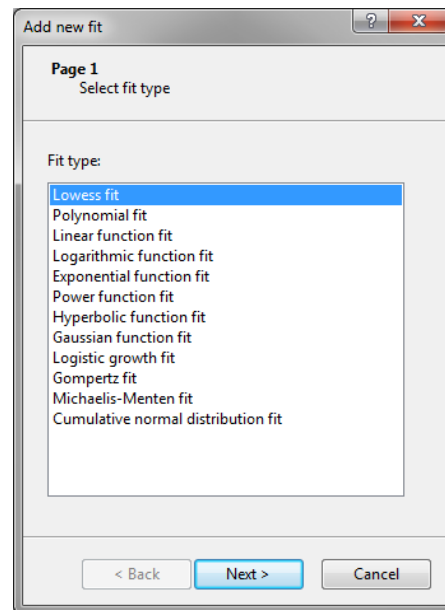


Figure 14.5.6: *Select fit type* wizard page shows the available fit models.

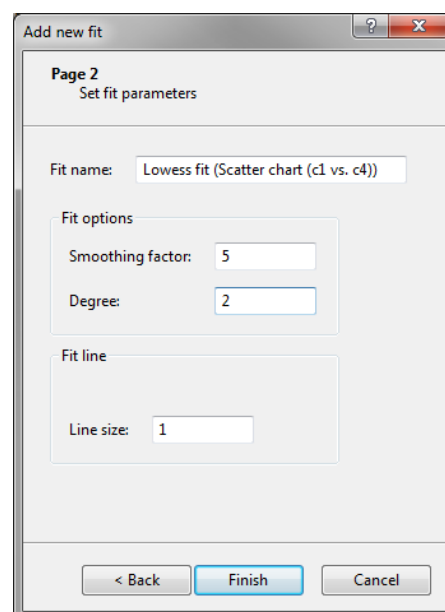


Figure 14.5.7: *Set fit parameters* wizard page: displaying additional options for the selected model.

Once created, a fit can still be edited by selecting it in the *Chart list* panel and choosing **Plot > Edit active plot properties...** (🔧). As a result, the *Polynomial fit properties* dialog box appears (Figure 14.5.9).

This dialog box allows the **Chart name** of the fit to be changed, the fit options to be edited (this part is specific for each fit), the **Line color** to be changed and the **Line size** to be set.

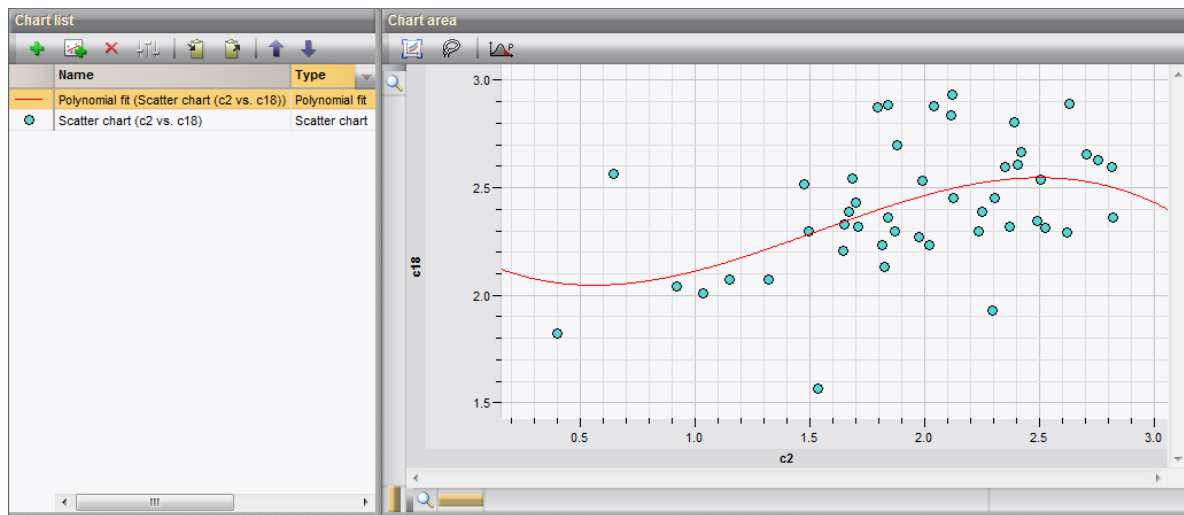


Figure 14.5.8: A polynomial fit of degree 3 on a scatter plot.

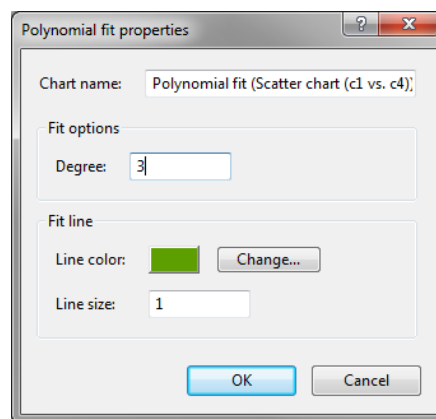


Figure 14.5.9: The *Polynomial fit properties* dialog box for a polynomial fit.

14.5.5.2 Available fit models

14.5.5.2.1 Lowess fit

Calculates a LOWESS fit (LOcally WEighted Scatterplot Smoothing). This method calculates a low-degree, locally weighted polynomial regression through each point of the data set. The polynomial degree and the weights (smoothing) are adjustable.

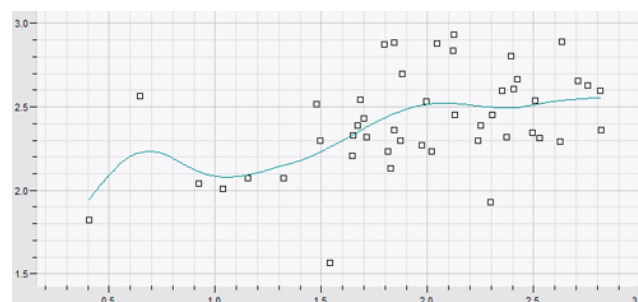


Figure 14.5.10: Lowess function with smoothing factor 10 and degree 2.

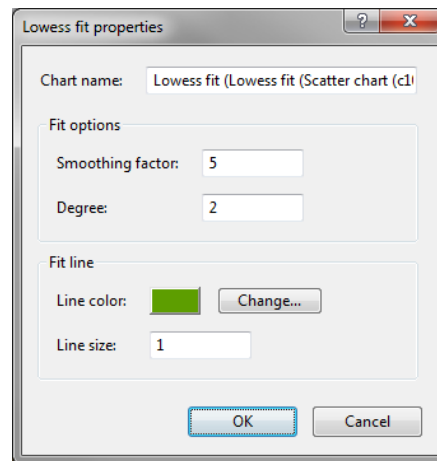


Figure 14.5.11: Lowess fit properties.

This fit has the following properties:

Smoothing factor Determines the smoothness of the LOWESS curve.

Degree Specifies the degree of the local polynomial regression.

14.5.5.2.2 Polynomial fit

A polynomial regression of degree n is given by: $y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n$.

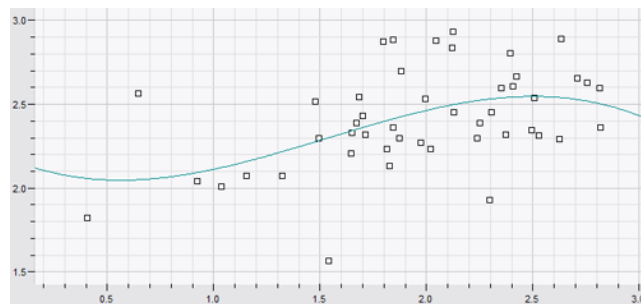


Figure 14.5.12: Polynomial fit of 3rd degree.

This fit has the following options:

Degree The degree or power of the function.

14.5.5.2.3 Linear function fit

A linear regression is given by: $y = A + Bx$.

This fit has the following options:

Force through zero Forces the fit to go through zero.

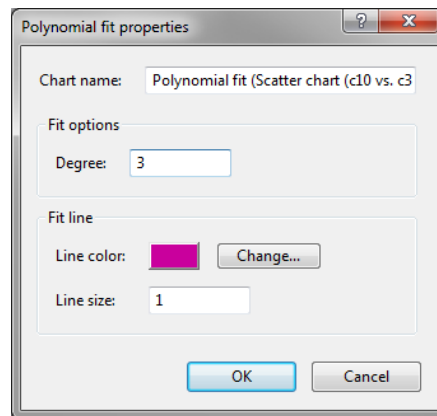


Figure 14.5.13: Polynomial fit properties.

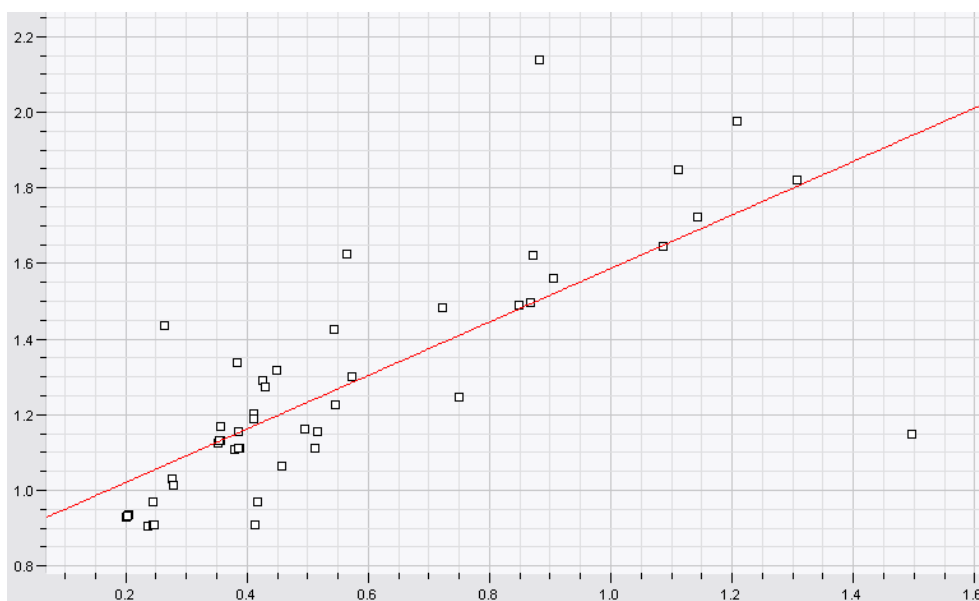


Figure 14.5.14: Linear fit.

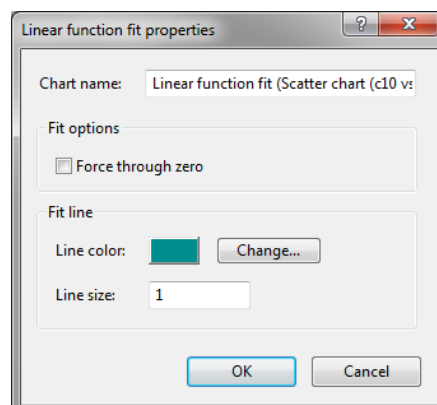


Figure 14.5.15: Linear function fit properties.

14.5.5.2.4 Logarithmic function fit

This fit model is given by the function $y = A + B \log(x)$ with A being the intercept.

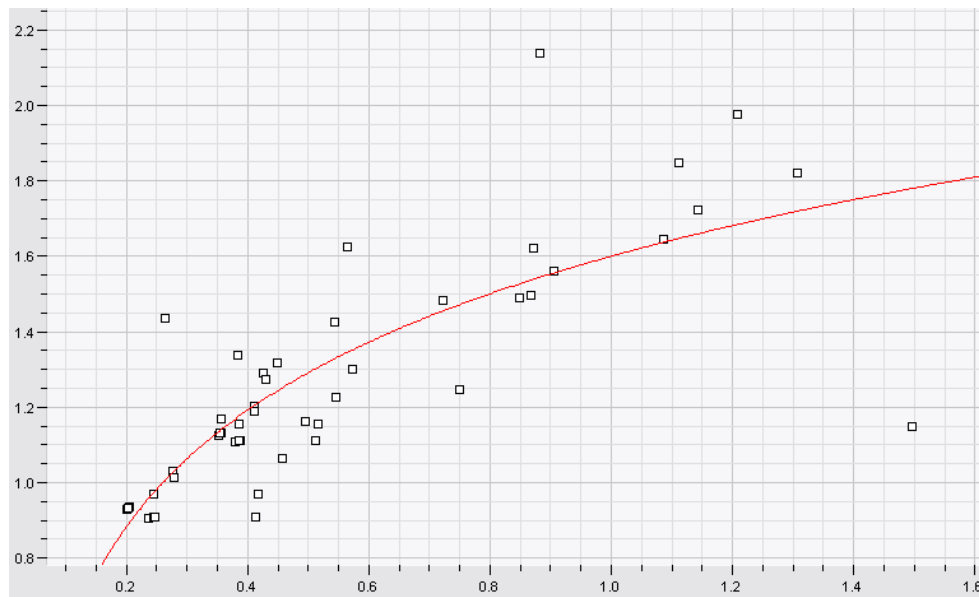


Figure 14.5.16: Logarithmic fit.

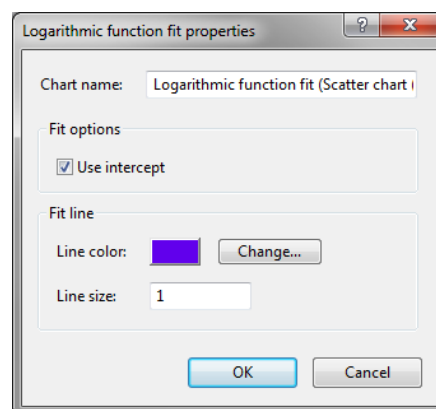


Figure 14.5.17: Logarithmic function fit properties.

This fit has the following options:

Use intercept Allows A to be different from zero.

14.5.5.2.5 Exponential function fit

This fit model is given by the function $y = O + Ae^{rx}$ with O being the offset.

This fit has the following options:

Use offset Allows the offset O to be different from zero.

14.5.5.2.6 Power function fit

This fit model is given by the function $y = O + Ax^B$, with O being the offset.

This fit has the following options:

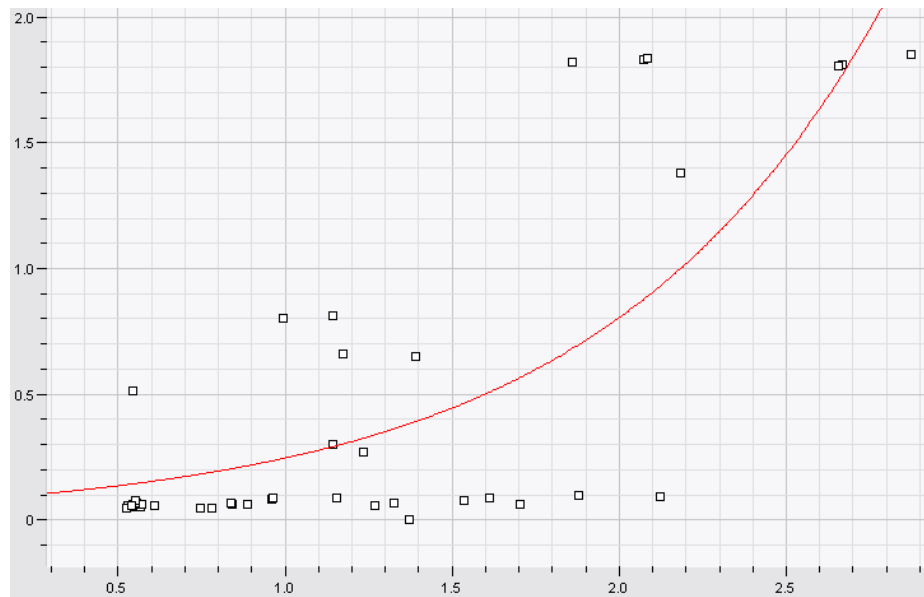


Figure 14.5.18: Polynomial fit of 3rd degree.

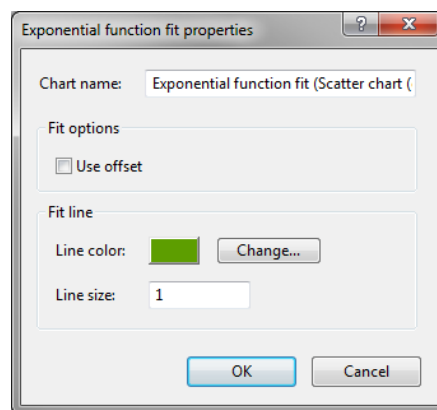


Figure 14.5.19: Exponential function fit properties.

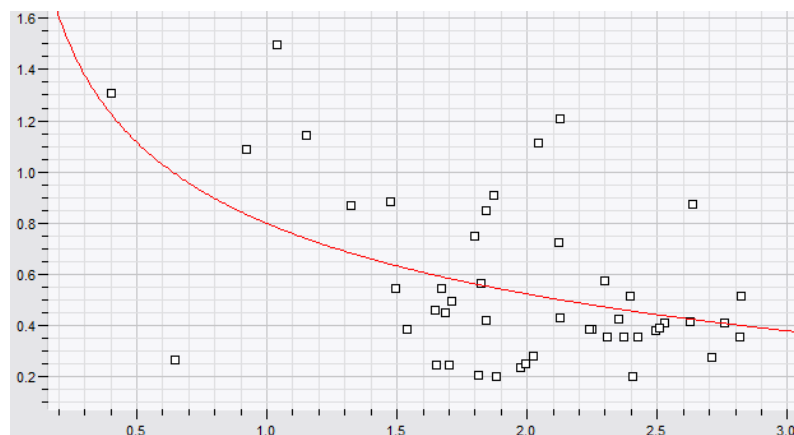


Figure 14.5.20: Power fit.

Use offset Allows the offset O to be different from zero.

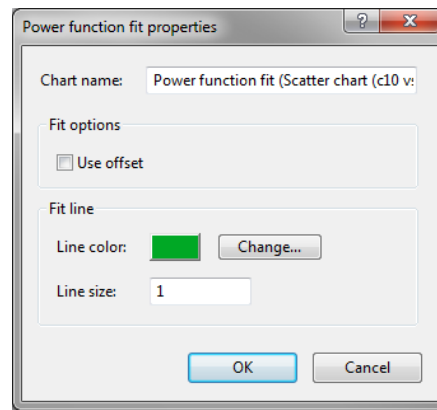


Figure 14.5.21: Power function fit properties.

14.5.5.2.7 Hyperbolic function fit

This fit is given by the function $y = A + \frac{B}{x-C}$ with C being the pole.

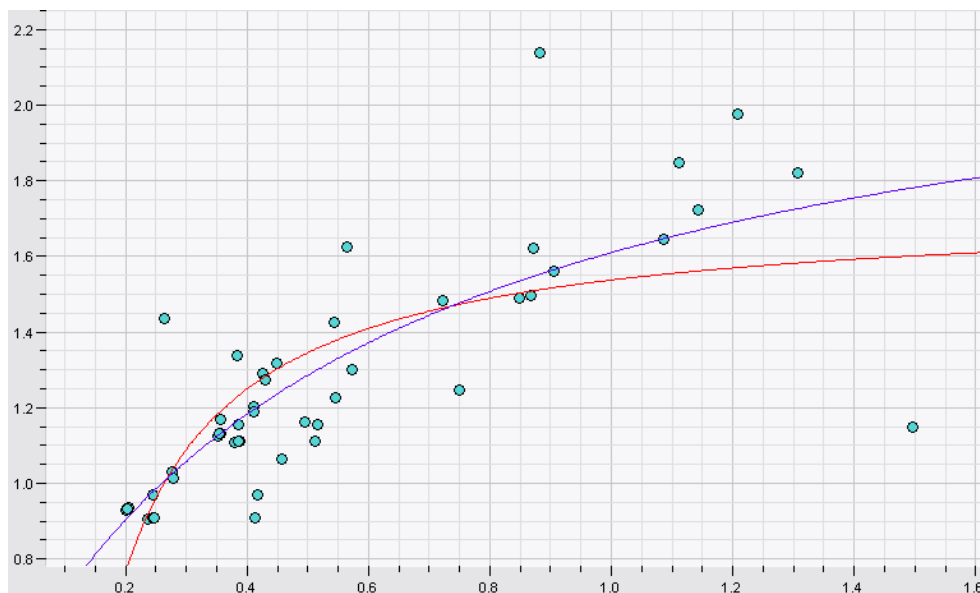


Figure 14.5.22: Hyperbolic fit without asymptote fit (red) and with asymptote fit (blue).

This fit has the following options:

Fit asymptote Allows the pole C to be different from zero.

14.5.5.2.8 Gaussian function fit

This fit is given by the function $y = O + Ae^{-\left(\frac{x-M}{S}\right)^2}$ with O being the offset.

This fit has the following options:

Use offset Allows the offset O to be different from zero.

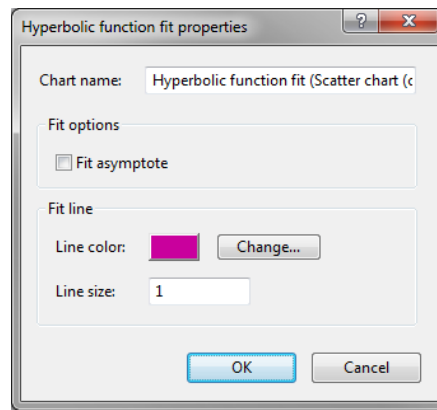


Figure 14.5.23: Hyperbolic function fit properties.

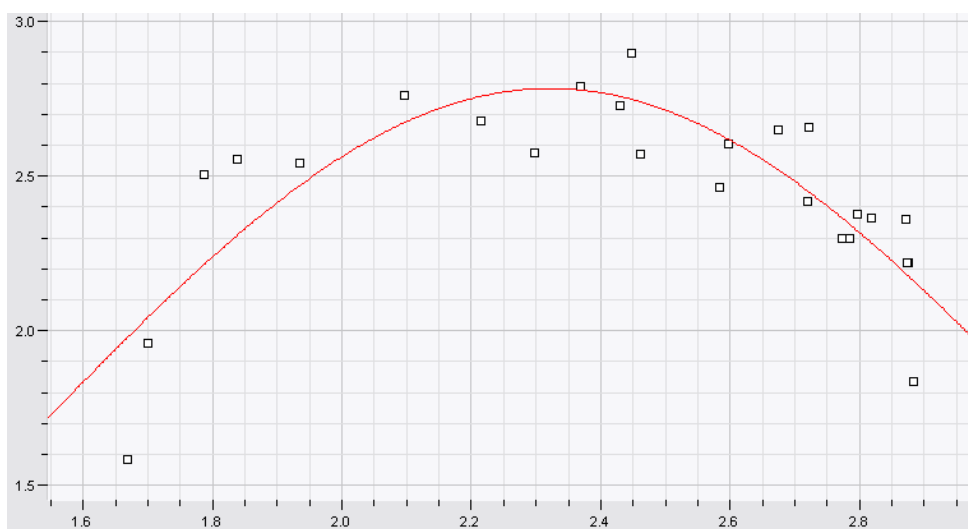


Figure 14.5.24: Gaussian fit.

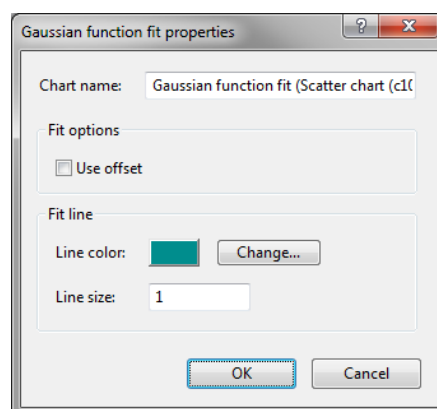


Figure 14.5.25: Gaussian function fit properties.

14.5.5.2.9 Logistic growth fit

This fit model is given by the function $y = A + \frac{C}{[1 + e^{Q-B(x-M)}]^{1/Q}}$ with A being the offset.

This fit has the following options:

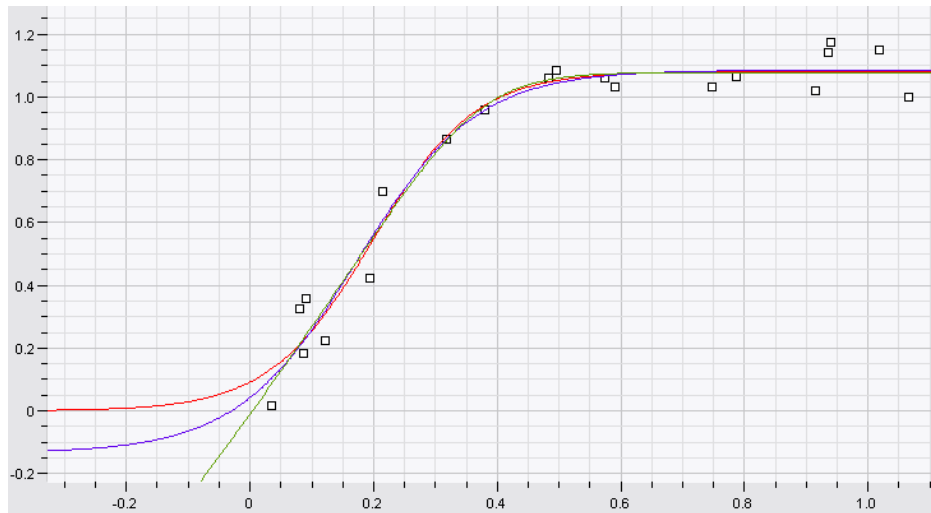


Figure 14.5.26: Logistic growth without offset (red), with offset (blue) and using generalized formula (green).

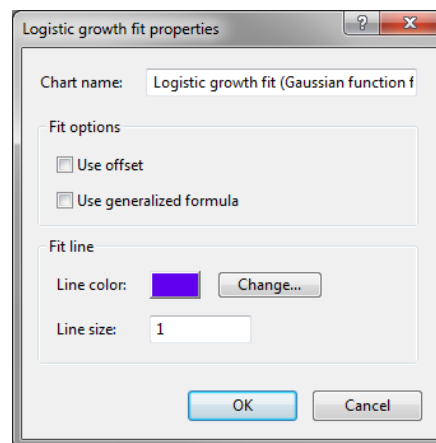


Figure 14.5.27: Logistic growth fit properties.

Use offset Allows the offset A to be different from zero.

Use generalized formula Allows the value Q to be different from zero.

14.5.5.2.10 Gompertz fit

This model fit is given by the function $y = A + Ce^{-[e^{B(x-M)}]}$, with A being the offset.

This fit has the following options:

Use offset Allows the offset A to be different from zero.

14.5.5.2.11 Michaelis-Menten fit

This fit model is given by the function $V_0 = A + \frac{V_{max}[x]}{K_m + [x]}$, with A being the offset.

This fit has the following options:

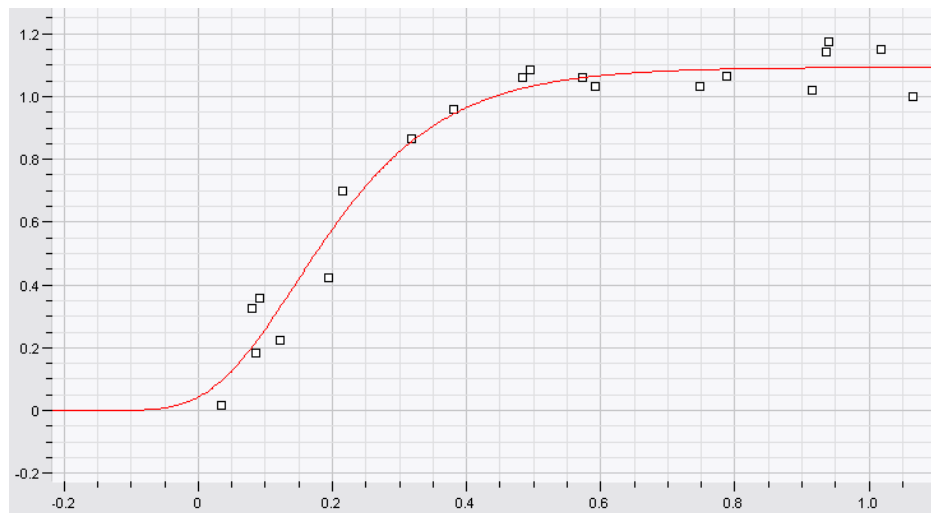


Figure 14.5.28: Gompertz fit.

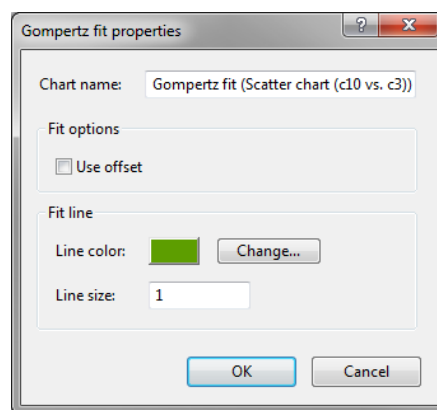


Figure 14.5.29: Gompertz fit properties.

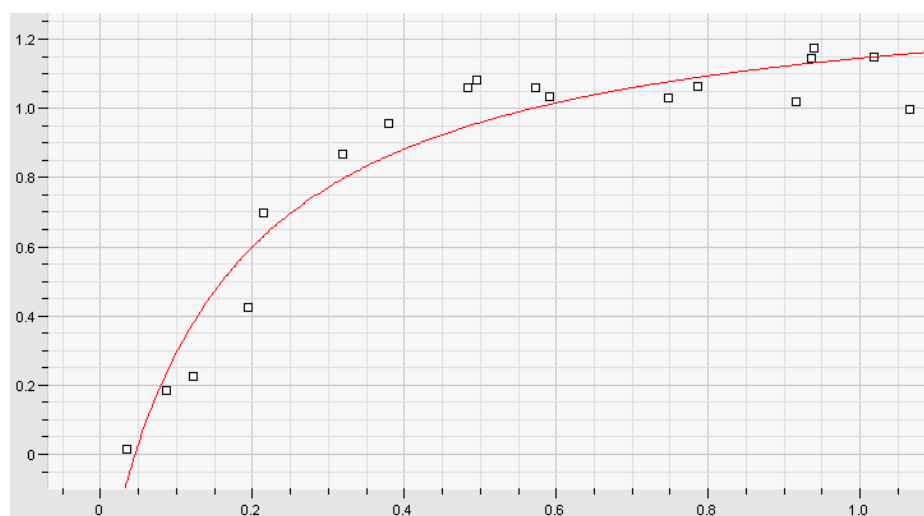


Figure 14.5.30: Michaelis-Menten fit.

Use offset Allows the offset A to be different from zero.

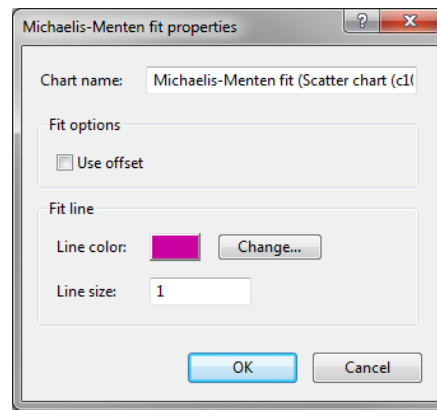


Figure 14.5.31: Michaelis-Menten fit properties.

14.5.5.2.12 Cumulative normal distribution fit

This fit model applies a cumulative distribution of the normal or Gaussian distribution function (see [14.5.5.2.8](#).

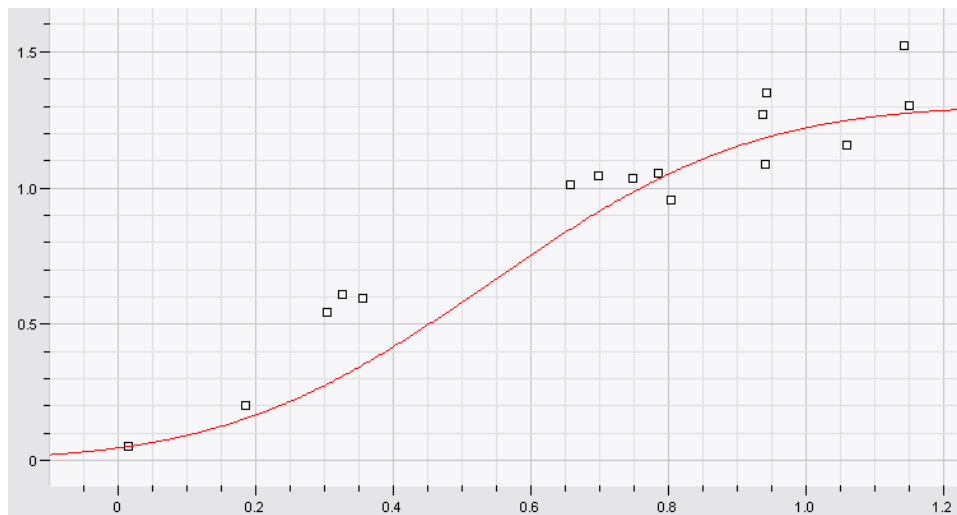


Figure 14.5.32: Cumulative normal distribution fit.

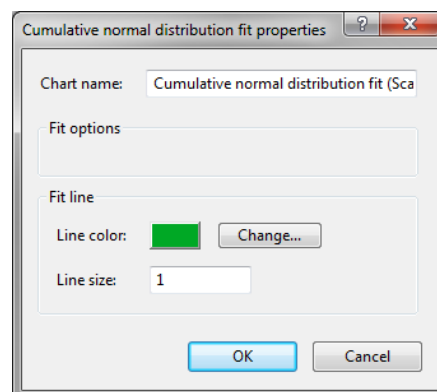


Figure 14.5.33: Cumulative normal distribution fit properties.

This fit has no specific options.

14.5.6 Displaying and managing multiple charts

It is possible to generate multiple charts within the same *Charts and statistics* window. All charts generated within the same window are displayed in the *Chart list* panel (Figure 14.5.34).

Charts that have a compatible visualization are displayed simultaneously in the *Chart area* panel. If the chart list contains incompatible charts, only the chart selected in the *Chart list* panel and the compatible charts are displayed. Charts that are currently displayed are indicated with "Yes" in the 'Visible' column of the chart list. For visible charts, the chart legend is displayed in the leftmost column of the *Chart list* panel.

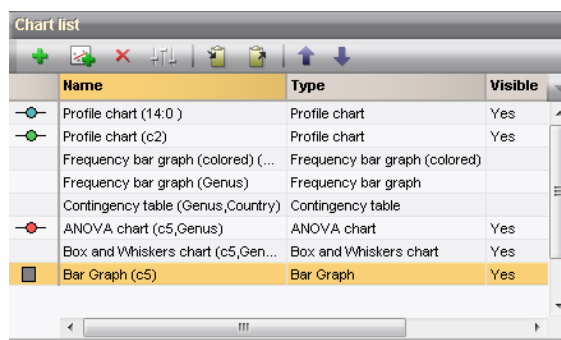


Figure 14.5.34: The *Chart list* panel.

Although charts may be indicated as visible, they may not be visible in reality in case they are hidden behind another chart. For example, a profile chart may be partially or fully hidden behind a bar graph. Even when the chart is selected, it may still be covered by another chart. The display order depends on the position in the chart list: a chart appearing above another chart in the list will be displayed on top of it in the chart area.

To move a chart up or down in the list, use **Plot > Move active plot up** (↑) or **Plot > Move active plot down** (↓), respectively. In case of a combined display of a bar graph and a profile chart, for example, it is recommended to move the profile chart upwards so that it is displayed on top of the bar graph.

A chart can be removed from the chart list using **Plot > Delete active plot** (✖).

14.5.7 Synchronizing selections between chart and database

Selections in the database and in the *Charts and statistics* window are automatically synchronized as long as the data source is in *live* mode (see 14.5.1 and 14.5.9). Selected items are indicated in orange (Figure 14.5.35). For example, in a profile or scatter chart, the dots of selected items are encircled in orange. In case of partial selections such as in frequency bar graphs and contingency tables, the length of the orange selection bars are proportional to the relative number of selected items in the bars or cells.

The selection shown in the chart depends on the data source chosen in the *Create chart* dialog box. If the data source is based on database entries, the selection for the entries is shown in the chart(s). If the data source is based on experiment character values, the selection of characters is shown.



The selection of characters for a character set is shown and can be edited in the *Comparison* window, in the *Experiment data* panel. It can also be displayed and edited in the *Character type* window, to be opened from the *Experiment types* panel in the *Main* window.



Synchronization with database entries cannot be made in case the data source is based on similarity data. The reason is that similarity data has entry pairs as elements, rather than individual entries.

Selections can be made in the *Chart area* panel by holding down the **Shift**-key and dragging a rectangle

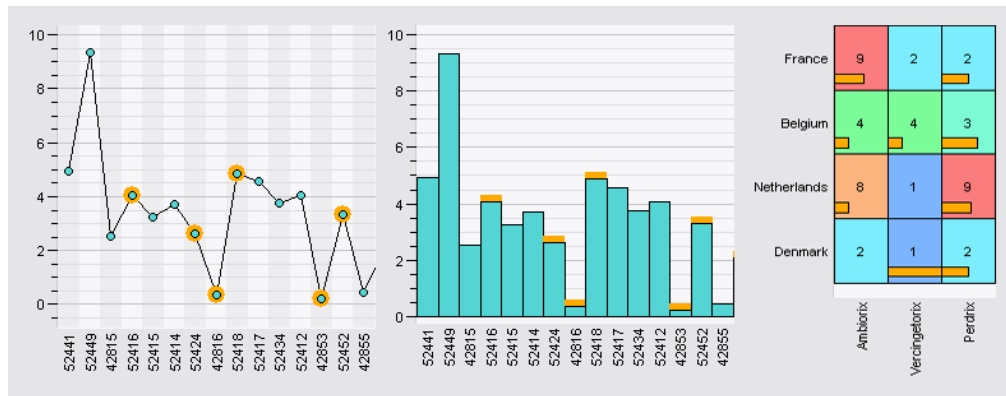


Figure 14.5.35: Examples of partial selections in different chart types.

over the items to select. Individual items can be selected or unselected by holding down the **Ctrl**-key while clicking on the items. In some chart types like scatter charts and profile charts, selections can also be made using the lasso selection tool, which is activated with **Plot > Lasso selection** (🔗). To make selections using the lasso selection tool, the **Shift**-key needs to be held down as well.

With **Plot > Clear selection**, the entire selection on the chart is cleared.



In case the *Charts and statistics* window has been uncoupled from the data source, i.e. if the underlying window has been closed or if the chart has been pasted from elsewhere (see 14.5.9), the chart data are not live anymore and it will not be possible to display or synchronize selections with the database.

14.5.8 Working with chart templates

All charts generated in a *Charts and statistics* window, along with the chart properties and fits defined, can be saved as a chart template with **File > Save as report template...** (⚙️). This causes the *Save chart template* dialog box to pop up (Figure 14.5.36).

The dialog box titled "Save chart template" contains the following fields and buttons:

- Template name: My template
- Template group:
- Description:
- OK button
- Cancel button

Figure 14.5.36: The *Save chart template* dialog box.

A unique name for the template should be entered as **Template name**.

Optionally, a name can be entered for a **Template group**. When a template group name is entered, the template group will appear as a node in the *Create chart* dialog box tree (Figure 14.3.3). Templates with the same group name will appear in the same node in the tree.

A description can optionally be specified that will appear in the right panel of the *Create chart* dialog box when selecting the template from the list.

An existing template can be loaded when a new *Charts and statistics* window is created by selecting the template name from the tree in the *Create chart* dialog box (Figure 14.3.3). All charts saved in the template

will appear for the entries and/or characters in the current data source.

To remove an existing template, create a *Charts and statistics* window that uses the template and select **File** > **Delete plot template...**

14.5.9 Copying and exporting charts

Charts can be copied between chart windows. A chart copied from another *Charts and statistics* window is decoupled from its data source and is therefore not live anymore. This means that the database selections are not indicated on the chart and that only those properties that are in use on the chart are still available. Derived properties can still be created (see 14.5.3). A chart can be copied by selecting it in the chart list panel and choosing **Plot** > **Copy active plot** (📄). The copied chart can be pasted in a *Charts and statistics* window with **Plot** > **Paste plot** (📄).

Chart data can also be copied as tab-delimited text data with **Plot** > **Copy active plot data to clipboard**. The data is stored on the clipboard and can be pasted in a text editor or a spreadsheet program.

A selected chart and its compatible charts can be printed using **File** > **Print...** The dialog box that appears is the standard Windows Print dialog box, allowing you to choose a printer and change the properties.

The chart graphics for the selected chart can be exported with **File** > **Save as...** This calls the *Export image* dialog box, as shown below.

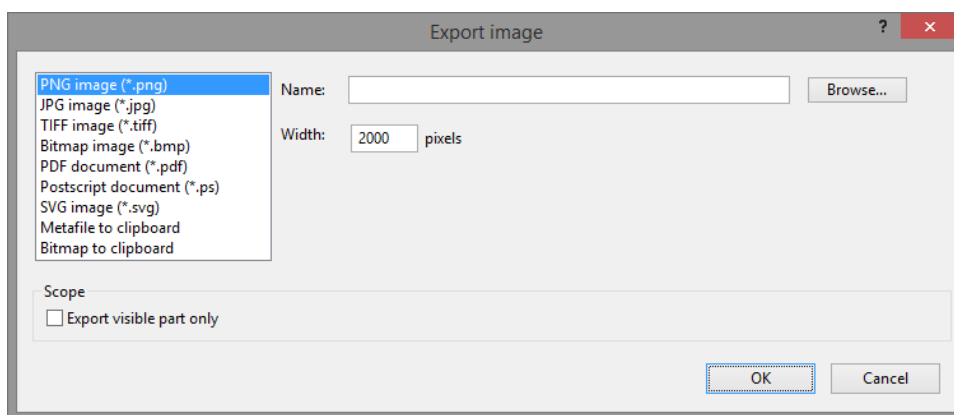


Figure 14.5.37: The *Export image* dialog box.

This dialog box allows you to export graphical information to a file or to the Windows clipboard in one of several available formats. In case a file is exported, a file **Name** should always be entered or browsed for via the <**Browse**> button. Exported files will open in their default editor. Information on the Windows clipboard can be pasted into other applications. Following export options are available:

- **PNG image (*.png):** exports to a Portable Network Graphics (PNG) file. PNG is a raster graphics file format that supports lossless data compression. A **Name** and **Width** (in pixels) should be specified; the height will be determined automatically.
- **JPG image (*.jpg):** exports to a Joint Photographic Experts Group (JPEG) file. JPEG or JPG is a raster graphics file format that uses a lossy data compression. A **Name** and **Width** (in pixels) should be specified, as well as a **Quality** parameter. With the latter, a tradeoff can be obtained between storage size and image quality.
- **TIFF image (*.tiff):** exports to a Tagged Image File Format (TIFF) file. TIFF is a raster graphics file format with optional lossless data compression. A **Name** and **Width** (in pixels) should be specified.

- **Bitmap image (*.bmp)**: exports to a BMP bitmap image or device independent bitmap (DIB) file. BMP is a raster graphics image file format used to store bitmap digital images, independently of the display device. A **Name** and **Width** (in pixels) should be specified.
- **PDF document (*.pdf)**: exports to a Portable Document Format (PDF) file. PDF is a file format used to present documents in a manner independent of application software, hardware, and operating systems. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **Postscript document (*.ps)**: exports to a PostScript (PS) file. PostScript is a computer language for creating vector graphics. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **SVG image (*.svg)**: exports to a Scalable Vector Graphics (SVG) file. SVG is an XML-based vector image format for two-dimensional graphics. A **Name** should be specified.
- **Metafile to clipboard**: copies the graphics as Windows enhanced metafile to the clipboard. Enhanced metafile is the standard clipboard exchange format between native Windows applications.
- **Bitmap to clipboard**: copies the graphics as a bitmap to the Windows clipboard. The **Width** (in pixels) should be specified.

Under **Scope**, one can choose to export the complete graphic or only the part that is currently displayed by unchecking or checking the **Export visible part only** option, respectively.

14.5.10 Pivot tables

Similar to many spreadsheet programs, the chart tools in BioNumerics offer *pivot tables* as possible data source. Using a pivot table, data from other sources can be summarized and visualized in a way that the original data sources do not allow. It offers different kinds of aggregation methods such as counting, summarizing and averaging of the original data. The aggregated results are displayed in a second table (the pivot table), on the basis of which charts can be plotted. Pivot tables can also be used for creating unweighted cross tabulations, in which categorical data are summarized into a contingency table.


To create a pivot table, first make a selection of the data sources that should be included. Next, select **Dataset > Create pivot table for selected properties...** to display the *Create pivot data set* wizard (see Figure 14.5.38).

The **Pivot property**, i.e. the data that will be displayed in the columns of the pivot table, should be selected in this dialog box. Only selected data sources of type string (text) are shown in the list.

Press <**Next**> to proceed to the *Summary method* wizard page (see Figure 14.5.39).

Three summarization methods are offered: **Average**, **Total** and **Count**. Please note that only numerical data sources can be summarized.

Pressing <**Finish**> will display the pivot table in the *Data set* window (see Figure 14.5.40).

This window contains the calculated pivot table, which can be sorted according to the highlighted column with **Edit > Sort by highlighted column**. The pivot table can be exported to text or MS Excel with **File > Save** or plotted in a new *Charts and statistics* window with **Edit > Chart and statistics...** (.

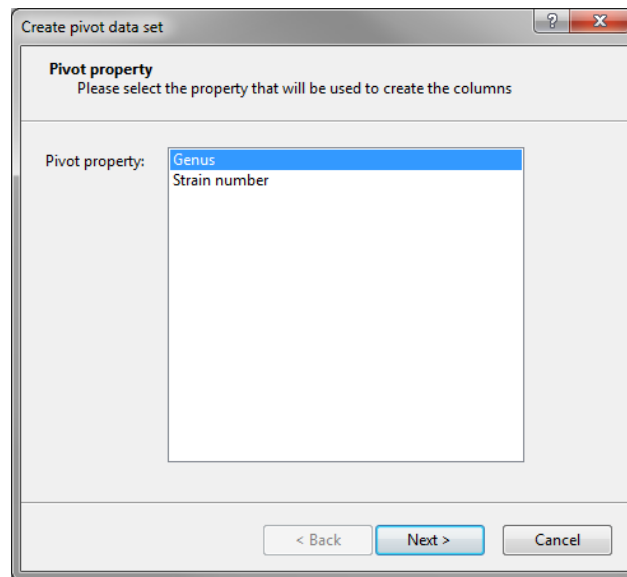


Figure 14.5.38: The *Pivot property* wizard page.

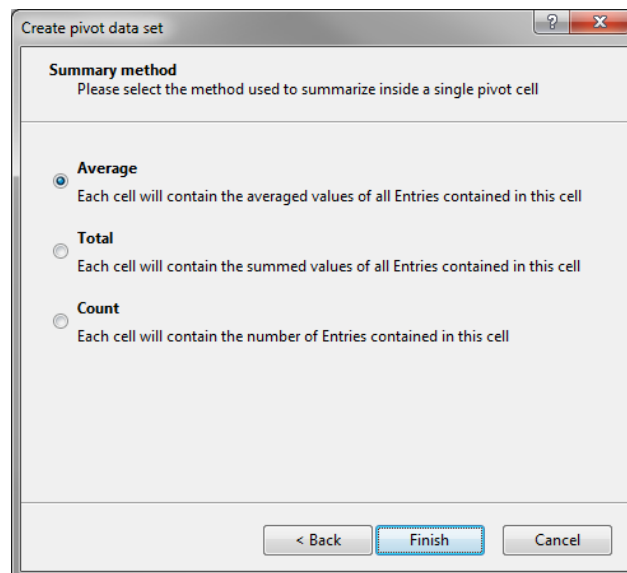


Figure 14.5.39: The *Summary method* wizard page.

Data set grid window

Identifier	Species	Count	Ambiorix (c3)	Vercingetorix (c3)	Perdrix (c3)
SYLVESTRIS	SYLVESTRIS	15	0.7868	0	0
ABERRANS	ABERRANS	4	1.085	0	0
PALUSTRIS	PALUSTRIS	3	0	2.391667	0
SP.	SP.	6	1.1245	0	1.21
NEMOROSUM	NEMOROSUM	3	0	2.344333	0
AQUATICUS	AQUATICUS	2	0	2.425	0
PSEUDOARCHAEUS	PSEUDOARCHAEUS	14	0	0	0.970929



Figure 14.5.40: The *Data set* window, showing a pivot table.

Chapter 14.6


Tutorials

14.6.1 Displaying character sets as bar graph

This example illustrates how character sets for one or more entries can be displayed as bar graphs.

- 1.1 Open the database **DemoBase Connected**.
- 1.2 Select a few entries the *Main* window (not those labeled as STANDARD in the *Genus* field) using the **Ctrl-key**. Alternatively, use the **space bar** to select a highlighted entry or click the ballot box next to the entry.
- 1.3 Highlight the *Comparisons* panel in the *Main* window and select **Edit > Create new object...** () to create a new comparison for the selected entries.
- 1.4 To clear the selection, press the **F4**-key.
- 1.5 In the *Comparison* window, select **Statistics > Chart and statistics...** (, **F7**).
- 1.6 Select **PhenoTest characters** under **Experiment character values** in the dialog box that pops up and press **<OK>**.
- 1.7 Choose **Bar Graph** in the next step and press **<Next>**.
- 1.8 Choose an entry from the list and press **<Next>**.
- 1.9 In the final step choose **Name** as **Label** and press **<Finish>**.

A bar graph is created in the *Chart area* panel of the *Charts and statistics* window displaying the character set of the selected entry. The rectangular bar of each character has a length proportional to the character value. We will now add a second bar graph to compare the characters between two entries.

- 1.10 In the *Data source overview* panel, select another entry under **Entry values**.
- 1.11 Select **Plot > Add new plot from selected properties...** ()
- 1.12 Click on **Bar graph** and press **<Next>**.
- 1.13 Press **<Finish>** again to obtain the bar graph.

The result looks as in Figure 14.6.1. More entries can be added to the chart in the same way.

- 1.14 Close the *Charts and statistics* window with **File > Exit**.
- 1.15 Close the *Comparison* window with **File > Exit** without saving the comparison.

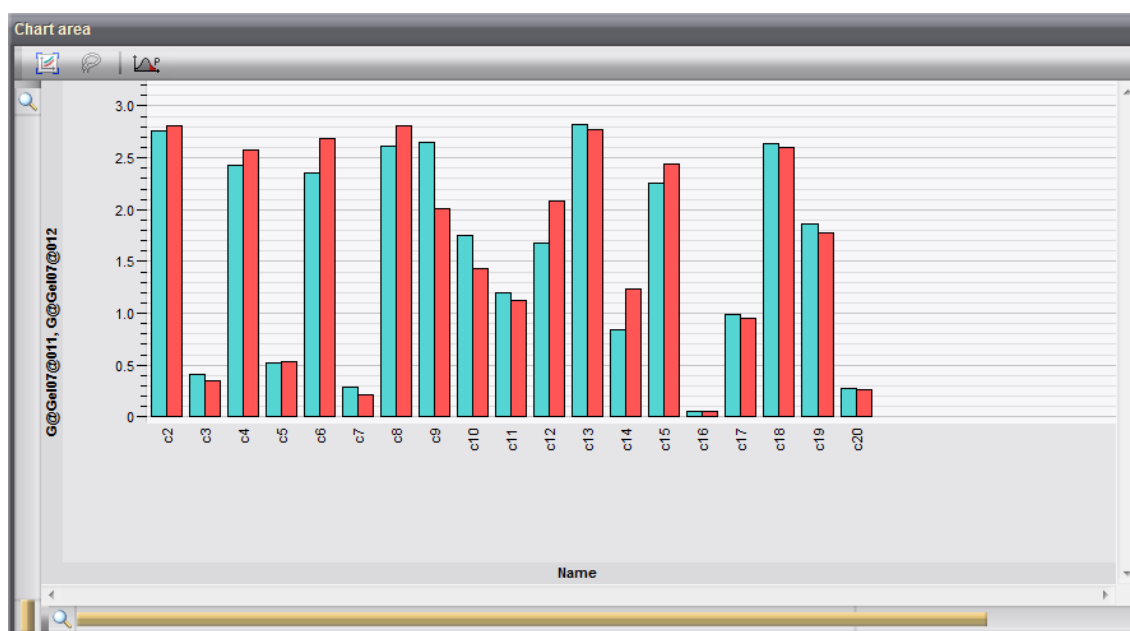


Figure 14.6.1: Bar graph displaying the character sets of two entries.

14.6.2 Plotting densitometric profiles

This example illustrates how densitometric curves from fingerprints can be plotted as profile charts.

- 2.1 Open the database **DemoBase Connected**.
- 2.2 In the *Main* window, double-click on an entry to open its *Entry* window.
- 2.3 Select **File > Charts and statistics...** (📊, F7).
- 2.4 In the *Create chart* dialog box, select **RFLP1 Curve** under **Fingerprint experiments** and press <OK>.
- 2.5 Select **Profile chart** in the next step and press <Next>.
- 2.6 Highlight **Value** in the next step and press <Next> and <Finish>.

A profile chart of densitometric values from the **RFLP1** profile for the selected entry appears in the *Chart area* panel (Figure 14.6.2).

We will now use the copy and paste functions to display another curve in the same *Charts and statistics* window.

- 2.7 Bring the *Main* window back into focus without closing the *Charts and statistics* window.
- 2.8 Double-click on another entry in the *Main* window to open its *Entry* window.
- 2.9 Select **File > Charts and statistics...** (📊, F7).
- 2.10 In the *Create chart* dialog box, select **RFLP1 Curve** under **Fingerprint experiments** and press <OK>.
- 2.11 Select **Profile chart** in the next step and press <Next>.
- 2.12 Highlight **Value** in the next step and press <Next> and <Finish>.

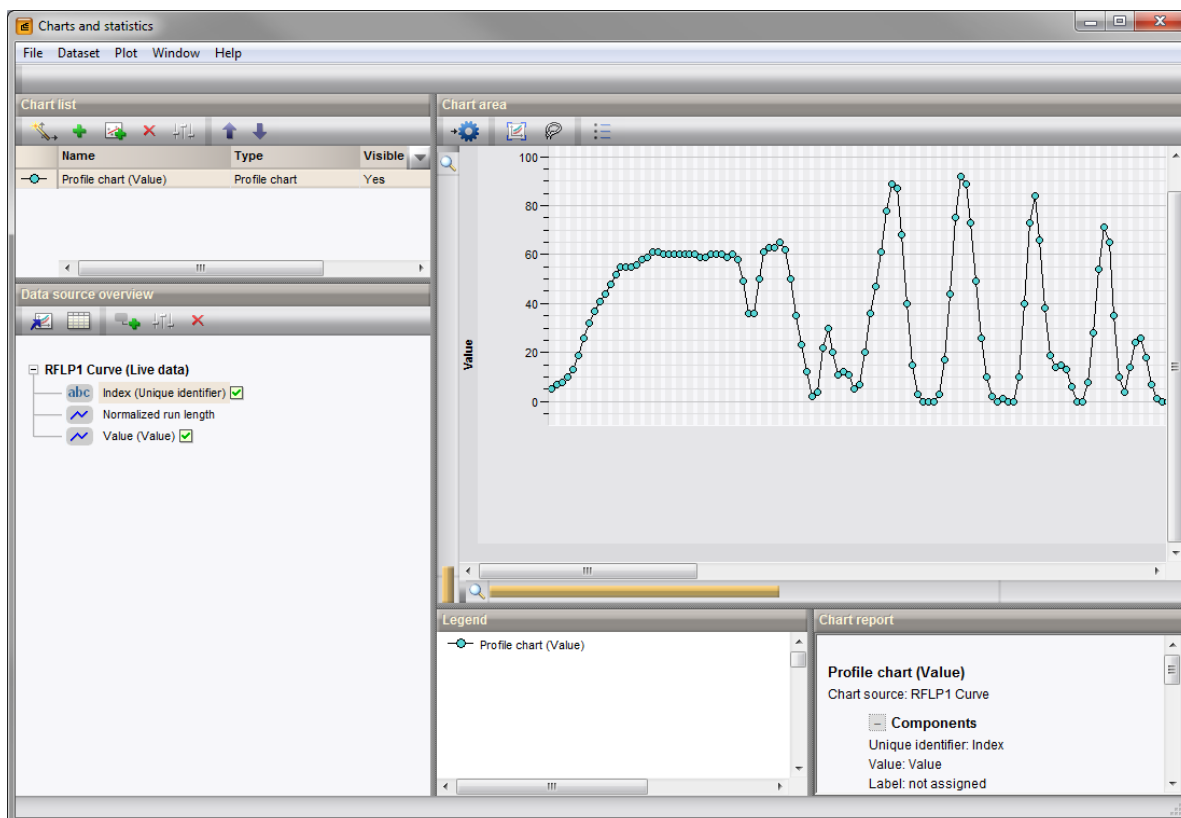


Figure 14.6.2: Profile chart from curve data.

A profile chart of densitometric values from the **RFLP1** profile for the selected entry appears in the *Chart area* panel.

- 2.13 Select **Plot > Copy active plot** (📄). Bring the first *Charts and statistics* window into focus and select **Plot > Paste plot** (📄).

The *Charts and statistics* window now contains two charts: the original one (cyan dots) and the pasted one (red dots) (see Figure 14.6.3). Note that the original chart has a live data source whereas the pasted chart does not, because it is just a copy which is not linked anymore to its parent *Entry* window.

- 2.14 Close the *Charts and statistics* window with **File > Exit** and close the *Entry* window.

14.6.3 Plotting a character for multiple entries

This example illustrates how a character can be plotted as a profile chart for a number of entries using the **DemoBase Connected** demonstration database.

- 3.1 Open the database **DemoBase Connected**.
- 3.2 Select all entries in the *Main* window with **Ctrl+A**, and unselect those labeled as STANDARD in the *Genus* field.
- 3.3 Highlight the *Comparisons* panel in the *Main* window and select **Edit > Create new object...** (+) to create a new comparison for the selected entries.
- 3.4 In the *Comparison* window, select **Statistics > Chart and statistics...** (📄, F7).
- 3.5 Select **Comparison entries** in the dialog box that pops up and press <OK>.

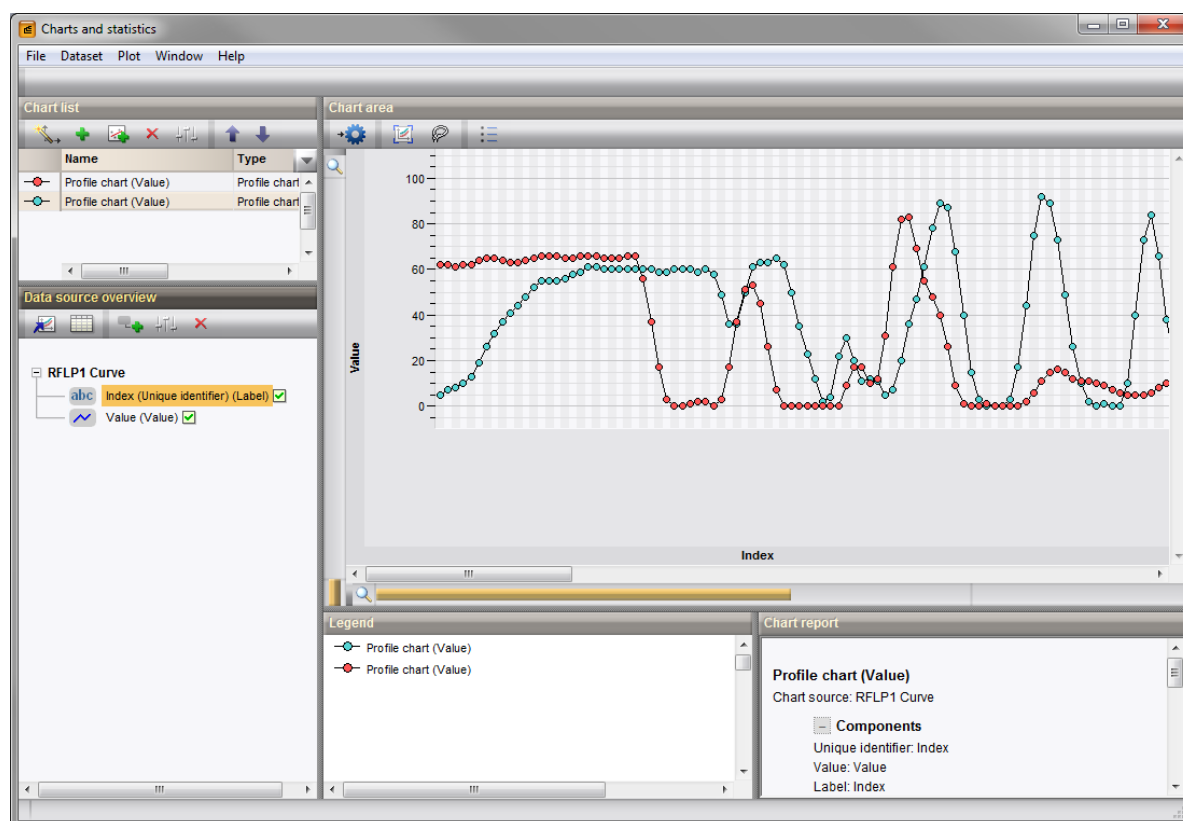


Figure 14.6.3: Two profile charts.

3.6 Select **Profile chart** in the next step and press <Next>.

3.7 In the next step choose **PhenoTest: Values: c17** and press <Next>.

3.8 In the last step choose **Entry information fields: Genus** as **Label** and press <Finish>.

The value for character c17 is now displayed for all entries in the *Chart area* panel of the *Charts and statistics* window. The label **Genus** identifies the entries on the chart.

A chart can be sorted according to a selected string or value property. As an example, we will sort the chart according to the character it is based upon.

3.9 Right-click on **c17 (Value)** in the *Data source overview* panel under **PhenoTest** and select **Use as Chart Sort** from the floating menu.

Now it becomes clear that there is a correlation between the intensity of the character c17 and the genus. Alternatively, the genus name can also be used as a sort property:

3.10 Right-click on **Genus (Label)** in the *Data source overview* panel and select **Use as Chart Sort** from the floating menu.


Ideally, we would like to see the genus and species information as chart labels, rather than only the genus name. Since only one chart label can be displayed at a time, we will create a *derived property*, containing the first character of the genus name and the full species name.

3.11 Highlight **Genus** in the *Data source overview* panel (probably indicated as **Genus (Label) (Sort)**) and select **Dataset > New derived property...** (🔧).

3.12 In the *Create derived property* wizard, select **Abbreviate** and press <Next>.

3.13 Leave the options on the second page to their defaults and press <Finish>.

A derived property **Abbreviate (Genus)** now branches off from the **Genus** property.

- 3.14 Select the derived property **Abbreviate (Genus)** and while holding down the **Ctrl-key**, select **Species**.
- 3.15 Select **Dataset > New derived property...** () to create a new derived property.
- 3.16 Select **Merge two strings** to concatenate the abbreviated genus name and the species name into a new string and press **<Next>**.
- 3.17 In the next page, choose **Abbreviate (Genus)** as **String 1** and **Species** as **String 2**. As **Template**, replace the comma between ^1 and ^2 by a space so that the template looks as "**^1 ^2**" (see Figure 14.6.4).

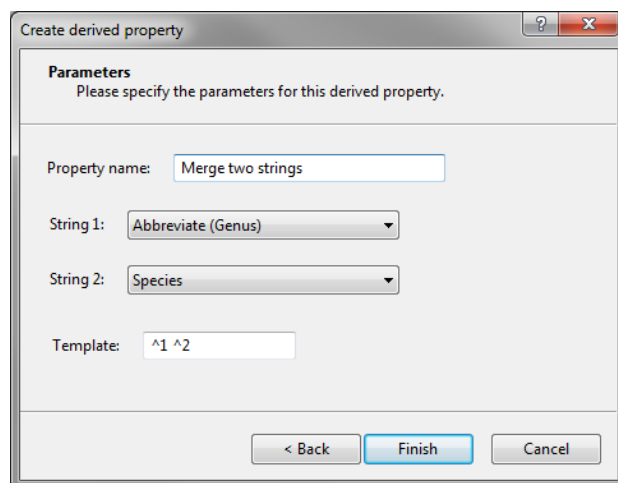


Figure 14.6.4: Merge two strings.

- 3.18 Press **<Finish>** to create the new derived property.
- 3.19 Right-click on the new derived property **Merge two strings** and select **Use as Chart label** from the floating menu.

This derived property can also be used to sort the chart:

- 3.20 Right-click on the derived property **Merge two strings** and select **Use as Chart Sort** from the floating menu.

The profile chart can further be supplied with additional information fields. For example, a *chart point label* can be added as follows:

- 3.21 Right-click on **Strain number** in the *Data source overview* panel and select **Use as Chart Point label** from the floating menu.

The strain numbers are now displayed as additional information on top of the chart points. The current profile chart displays just one character for the selected set of entries. The legend shows the symbol and color used for this chart (see Figure 14.6.5).

By selecting additional numerical properties as chart values, a profile chart can display multiple characters in the same graph:

- 3.22 Right-click on **c19** in the *Data source overview* panel under **PhenoTest** and select **Use as Chart Value**.

The chart now displays a profile for the two characters. The legend still lists one profile chart, but containing multiple components. Alternatively, you can also display the second character profile in a separate chart, which has the advantage that you can assign different colors or symbols to the profiles.

- 3.23 First, remove character c19 from the current profile by right-clicking on **c19 (Value)** in the *Data source overview* panel under **PhenoTest** and selecting **Use as Chart Value** from the floating menu.

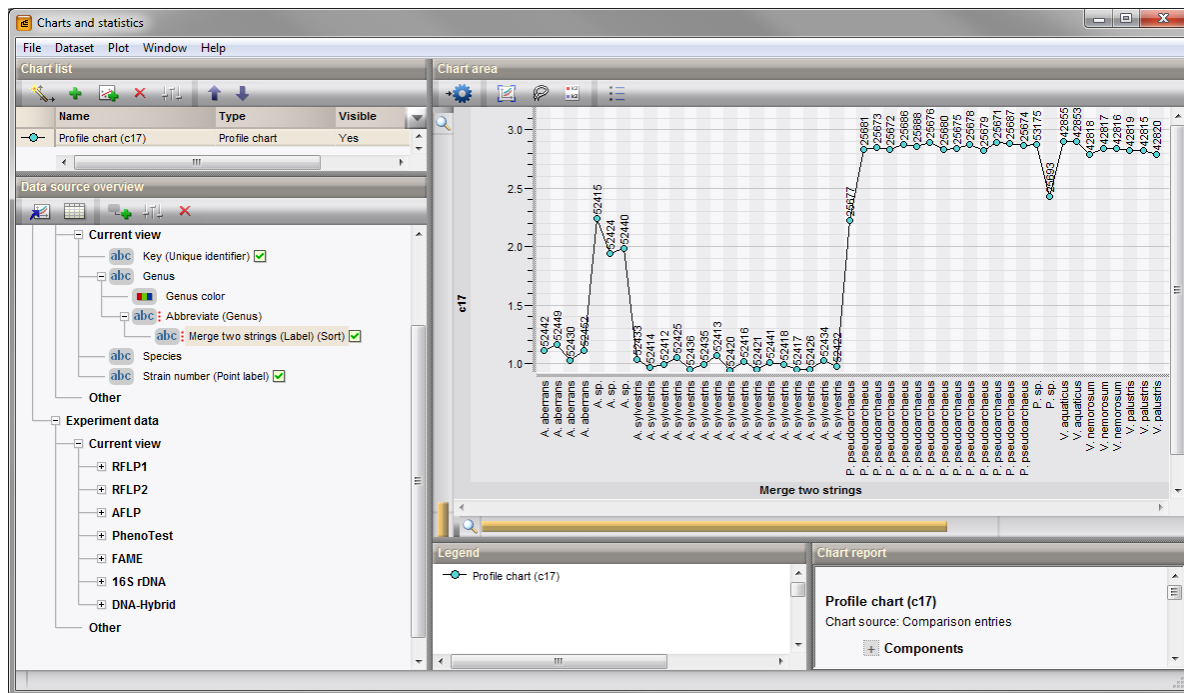


Figure 14.6.5: One profile chart.

3.24 Click on *c19*, hold down the **Ctrl**-key and click on the derived property *Merge two strings*.

3.25 Create a new chart with **Plot > Add new plot from selected properties...** (+).

3.26 Select **Profile chart** and press <Next>.

3.27 Press <Finish> to create the profile chart.

The chart area now looks as in Figure 14.6.6 and the two characters are listed with a different color in the legend panel.

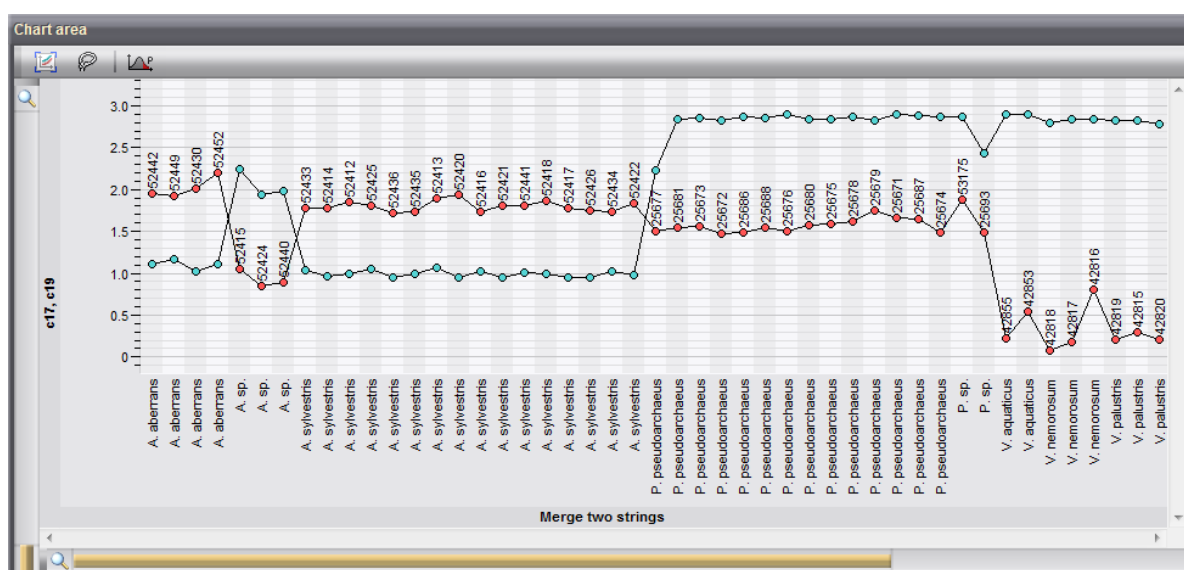




Figure 14.6.6: Profile charts of two characters over a set of entries.

3.28 Close the *Charts and statistics* window with **File > Exit**.

14.6.4 Creating a scatter chart for two characters

This example illustrates how two characters can be compared in a scatter chart for a number of entries. The **DemoBase Connected** demonstration database is used.

- 4.1 Open the database **DemoBase Connected**.
- 4.2 Select all entries in the *Main* window except those labeled as STANDARD in the *Genus* field.
- 4.3 Highlight the *Comparisons* panel in the *Main* window and select **Edit > Create new object...** () to create a new comparison for the selected entries.
- 4.4 In the *Comparison* window, select **Statistics > Chart and statistics...** (, F7).
- 4.5 Select **Comparison entries** in the dialog box that pops up and press <OK>.
- 4.6 Select **Scatter chart** in the next step and press <Next>.
- 4.7 In the next step choose **PhenoTest: Values: c17** as *X axis* and **PhenoTest: Values: c19** as *Y axis* and press <Next>.
- 4.8 In the last step, select **Entry information fields: Strain number** as *First Label* and press <Finish>.

The two characters are now displayed in an X-Y coordinate system with c17 as X axis and c19 as Y axis. Each dot in the scatter chart represents an entry plotted according to its respective values for c17 and c19. It can be seen from this chart that the scatter for these two characters is not random.

The *Entry* window can be revealed from the chart area by double-clicking on a dot. Additionally, it is also possible to select entries in the chart area using **Ctrl+click** or by holding down the **Shift**-key while dragging the mouse arrow over the dots to select.

In dense areas with on the scatter chart, the **Strain number** information may become illegible, and therefore using symbols or colors may be more informative.


- 4.9 First, remove the entry labels from the scatter chart by right-clicking on **Strain number** in the *Data source overview* panel and selecting **Use as Chart First Label**.
- 4.10 Right-click on **Group colors** under **Group names** in the *Data source overview* panel and select **Use as Chart Color** from the floating menu.

The chart area now looks as in Figure 14.6.7.

- 4.11 Close the *Charts and statistics* window with **File > Exit**.

14.6.5 Creating a contingency table

This example illustrates how a set of entries can be compared based upon two string variables, each dividing the entry set into a number of categories. The entry numbers per combination of categories are displayed in a contingency table. Database **DemoBase Connected** is used as a basis and some extra field information is imported.

- 5.1 Download the file `Chart_data.zip` from the Applied Maths website and extract the content.
- 5.2 Open the database **DemoBase Connected**.
- 5.3 Select **File > Import...** (, **Ctrl+I**) in the *Main* window to call the *Import* dialog box.

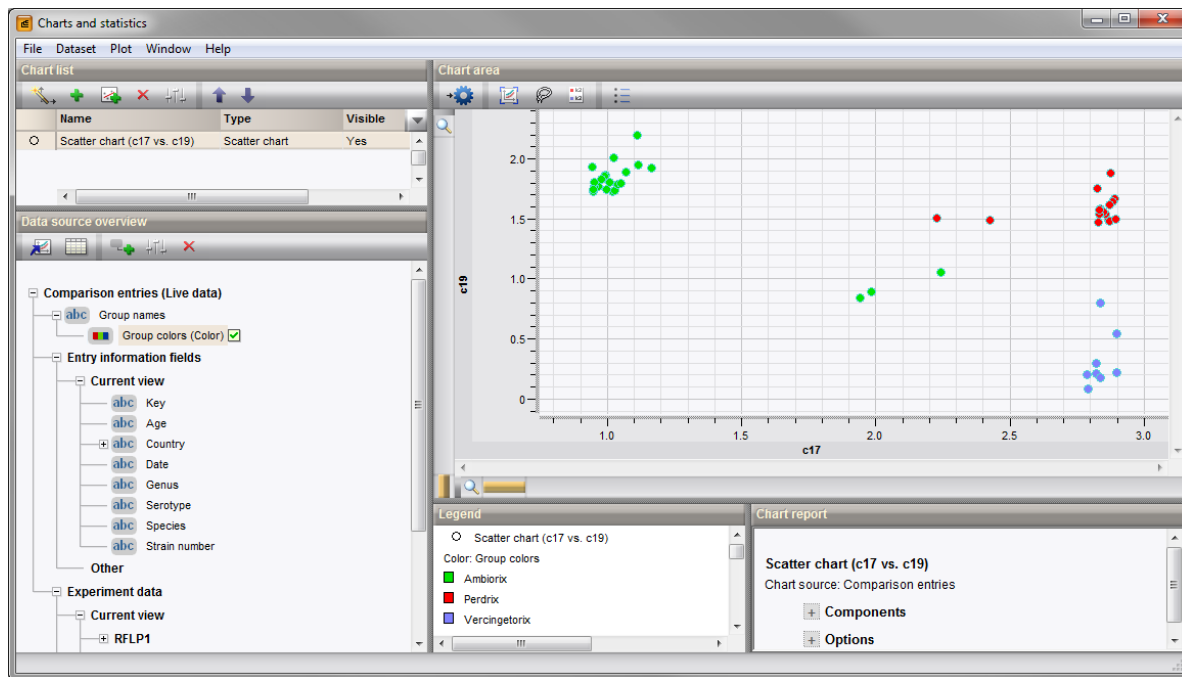


Figure 14.6.7: Scatter plot of two characters.

5.4 Choose the option **Import fields (text file)** under the **Entry information data** item in the tree and click **<Import>**.

5.5 Browse for the downloaded and unzipped Chart data.txt file and press **<Next>**.

5.6 In the **Import rules** dialog box highlight the row that corresponds to **Key** and press **<Edit destination>**.

5.7 In the **Edit data destination** dialog box, highlight **Key** and press **<OK>**.

5.8 Select the fields **Age**, **Date**, **Country** and **Serotype** using the **Ctrl-key** and press **<Edit destination>** again.

5.9 In the **Edit data destination** dialog box, highlight **Entry info field** and press **<OK>**.

5.10 In the **Create new** dialog box that appears, leave the default names unaltered and press **<OK>** and confirm the action.

The **Import rules** dialog box should now look like in Figure 14.6.8.

5.11 Press **<Next>** to move to the next page and press **<Finish>**.

5.12 Enter **Chart data** as name for the template and press **<OK>**.

5.13 Make sure the created template is selected and press **<Next>** to move to the next page.

5.14 Press **<Finish>**.

The entry information is imported and displayed in the **Database entries** panel.

5.15 In the **Main** window, make sure all entries are selected except the "STANDARD" entries. With this selection, create a **Charts and statistics** window with **Analysis > Chart and statistics...** (🖨️, F7).

5.16 Select **Currently selected entries** and press **<OK>**.

5.17 Select **Contingency table** in the next step and press **<Next>**.

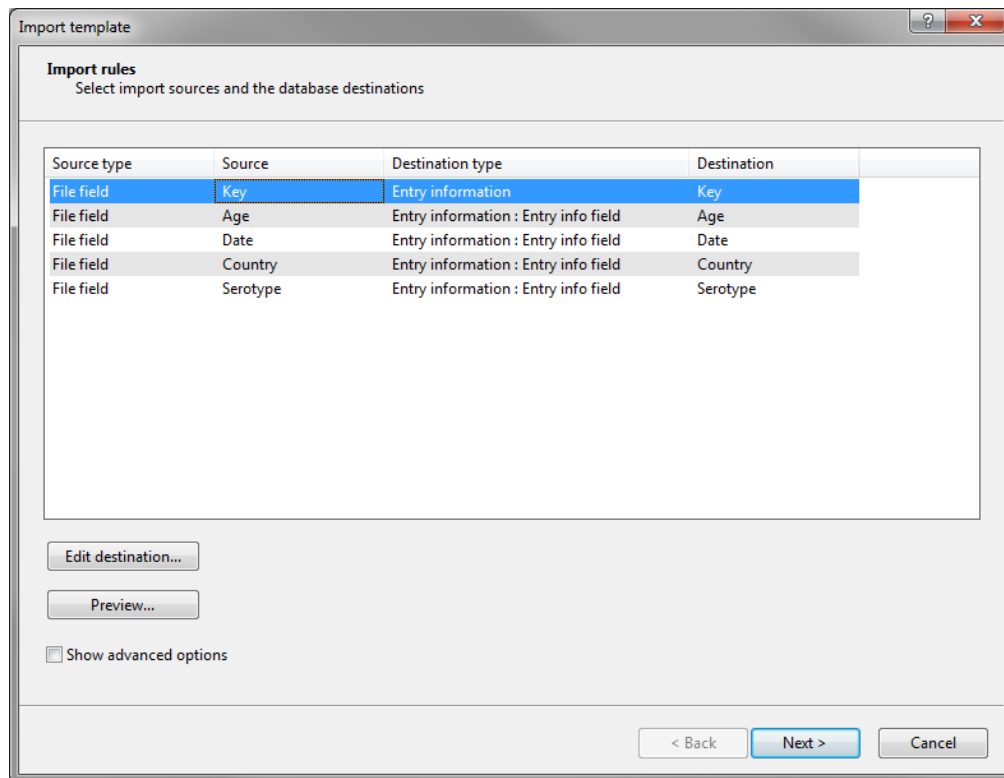


Figure 14.6.8: The import template mapping rules.

- 5.18 Choose *Entry information fields: Country* as *Column* and *Entry information fields: Serotype* as *Row* and press <*Finish*> (see Figure 14.6.9).

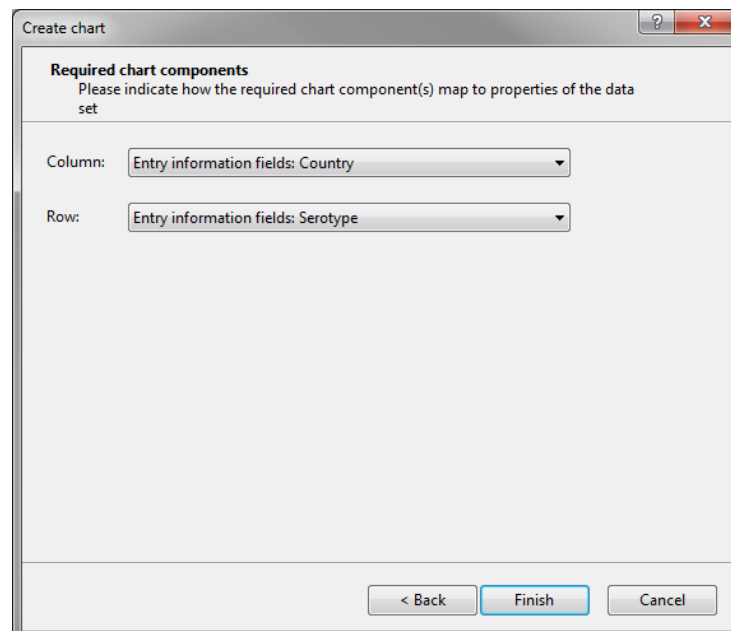


Figure 14.6.9: Chart components.

By default, the contingency table displays the member counts per cell and the cells are colored according to the member counts using a color scale blue-green-yellow-orange-red. In addition, the order of the cells is determined by the appearance in the database (see Figure 14.6.10).

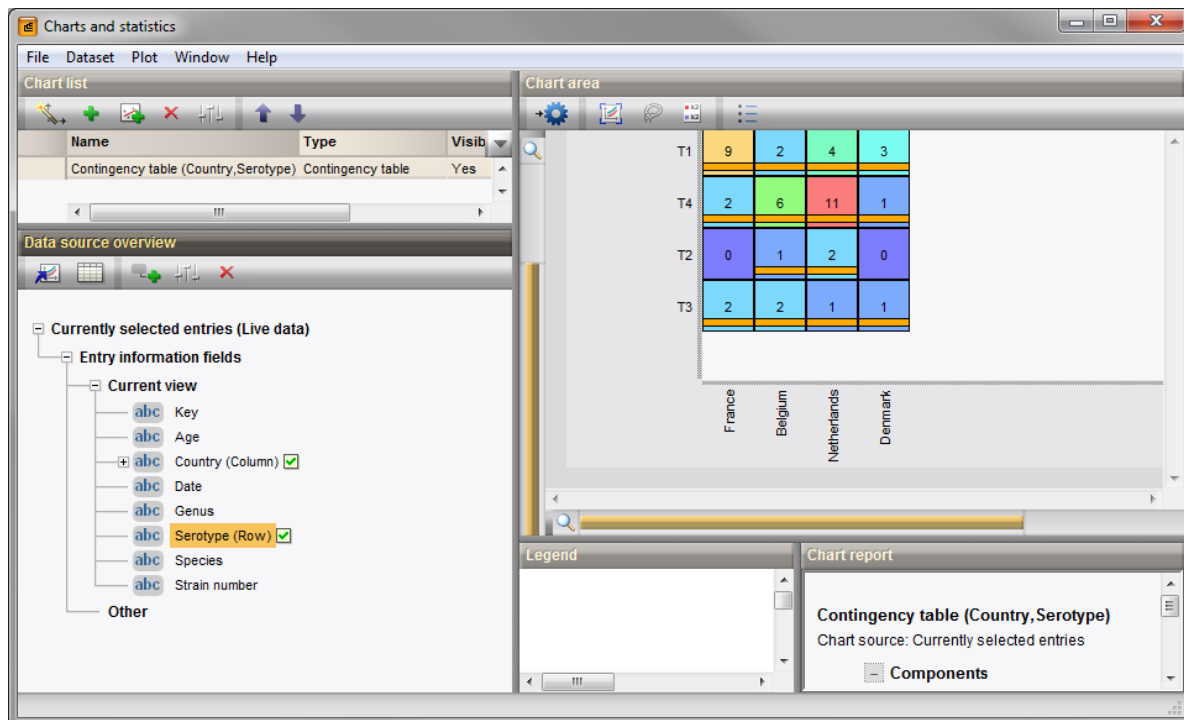


Figure 14.6.10: Contingency table.

The currently selected entries are shown in the contingency table as yellow bars in the cells. The width of the bar in a cell is proportional to the relative number of entries selected within the cell.

- 5.19 To clear the selection, press the **F4**-key.
- 5.20 Select the entries of a particular cell by clicking in the cell while holding down the **Ctrl**-key. To select multiple cells at once, hold down the **Shift**-key while dragging the mouse over the cells to select.
- 5.21 To change the appearance settings, select **Plot > Edit active plot properties...** (⚙️).
- 5.22 Select **Alphabetical** as **Column sort** and **Row sort**.
- 5.23 Choose **Row percentages** as **Quantification** and **Gray scale** as **Cell background**. Press **<OK>** to apply the changes.
- 5.24 Close the *Charts and statistics* window with **File > Exit**.

Part 15

Identification

Chapter 15.1

Principles

Identification (also called supervised learning or classification) can be performed in different ways in BioNumerics:

- A simple and straightforward way is to paste unknown entries into an existing cluster analysis, as described in [13.3.12](#). While this method is simple in principle, it does not offer much statistical confirmation and becomes slow and tedious in case of many unknowns.
- For several experiment types, fast matching methods (see [15.2](#)) are available that match entries based on a experimental data against a *complete* database. Since no pairwise comparisons need to be performed, the methods are fast indeed, but they do not allow to select a subset of reference entries to identify against.
- If it is important to specify a list of reference organisms to identify against, identifications should be performed against comparisons or stored classifiers (see [15.5](#)). In this case, a comparison or identification project (see [15.3](#)) provides both the list of reference entries and the class labels. When using comparisons, only similarity-based classification methods can be used. The use of identification projects and classifiers (see [15.4](#)) allows the use of stored and trained classifiers, such as Naive Bayesian classifiers and Support Vector Machines.
- Finally, identification can be automated to a large extent when using decision networks (see [15.6](#)).



The functionality described in this Chapter requires the Classifiers and Identification module (ID) to be present in your BioNumerics configuration.

Chapter 15.2

Fast matching methods

15.2.1 Fast band-based database screening of fingerprints

In case of large databases of fingerprint patterns, the most time-consuming part of a quick database screening of new or unknown patterns is reading or downloading all the fingerprint information. BioNumerics offers a tool that overcomes this bottleneck by generating a cache containing band information of all available fingerprints belonging to a fingerprint type. When a database screening is performed, this cache is loaded rather than the full gel information. This cache-based fingerprint screening is extremely fast, even for the largest databases, but is limited to band-based comparisons of fingerprint patterns. Please note that this tool requires the Fingerprint data module (**FP**) and the Classifiers and Identification module (**ID**) to be present in your BioNumerics configuration.

The fast band-based identification can be enabled in the *Fingerprint type* window, by selecting **Settings** > **Enable fast band matching**. A question pops up "Do you want to generate cached patterns for all current fingerprints?". By answering <Yes>, a cached pattern will be generated for all patterns present in the database that belong to the selected fingerprint type. If you answer <No>, a cached pattern will be created only for new patterns that are added to the database.



For the fast band matching identification tool to work, metrics information (molecular weight regression) needs to be available for the active reference system of the selected fingerprint type (see 4.1.5).

The fast band matching identification tool is launched from the *Main* window, where a set of selected entries will be identified against all other database entries. Select **Analysis** > **Fast matching** > **Fast band matching...** to open the *Fast band matching* dialog box (see Figure 15.2.1).

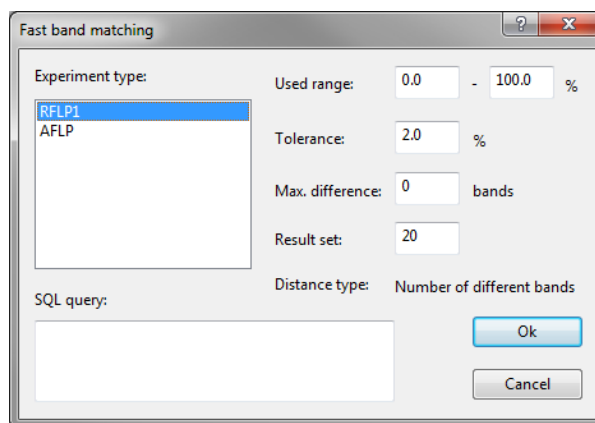


Figure 15.2.1: The *Fast band matching* dialog box.

Under **Experiment type**, select the fingerprint type you want to use for the band matching. With **Used range**, you can specify a range of the pattern (in percentage distance from top) within which bands will be compared. The **Tolerance** is the same as the position tolerance explained in 4.2.1. With **Maximum difference**, you can specify the maximum number of different bands between the unknown pattern and a database pattern to be included in the result set. Furthermore, the **Result set** can be limited to a certain number (default 20). In the input box **SQL query**, it is possible to enter a WHERE clause of a Structured Query Language (SQL) database query, to limit the search to a subset of entries that match a specific string entered for an information field.

The typical syntax of a restricting SQL query WHERE clause is: "GENUS"='Ambiorix'

One can also combine statements, for example: "GENUS"='Ambiorix' AND "SPECIES"='sylvestris'
"GENUS"='Ambiorix' OR 'Perdrix'

By pressing <OK>, the fast band matching is executed and the identification result pops up in the *Fast matching* window (Figure 15.2.2).

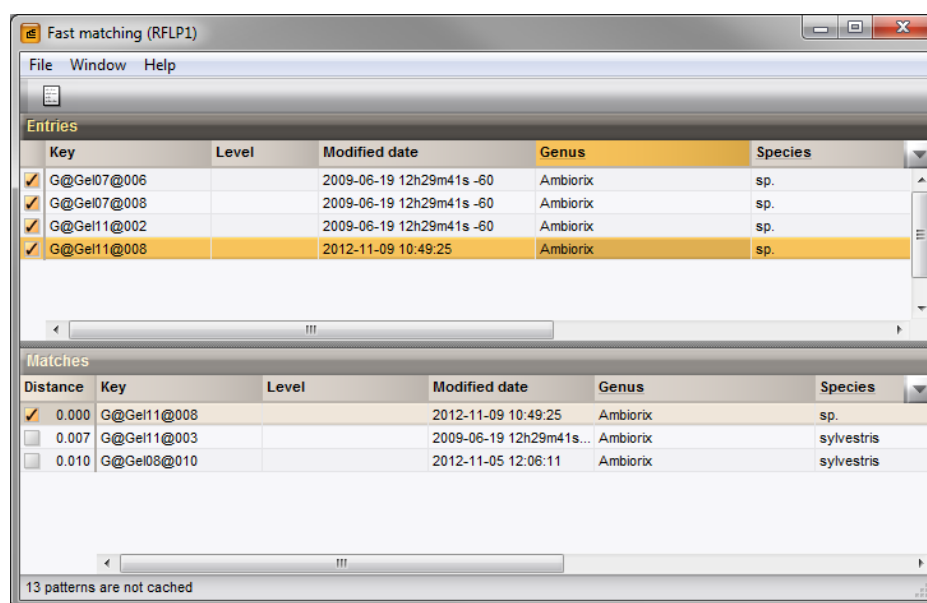



Figure 15.2.2: The *Fast matching* window.

This window is subdivided in two dockable panels, of which the *Entries* panel lists the entries to be identified, and the *Matches* panel lists the result set for the selected entry in the *Entries* panel (for display options of dockable panels, see 2.3.4). The only matching criterion used is the number of different bands, which is listed in the 'Distance' column of the *Matches* panel.



In cases where matching patterns are identical, there may be a small decimal distance. For each identical match, the software uses the band pair with the highest shift and adds this shift value to the match (i.e. to zero). This is an additional feature to sort identical patterns according to distance based upon shifts within the defined position tolerance.



In the *Entries* panel and *Matches* panel of the *Fast matching* window, the same information fields are displayed as in the *Database entries* panel of the *Main* window. To display or hide other information fields in a panel, click on the column properties button  in the information fields header and select **Set active fields**.

In both panels of the *Fast matching* window, you can select or unselect entries using the mouse in combination with the **Shift** or **Ctrl**-keys. You can also pop up the *Entry* window by double-clicking on an entry or pressing **Enter**.

A text report can be exported with **File > Export...** . The exported file lists the matched entries together

with the best matching database entries, sorted according to number of different bands.

15.2.2 Fast character-based identification

Similar as for fingerprints (15.2.1), the software offers a tool for screening an entry against the database, based upon a character type experiment. This identification tool benefits from a bulk-fetching mechanism, which makes it many times faster for identification against large databases. Unlike for a fingerprint type, there is no indexing of the experiment information needed to optimize the speed. Please note that this tool requires the Character data module (CH) and the Classifiers and Identification module (ID) to be present in your BioNumerics configuration.

To perform a fast character matching on a number of selected entries in the database, select **Analysis > Fast matching > Fast character matching...** The *Fast character set matching* dialog box appears (Figure 15.2.3).

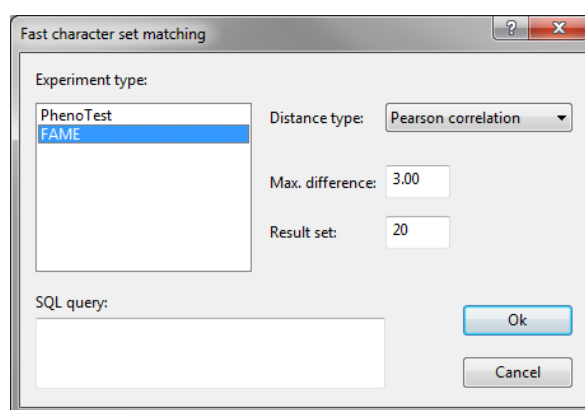


Figure 15.2.3: The *Fast character set matching* dialog box.

This dialog box displays the available character type experiments under **Experiment type**. With **Distance type**, the coefficient on which the distance is based can be chosen. Available coefficients include **Pearson correlation**, **Cosine correlation**, **Canberra metric**, **Euclidean distance**, **Manhattan distance** and the **Categorical coefficient**. The distances are calculated as $100 - [\% \text{ similarity or correlation}]$. Under **Max. difference**, a maximum percentage distance can be entered for database entries to be listed as matching. The maximum number of matching database entries to be displayed can be specified under **Result set**.

Similar as for fingerprints, an **SQL query** WHERE clause can be entered, to limit the search to a subset of entries that match a specific string entered for an information field. See 15.2.1 for examples of such SQL queries.

By pressing <OK>, the fast character matching is executed and the identification result pops up in the *Fast matching* window (see 15.2.1).

15.2.3 Fast sequence-based identification

Similar as for fingerprints (15.2.1) and characters (15.2.2), the software offers a tool for screening an entry against the database, based upon a sequence type experiment. Unlike for a fingerprint type, there is no indexing of the experiment information needed to optimize the speed. Please note that this tool requires the Sequence data module (SQ) and the Classifiers and Identification module to be present in your BioNumerics configuration.

To perform a fast sequence matching on a number of selected entries in the database, select **Analysis > Fast**

matching > *Fast sequence matching*.... This will display the *Fast sequence matching* dialog box (Figure 15.2.4).

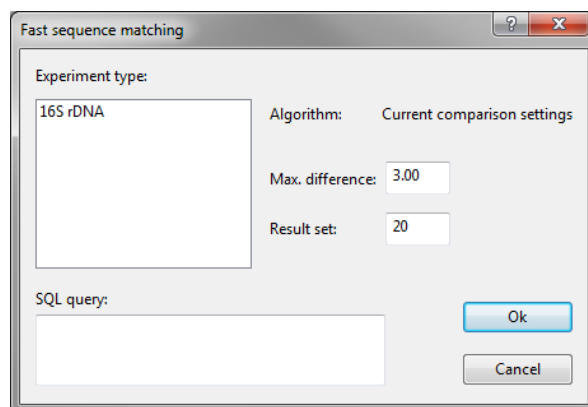


Figure 15.2.4: The *Fast sequence matching* dialog box.

This dialog box displays the available sequence type experiments under **Experiment type**. The similarity type is the current default setting specified for the experiment type. The distances are calculated as $100 - [\% \text{ similarity or correlation}]$. Under **Max. difference**, a maximum percentage distance can be entered for database entries to be listed as matching. The maximum number of matching database entries to be displayed can be specified under **Result set**.

Similar as for fingerprints and character sets, an **SQL query** WHERE clause can be entered, to limit the search to a subset of entries that match a specific string entered for an information field. See 15.2.1 for examples of such SQL queries.

By pressing <OK>, the fast sequence matching is executed, and the identification result pops up in the *Fast matching* window (see 15.2.1).

Please note that fast sequence matching in BioNumerics can also be achieved using BLAST projects (see 8.9).



Chapter 15.3

Identification projects

15.3.1 Creating an identification project

Any *Identification project* defined in BioNumerics contains two crucial components:

- The list of entries to identify against, i.e. a set of reference organisms of which their assignment to *classes* (see below) is unambiguously known.
- The classes, as described in the *class labels*. A class is a definable *taxon* and a possible outcome of an identification. Depending on the purpose of the identification project, this corresponds to e.g. a species, serotype, variety, pathovar, etc..

To create a new identification project (requires the Classifiers and Identification module ) , click on the tab of the *Identification projects* panel to bring this panel into focus and select **Edit > Create new object...** (). The *New identification project* wizard appears (see Figure 15.3.1).

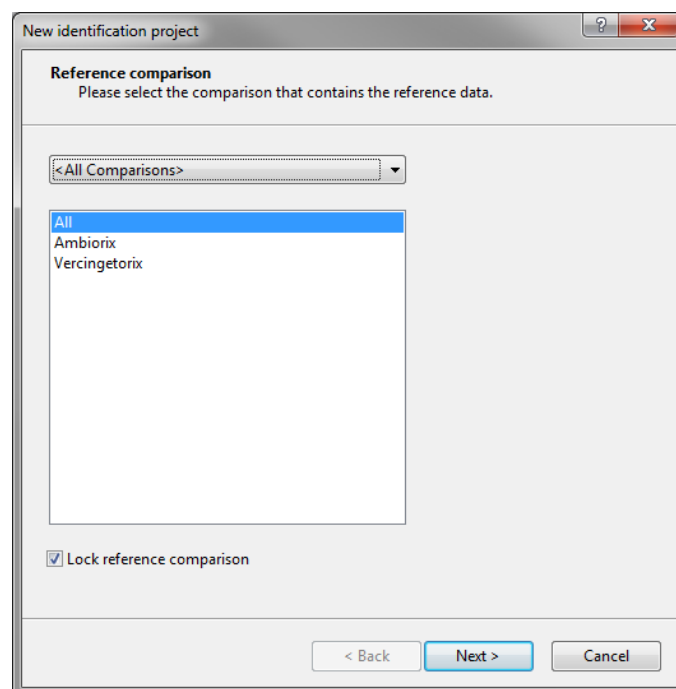


Figure 15.3.1: The *Reference comparison* wizard page.

In this page, a **Reference comparison** that contains the entries to identify against, should be selected. A list of comparisons that are present in the database are shown in the list on the left. This list can be further filtered by selecting a *view* (see 3.2.2) from the drop-down list above the comparison list.

When the option **Lock reference comparison** is checked, the comparison that is selected here will be locked to protect it from being removed by accident (see 3.2.3 for more information about locking objects).

Pressing <Next> will display the *Class labels* wizard page (see Figure 15.3.2).

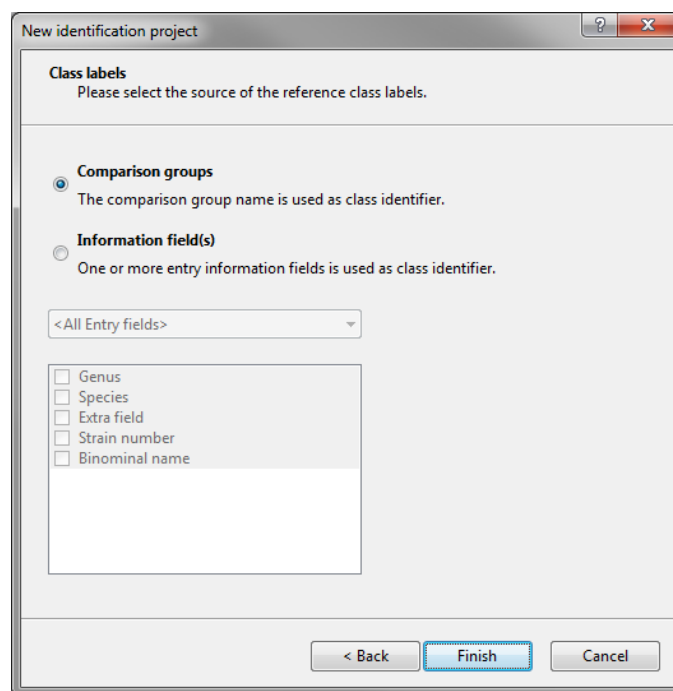


Figure 15.3.2: The *Class labels* wizard page.

The *Class labels* can either come from **Comparison groups** (see 13.3.4) or from one or more **Information fields**.

If **Information fields** is selected, you need to specify which information fields need to be used. This can be done by checking one or more check boxes left of the information fields in the list. By default, all entry information fields are listed, but this list can be filtered by selecting a view from the drop-down list.

When more than one entry information field is selected, any unique combination will be considered a class.

Pressing <Finish> will display the *Identification project name* dialog box.

In this dialog box, a name for the identification project can be entered. A default name, based on the name of the reference comparison, is suggested.

The new identification project will now be listed in the *Identification projects* panel and opened in the *Identification project* window to create *classifiers* for it (see 15.4.1).

15.3.2 Editing an identification project

To edit an identification project (e.g. to add a classifier, see 15.4.1), click on the identification project in the *Identification projects* panel of the *Main* window and select **Edit > Open highlighted object...** (🖱️, Enter). This opens the *Identification project* window (see Figure 15.3.3).

The *Classifiers* panel lists any classifier that is defined for the identification project. For each classifier, the Classifier type is displayed and the Experiment type and Data type on which it is based (see 15.4.1). It is

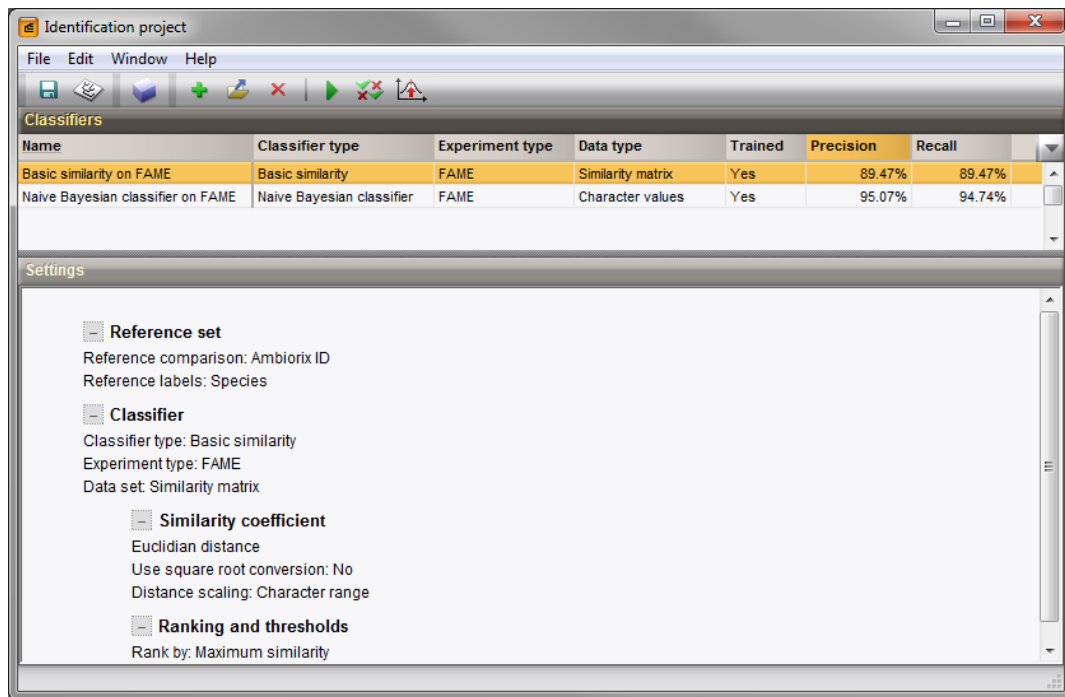


Figure 15.3.3: The *Identification project* window of an identification project containing two classifiers.

furthermore indicated whether this classifier has already been trained or not (see 15.4.2). If a cross-validation has been performed (see 15.4.4), the Precision and Recall of the classifier are mentioned.

In the *Settings* panel, the settings from the Reference set and the highlighted classifier are summarized.

The reference labels that were defined during creation of the identification project (see 15.3) can be changed with **Edit > Edit reference label definition....** This shows the *Identification reference labels* dialog box (see Figure 15.3.4).

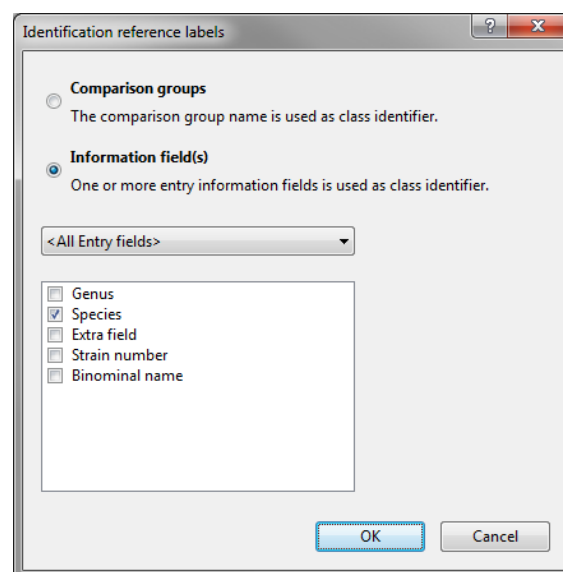




Figure 15.3.4: The *Identification reference labels* dialog box.

The identification reference labels can either come from **Comparison groups** (see 13.3.4) or from one or more **Information fields**.

If **Information fields** is selected, you need to specify which information fields need to be used. This can be done by checking one or more check boxes left of the information fields in the list. By default, all entry information fields are listed, but this list can be filtered by selecting a view from the drop-down list.

When more than one entry information field is selected, any unique combination will be considered a class.

The reference comparison on which the identification project is based, can be opened with **File > Open reference comparison** (). This is useful to create e.g. a cluster analysis or chart based on the entries in the reference set.

An identification project can be saved with **File > Save** (, **Ctrl+S**) and the *Identification project* window closed with **File > Exit**.

Chapter 15.4

Classifiers

15.4.1 Creating a new classifier

A classifier is the tool or method that assigns a class to unknown entries during the process called identification. An identification project can contain one or more classifiers.

To create a new classifier in an identification project, select **Edit > Create new classifier...** (+) to start the *New classifier* wizard (see Figure 15.4.1).

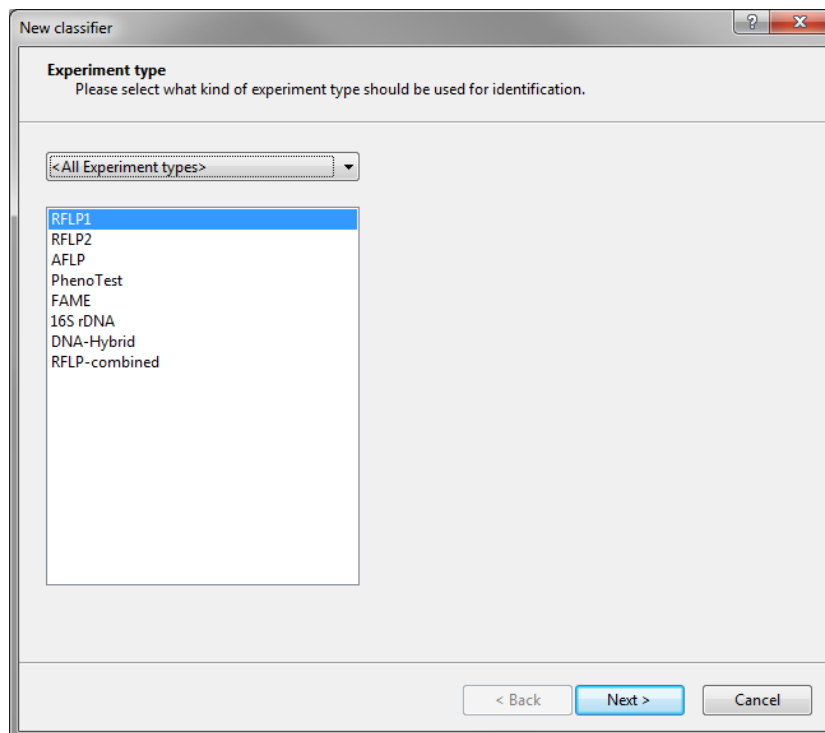


Figure 15.4.1: The *Experiment type* wizard page.

In this page, the **Experiment type** on which the identification should be based, needs to be selected. By default, all experiment types that are present in the database are listed. This list can be filtered by selecting a view (see 3.2.2) from the drop-down list.

Pressing **<Next>** will display the *Classifier method* wizard page (see Figure 15.4.2).

This page allows you to determine the **Classifier method** for identification.

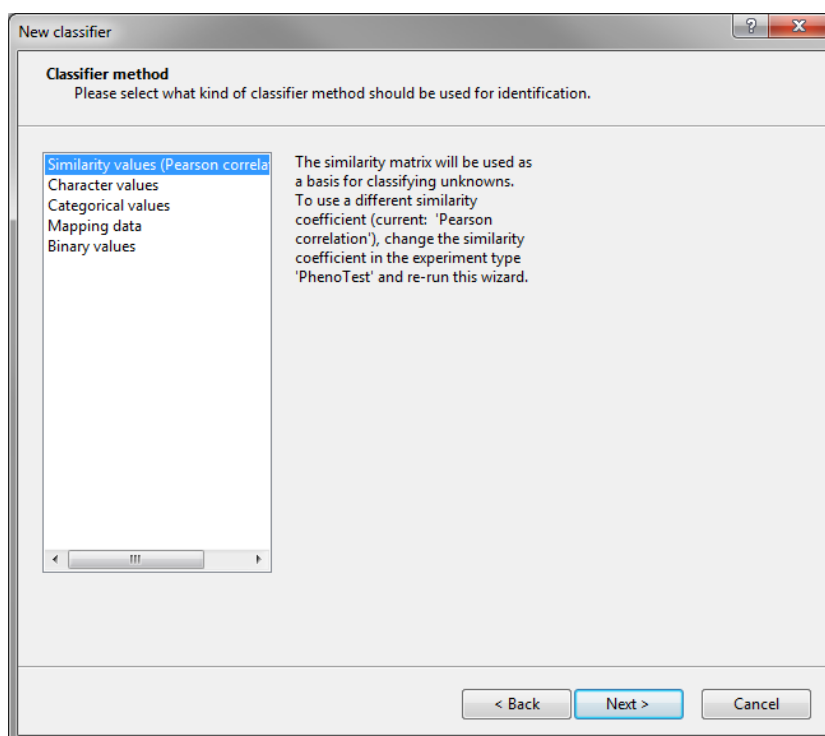


Figure 15.4.2: The *Classifier method* wizard page.

For any experiment type selected in the previous page, *Similarity values* will be available to base the identification on. These are the pairwise similarity values, calculated between the unknown and each individual class reference member. The similarity coefficient used to calculate these similarity values is indicated in between brackets and corresponds to the default coefficient as defined in the experiment type.

Depending on the experiment type, additional classifier methods might be available:

- **Fingerprint types:**

- *Densitometric curve (full)*: The complete normalized densitometric curve from a fingerprint.
- *Densitometric curve (active)*: The normalized densitometric curve from a fingerprint, limited to the active zones that were defined for that fingerprint type (see 4.1.5.9).

- **Character types:**

- *Character values*: The actual character values. Choose this option if the values should be treated as numerical.
- *Categorical values*: The character values in case they should be treated as categorical (multi-state).
- *Mapping data*: The character mappings (see 6.1.2.7).
- *Binary values*: Character values converted into binary data according to the binary conversion settings (see 6.1.2.2).

- **Nucleic acid sequence types:**

- *Sequence nucleotides (unaligned)*: The full, non-aligned nucleic acid sequence.

- **Spectrum types:**

- *Spectrum curve data*: The full spectrum curves.

- **Peak class presence:** Binary (presence/absence) values for all peak classes, belonging to the indicated peak class type (see 5.5).
- **Peak class height:** The peak heights of all peak classes, belonging to the indicated peak class type.
- **Sequence read sets:**
 - **Keyword presence:** Presence/absence values for sequence read sets keywords (see 9.1.3).
 - **Keyword frequency:** Frequencies of occurrence for sequence read sets keywords.

Pressing <Next> will display the *Scoring method* wizard page (see Figure 15.4.3).

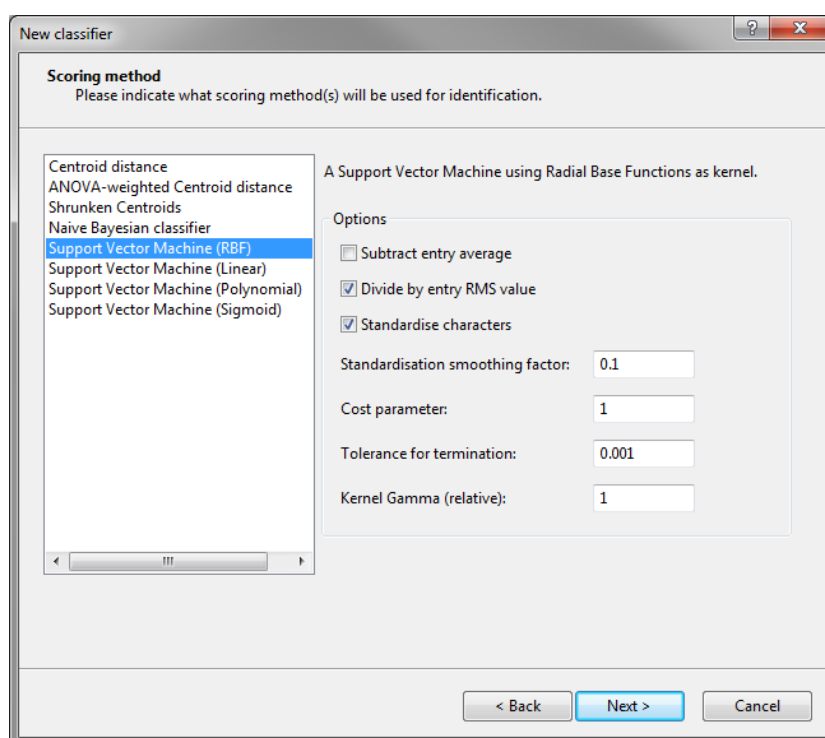


Figure 15.4.3: The *Scoring method* wizard page.

In this page, the *Scoring method* should be selected.

Following scoring methods are available:

Basic similarity:

Based on the similarity values between the unknown and each individual class reference member, this scoring method calculates the Maximum similarity, Average similarity, Minimum similarity and Normalized distance of the unknown with the class. While the former three are more or less self-explanatory, the *Normalized distance* is an indication for the confidence of the identification. It is calculated by comparing the average similarity between the unknown entry and the class reference members with the average similarity between the class reference members. This quality indication therefore takes the internal heterogeneity of the class into account.

k-Nearest Neighbors:

With the option *K-Nearest Neighbor*, the user has to specify a *K value*, which is a number of entries from the whole library having the highest similarity with the unknown. Suppose that 10 is entered for *K*, the 10 best matching entries from the whole library will be retained. The class having the largest number of entries

belonging to these K nearest neighbors is considered the best matching, and gets the highest score. The score is simply the number of entries of the class that belong to the K nearest neighbors.

The **K value** is supposed to be smaller than the number of entries contained in each of the classes. If this is not the case, the program will warn you for this conflict when the identification is executed.

Balanced similarity:

The Balanced similarity classifier scores the match of an unknown to a class using a weighted sum of the maximum and average similarities of the class entries with the unknown:

$$\text{Score} = wS_{\max} + (1 - w)S_{\text{avg}},$$

where w is the **Max. similarity weight** parameter, i.e. the weight for the maximum similarity. The weight for the average similarity is one minus the **Max. similarity weight**. Note that with a value of 1 this classifier becomes the Maximum similarity classifier, with a value of 0 it is the Average similarity classifier.

Weighted average similarity:

The weighted average similarity classifier computes scores using a weighted sum of class similarities to the unknown, with weights exponentially decreasing with increasing distance to the unknown:

$$\text{Score} = \frac{\sum_{i=1}^n w_i S_i}{\sum_{i=1}^n w_i},$$

with S_i the similarity of the i^{th} reference sample ($0 \leq i \leq 1$) and

$$w_i = \exp -F(1 - S_i),$$

where F is the **Weight factor** parameter. The **Weight factor** controls the speed with which the weights go down as similarities decrease. Increasing the weight factor thus skews the scores towards using only high class similarities to the unknown, decreasing the weight factor on the other hand means that lower similarities are more and more present in the overall weighted sum.

Centroid distance:

The Centroid distance classifier computes scores using the geometrical distance of the unknown to the centroid (average) class values. For each character in a class, the average value is computed. The score of this class for an unknown is then the distance of this set of values to the unknown character values.

If **Subtract entry average** is checked, the scalar values for each entry and/or unknown are centered on zero (i.e. the average value of the scalar values is equal to zero) before using them for training/identification. Note that this happens for each entry/unknown individually.

Check the option **Divide by entry RMS value** to apply a per-entry/unknown scaling to the respective scalar values, making the standard deviation of the entry/unknown scalar values equal to one.

ANOVA-weighted Centroid distance:

An extension of the Centroid distance classifier where the centroid distances are weighted by an ANOVA factor, consisting of the per-character variance between classes divided by the per-character class variance.

If **Subtract entry average** is checked, the scalar values for each entry and/or unknown are centered on zero (i.e. the average value of the scalar values is equal to zero) before using them for training/identification. Note that this happens for each entry/unknown individually.

Check the option **Divide by entry RMS value** to apply a per-entry/unknown scaling to the respective scalar values, making the standard deviation of the entry/unknown scalar values equal to one.

The ANOVA weight factor can be smoothed by choosing a nonzero **Variance smoothing factor**. If so, the numerator and denominator of the per-character weight factors are incremented with the smoothing factor multiplied with the average per-character class variance. The effect of the smoothing factor is thus to mitigate the effects of the ANOVA weight factor.

Shrunk Centroids:

An extension to the Centroid distance classifier, each of the class centroids (per-character averages) are moved towards the overall centroid for all the classes (i.e. average per-character value over all the classes) by an amount regulated by the **Shrink factor** parameter. Choosing a nonzero shrink factor thus reduces the effect of noise. It also does automatic character selection because characters that are shrunk to zero for all classes are effectively eliminated from the prediction rule.

Following parameters are available:

- **Subtract entry average:** If this option is checked, the scalar values for each entry and/or unknown are centered on zero (i.e. the average value of the scalar values is equal to zero) before using them for training/identification. Note that this happens for each entry/unknown individually.
- **Divide by entry RMS value:** Check this option to apply a per-entry/unknown scaling to the respective scalar values, making the standard deviation of the entry/unknown scalar values equal to one.
- **Per-character standard deviation smoothing:** The per-character standard deviation, which is used as weight in the weighted square distance scoring method, is linearly interpolated between its actual value and the class average per-character standard deviation. A value of zero means using the actual value (but see also the overall standard-deviation smoothing), a value of one means using the class averaged value.
- **Overall standard-deviation smoothing:** Similar to the per-character standard deviation smoothing, now using the class and character averaged standard deviation instead of the class averaged per-character standard deviation. Note that the overall standard-deviation smoothing is applied after the per-character standard deviation smoothing.
- **Shrink factor:** Determines the rate (linear) at which class centroids converge to the overall class centroid.

Naive Bayesian classifier:

A classifier based on Bayesian principles, the Naive Bayesian classifier (NBC) finds the highest probability of a class for the set of classes, given a data-set. It exists in three flavors: scalar, categorical and binary. With a NBC, the per-class chance is given by the product of the individual character data probabilities (naive assumption: the data is assumed to be uncorrelated) multiplied by the class prior. These individual probabilities are determined by a model, which needs to be trained. The model consists of either character or category frequencies for binary and categorical data or a Gaussian distribution for scalar data. The class priors depend on the value of the **Use prior distribution** parameter.

Following parameters are available:

- **Subtract entry average** (scalar): If this option is checked, the scalar values for each entry and/or unknown are centered on zero (i.e. the average value of the scalar values is equal to zero) before using them for training/identification. Note that this happens for each entry/unknown individually.
- **Divide by entry RMS value** (scalar): Check this option to apply a per-entry/unknown scaling to the respective scalar values, making the standard deviation of the entry/unknown scalar values equal to one.

- **Use prior distribution** (binary, scalar and categorical): If this option is checked, the class prior knowledge factors are the class frequencies. If not, then the priors of all classes are set to the inverse of the total class count.
- **Per-character standard deviation smoothing** (scalar): The per-character standard deviation, which is used as weight in the weighted square distance scoring method, is linearly interpolated between its actual value and the class average per-character standard deviation. A value of zero means using the actual value (but see also the overall standard-deviation smoothing), a value of one means using the class averaged value.
- **Overall standard-deviation smoothing** (scalar): Similar to the per-character standard deviation smoothing, now using the class and character averaged standard deviation instead of the class averaged per-character standard deviation. Note that the overall standard-deviation smoothing is applied after the per-character standard deviation smoothing.
- **Frequency offset** (binary and categorical): The computation of a class probability uses a product of per-character probabilities. For binary and categorical data, these are the per-class character frequency and the per-class per-character category frequencies, respectively. If one of these frequencies is equal to zero i.e. if none of the members of this class have this character/category value, the entire product is zero and the resulting class probability is zero. Using a non-zero frequency offset prevents one or a few characters/categories frequencies from ruling out their respective classes by adding a fixed value to each of the character/category frequencies.
- **Minimum character probability** (scalar): The function of this parameter, available for scalar data, is equal to that of the frequency offset for binary and categorical Naive Bayesian classifiers: set a lower limit on the per-class per-character probability to prevent the probability of that class becoming very small due to very small probabilities of a limited number of characters.

Support Vector Machine:

Support vector machines (SVMs) are modeling methods for binary classification of data. The algorithm consists of finding a set of (hyper)planes that optimally separate two input classes in character space, the data points that are part of these separating planes are called the support vectors. Because the distribution of the input data in the character space is often too complex to be able to separate them in two by using planes, the data is transformed to a higher dimensional so-called feature space where finding a set of separating hyper-planes is feasible.

Several SVM flavors have been implemented: Linear, RBF (Gaussian Radial Basis Function), Polynomial and Sigmoid SVM kernels. Note that it is important to optimize the parameters of the chosen SVM model for each specific set of input data (see 15.4.5).

Following parameters are available for all SVMs:

- **Subtract entry average** (for scalar data only): If this option is checked, the scalar values for each entry and/or unknown are centered on zero (i.e. the average value of the scalar values is equal to zero) before using them for training/identification. Note that this happens for each entry/unknown individually.
- **Divide by entry RMS value** (for scalar data only): Check this option to apply a per-entry/unknown scaling to the respective scalar values, making the standard deviation of the entry/unknown scalar values equal to one.
- **Cost parameter**: The penalty parameter for the SVM error term (C-SVM algorithm, see <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>). Also called the soft margin parameter, the cost parameter allows SVM solutions that do not cleanly separate all the data. Choosing the cost parameter is thus a trade-off between a larger search margin and a small error penalty, increasing the cost parameter increases the cost of misclassifying points, creating a more accurate model that may not generalize well however.

- **Tolerance for termination:** Solving the SVM problem is done with an iterative loop (using working set selection, see <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>). With the tolerance for termination, we specify when the solution has converged enough to be retained.

Description and parameters of the different SVM flavors:

- **Linear SVM kernel:** The simplest of the different kernel functions, it only allows a linear classification of the data (i.e. the data is not transformed to a higher-dimensional feature space). Its shape is given by:

$$K(x, y) = x^T y.$$

- **RBF SVM kernel:** The implemented Radial Basis Function (RBF) kernel is a Gaussian kernel:

$$K(x, y) = e^{-\gamma \|x - y\|^2}.$$

The γ parameter should be finely tuned to the problem at hand, overestimating it causes the RBF kernel to behave like the linear kernel and thus lose its nonlinear power, underestimating it makes the SVM classifier more sensitive to noise in the training data.

- **Kernel Gamma (relative):** The gamma parameter of the RBF kernel is given by:

$$\gamma = \frac{\text{KernelGamma(relative)}}{\text{\#characters}}.$$

- **Sigmoid SVM kernel:** The sigmoid kernel (also known as the hyperbolic tangent kernel) is given by

$$K(x, y) = \tanh(\gamma x^T y + c),$$

where c is the kernel offset. Note that a SVM model using a sigmoid kernel is equivalent to a two-layer, perceptron neural network.

- **Kernel Gamma (relative):** See Kernel Gamma (relative) of the RBF kernel
- **Kernel Offset**

- **Polynomial SVM kernel:** The polynomial kernel is given by:

$$K(x, y) = (\gamma x^T y + c)^d,$$

where c is the kernel offset, d is the kernel degree.

- **Kernel Gamma (relative):** See Kernel Gamma (relative) of the RBF kernel
- **Kernel Offset**
- **Kernel Degree:** Increasing the kernel degree gives the SVM algorithm more flexibility in separating the data into classes. It also increases the risk of over-fitting, however.

Pressing <Next> will display the *Scoring method* wizard page (see Figure 15.4.4).

From the **Rank by score** drop-down list, the score can be selected that will be used for the default ranking of the identification results in the *Identification* window.

Under **Reported scores**, all available scores are listed. Using the check boxes, one can determine whether a score will be reported or not. For each of the reported scores, a **Threshold** and a **Minimum difference** can be entered. The **Threshold** corresponds to the minimum score needed for a valid identification. The **Minimum difference** corresponds to the difference between the score of the best identification result and the runner-up. If this difference is lower than the value specified here, the identification is considered invalid.

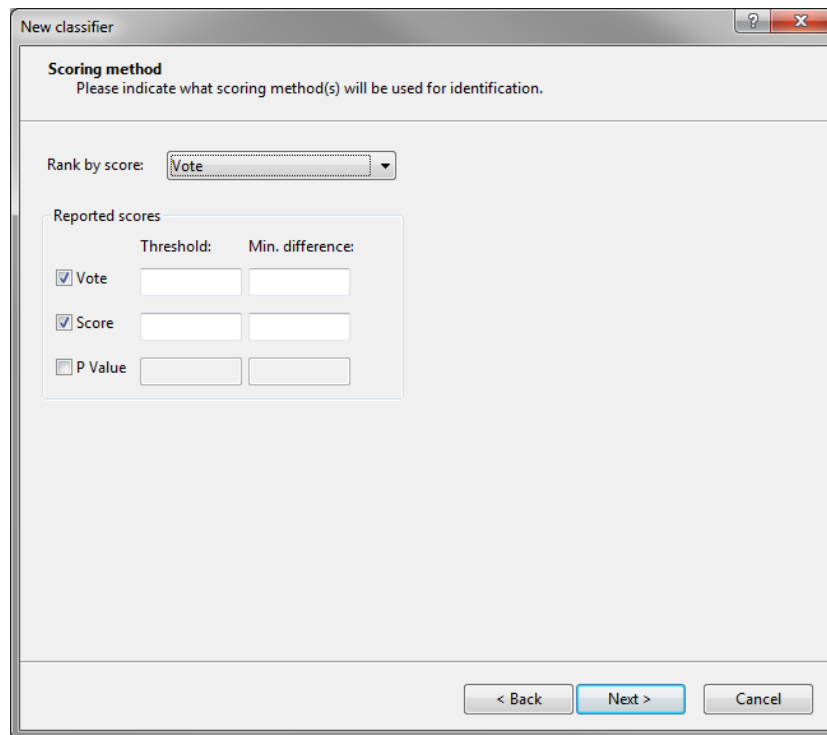


Figure 15.4.4: The *Scoring method* wizard page.

When *<Next>* is pressed, the *Classifier name* dialog box pops up.

In this dialog, a name for the new classifier can be entered. A classifier name is suggested based on the scoring method and experiment type. The suggested name can be overridden by any other user-defined name.

If the created classifier requires training before it can be used for identification, the software will ask if you want to train the classifier now. See 15.4.2 for more information.

15.4.2 Training classifiers

Most classifiers need to be *trained* before they can be used. This is not the case for similarity-based classifiers, since the similarity values are calculated on-the-fly during the classification process.

The question whether or not to train a classifier, pops up after creation of a classifier that needs training (see 15.4.1). When “No” has been answered to that question or to re-train a classifier after the underlying data have been modified, select *Edit > Train classifier* (▶) to train the highlighted classifier. Depending on the type of classifier and the number of entries and classes in the identification project, this takes a few seconds up to several minutes.

Selecting *Edit > Export training data* will export the training data to a file.

15.4.3 Classifier settings

The settings of the highlighted classifier in the *Classifiers* panel can be edited with *Edit > Edit classifier settings...* (🔧). This action calls the *Classifier settings* dialog box (see Figure 15.4.5).

The *Options* tab in the *Classifier settings* dialog box contains the parameters as entered in the *Scoring*

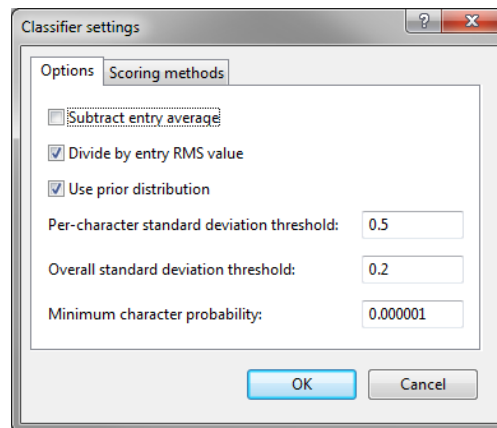


Figure 15.4.5: The *Classifier settings* dialog box.

method wizard page of the *New classifier* wizard (see 15.4.1). The *Scoring methods* tab summarizes the parameters as discussed in the *Scoring method* wizard page of the same wizard. Each of the parameters can be modified here.

15.4.4 Cross-validation analysis

Cross-validation is a method to assess the consistency of a classifier. The validation is done by identifying the class members from the identification project against the classifier. The class members are thereby split into k parts, then $k - 1$ of these parts are used for training and the remaining part for testing. This step is repeated k times, each time using a different part for testing and the remaining parts for training.

To perform a cross-validation analysis, select **Edit > Cross-validation analysis...** (🔧). This will call the *Cross-validation* dialog box (see Figure 15.4.6).

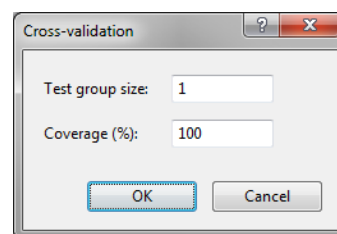


Figure 15.4.6: The *Cross-validation* dialog box.

The **Test group size** (or "validation set" size) corresponds to the number of entries that are omitted from the total number of entries before the classifier is trained (the remaining entries are sometimes referred to as the "training set"). Subsequently, the entries from the test group are being matched and it is checked if these are classified correctly. The **Test group size** always needs to be smaller than half of the number of entries in the reference set.

A **Coverage (%)** of 100% (the default value) means that all entries will eventually be used in a test group. To speed up the analysis, the coverage can be lowered. This means of course that the validation results are slightly less reliable, as not all combinations are being considered.

Pressing <OK> will perform the cross-validation and the results will be displayed in the *Identification cross validation* window (see Figure 15.4.7).

The *Identification cross validation* window consists of three dockable panels:

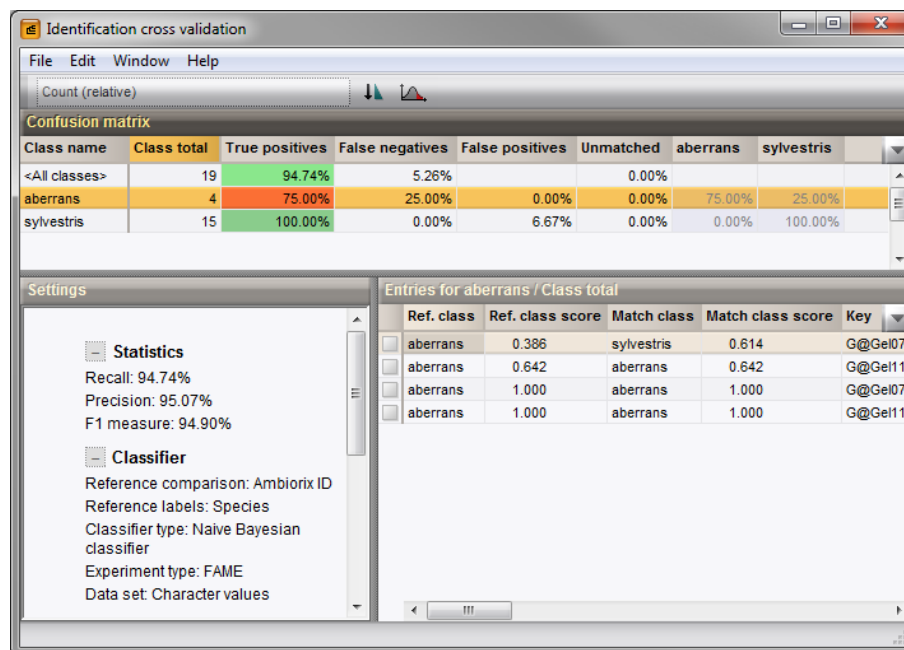


Figure 15.4.7: The *Identification cross validation* window.

- The *Confusion matrix* panel shows all reference classes from the classifier in a matrix. Each cell in the matrix contains the percentage of entries from a certain reference class (in rows) that are identified as belonging to a certain reference class (in columns). Based on these identification results, the number of True positives, False negatives, False positives and Unmatched is calculated and reported for each reference class.
- Clicking on a cell in the *Confusion matrix* panel will display the corresponding entries in the *Entries* panel. In addition to the entry information, the Reference class, Reference class score, Match class and Match class score are reported.
- The *Settings* panel summarizes the settings used for the classifier.

Using the **Count (relative)** drop-down list, the identification results can be displayed as a percentage of entries (*Count (relative)*), number of entries (*Count (absolute)*), average identification score of the entries (*Average score*), average identification score of the entries relative to the highest score (*Average score (relative)*) or score quantile range (*Score quantile range*).

Via **Edit > Chart and statistics...** (📊), a number of useful charts can be created for assessment of the cross-validation results. The *Create chart* dialog box that pops up offers a number of predefined chart templates and the "CrossValidationResults" data source for creating custom charts.

15.4.5 Optimizing classifier parameters

Most classifiers have several *parameters* that all have their influence on the outcome of the classification.

Select **Edit > Optimise classifier parameter...** (🔧) to display the *Classifier parameter optimization* dialog box (see Figure 15.4.8).

Parameters should be optimized one by one. The highlighted parameter in the *Parameter to optimize* list is the one that will be optimized. A *Minimum value* and *Maximum value* for the parameter values can be specified. The highlighted criterion in the *Criterion* list will be maximized.

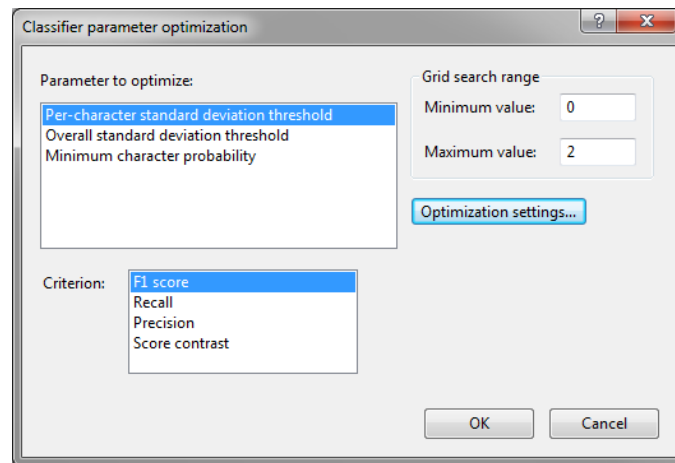


Figure 15.4.8: The *Classifier parameter optimization* dialog box, here shown for a Naïve Bayesian classifier.

Further optimization settings can be specified after pressing the *<Optimization settings...>* button.

When *<OK>* is pressed, the parameter optimization calculation starts.

When an optimal parameter value is found, this is reported and used in the classifier settings. Next, the *Classifier parameter optimization* window pops up.

The *Classifier parameter optimization* window contains a grid (or table), consisting of the calculated values for each of the classification criteria (in columns) for all the parameter values evaluated (in rows).

Using *Edit > Chart and statistics...* (📊), a number of useful graphs can be created.

Chapter 15.5

Identifying entries using classifiers

Identifications in BioNumerics are always performed on the selected entries (see 3.2.4 on how to make a selection).

With a selection of unknown entries made, use *Analysis > Identify selected entries...* (🔍) to start the *Identify* wizard (see Figure 15.5.1).

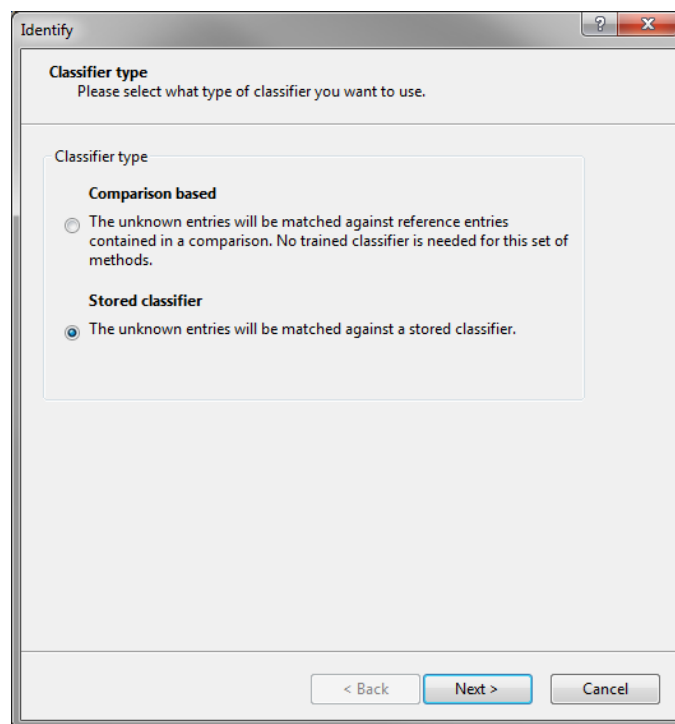


Figure 15.5.1: The *Classifier type* wizard page.

The first page of the *Identify* wizard deals with the **Classifier type**. Basically, there are two different methods available for identification:

- **Comparison based:** This method uses any of the saved comparisons. The comparison contains the list of entries to identify against and the classes are obtained either from comparison groups or information fields. Only similarity-based identifications are possible.
- **Stored classifier:** This method uses previously created identification projects (15.3) and classifiers (15.4). All identification methods available in BioNumerics (based on similarity values as well as on trained classifiers) can be used.

When one or more identification projects are present in the database, the second option will be checked by default.

If **Stored classifier** is checked and <Next> pressed, the *Stored identification project* wizard page (see Figure 15.5.2) will appear. If, in contrast, **Comparison based** is checked, the *Reference comparison* wizard page will appear after pressing <Next> (see Figure 15.5.3).

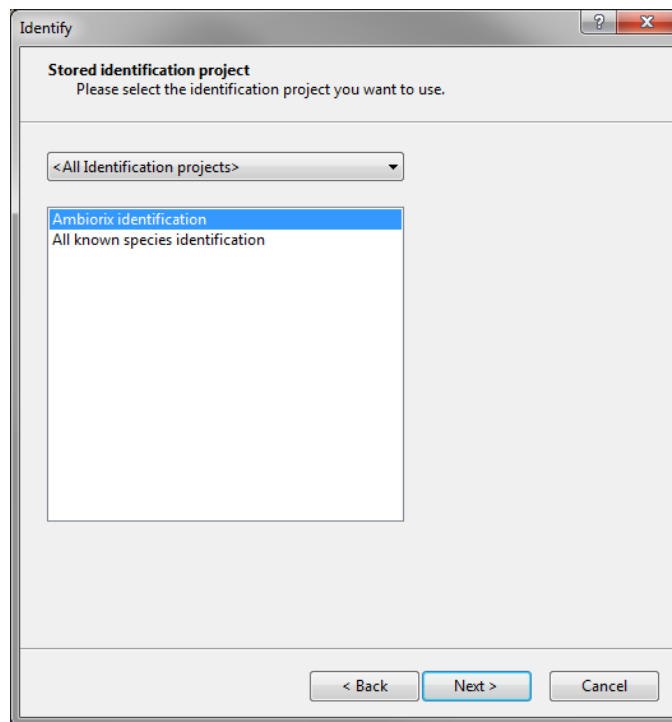


Figure 15.5.2: The *Stored identification project* wizard page.

This dialog displays by default a list of all saved identification projects in the database. If needed, this list can be filtered using views (see 3.2.2). The highlighted identification project will be used.

As this is all the information needed for the identification, pressing <Next> will open the *Identification* window with the identification results.

This dialog displays by default a list of all saved comparison in the database. If needed, this list can be filtered using views (see 3.2.2). The comparison containing the reference data, i.e. the list of entries to identify against and the reference classes, needs to be selected from the list.

Pressing <Next> will open the *Class labels* wizard page (see Figure 15.5.4).

This dialog box deals with the **Class labels**, which can either come from **Comparison groups** (see 13.3.4) or one or more **Information fields**.

If **Information fields** is selected, you need to specify which information fields need to be used. This can be done by checking one or more check boxes left of the information fields in the list. By default, all entry information fields are listed, but this list can be filtered by selecting a view from the drop-down list.

When more than one entry information field is selected, any unique combination will be considered a class.

Pressing <Next> will open the *Experiment data* wizard page (see Figure 15.5.5).

In this dialog, the **Experiment data** on which the identification should be based, need to be selected. By default, all experiment types that are present in the database are listed. This list can be filtered by selecting a view (see 3.2.2) from the drop-down list. More than one experiment type can be checked

Pressing <Next> will display the *Identification method* wizard page (see Figure 15.5.6).

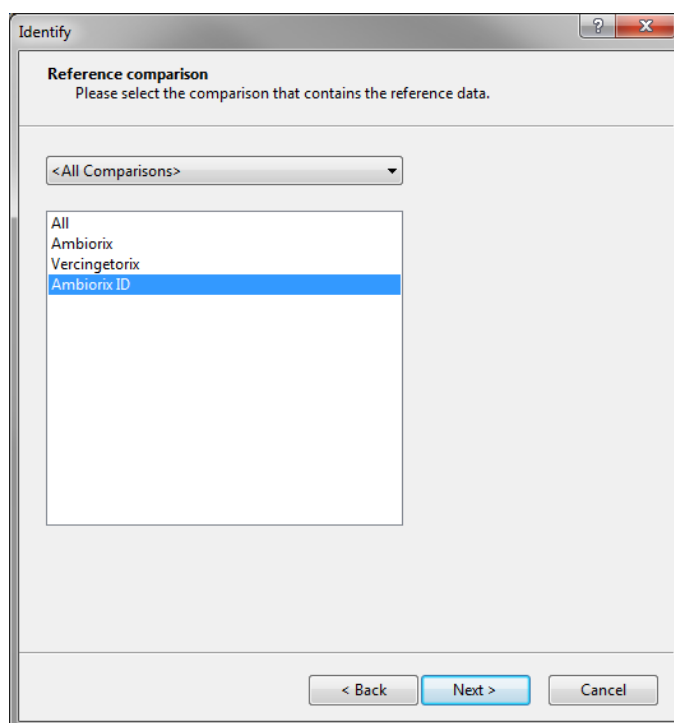


Figure 15.5.3: The *Reference comparison* wizard page.

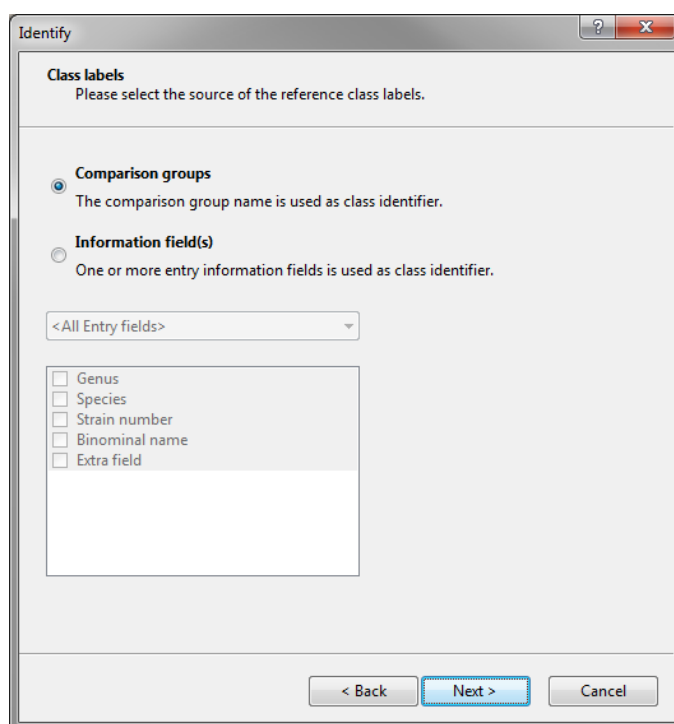


Figure 15.5.4: The *Class labels* wizard page.

In this dialog, the (similarity-based) *Identification method* needs to be selected. Following methods are available:

Basic similarity:

Based on the similarity values between the unknown and each individual class reference member, this scor-

Figure 15.5.5: The *Experiment data* wizard page.

Figure 15.5.6: The *Identification method* wizard page.

ing method calculates the Maximum similarity, Average similarity, Minimum similarity and Normalized distance of the unknown with the class. While the former three are more or less self-explanatory, the *Normalized distance* is an indication for the confidence of the identification. It is calculated by comparing the average similarity between the unknown entry and the class reference members with the average similarity between the class reference members. This quality indication therefore takes the internal heterogeneity of

the class into account.

k-Nearest Neighbors:

With the option *K-Nearest Neighbor*, the user has to specify a **K value**, which is a number of entries from the whole library having the highest similarity with the unknown. Suppose that 10 is entered for *K*, the 10 best matching entries from the whole library will be retained. The class having the largest number of entries belonging to these *K* nearest neighbors is considered the best matching, and gets the highest score. The score is simply the number of entries of the class that belong to the *K* nearest neighbors.

The **K value** is supposed to be smaller than the number of entries contained in each of the classes. If this is not the case, the program will warn you for this conflict when the identification is executed.

Balanced similarity:

The Balanced similarity classifier scores the match of an unknown to a class using a weighted sum of the maximum and average similarities of the class entries with the unknown:

$$\text{Score} = wS_{\max} + (1 - w)S_{\text{avg}},$$

where *w* is the **Max. similarity weight** parameter, i.e. the weight for the maximum similarity. The weight for the average similarity is one minus the **Max. similarity weight**. Note that with a value of 1 this classifier becomes the Maximum similarity classifier, with a value of 0 it is the Average similarity classifier.

Weighted average similarity:

The weighted average similarity classifier computes scores using a weighted sum of class similarities to the unknown, with weights exponentially decreasing with increasing distance to the unknown:

$$\text{Score} = \frac{\sum_{i=1}^n w_i S_i}{\sum_{i=1}^n w_i},$$

with S_i the similarity of the i^{th} reference sample ($0 \leq i \leq 1$) and

$$w_i = \exp -F(1 - S_i),$$

where *F* is the **Weight factor** parameter. The **Weight factor** controls the speed with which the weights go down as similarities decrease. Increasing the weight factor thus skews the scores towards using only high class similarities to the unknown, decreasing the weight factor on the other hand means that lower similarities are more and more present in the overall weighted sum.

Pressing <Next> will display the *Scoring method* wizard page (see Figure 15.5.7).

From the **Rank by score** drop-down list, the score can be selected that will be used for the default ranking of the identification results in the *Identification* window.

Under **Reported scores**, all available scores are listed. Using the check boxes, one can determine whether a score will be reported or not. For each of the reported scores, a **Threshold** and a **Minimum difference** can be entered. The **Threshold** corresponds to the minimum score needed for a valid identification. The **Minimum difference** corresponds to the difference between the score of the best identification result and the runner-up. If this difference is lower than the value specified here, the identification is considered invalid.

Pressing <Next> will open the *Identification* window with the identification results (see Figure 15.5.8).

The *Identification* window is divided in five dockable panels:

- The *Entries* panel lists the unknown entries that were selected for identification.

Figure 15.5.7: The *Scoring method* wizard page.


Key	Genus	Species	Strain number	Consensus	Basic similarity on FAME (Max...)	Basic similarity on RFLP1 (Max...)	Naive Bayesian classifier
✓ G@Gel07@006	Ambiorix	sp.	52415	Ambiorix, aberrans	97.6	87.5	1.000
✓ G@Gel07@008	Ambiorix	sp.	52424	Ambiorix, sylvestris	96.8	87.5	0.000
✓ G@Gel11@002	Ambiorix	sp.	52440	Ambiorix, aberrans	97.4	91.7	1.000
✓ G@Gel11@008	Ambiorix	sp.	52425	Ambiorix, sylvestris	97.3	89.7	0.000

Class	Maximum simil...	Average simila...	Minimum simil...	Normalized dis...
Ambiorix, aberrans	97.6	97.1	96.7	1.11
Ambiorix, sylvestris	96.8	95.4	93.0	1.48
Ambiorix, sp.	96.3	96.3	96.3	



Figure 15.5.8: The *Identification* window.

- The *Results* panel lists for each classifier (organized in columns) the class that matches best with the unknowns.


- The *Result details* panel shows the identification details for the highlighted unknown entry / classifier combination.
- The *Match comments* panel lists any comments on the match shown in the *Result details* panel, such as a comparison with cross-validation results (see 15.4.4).
- The *Settings* panel lists the settings for the selected classifier.

The separator lines between the panels can be moved to make optimal use of the display. Information fields in the *Entries* panel can be displayed or hidden by pressing the column properties button  in the information fields header and selecting **Set active field**. For detailed information about the display options available for object grid panels, see 3.2.7.


The columns in the *Results* panel contain the name of the best matching classes and their identification score. The identification scores of the classifier are obtained using the settings specified in the *Settings* panel. Normalized distances or *p*-values appear as colored squares next to the identification scores. They range from red (improbable identification) over orange, yellow (doubtful identification) to green (faithful identification).

Using **View > Show more matches** () , the second, third, etc. best match can be shown in the *Results* panel for each unknown. To display fewer matches per unknown, select **View > Show less matches** ().

The *Result details* panel lists the best matching classes for the selected unknown entry / classifier combination, ranked by their identification score. The normalized distances and *p*-values are displayed here as a number. Clicking in the *Entries* panel or *Results* panel updates the *Result details* panel with the information of the newly selected unknown entry / classifier combination.

Selecting **View > Open in comparison window** () opens a *Comparison* window, listing the unknown entry and the entries of the class.

The identification overview can be exported to a text or CSV file with **File > Export match results** or a detailed report can be created with **File > Export match result details**.

For routine identification purposes, it can be useful to store the identification results for each unknown entry. It is recommended to first create a dedicated field for this purpose in the database (see 3.3.3). Next, highlight this field in the *Entries* panel and select **File > Transfer results to database** (). The *Transfer results to database* dialog box pops up (see Figure 15.5.9).

From the **Classifier** drop-down list, the classifier from which the results should be transferred to the database (or the consensus identification of all classifiers), can be selected.

The results can either be written to the same information fields that delivered the **Reference label(s)**, or a **Specific information field** can be selected from the list below.

Pressing <OK> will write the results to the database.

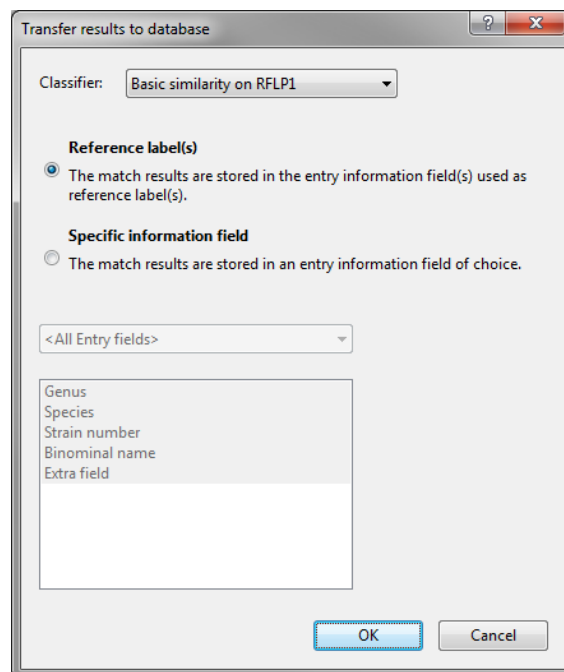


Figure 15.5.9: The *Transfer results to database* dialog box.


Chapter 15.6

Decision networks

15.6.1 Introduction


Decision networks are operational work flows that carry out [logical] operations and/or actions on the database. The networks consist of *Operators* as building blocks, that form the *Nodes* of the network:

- **Input operators** retrieve specific information, usually experimental data, from the database;
- **String, Value and Sequence operators** perform an action on data types, for example find subsequences, count bands, or evaluate character values;
- **Boolean operators** have one or more binary states as input and can e.g. combine them into a new binary state or a string;
- **Output actions** can perform a specific action on the database, for example, write the result of a decision into a database field.


Decision networks should be seen as a construction kit that allows you to build your own automated decision or action work flows, with practically endless possibilities. They can be used to make decisions, predict features, perform queries, fill in fields, create graphs and plots, and much more. Please note that decision networks are only available when the Classifiers and Identification module () is present in your BioNumerics configuration.

15.6.2 Creating a new decision network

In the default configuration of the *Main* window (see [2.3.2](#) for features and display options of the *Main* window), the *Decision networks* panel is seen as a tab behind the *Comparisons* panel. Click on the tab to bring the *Decision networks* panel to the top (Figure [15.6.1](#)).

To create a new empty decision network, highlight the *Decision networks* panel and select **Edit > Create new object...** () . In the *Create new object* dialog box that pops up, enter a name for the decision network and press <OK>.

The new decision network is now listed in the *Decision networks* panel. When a decision network is opened, it contains by default the current selection of entries. Therefore, it is practical to make a selection of entries you want to use in the decision network before opening it.

A decision network can be opened by selecting **Edit > Open highlighted object...** (, **Enter**) or by double-clicking on its name.

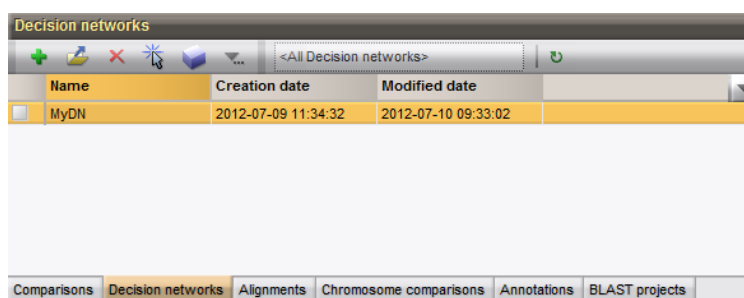


Figure 15.6.1: The *Decision networks* panel in the *Main* window.

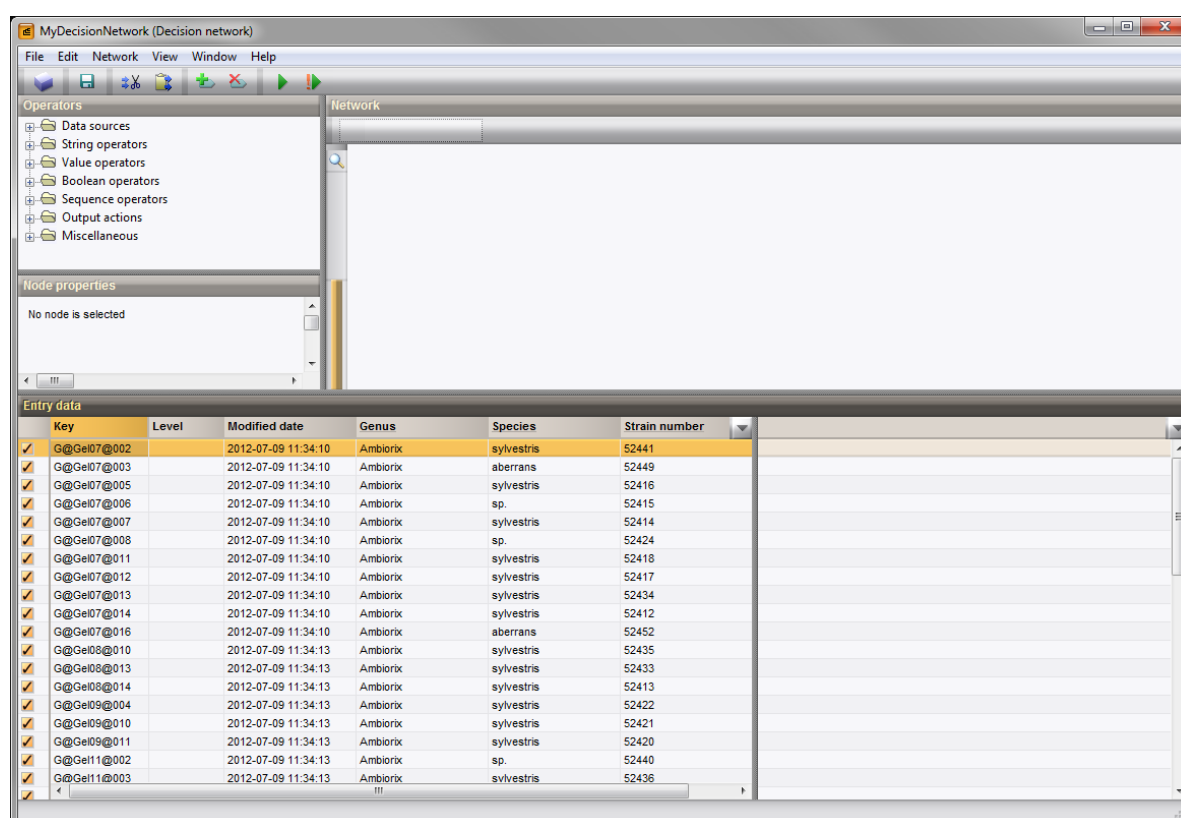


Figure 15.6.2: The *Decision Network* window with a new empty decision network.

The empty *Decision Network* window looks as in Figure 15.6.2 in its default configuration.

The window contains four panels, of which the main *Network* panel displays the network scheme. The *Operators* panel lists a tree of all operators that are available to construct the decision network (the building blocks). In *Node properties* panel, the properties and data of the current selected node is given. The *Entry data* panel lists the entries currently used in the network and their selection status. In the right hand sub panel, the output(s) from the network are listed for the entries (currently empty).

Similar as in a *Comparison* window (see 13.2.1), it is possible to add or remove entries from the list.

To add entries, copy selected entries from the *Main* window or from the *Comparison* window using **Edit > Views > Copy selection** (Ctrl+C), and paste them in the *Decision Network* window using **Edit > Paste selection** (Ctrl+V).



To remove entries from the decision network, first select the entries to remove, and then use **Edit > Cut selection** (Ctrl+X).



As opposed to a comparison (see 13.2.1), the selection of entries in a decision network is dynamic: each time you open or run the decision network, it will act on the current selection.

15.6.3 Operators


Operators are the building blocks of a decision network. They can be categorized in different groups according to their function. These groups are represented in the expandable tree in the *Operators* panel. Each operator requires a compatible output from another operator as input and delivers a result as output to the network (only operators of type *Data sources* do not require output from an operator).


- **Data sources:** These operators request data components from the database or from the user and deliver it to the network. The component can be a database field, attachment, fingerprint fields, fingerprint bands, a character value, or a sequence. A special subcategory contains the *Fixed values*, which can either be a constant value or a constant string. The subcategory *User prompt* contains operators that prompt the user to enter information of a defined data type.
- **String operators:** perform an operation on a string. They include finding a text match, defining regular expressions, comparing two strings, concatenating strings, getting substrings, or converting a string into a value. Note that more powerful string operators exist specifically for sequences (Sequence operators).
- **Value operators:** perform an operation on one or more values. The result can be a value (in case of calculations and functions), a boolean (Comparison, Value range), or a string (Value to string).
- **Boolean operators:** have one or more boolean states as input. Besides the basic operators AND, OR and NOT, there are more advanced boolean operators such as TRUECOUNT, which evaluates the number of true states between multiple outputs (see 15.6.7). The Categorical combiner will evaluate multiple boolean outputs and list the true output(s) as its own output. Boolean to string and Boolean to value operators will convert a boolean state into a string or value, respectively.
- **Sequence operators:** are specifically designed for sequence data. Find subsequence searches for a subsequence in a sequence data type, allowing for mismatches, gaps, and IUPAC notations. Amino acid translation translates a nucleic acid sequence into an amino acid sequence using a defined translation table.
- **Output actions:** perform an action on the database, which can be writing a result in a field, changing the selection status according to the result, writing a value in a character type experiment, writing into a sequence, or writing to an attachment. Output actions are only executed if the Execute button  is pressed.
- **Charts:** will produce a chart in the *Chart & statistics* window from the outputs of selected nodes. A chart will only be created if the Execute button  is pressed.
- **Miscellaneous:** contains a Duplicator, which allows one to duplicate a selected operator, e.g. to split up complex networks. The Is present operator returns whether a data component is present for an entry (a character, a sequence, or a fingerprint). The Execute script operator is beyond the scope of this manual.

15.6.4 Building a decision network

As an example, we will create a simple decision network that discriminates between the three genera in **DemoBase Connected**, based upon the 16S rDNA sequences. The **DemoBase Connected** can be downloaded directly from the *BioNumerics Startup* window or restored from the back-up file available on our website:

- To download the database directly from the *BioNumerics Startup* window, click the **Download example databases** link, located in the lower right corner of the *BioNumerics Startup* window. Select **DemoBase Connected** from the list and select **Database > Download**.
- To restore the database from the back-up file, first download the file `DemoBase_Connected.bnbk` from <http://www.applied-maths.com/download/sample-data>, under 'DemoBase Connected'.

In the *BioNumerics Startup* window, press the  button, select **Restore database**, browse for the downloaded file and select **Create copy**. Specify a name and click **<OK>**.

4.1 In the newly created decision network, open the Data sources group in the *Operators* panel by clicking on its  icon.

4.2 Double-click on Sequence, which opens a *Decision network operator* dialog box (Figure 15.6.3).

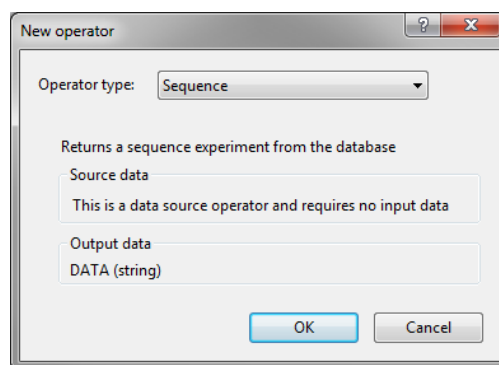


Figure 15.6.3: The *Decision network operator* dialog box for a Sequence operator.

This dialog box describes the operator and mentions the source data needed and the output data delivered to the network.

4.3 Press **<OK>** to edit the node properties for the sequence input node (Figure 15.6.4).

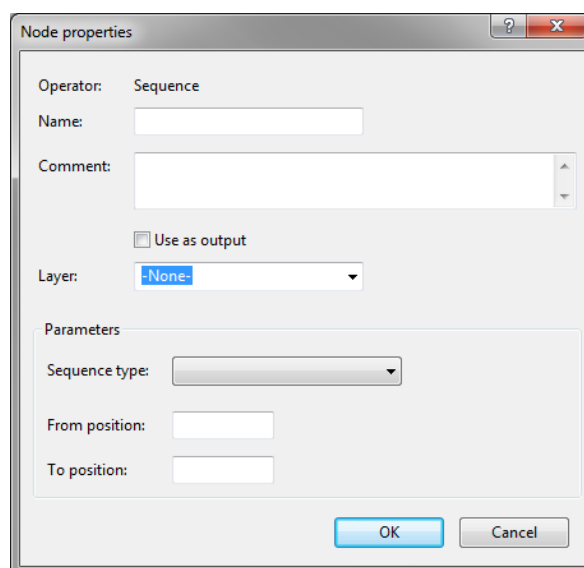


Figure 15.6.4: The *Node properties* dialog box for a Sequence operator.

Optionally, a **Name** can be entered for the node. If this is not done, it will be named automatically using a sequential number.

A **Comment** field can also be entered; this field will be shown in the *Node properties panel* (15.6.2).

When **Use as output** is checked, the result of the node is shown in the *Entry data panel* (right hand sub panel; see 15.6.2). This option makes little sense for a Sequence operator, as only sequences would be returned.

The **Layer** option is explained in 15.6.6.

Parameters lists some parameters that are specific for each operator type. For a Sequence operator, the **Sequence type** to feed the sequences can be selected from the drop-down list. With **From position** and **To position**, a range within the full sequences can optionally be specified to deliver to the network. This option only makes sense if the sequences are pre-aligned.

4.4 Enter e.g. "16S" as a **Name**, select **16S rDNA** (the only sequence type available in **DemoBase Connected**) as **Sequence type** and press <OK>.

The network now contains one node, i.e. "16S".

4.5 If you click on an entry in the *Entry data panel*, the node and the *Node properties panel* are updated with the sequence data of the highlighted entry.

4.6 Select the node "16S" in the network (a selected node is bordered by a red line).

4.7 Open the Sequence operators group in the *Operators panel* and double-click on Find subsequence.

The *New operator dialog box* appears, showing that this operator delivers multiple output data:

- **IsMatch** is a boolean reporting whether the subsequence occurs;
- **Start** and **End** are value-type data that return the start and end positions of the matching subsequence on the sequence;
- **Seq1** and **Seq2** return the query sequence and the matching subsequence on the data sequence, respectively. The latter can be different as the operator allows mismatches and gaps.

4.8 Press <OK> to edit the node properties (Figure 15.6.5).

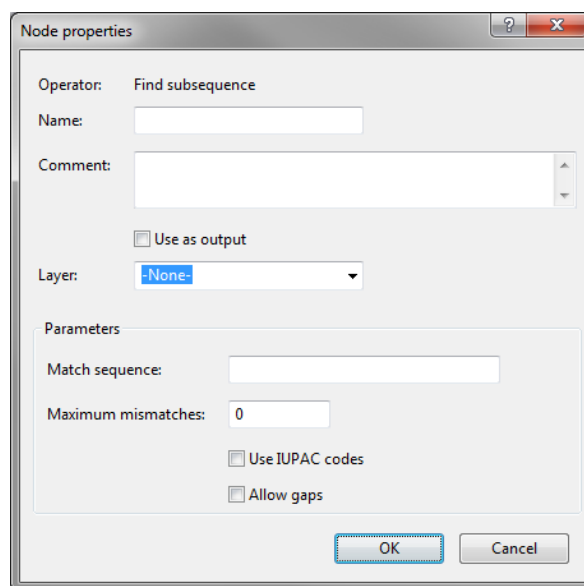


Figure 15.6.5: The *Node properties dialog box* for a Find subsequence operator.

Parameters specific for a Find subsequence operator are:

- **Match sequence:** in this text box you can enter the subsequence to search for
- **Maximum mismatches:** the maximum number of mismatches allowed
- **Use IUPAC codes:** if checked, IUPAC code for ambiguous base positions is taken into account
- **Allow gaps:** checking this option enables the introduction of gaps into the **Match sequence** or sequence that is being queried. A gap is considered as a mismatch.

4.9 Enter as **Name** “Signature 1”, and as **Comment** “Recognizes Ambiorix”, and check **Use as output**.

4.10 Enter GGGTGTAG as **Match sequence**, with zero mismatches allowed.

4.11 Press <OK> to confirm the node properties. The network is now ready to produce a first result.

4.12 In the *Decision Network* window, press the  button to calculate the network.

The *Entry data* panel now contains one output column, Signature 1, showing a boolean TRUE or FALSE for each entry. All *Ambiorix* entries have the boolean TRUE, the others FALSE.

In the decision network (Figure 15.6.6), the node “Signature 1” is marked with a green flag, indicating that it is an output node, resulting in a column in the *Entry data* panel.

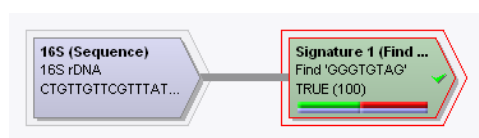


Figure 15.6.6: Simple decision network with one boolean output.

For each highlighted entry in the *Entry data* panel, the output node is either colored green (true) or red (false). The percentage of true and false entries in the entry data panel is indicated as a green and red bar, respectively. In addition, the percentage of selected entries is indicated with a blue bar.

For each highlighted sequence, the *Node properties* panel displays detailed information about the selected node: the input parameters as a first group and the output data as a second group.

4.13 Continue to build the network by selecting the data node “16S” again and adding a second “Find subsequence” node to it.

4.14 Enter “Signature 2” as **Name**, “Recognizes Vercingetorix” as **Comment**, and CGATCTCACG as **Match sequence**, with zero mismatches allowed.

4.15 Check **Use as output** and press <OK>.

4.16 Press the  button to calculate the network. The network now looks as in Figure 15.6.7.

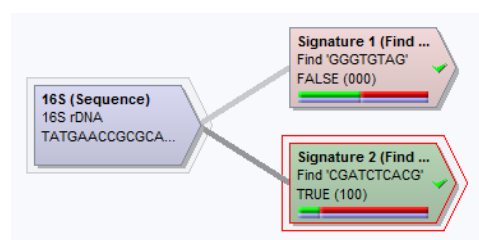


Figure 15.6.7: Decision network with two boolean output nodes.

A second column, Signature 2, is added to the *Entry data* panel. The *Ambiorix* entries have "Signature 1" true and "Signature 2" false, whereas the *Vercingetorix* entries have "Signature 2" true and "Signature 1" false. Entries of *Perdrix* have both signatures false.

Although this type of network might allow you to predict the properties for new and unknown entries, the output is not very descriptive or easily interpretable. We will now turn the network into a more descriptive result.

This time we will make use of a Duplicator operator. This operator duplicates a node in the network, i.e. one output parameter from it, which can be chosen. This tool is useful if a node is to be used in more than one independent operations in the network. Theoretically one could branch the different operations from the same node, but the network could easily become unsurveyable. As we need to check "Signature 1" and "Signature 2" to be both negative for *Perdrix*, we will duplicate both booleans and branch the new operation from there.

4.17 Select the subsequence search node "Signature 1" and, in the *Operators* panel, double-click on the Duplicator operator in the Miscellaneous group.

4.18 Select "Signature 1 (IsMatch)" as the parameter to be duplicated and press <OK>.

4.19 In the *Node properties dialog box* of the Duplicator operator, leave **Name** and **Comment** fields blank and leave **Use as output** unchecked, as this is an intermediate node.

4.20 Make sure **Hide link** is checked and press <OK>. This will place the node on a new line, separated from the other operators.

4.21 Select the duplicator node (red border) and create a new node using the NOT operator from the Boolean operators group.

4.22 Enter "No Signature 1" as **Name** for this node and press <OK>.

4.23 Repeat Instruction 4.17 to Instruction 4.22 for node "Signature 2". However, for the NOT boolean operator, enter this time "No Signature 2".

4.24 Select both boolean nodes "NOT" by clicking the first and then, while holding down the **Ctrl** key, clicking the second. Both nodes are now bordered in red.



You can also select multiple nodes by dragging the mouse over the nodes to select, while holding down the left mouse button.

4.25 Combine the two nodes with an AND operator from the Boolean operators group.

4.26 Enter "Perdrix" as **Name** for this node and press <OK>.

The network now looks as in Figure 15.6.8.

We now already have one boolean node called "Perdrix"; we still need to create similar nodes for the two other groups.

4.27 Select both nodes "Signature 1" and "No Signature 2" and combine them with a boolean operator AND.

4.28 As **Name** for this node, enter "Ambiorix" and press <OK>.

4.29 Select both nodes "Signature 2" and "No Signature 1" and combine them with a boolean operator AND.

4.30 As **Name** for this node, enter "Vercingetorix" and press <OK>.

The network now looks as in Figure 15.6.9. This network contains cross-branching operators, but is still surveyable thanks to the duplicator nodes.

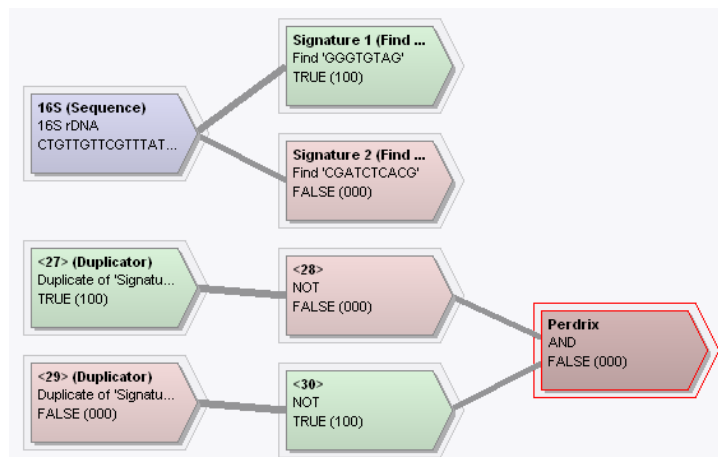


Figure 15.6.8: Decision network with duplicated nodes.

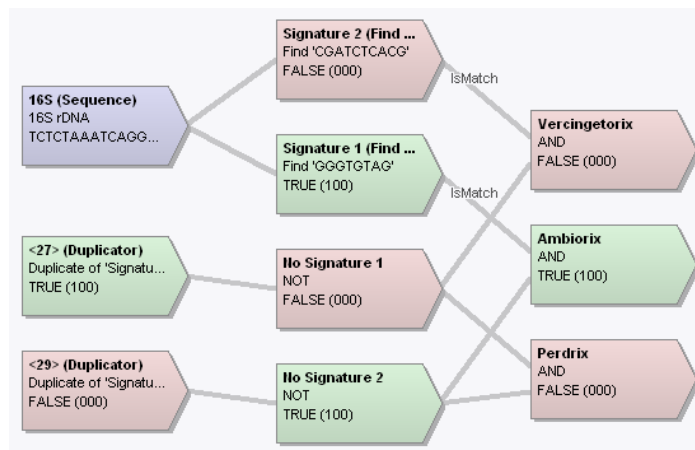


Figure 15.6.9: Decision network leading to three categorical boolean nodes.

Note that the connectors branching from the "Signature 1" and "Signature 2" nodes contain a tag "IsMatch", because these nodes have multiple outputs. "IsMatch" is the boolean that tells whether there is a match. The duplicator nodes do not contain this tag, because a duplicator can only contain one parameter from its parent, for which we chose "IsMatch".

If you click on any of the entries in the network, the end node with its name should be TRUE (green) whereas both others should be FALSE (red).



One of the advantages of a decision network over the advanced query tool (see 3.3.9) is that you can inspect the state for each evaluation or action in the network on the fly. The *Node properties* panel thereby shows all the details for the non-boolean operations.



If you click on any node, all the connector lines in the network that connect the node to either parent or descendant nodes are shown as bold dark lines. This makes it easier to inspect dependencies.

4.31 In the *Decision Network* window, press the  button to calculate the network.

The network now contains 3 mutually exclusive boolean nodes (only one out of the three can be true). Such nodes can be combined into a categorical set, i.e. a set of multiple states (categories), of which one state is

true for each individual. We will combine these categories into one node that tells the name of the genus:

4.32 Select all three end nodes ("Vercingetorix", "Ambiorix" and "Perdrix") and create a new node using the boolean operator Categorical combiner. The output for this node is a string, containing the category that is true.

4.33 Enter "Genus" as *Name* and check *Use as output*.

The parameter *Single choice (highest confidence only)* will allow the node to take out the category with the highest confidence value (see 15.6.7), in case more than one category turns out to be true. In this specific case, we do not need to enable this option.

4.34 Press <OK> to confirm the node properties.

4.35 In the *Decision Network* window, press the  button to calculate the network.


The *Entry data* panel now contains a new output column, Genus, showing the genus name for all entries selected in the decision network.

Finally, we will add an output action operator to write the result of the network in a database field.

4.36 In the *Main* window, add a new information field, e.g. "Name by DN".

4.37 In the *Decision Network* window, select the categorical combiner node "Genus" and double-click the Write to field operator from the Output actions group.

4.38 In the *Node properties dialog box* of the Write to field operator, enter a name (optional), e.g. "Write to database", and select field "Name by DN" as the *Database field name*.

The finished decision network now looks as in Figure 15.6.10. The nodes that perform an output action are orange, to indicate that these nodes can perform changes to the database. For safety reasons, the output actions are not executed automatically when the network is calculated using the  button.

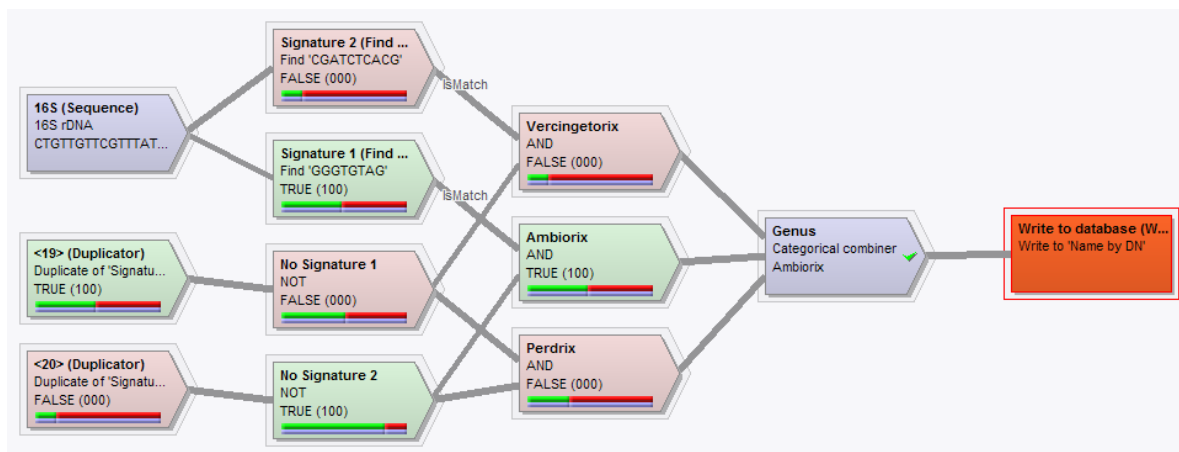



Figure 15.6.10: Decision network that decides between three groups and writes the output to the database.

4.39 To calculate the network and execute the output action(s), press the  button.

To alert you that the network will now perform changes to the database, the following warning box appears (Figure 15.6.11).

4.40 Press <OK> to confirm the execution. When finished, the database field "Name by DN" contains the genus names as defined by the decision network.

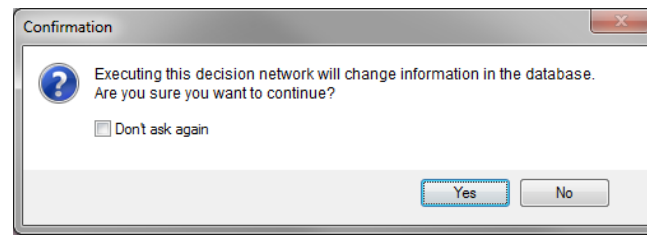


Figure 15.6.11: Warning box that appears if a network containing output nodes is executed.

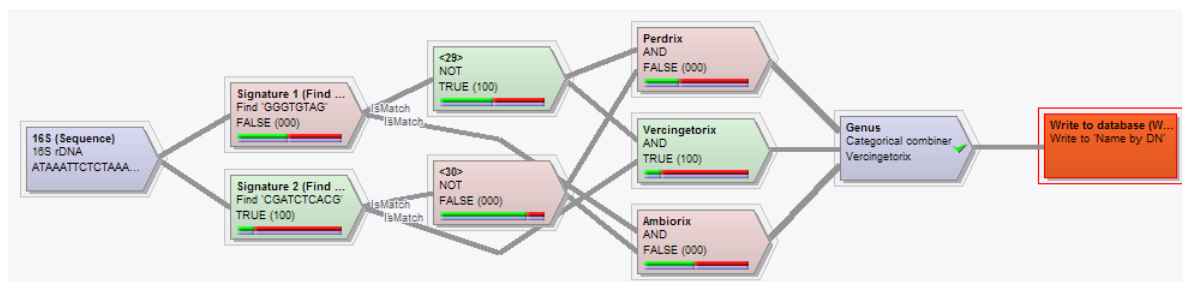


Figure 15.6.12: The same network as in Figure 15.6.10, without duplicator nodes.



The entire decision network could have been build without the duplicator nodes. In that event, more cross-connectors would exist and the network would be less surveyable (Figure 15.6.12). However, the result would be the same.

15.6.5 Display and output options for decision networks

It is possible to zoom in or out on a decision network using **View > Zoom in** (🔍) and **View > Zoom out** (🔍) or by extending or compressing the zoom slider in the *Network* panel (see 2.3.7 for instructions on the use of zoom sliders).

Select **File > Print...** to print the decision network. One will be prompted for the printer to use and basic printer settings. Only the content of the *Network* panel is sent to the printer.

The content of the *Network* panel can also be copied to the Windows clipboard for import in other programs, e.g. to create reports:

Select **File > Copy to clipboard (metafile)** to export the network as a metafile. Paste the clipboard in a program such as MS Word or PowerPoint to see the exported graphical representation of the network.


Select **File > Copy to clipboard (bitmap)...** to export the network as a bitmap. The program will prompt for the bitmap resolution, enter e.g. 1,000. The clipboard can now be pasted in a graphics editor such as Adobe Photoshop.


15.6.6 Working with layers in a decision network

The use of *layers* in a decision network is to structure and organize separate sub-flows in complex networks. Whereas by means of *duplicators* one can duplicate a node to start at a new line and continue the flow from there, a layer is a *sub-flow* of the network that can be visualized separately from the others.

A layer can only be created along with the creation of a new node. To illustrate the use of layers we will add a sub-flow to the existing decision network **My DN**. Suppose we will evaluate two character values to

define resistance of the entries.

- 6.1 Create a "Character value" node by double-clicking on the Character value operator in the Data sources group.
- 6.2 In the *Node properties dialog box*, type "Resistance" in the input field **Layer**.
- 6.3 Choose **PhenoTest** as **Experiment** and select "c4" as **Character**.
- 6.4 Enter "Char 1" as **Name**, and press <OK>.
- 6.5 Repeat Instruction 6.1 to Instruction 6.4 for a second character "c12" from **PhenoTest**, entering "Char 2" as **Name**.
- 6.6 Select the character value node "Char 1" and double-click on the Value range operator in the Value operators group.
- 6.7 In the *Node properties dialog box*, select "Resistance" from the drop-down box under **Layer**.
- 6.8 Enter "2" as **Minimum value** and press <OK>.
- 6.9 Repeat actions Instruction 6.6 to Instruction 6.8 for node "Char 2", entering the same minimum value "2".
- 6.10 Select both "Value range" nodes and combine them with a boolean operator AND.
- 6.11 In the *Node properties dialog box* of the "AND" node, specify "Resistance" as **Layer** and enter "Multi resistant" as **Name**.
- 6.12 Check **Use as output** and press <OK>.
- 6.13 Calculate the network with ; a new column is added to the output sub panel of the *Entry data panel*.

The toolbar in the *Decision Network* window contains a drop-down box  that allows you to select a layer to visualize. By default, the complete network is visualized.

- 6.14 Select "Resistance" from the drop-down box. Only the nodes that belong to the resistance flow are now shown (Figure 15.6.13).

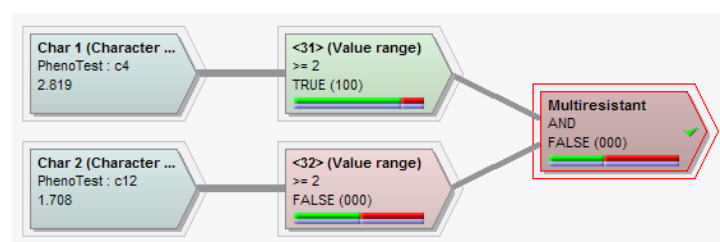


Figure 15.6.13: Decision network with one layer visualized (see text for explanation).

Nodes from a layer can be connected to other nodes belonging to the complete network or a different layer. In that case, the subpart of the complete network or the other layer(s) that contribute to the outcome of the layer are shown along with the layer.

- 6.15 As an example, select "Complete network" again from the drop-down box.
- 6.16 Select the boolean nodes "Multi resistant" and "Ambiorix", and connect them with a boolean AND operator.
- 6.17 In the *Node properties dialog box* of the AND operator, specify a **Name**, e.g. "Multi resistant Ambiorix" and choose "Resistance" as **Layer**.

6.18 Select "Resistance" from the drop-down box. The network now looks as in Figure 15.6.14.

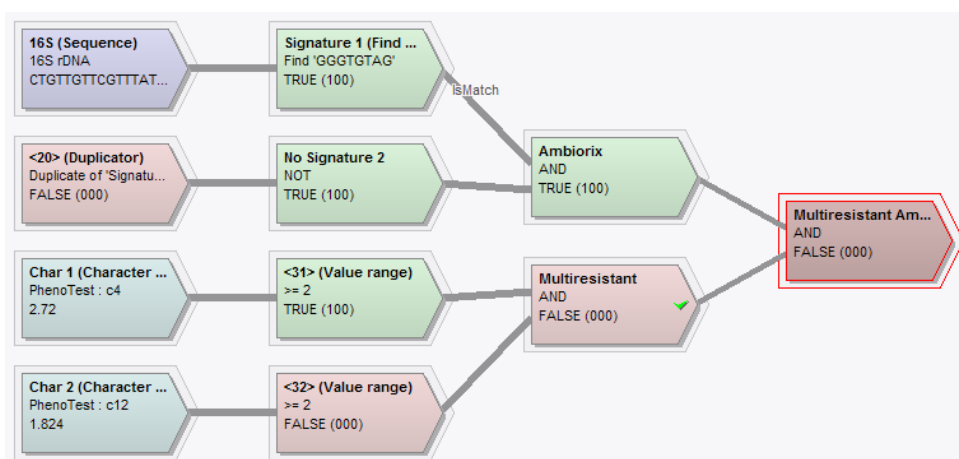


Figure 15.6.14: Example of a layer visualized along with nodes that contribute to the output.

15.6.7 Using confidence values

Confidence values are useful in cases where answers cannot be formulated clearly as either yes or no. For example, in the network we created in 15.6.6, the decision whether to label the entries as "resistant" or not, depends on the fact that a value is more or less than 2. In reality, however, states cannot be clearly defined from non-binary measurements. Therefore, it is possible to enter a **Fuzzy zone** in a value range operator.

7.1 In the network layer created in 15.6.6, double-click on the value range node "Char 1" to re-edit it.

7.2 In the *Node properties dialog box*, enter "1" as **Fuzzy zone**. Press <OK> to confirm the change.

7.3 Repeat Instruction 7.1 to Instruction 7.2 for the value range node connected to "Char 2", also entering "1" as **Fuzzy zone**.

The **Fuzzy zone** extends equally to both sides of the limit(s) entered for the range (see Figure 15.6.15). For example, if you entered 2 as limit, the answer will still be FALSE for all entries that have the value below 2 and TRUE for all those that have more than 2. However, all values between 1.5 and 2.5 will exhibit a confidence that is bigger than 0 and lower than 100. A value of exactly 2 will have a confidence of 50.

After recalculating the network with , each output TRUE or FALSE contains a value that ranges between 0 and 100%.

The confidence values are preserved throughout the flow of the network: when, for example, two booleans are combined with AND (Figure 15.6.14), the lowest of the two values is used. In case two booleans are combined with OR, the highest of the two values is retained.

Confidence values are also optionally used in the Categorical combiner operator (Instruction 4.32). If the **Single choice (highest confidence only)** option is enabled, and in case multiple categories appear to be true, the network will look for the category with the highest confidence value to retain. If this option is not enabled, multiple true categories will be listed together, separated by semicolons.

15.6.8 Building decisions relying on multiple states

In 15.6.4 we have already described the Categorical combiner operator (Instruction 4.32), which evaluates a number of boolean nodes and uses the name of the most true node as its output. In a number of cases,

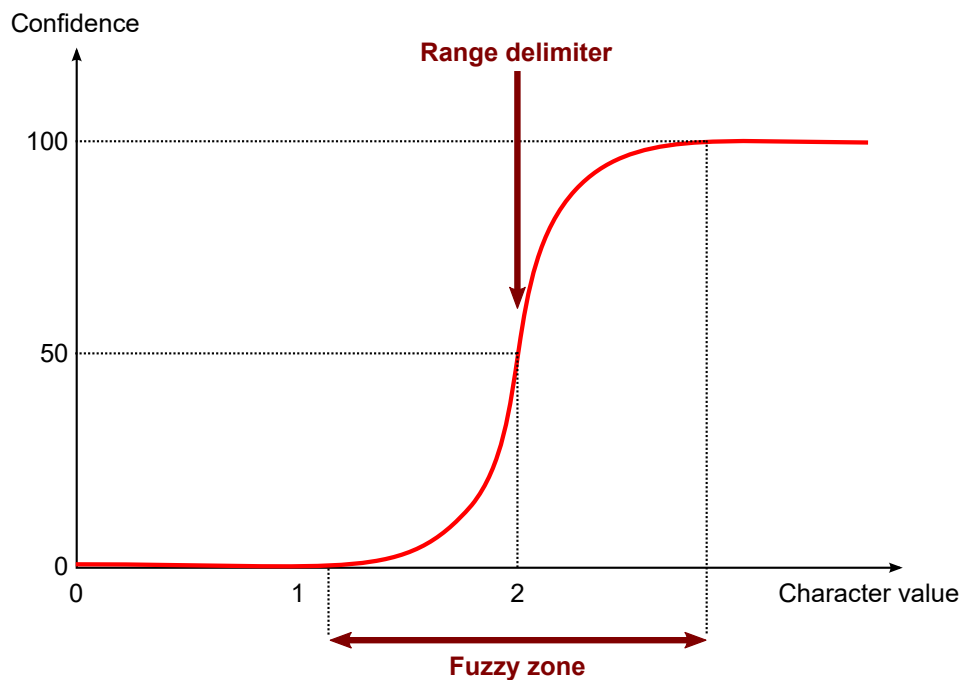


Figure 15.6.15: Graphical illustration of the effect of a fuzzy zone on the confidence value of a boolean decision.

however, it might be required to build decisions upon conditions such as "at least x true states", and/or "at most y true states". This can be achieved using the TRUECOUNT operator.

The TRUECOUNT operator (Figure 15.6.16) combines a number of boolean nodes, and is set to TRUE if at least x booleans are true (*At least true*) and/or at most y booleans are true (*At most true*) (with $y \geq x$).

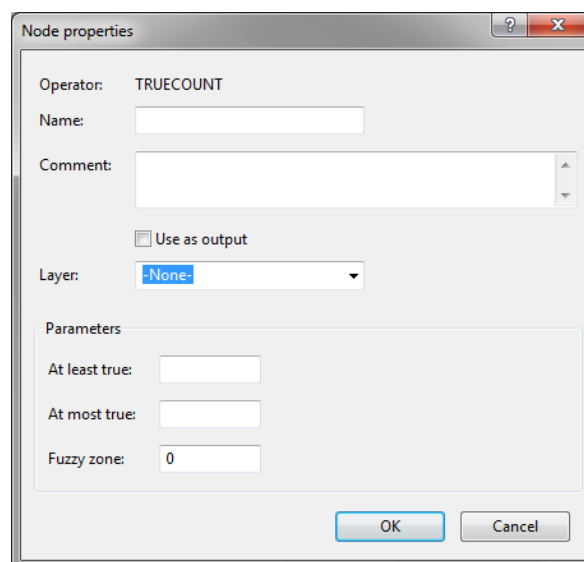


Figure 15.6.16: The *Node properties* dialog box of the TRUECOUNT operator.

For example, suppose that a bacterial strain is multi-drug resistant if it exhibits resistance for at least 12 antibiotics out of a set of 20. A network can be built that calculates multi-drug resistance by using value range nodes for each of the antibiotics, and combining them using a TRUECOUNT operator that has "12" as *At least true* parameter.

The boolean operator TRUECOUNT also makes use of confidence values. Optionally, one could enter "11"


for *At least true* and use "2" for *Fuzzy zone*, which would mean that entries with resistance towards 12 antibiotics are shown as multi-drug resistant with 100% confidence, and entries with resistance towards 11 antibiotics are shown as multi-drug resistant with 50% confidence.







Since the TRUECOUNT operator only recognizes integer values as input, the *fuzzy zone* value has to be an even number of at least 2. A fuzzy value of 1 would be divided into 0.5 at either side of the delimiter (see Figure 15.6.15) and would be truncated to zero.

15.6.9 Creating charts from a decision network


BioNumerics can plot the results of a decision network in a chart (scatter plot, bar graph, contingency table, etc.) by using its chart and statistics tools. There are two ways a chart can be generated from a decision network:

- The result of every node that is specified as output node can be plotted using the  button. Depending on the content of the node, the *Chart & statistics* window automatically generates the suitable plot type.
- Charts can also be created as an output action, in which case a chart is automatically generated when the network is executed.

To illustrate the manual creation of a graph from a node, we will make some graphs from nodes in the network we created in the previous paragraphs.

- 9.1 Make sure "Complete network" is selected from the drop-down menu of the *Decision Network* window (see Instruction 6.13).
- 9.2 Specify the categorical combiner node "Genus" as an output node (if not already specified this way) by enabling *Use as output* in the *Node properties dialog box*.
- 9.3 Recalculate the network with .
- 9.4 Select the "Genus" node and press the *Network > Plot in chart window...* (). A bar graph appears, showing the relative occurrences of the three genera (Figure 15.6.17).
- 9.5 Close the *Chart & statistics* window with *File > Exit*.
- 9.6 Select the data source node "Char 1" and specify it to be an output node, similar as in Instruction 9.2.
- 9.7 Recalculate the network with .
- 9.8 Select *Network > Plot in chart window...* (). A cumulative distribution of the character values appears (Figure 15.6.18).
- 9.9 Close the *Chart & statistics* window with *File > Exit*.

To create a graph each time the network is executed, we will use the nodes "Char 1" and "Char 2" as input, and create a scatter plot from them.

- 9.10 Select both nodes "Char 1" and "Char 2", and create a new output action node by double-clicking on the 2D scatter plot operator in the Charts group under Output actions.
- 9.11 Execute the network by pressing the  button. A scatter plot is generated in a *Chart & statistics* window, comparing the two characters for each entry used in the network (Figure 15.6.19).
- 9.12 Close the *Chart & statistics* window with *File > Exit*.

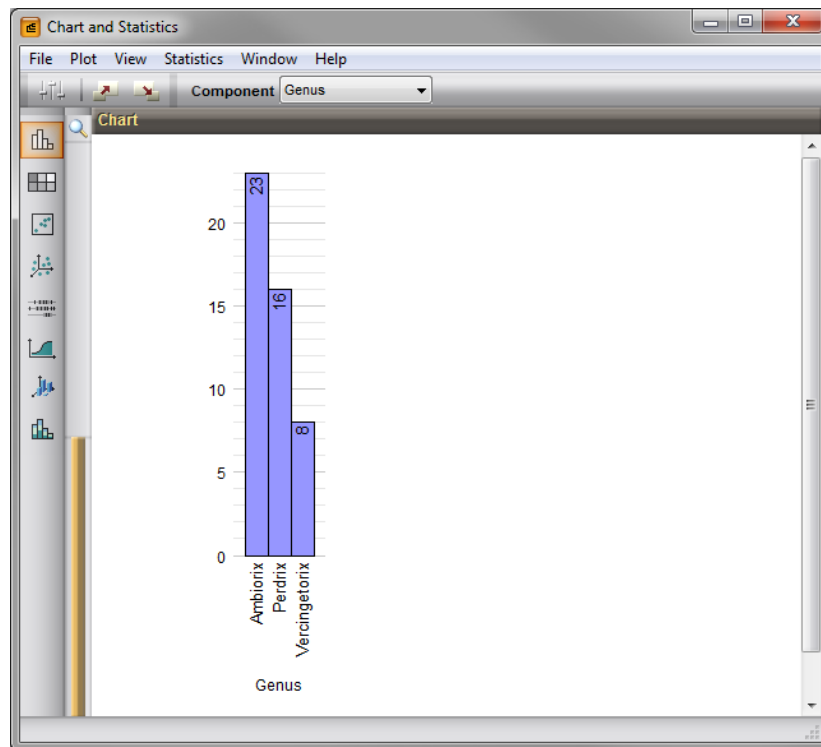


Figure 15.6.17: Bar graph popped up from categorical combiner output node.

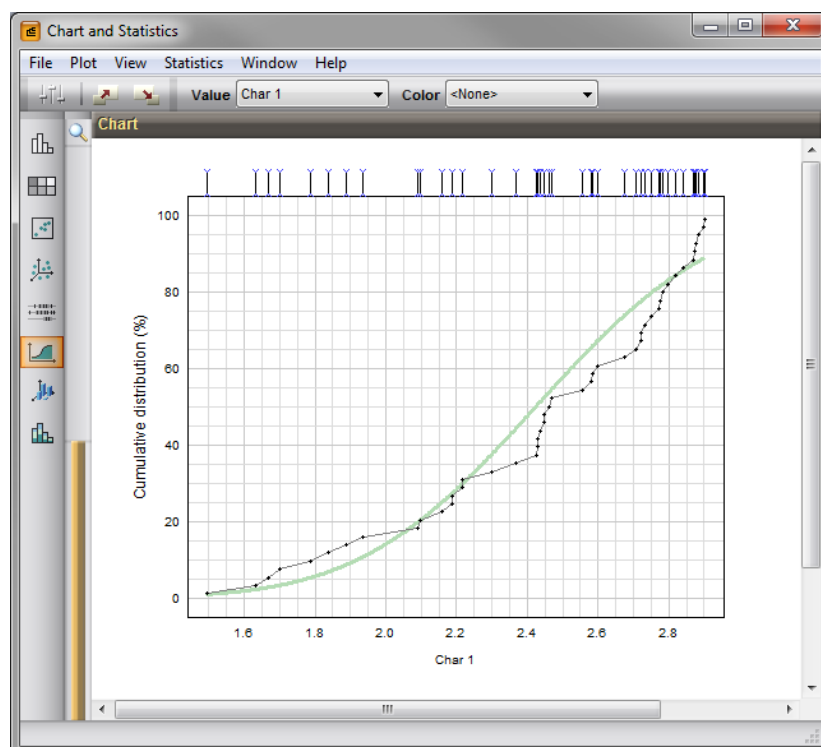


Figure 15.6.18: Cumulative distribution popped up from a character value output node.

15.6.10 Executing a decision network from the main window

Once built and saved, a decision network can be used as a tool to perform certain manipulations on the database in an automated way. These manipulations are the output actions defined in the network.

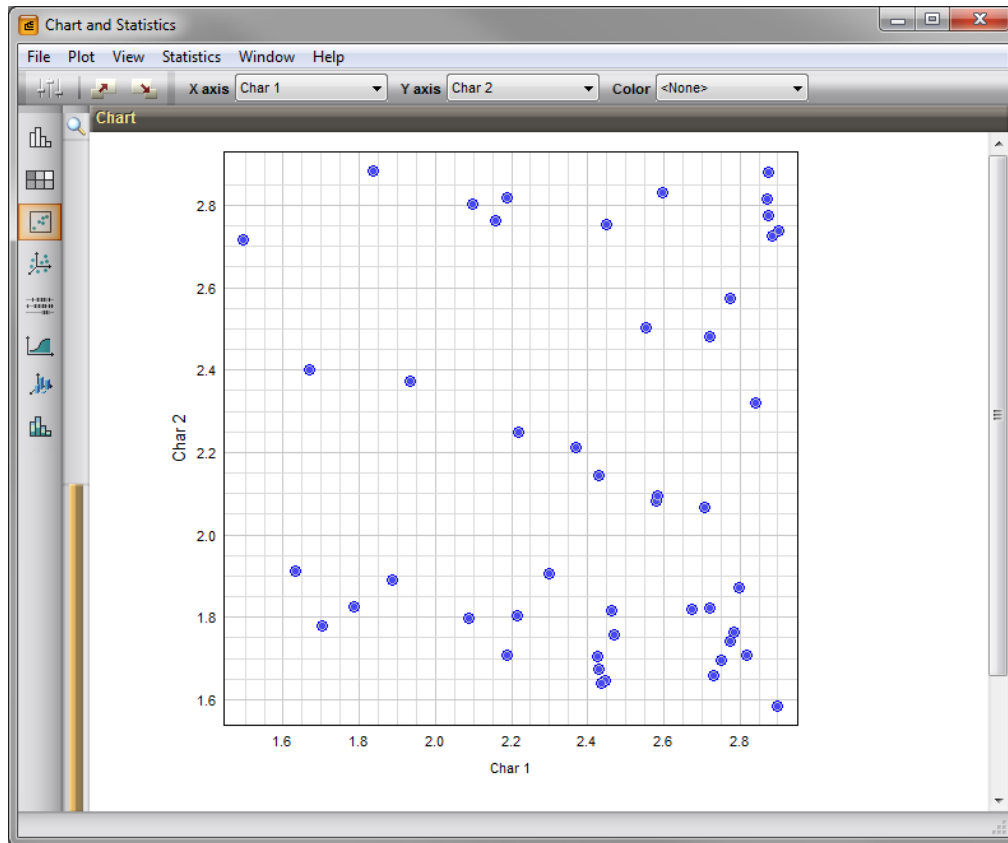



Figure 15.6.19: Scatter plot graph obtained from an output action node.

For example, if you save and quit the network created in the previous paragraphs, you will notice that the network is listed in the *Main* window, in the *Decision networks* panel (see Figure 15.6.2).

10.1 You can directly execute the network from the *Main* window by pressing the  button in the toolbar of the *Decision networks* panel.

A dialog box pops up (Figure 15.6.20), offering three choices for executing the decision network: on *All entries*, on *Currently selected entries only*, or on *Non-selected entries only*.

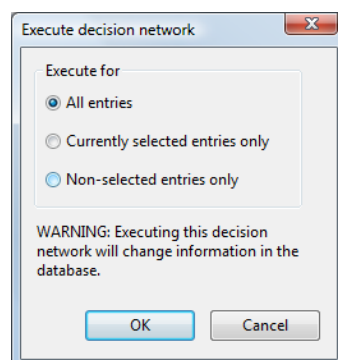


Figure 15.6.20: Choices for executing a decision network directly from the *Main* window.

10.2 Check *All entries* and press <OK>. All output actions defined in the network are executed on all entries in the database: output strings are written in defined information fields, and charts are generated.

One obvious application for executing a decision network directly from the *Main* window is to use it as a kind of an advanced query tool: the output action operator Change selection will change the selection status

of the entries into selected if the input for the "Change selection" node is TRUE, or non-selected if the input for the "Change selection" node is FALSE.

15.6.11 Decision trees

A decision tree is mainly used to build bifurcating decision schemes based on a number of TRUE/FALSE evaluations. A decision tree does not provide possibilities which cannot be achieved in a normal decision network as described in the previous paragraphs. However, it allows the scheme to be presented in a more intuitive way and can be a suitable asset for e.g. taxonomic identification schemes.

The operators to build a decision tree can be found in the Boolean operators group, where they are grouped in a category Decision trees. The tree always starts with a Decision tree root, which will contain the output of the tree as well.

In the example below, we will create a decision tree that performs the same task as the decision network created in 15.6.4, i.e. identifying entries at the genus level based upon a signature sequence.

11.1 Create a new decision network as described in 15.6.2. Enter "Decision tree" as *Name*.

11.2 Select a number of database entries and open the decision tree.

11.3 Under Boolean operators, open the category Decision trees and double-click on the Decision tree root operator. Enter "Genus name" as *Name* and select *Use as output*.

The data evaluations and the definitions of the criteria have to be done in a separate flow of the network, and the decision tree is built on the outcome of those evaluations. We will now create the criteria needed for the identification tree (see also 15.6.4).

11.4 Under Data sources, double-click on Sequence to create a sequence data source node.

11.5 Select 16S rDNA as *Sequence type* and press <OK>.

11.6 With the "Sequence" node selected, double-click the Find subsequence operator in the Sequence operators group. Enter "Ambiorix signature" as *Name*, and GGGTGTAG as *Match sequence*.

11.7 Again with the "Sequence" node selected, create a second "Find subsequence" node. Enter "Vercingetorix signature" as *Name*, and CGATCTCAG as *Match sequence*.

Back in the decision tree, we will create a bifurcation based upon the presence of the Ambiorix signature, as follows:

11.8 Select both the decision tree root and the "Ambiorix signature" node and create a Bifurcation (under Boolean operators > Decision trees).

The *New operator dialog box* looks as in Figure 15.6.21.

As *Input (boolean)*, the tree root should be selected, whereas as *Criterion (boolean)*, the Find subsequence node "Ambiorix signature" should be selected.

As a *Name*, you can enter "Is Ambiorix". The decision tree network now looks as in Figure 15.6.22.

We will now create two *leaves* branching off from this bifurcation: one for TRUE and one for FALSE.

11.9 Select the bifurcation "Is Ambiorix" and double-click on the Decision tree leaf operator. The *Source data* can either be "Is Ambiorix (POS)" or "Is Ambiorix (NEG)", standing for a TRUE and FALSE condition, respectively.

11.10 Select "Is Ambiorix (POS)" as *Source data*, and enter "Ambiorix" as *Name*.

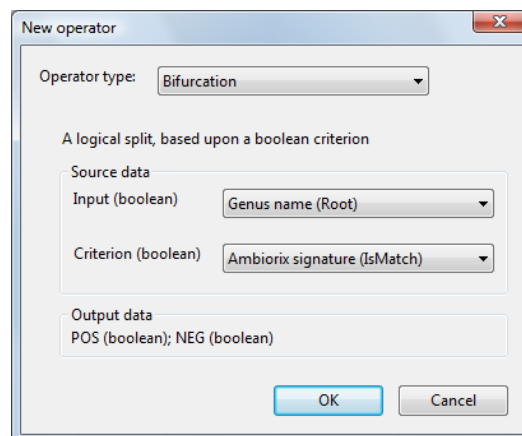


Figure 15.6.21: The *New operator* dialog box for a bifurcation.

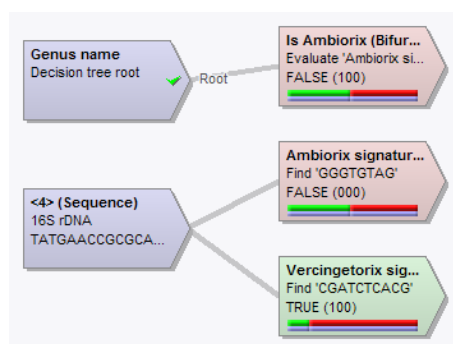


Figure 15.6.22: Decision tree in construction, containing one bifurcation.

11.11 To create the second bifurcation, select the bifurcation "Is Ambiorix" again and create a second decision tree leaf.

11.12 Select "Is Ambiorix (NEG)" as *Source data*, and enter "Not Ambiorix" as *Name*.

This is an example of the simplest decision tree that exists, evaluating one criterion and identifying entries as belonging to a taxon or not (Figure 15.6.23).

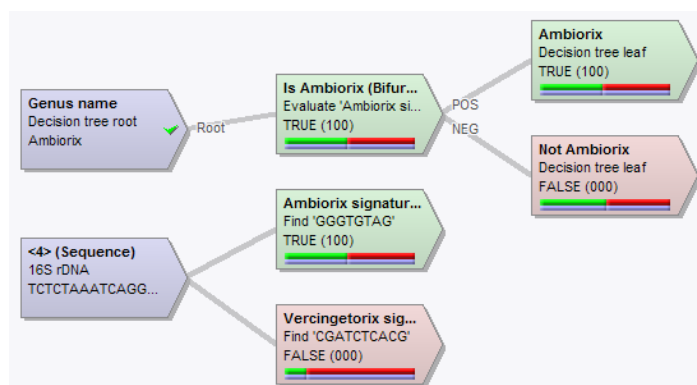



Figure 15.6.23: Simple decision tree based upon one criterion.

11.13 If you calculate the decision tree with , you will notice that for each entry you select, the decision tree root either shows "Ambiorix" or "Not Ambiorix".

As we defined the root as an output node (see Instruction 11.3), the right sub panel of the *Entry data panel*

also displays this result for all the entries used in the network.

To create a decision tree that identifies the three genera, we have to insert further criteria at the first bifurcation, rather than a leaf:

11.14 Delete the leaf node "Not Ambiorix".

11.15 Select both the bifurcation "Is Ambiorix" and the Find subsequence node "Vercingetorix signature", and create a new bifurcation from these nodes.

11.16 Select "Is Ambiorix (NEG)" as **Input (boolean)** and "Vercingetorix signature (IsMatch)" as **Criterion (boolean)**.

11.17 Enter "Is Vercingetorix" as **Name**.

To finalize the tree so that it identifies the three genera, we further have to insert two leaves:

11.18 Select the "Is Vercingetorix" bifurcation node and create a new decision tree leaf node.

11.19 As **Input (boolean)** for the leaf, select "Is Vercingetorix (POS)".

11.20 Enter "Vercingetorix" as **Name**.

11.21 Finally, select the "Is Vercingetorix" bifurcation node again and create a second decision tree leaf node.

11.22 As **Input (boolean)** for the leaf, select "Is Vercingetorix (NEG)".

11.23 Enter "Perdrix" as **Name**.

The finalized tree now looks as in Figure 15.6.24, which is simpler to interpret than the comparable decision network depicted in Figure 15.6.10.

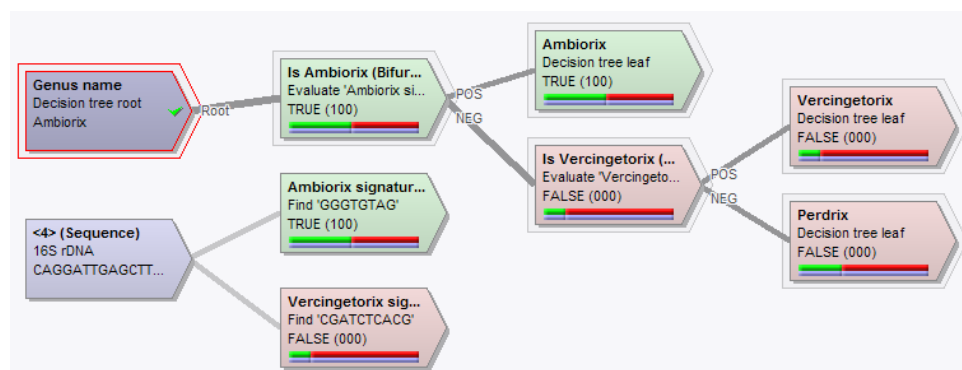


Figure 15.6.24: Decision tree that decides between three possible genera, in two dichotomic steps.

Optionally, you can add an output action node to the root, writing the result to a database field, as explained in Instruction 4.38.

The criteria defined for this decision tree could have been created in a separate layer (see 15.6.6), which would allow us to better separate the tree from its criteria and display either the tree or its criteria.

Part 16

Advanced cluster analysis

Chapter 16.1

Background information

16.1.1 Introduction

The **Advanced cluster analysis** tools in BioNumerics provide a powerful platform for all major hierarchical clustering schemes applied in biology. Besides offering a flexible interface with unparalleled display, editing and analysis options, the greatest innovation comes from the new method to provide reliability values for branches based upon uncertainty of data and degeneracy of tree solutions. Rather than being implemented within an algorithm, the method builds a statistical framework around the algorithm so that (1) the algorithm itself does not have to be rewritten and (2) the method can be applied to any existing hierarchical clustering scheme. In the current version, the advanced cluster analysis tools include the following clustering schemes:

- **Global closest pair methods:** unweighted pair group method with arithmetic mean (UPGMA), weighted pair group method with arithmetic mean (WPGMA), single linkage method, complete linkage method, Ward's method, centroid method, median method;
- **Neighbor Joining methods:** classical Neighbor Joining, Bio-Neighbor Joining, NeighborNet;
- **Parsimony and maximum likelihood methods:** basic maximum parsimony, optimized maximum parsimony, recomposed maximum parsimony (quartet puzzling), basic maximum likelihood, optimized maximum likelihood;
- **Minimum spanning tree methods:** Prim's MST algorithm, MST with hypothetical nodes.

Note that not all methods above result in *trees* or *dendrograms*. Some of them, e.g. NeighborNet, produce *networks*. As will be seen in the sequel, the results of a tree inferring algorithm after analysis using the statistical framework can also be a *network*. Since trees are only a special form of networks, which is the more general term, the term network will often be used to include both networks in the strict sense and trees.

In order to have a good understanding of all the options and parameters, some basic knowledge on the reliability issues that are inherent to clustering schemes and the solution offered in BioNumerics is required. All of the network methods mentioned above are subject to two important deficiencies.

- The first deficiency is called the *degeneracy* problem, which means that a solution obtained for a clustering method is usually not unique. In most cases, a clustering method provides a whole set of equivalent solutions for the same input data. The solution returned can differ depending on biologically irrelevant factors such as the algorithm used and the order of the input entries.
- The second deficiency is a consequence of the fact that the input data itself is not perfect, but is implicitly considered to be perfect by the algorithm. Minor variations in the input data values, for example within the boundaries of the known experimental error, can cause the same algorithm to produce trees with different clusters. This problem is denoted as the *input data imperfection*.

16.1.2 The problem of degeneracy

A solution of a cluster analysis method is a network which has been constructed according to the rules by which the method is defined. However, in many cases, these rules may lead to several equivalent solutions, which means that the method has a set of solutions rather than a unique solution. Consider the following simple example in which three electrophoresis patterns are clustered using the UPGMA method (Figure 16.1.1):

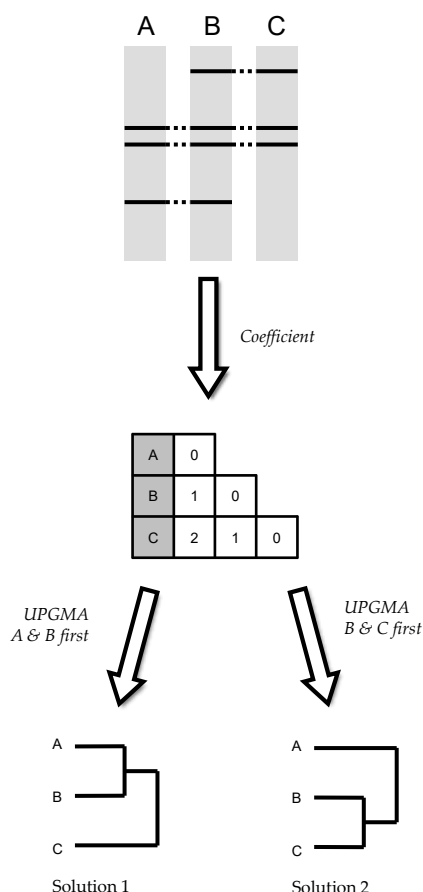


Figure 16.1.1: Example of degeneracy of UPGMA tree solutions.

The UPGMA iteration rule specifies that the two clusters with minimal distance should be merged. In the present example, both [A,B] and [B,C] have minimal distance, which means that two degenerated solutions result from this data set. Obviously, larger data sets will result in distance matrices with numerous equal distances. The accumulation of equivalent choices in the UPGMA iteration results in a multitude of possible combinations and a large number of UPGMA solutions.

The degeneracy problem appears in all methods listed above. It is particularly prominent with binary or categorical data as input data, e.g. nucleotide sequences, MLST allele data, MLVA typing, electrophoresis fingerprints recorded as band presence/absence, binary character data, etc.. With extremely simple data as input, e.g. MLST allele numbers from 7 housekeeping genes, a single random solution of a hierarchical clustering method is biologically uninformative. Part of the degeneracy problem can be solved by adding criteria to the rules of the clustering method. Those additional criteria may have a biological interpretation so that a biologically more relevant solution is presented out of a large number of theoretically possible solutions. To cluster MLST data for example, so-called priority rules have been introduced to restrict the set of solutions to those that have a meaningful interpretation in view of population dynamics. However, this approach does not solve the degeneracy problem.

16.1.3 The problem of input data imperfection

Data imperfection typically occurs in case of quantitative measurements, e.g. enzymatic activity, spot intensities, antibiotic resistance measurements using inhibition zones, etc.. Such data is usually transformed into a distance (or similarity) matrix prior to clustering, which means that the clustering algorithm ideally should take into account a certain amount of error on each individual distance value. However, an implementation of a clustering method makes the implicit assumption that the slightest differences in distance are significant. For example, a distance of 1.497 will always be considered less than 1.502 by any implementation of a clustering scheme. If we now assume that each distance value has a certain error, we have to consider every decision of the clustering algorithm where multiple possibilities occur within the boundaries of that error, as leading to an equivalent solution. If the error on the distance values in the above example is 0.05, these distances, and perhaps others as well, should be treated as equal. As such, the data imperfection problem leads to a degeneracy problem as described above, where occurrence of multiple error-based degeneracies may accumulate into numerous possible solutions.

Note that, in terms of the input data, the data imperfection problem is more or less complementary to the degeneracy problem. Pure degeneracies occur frequently with simple binary or categorical data. A set of 7 housekeeping genes, for example, results in 7 alleles which can only be the same or different. Hence, a matrix of distances can only contain 7 discrete distance values. It is obvious that many pairs of entries will have the same distance value, so that degeneracy is very prominent. Such pure degeneracies will almost never occur when distances are calculated from quantitative measurements, especially if the values are measured with decimal accuracy. On the other hand, data imperfection (e.g. experimental error) is not an issue with binary or categorical data whereas it is usually inherent to the measurement process involved in obtaining quantitative data.

16.1.4 The resampling framework

There are various reasons why it is practically impossible to design an algorithm that produces an exhaustive list of solutions given a clustering method and a data set. Most notably, it would require a specific algorithm to be developed for every existing clustering method. In addition, the amount of computer memory and computing time required might easily exceed the capacity of a normal workstation.

Therefore, rather than attempting to calculate all possible solutions, BioNumerics uses a sampling approach which produces a representative set of solutions for a given set of input data. Based upon the set of solutions, a maximum score tree or consensus tree or network is calculated for which the significance of each branch can be indicated. The method is built as a framework around a clustering method, according to the scheme in Figure 16.1.2.

From the input data, which can be a data matrix as well as a distance (similarity) matrix, a number of resampled data sets are generated by the *data factory*. Each resampled data set is clustered using the chosen clustering method. The resulting trees are summarized into a consensus network by the *summary factory*. The *resampling framework* is the entire framework that includes data factory, the encapsulated clustering algorithm and the summary factory.

16.1.5 Data resampling techniques

At the level of the *data factory*, three types of resampling can be applied, depending on the type of data:

- **Permutation resampling:** The entries of the input data set are rearranged randomly into a new data set. A large number of thus resampled data sets are generated and clustered. The resulting set of trees is a representative sample of the total space of degenerated solutions and is processed by the *summary*

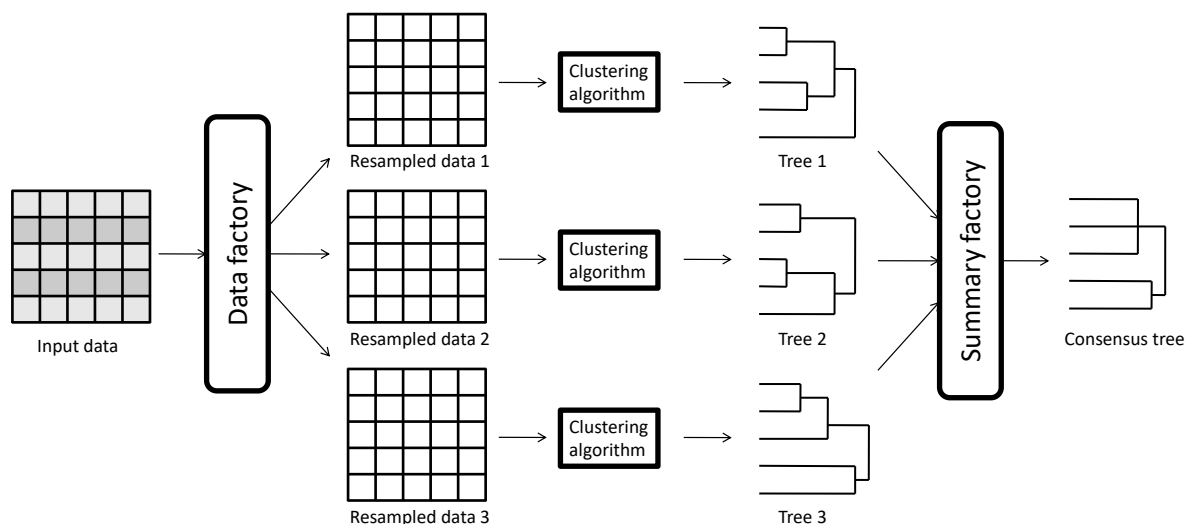


Figure 16.1.2: Schematic representation of the resampling framework in the BioNumerics tree and network inference tools.

factory into a summary tree. Permutation resampling can be performed on character data sets as well as distance matrices. The purpose of permutation resampling is to assess the independence of a particular degenerated solution for each cluster of a tree: those clusters that appear in the majority or in all of the evaluated solutions are largely independent of the degeneracy and can be considered significant in view of the degeneracy problem. Permutation resampling is very useful in clustering binary and categorical data, especially when the number of characters is limited. Examples are binary band-based similarity matrices of electrophoresis fingerprints (e.g. Dice, Jaccard), allele-based MLST data, allele-based MLVA data, binary character data etc..

- **Bootstrap resampling:** A new input data matrix is created by bootstrapping the original input data in the sense of Efron (1979) [14]. This resampling method can only be applied to character data matrices, not to distance matrices. Bootstrap resampling has been applied to estimate the significance of the branches from a single existing tree (Felsenstein, 1985) [16]. In the BioNumerics implementation, all trees resulting from a large number of bootstrap resamplings are processed by the *summary factory* to obtain a more reliable summary tree. Bootstrap resampling has the same goal as the Felsenstein bootstrap, i.e. providing a statistical measure of the robustness of the clusters in function of the input data. However, rather than using one individual tree and assigning its robustness, BioNumerics uses all obtained trees to calculate a maximum score tree or a consensus tree that summarizes all the most reliable branches (see 16.1.6). If one individual tree calculated from the original data set is used, the framework is reduced to a Felsenstein bootstrap. The data types suitable for the bootstrap resampling are the same as for the Felsenstein bootstrap, i.e. matrices of independent binary or categorical characters such as DNA or protein sequences.
- **Error resampling:** A new distance matrix is created by adding an error to each value from the original distance matrix. The errors are randomly drawn from a normal probability distribution with a mean of zero and user-defined standard deviation. All trees resulting from a large number of error resamplings are processed by the summary factory into a more reliable summary tree. Error resampling assesses the independence of a tree solution with respect to error on the input data. Clusters that are found in the majority or all of the resampling solutions are largely independent of the error on the input data and can be considered significant in view of the error problem. Error resampling is useful for all experiment types that produce non-exact data values, usually quantitative. Unfortunately, it is much easier to determine an error on quantitative character measurements than on distance values. The proposed solution is to repeat the same experiments for a given number of entries and to calculate a distance matrix for each repeated data set. The average of observed fluctuations on the corresponding

distance values is a good indication of the size of the error to be used for the distance matrix.

16.1.6 Tree summarizing techniques

The *summary factory* (see Figure 16.1.2) is a mechanism that summarizes a number of trees with different topologies into a single representation. Depending on the summary method used, this representation can be a tree or a network. When the outcome is a tree, depending on the method used it can be a true solution for the clustering method or a consensus tree. The summarizing techniques as well as the calculation of the reliability values (i.e., the *resampling support*) are based upon branch counting. Without dealing with the technical details, it should be mentioned that a branch (or a splitting branch set) is a set of nodes that can be split from the rest of the tree and that is identified by the entries it contains. A branch is considered the same in two trees if in both trees the same entries can be split from the rest of the tree.

The following summarizing methods are used:

- **Top score tree:** To obtain a top score tree, an overall resampling support score is calculated for each resampling tree by multiplying the frequencies of all the branches. The overall resampling support can be seen as a global reliability score of the tree. The tree with the highest overall reliability score is returned as the top score tree. This method produces the most reliable tree that is a true solution for the chosen clustering method, and can be interpreted as such. However, the fact that the entire tree has the highest reliability score does not mean that every branch has a high reliability score too. In order to obtain a tree with only reliable branches, a consensus tree needs to be calculated (see below).
- **Consensus approach:** In this approach, all branches found in the resampled trees are considered. However, a reliability threshold is applied to exclude branches with low frequency. The reliability threshold is entered as a frequency percentage. Although any value between 0 and 100% can be entered, three values are of particular interest. A threshold value of 0% results in an *include-all summary*, i.e., a consensus that includes all possible solutions. A threshold value of 100% leads to a *strict summary*, containing only those branches that occur in all resampling trees. A threshold value of 50% results in a *majority summary*, where each branch occurs in at least 50% of the resampling trees. This is the most useful option for general purposes.

16.1.7 Trees and networks

For a good understanding of the paragraphs that follow, it is important to realize that two, fundamentally different, types of networks exist:

- Separation type networks
- Connection type networks

A network is of *separation type* when the branches in the network model *separation* between two clusters of nodes in the network, corresponding to input entries. Examples of separation networks are rooted trees, NeighborNets and all types of Neighbor-Joining trees. Leaving out a branch in a separation tree (or a set of branches in a separation network) causes the network to fall apart in two components. The branch that has been left out, is then interpreted as a *bipartition* of the set of input entries, putting all input entries of one component in the same group.

A network is of *connection type* when the branches in the network model *connections* between the nodes in the network. All nodes in a connection type network are entry nodes. A typical example of a connection type network is a minimum spanning tree: two nodes in the tree are connected when the distance between

these nodes is appropriately small. Therefore, a branch in a minimum spanning tree models the closeness of its source and target node. Other examples of connection type networks are all kinds of networks that try to eliminate partial correlation between nodes.

A consensus summary network is not necessarily a tree. In case of a minimum spanning tree, which is a *connection type* tree, a consensus network simply connects all branches that fulfill the threshold criteria. If an *include-all summary* was calculated (0% threshold), the result will be one single network containing a number of nodes that have multiple connections, representing the degeneracies. If a *strict summary* was calculated (100% threshold), a number of separate small networks may occur. Any other threshold value is likely to produce a combination of both, i.e. one or more networks of which some nodes may contain multiple connections (see Figure 16.1.3).

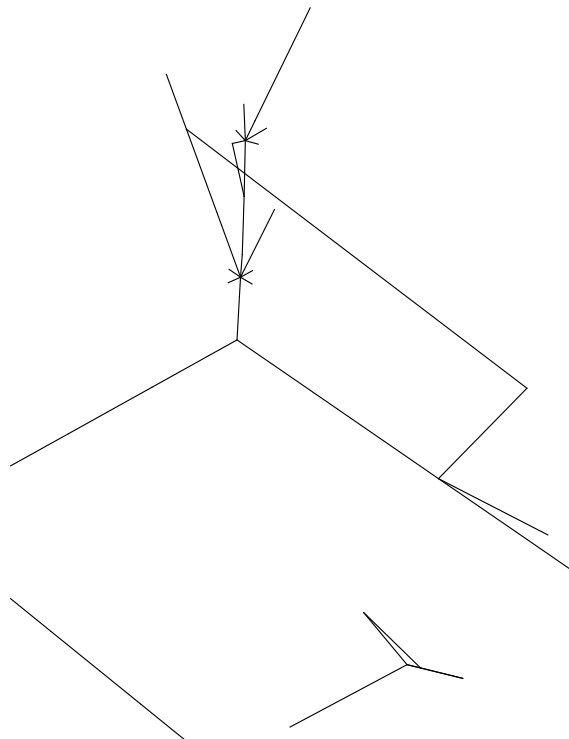


Figure 16.1.3: Minimum spanning tree with a resampling threshold of 50%.

For *separation type* trees (all global closest pair methods, neighbor joining methods and maximum parsimony/likelihood methods) a so-called *planar split network* is produced. In case an *include-all summary* was calculated (no threshold) or using a low threshold value, many branches may be represented as parallelograms (see Figure 16.1.4).

The meaning of the parallelograms is easily understood by looking at the steps in Figure 16.1.5 that lead to the *consensus planar network* (bottom). Every parallelogram in the consensus planar network depicts two alternative solutions with a higher frequency than the threshold: either one of the two pairs of parallel lines can be taken together to form an alternative solution. Note that a planar network is a simplification of a multidimensional network, which implies that in case of highly degenerate solutions it will not always be capable of representing all degeneracies in a consensus graph. Obviously, this is often the case when using settings that include a lot of degeneracies (e.g. *include-all summary* or a low threshold). In Figure 16.1.4, the parallelogram branching (e.g. as found in the purple entries) indicates that multiple solutions occur with more than 10% frequency within that group. Conversely, the star-like branching (e.g. as seen in the green entries) indicate that no solutions were found with more than 10% frequency. These degenerate solutions are not displayed because they did not fulfill the threshold criterion. Instead, they are left unresolved by taking them together.

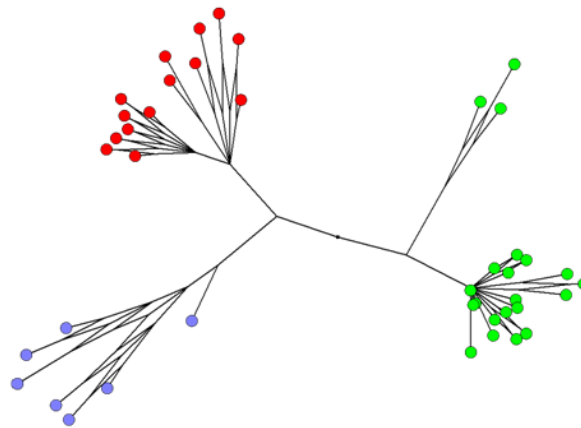


Figure 16.1.4: Planar consensus network with a resampling threshold of 10%.

Interestingly, since a *majority consensus* only considers splits that occur in at least 50% of the cases, the alternative solution to a split with more than 50% frequency always has a frequency below 50% and is therefore never retained. Consequently, a majority consensus will always be a *tree*, i.e. it will never display parallelograms as in a planar network.

Planar networks are often difficult to evaluate. Only in case of small and simple clusterings, the parallelogram representation may offer some added value. However, with large clusterings that have numerous solutions, the nets of parallelograms only confuse the picture and do not summarize the most important groups in an easily interpretable graph. BioNumerics therefore offers an alternative solution whereby the consensus graph is represented as a tree. This is also schematically represented in Figure 16.1.5 (*consensus tree*). The alternative solutions are represented in one branch for which the average distance is used. Interestingly, for separation type networks, an *include-all consensus* and a *strict consensus* are exactly the same in a tree representation. Indeed, in an include-all consensus all splits that occur in one single solution are taken into account, so with exception of the splits that occur in 100% of the solutions, all branches are averaged. In a strict consensus, by definition only the branches that occur in 100% of the solutions are shown, the remaining branches are averaged as well. Include-all and strict consensus trees exhibit large "flat clusters" due to the accumulation of averaged splits.

For most purposes, a tree representation will offer a more satisfactory summary of the relationships than a network. As mentioned before, a majority consensus is the most valuable compromise between summarization and information content. Figure 16.1.6 shows an example of a strict consensus tree exhibiting the typical flat branches.

Note that a consensus tree does not correspond to a particular solution for the clustering method used. Although a consensus tree is based on a straightforward and easily interpretable algorithm, this might be seen as an inconvenience. This disadvantage can be overcome by calculating a *top score tree* as described in 16.1.5. A top score tree is the solution for the clustering method that has the highest overall resampling support and can thus be considered as the most preferred tree among all solutions of that method in terms of overall reliability. However, a lot of information with regard to alternative branches with possibly higher support will remain undetected in a top score tree.

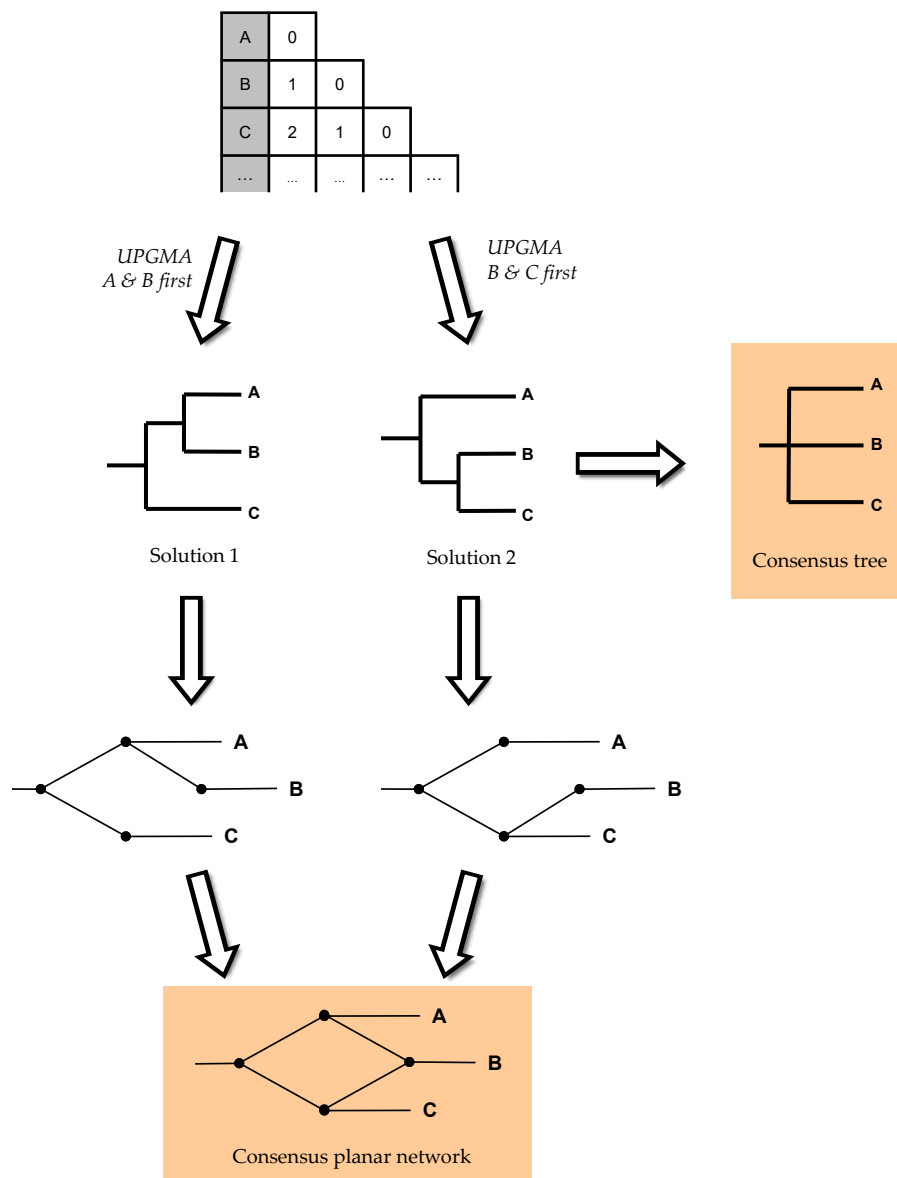


Figure 16.1.5: Schematic representation explaining the consensus representation of degeneracies in a planar network and a tree.

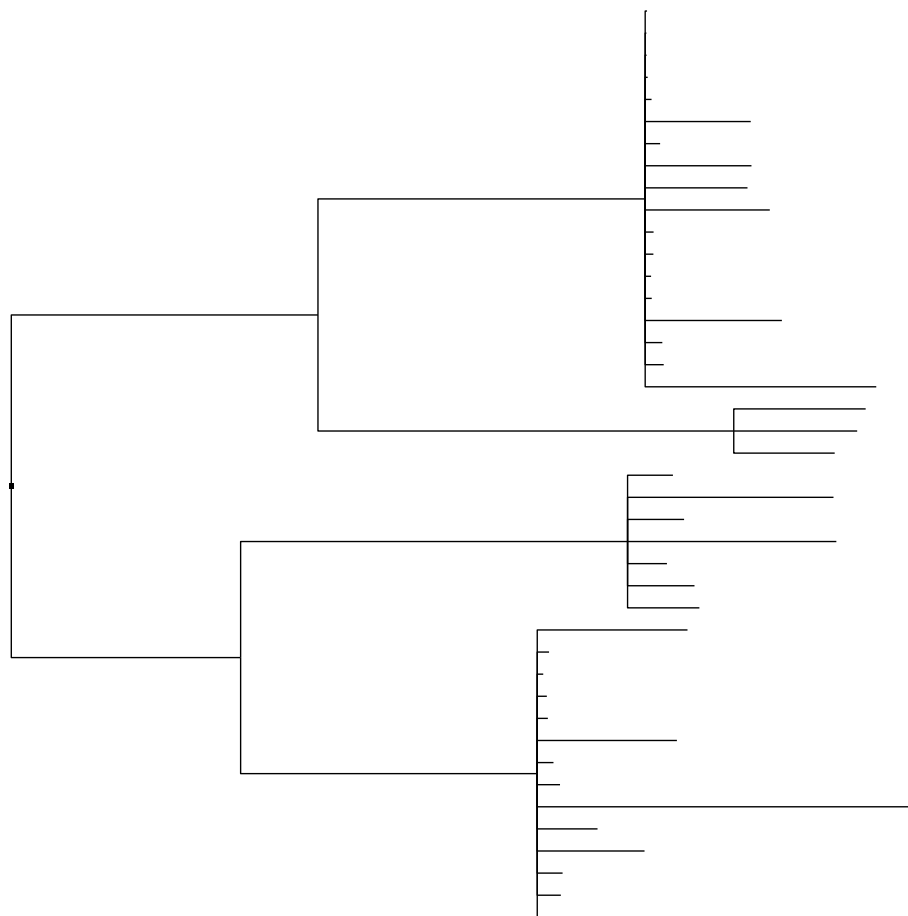


Figure 16.1.6: Strict consensus summary in a tree representation.

Chapter 16.2

The Advanced clustering wizard

16.2.1 Understanding the work flow of the advanced clustering wizard

The advanced clustering application is very comprehensive in terms of available clustering methods. As this can be combined with a variety of data input options, resampling strategies, and tree summarizing methods, it is obvious that the number of tree outputs that can be generated is enormous. However, not all combinations of choices make sense and some are even impossible. To make it possible for the user to build a clustering scheme gradually and with meaningful combinations of settings, the entire clustering setup is contained in a wizard. The basic successive choices correspond to steps in the wizard and are depicted in Figure 16.2.1.

Obviously, not all steps listed in Figure 16.2.1 are applicable to every combination of choices. For example, a UPGMA clustering has no additional parameters to be specified, so that the step "Define algorithm settings" will be skipped if UPGMA is chosen. Also, the last step, "Define summarizing strategy", is not applicable if no resampling is done, and will therefore be skipped in that case. In addition, within each step, one or more specific options or parameters may not be compatible with the (combination) of choices taken previously. Such options or parameters are then made inaccessible and appear as grayed items in the wizard.

Figure 16.2.2 shows a full diagram of all major clustering options. As can be seen from the figure, the wizard makes use of analysis templates, which contain all the settings for a specific analysis work flow. The software comes with a number of predefined templates to make commonly used clustering work flows easy to calculate. We will explain the use of templates in more detail later in this chapter. The blue figures depicting a tree or a network indicate that the wizard finishes at this point and starts calculating the result. For example, if a template is chosen, a tree or network is calculated immediately after step 1. Some options in the diagram are written in color. This indicates that further on in the diagram, another option can only be applied if it has the same color. For example, **Similarity** in step 2 is written in red, and **Error resampling** in step 5 is also written in red. This means that **Error resampling** can only be done when **Similarity** was chosen. Likewise, **Bootstrap resampling** in step 5 (green), can only be chosen if **Characters** was selected in step 2 (also green).

16.2.2 Steps in the advanced clustering wizard

16.2.2.1 Basic analysis settings and analysis template

Under *Analysis name*, you can enter a name for the analysis, which is used to save this clustering along with the comparison. The name of this clustering will appear in the *Analyses* panel of the *Comparison* window, so that the clustering can be opened again at any time by double-clicking on the name.

Under *Experiment*, you can choose the experiment type (aspect) to cluster. By default, the experiment type

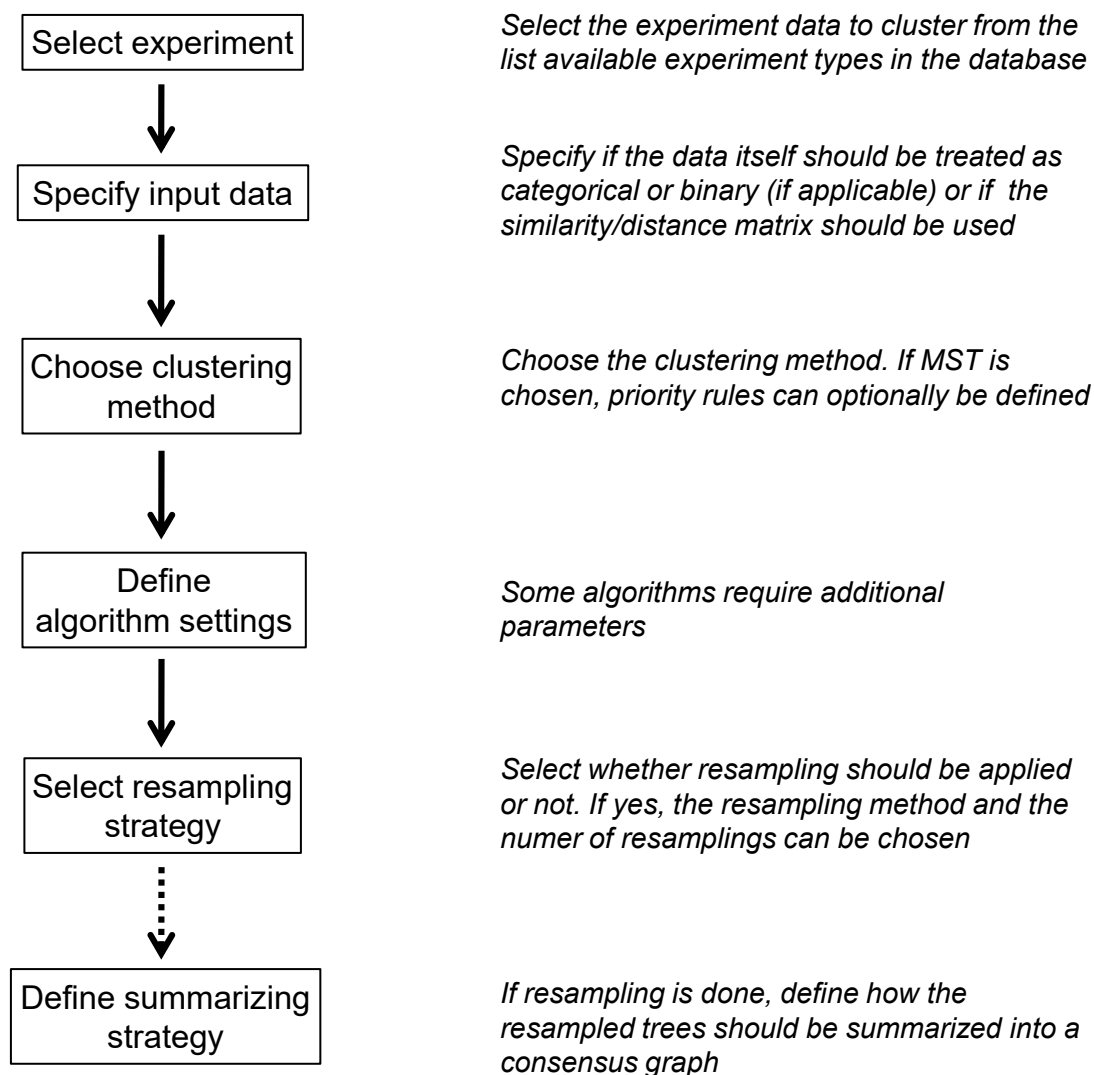


Figure 16.2.1: Consecutive steps in the clustering wizard.

selected in the *Comparison* window is taken. It is also possible to change the experiment settings. The *<Experiment settings>* button will open the corresponding experiment type window. In case of similarity-based clustering, this option has no use at this point, since the similarities have already been calculated. In case of binary, categorical or sequence data however, you might want to change some settings.

The *Analysis template* box allows you to choose one of the *Predefined templates*:

- **MST for categorical data** calculates a minimum spanning tree for categorical (multi-state) characters. It is obvious that this option is only meaningful in combination with categorical input data, not for RFLP fingerprint data as in this example. The template includes SLV and DLV priority rules.
- **Maximum parsimony tree** calculates a standard maximum parsimony tree from categorical data such as sequence data.
- **Maximum likelihood tree** calculates a standard maximum likelihood tree from sequence data.
- **Top score UPGMA** calculates the UPGMA tree with the highest resampling support from a similarity matrix and using permutation resampling (changing the order of entry input).

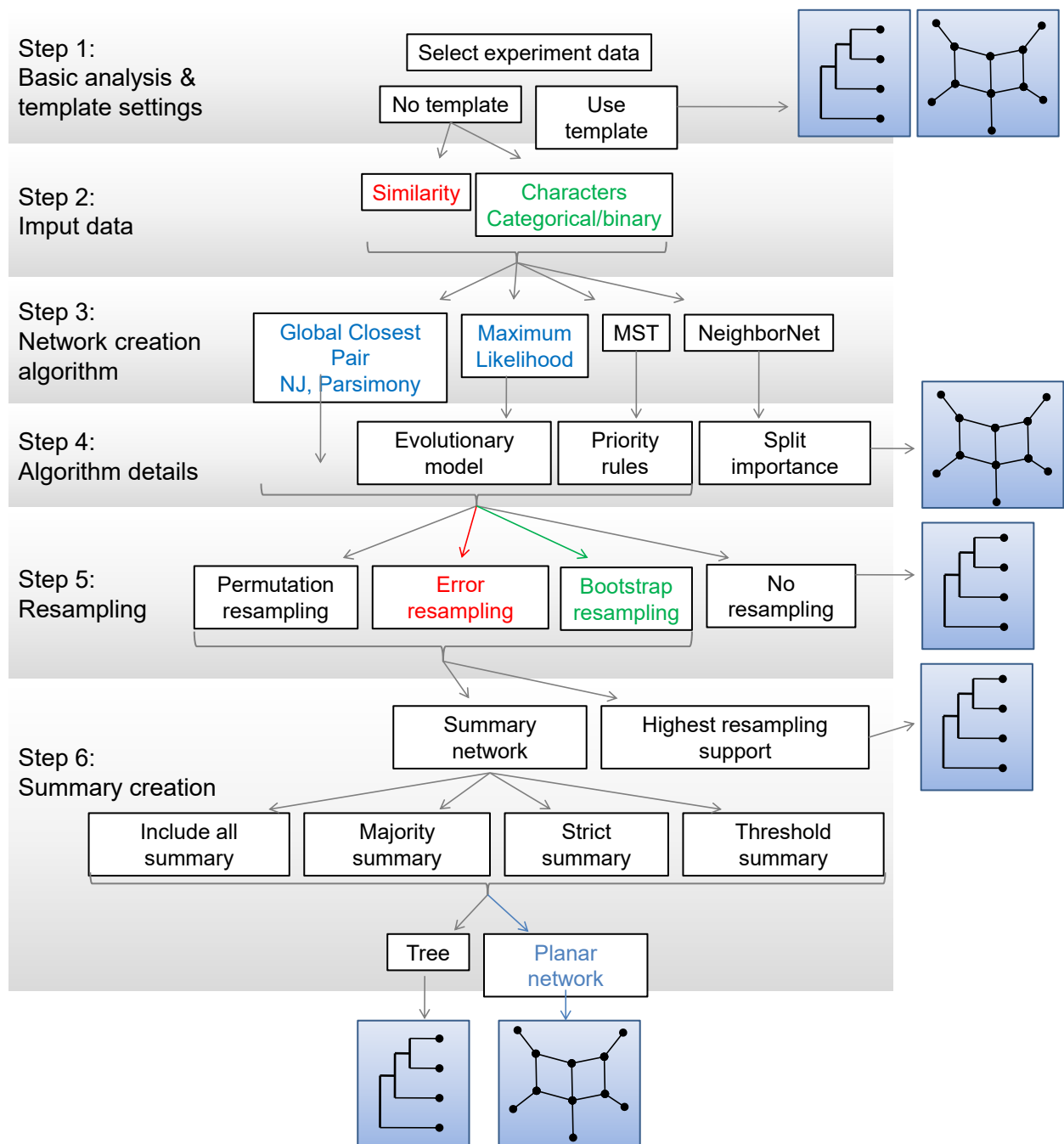


Figure 16.2.2: Detailed diagram of the six steps in the cluster analysis wizard. Options in color can only be applied in combination with an option in a previous step that has the same color. A blue tree/network figure means that a clustering is calculated at this point.

- **Majority UPGMA** calculates a majority summary tree based on UPGMA from a similarity matrix and using permutation resampling (changing the order of entry input).

In case resampling is applied, the templates always use a sampling size of 200.

When a template is chosen, the program skips all further steps in the wizard and starts the calculations as soon as the *<Next>* button is pressed. However, a check box **Modify template settings for new analysis** allows an existing template to be selected and modified by going through the successive steps of the wizard. Afterwards, the changed template can be saved as a new template or the original template can be overwritten.

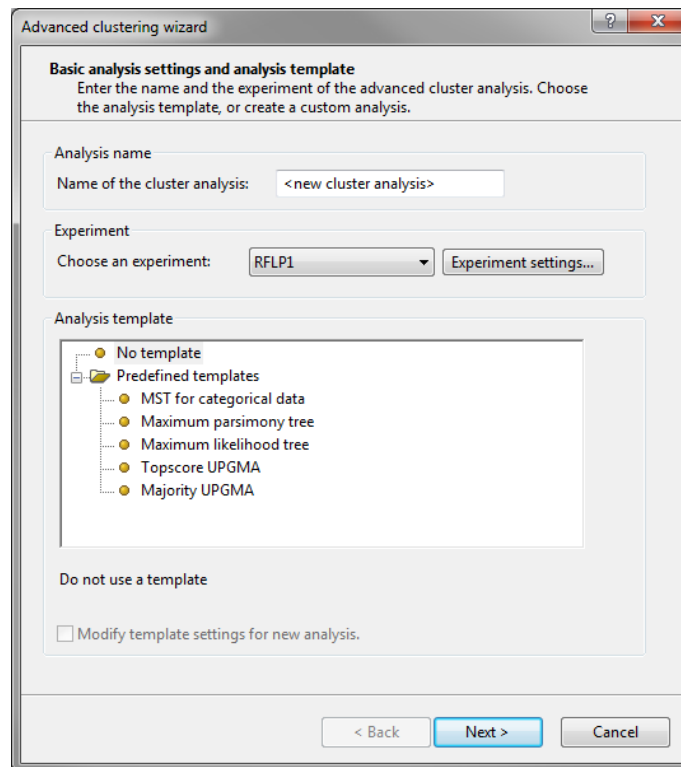


Figure 16.2.3: Basic analysis settings and analysis template.

16.2.2.2 Input data and treatment

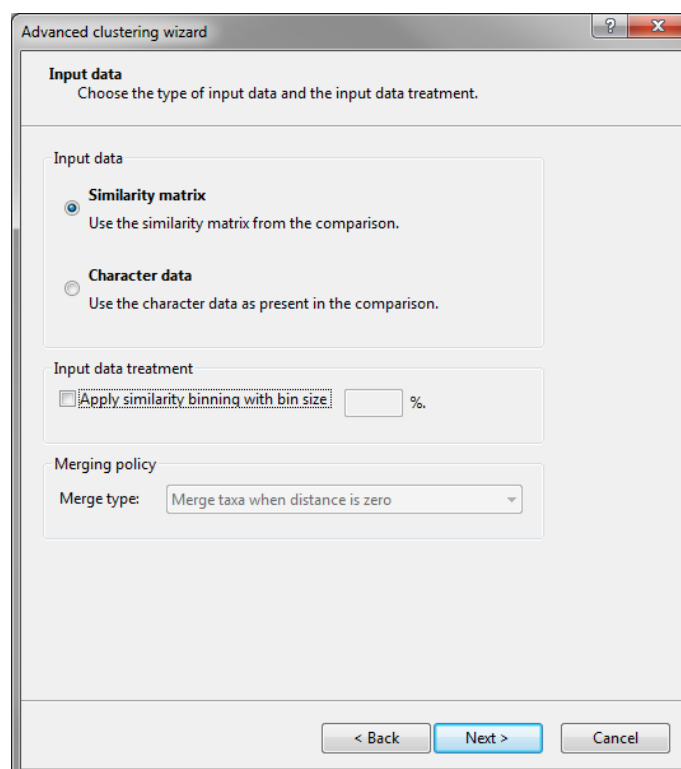


Figure 16.2.4: Choose type of input data and treatment.

In the second step, the type of input data needs to be selected. The input can be the *Similarity matrix* or the

Character data of the selected experiment type.

In case **Similarity matrix** is checked, a binning can be performed on the similarity values so that all values in a specified bin size are considered identical (check **Apply similarity binning with bin size**).

In case **Character data** is checked, the characters can be treated as categorical data or binary data.

The **Merging policy** is only applicable for character based data. The default setting for **Merging policy** is to **Merge taxa when distance is zero**. Using this setting, identical entries are displayed in one node without segmentation.

16.2.2.3 Network creation algorithm

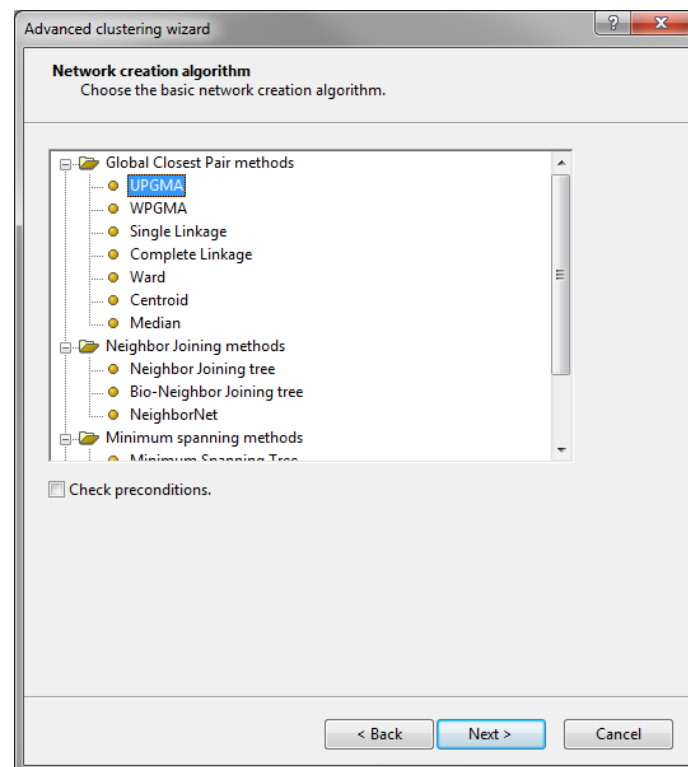


Figure 16.2.5: Choose network creation algorithm.

The clustering methods are subdivided in five principal groups. However, only those methods that can be applied to the selected type of data are displayed. For instance, if the **Similarity matrix** option was selected as input data in the previous step, the parsimony and maximum likelihood methods are not shown, since the data is not of character type. This is an overview of the available methods:

- **Global closest pair methods:** unweighted pair group method with arithmetic mean (**UPGMA**), weighted pair group method with arithmetic mean (**WPGMA**), **Single linkage** method (also called Nearest Neighbor), **Complete linkage** method (also called Furthest Neighbor), **Ward's** method, **Centroid** method, and **Median** method.
- **Neighbor-Joining methods:** Classical **Neighbor Joining tree**, **Bio-Neighbor Joining**, and **Neighbor-Net**.
- **Minimum spanning methods:** Minimum spanning tree algorithm and Prim's MST algorithm, with (**Minimum Spanning Tree with hypothetical nodes**) or without hypothetical nodes (**Minimum Spanning Tree**).

- **Maximum parsimony methods:** *Basic maximum parsimony tree*, *Optimized maximum parsimony (Simulated Annealing)*, and *Recombined maximum parsimony (Quartet Puzzling)*.
- **Maximum likelihood methods:** *Basic maximum likelihood*, *Optimized maximum likelihood (Simulated Annealing)*, and *Recombined maximum parsimony (Quartet Puzzling)*.
- **Correlation methods:** *Correlation eliminator* and *Partial correlation eliminator*.

The option **Check preconditions** makes it possible to check the validity of the distance matrix in function of the chosen algorithm. For **Global closest pair methods**, the matrix is checked for *ultrametricity*, for **Neighbor Joining methods** it is checked for *additivity*, and for **Minimum spanning methods** it is checked for *triangle inequality*. If the preconditions are not met, the clustering can be performed as well, but the interpretation of the result will be mathematically less elegant, and sometimes needs to be done with more care (e.g. MST without triangle inequality).

16.2.2.4 Hypothetical nodes

Figure 16.2.6: Specify the rules to create hypothetical nodes.

When creating a minimum spanning tree with hypothetical nodes, you can specify that a hypothetical type should only be created when:

The total network length is decreased with at least (default 1): Only in the case the introduction of a hypothetical type decreases the total spanning of the tree with default one change, the hypothetical type will be accepted.

If there are at least (default 1) neighbors at distance at most: The algorithm will only accept hypothetical types that have at least default one neighbor that has no more than default 1 change.

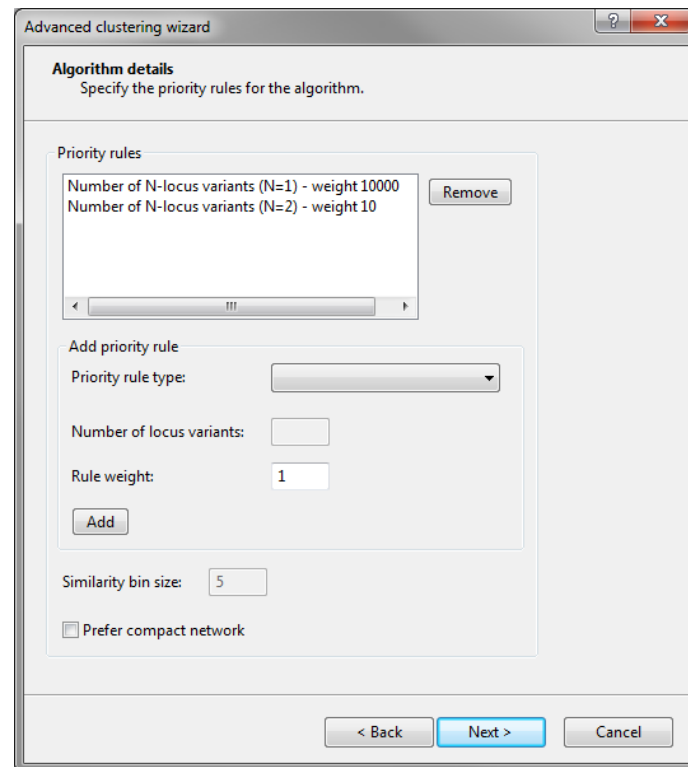


Figure 16.2.7: Specify priority rules.

16.2.2.5 Priority rules

The minimum spanning tree method usually provides many equivalent solutions for the same problem, i.e. one data set can be clustered in to many MSTs with a different topology but with the same total distance. Therefore, a number of priority rules, with respect to the linkage of types in a tree, have been adopted from the BURST program to reduce the number of possible trees to those that have the most probable evolutionary interpretation.

One or more rules can be added (<Add>) or removed (<Remove>) from the list.

Following rules can be selected from the *Priority rule type* list:

Number of N-locus variants: N has to be chosen by the user (*Number of locus variants*). For example, if N is 1, this means that, in case two types having an equal distance to a linkage position in the tree, the type that has the highest number of single locus variants (i.e. other types that differ only in one state or character) will be linked first.

Minimum average distance: The type that has the lowest average distance with the other types will be linked first, in case of equivalent solutions.

Maximum state frequency: The program calculates a frequency table for each state of each character. Types are ranked based upon the product of frequencies of their characters. In case of equivalent possibilities, types that have the highest state frequency rank are linked first.

The check box **Prefer compact network** is a setting that corresponds to an implicit priority rule in the standard Prim's algorithm, namely to compact the network as much as possible. By unchecking this option, a modified algorithm is used which does not apply this implicit priority rule.

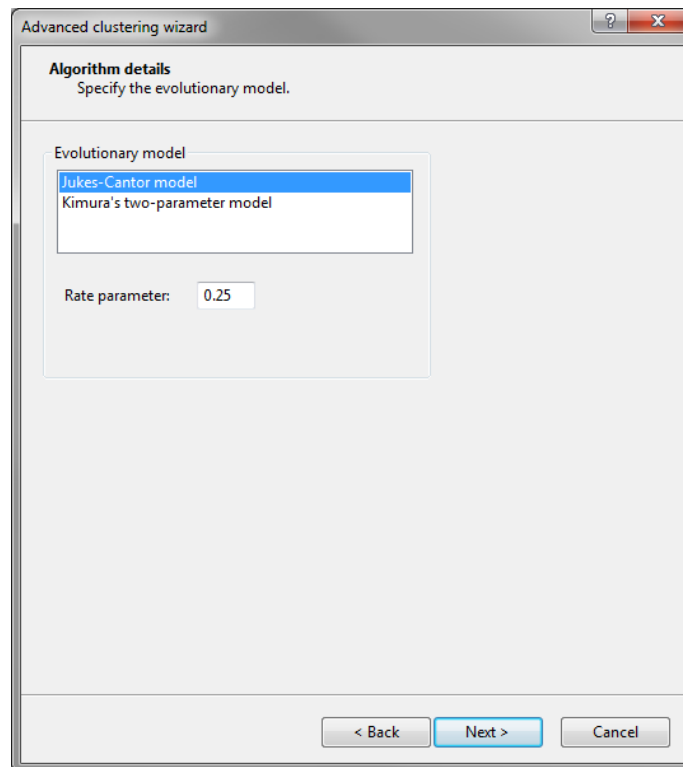


Figure 16.2.8: Specify evolutionary model.

16.2.2.6 Evolutionary model

As an evolutionary model, one can select the *Jukes and Cantor* correction [21], a *one parameter* correction for the evolutionary distance as calculated from the number of nucleotide substitutions.

Alternatively, the *Kimura 2 parameter* correction [22] can be selected.

In either case, the resulting tree displays a distance scale which is proportional to an evolutionary time, rather than a similarity scale.

16.2.2.7 Split importance

In case of a NeighborNet, splits can be excluded based on their importance, to reduce the complexity of the tree.

16.2.2.8 Resampling procedure

This step deals with the resampling strategy.

No resampling performs a cluster analysis without any resampling method, which leads to just one solution without an indication for support on the branches.

Permutation resampling uses permutations of entry input order to obtain different solutions.

Error resampling adds random error to the similarity values within specified boundaries to obtain different solutions.

Bootstrap resampling uses bootstrap of input character data to obtain different solutions.

Depending on whether the input data are similarities or characters, either bootstrap resampling or error

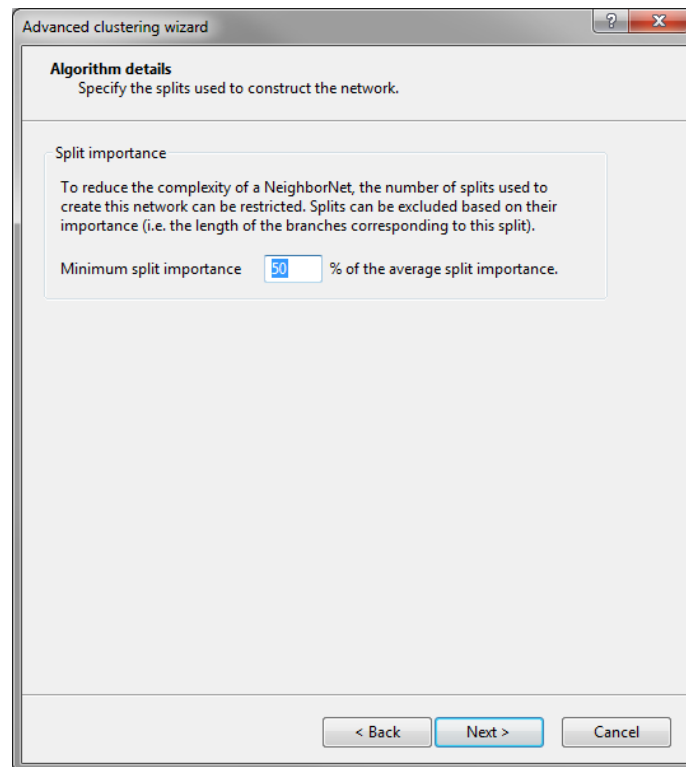


Figure 16.2.9: Specify the splits used to construct the network.

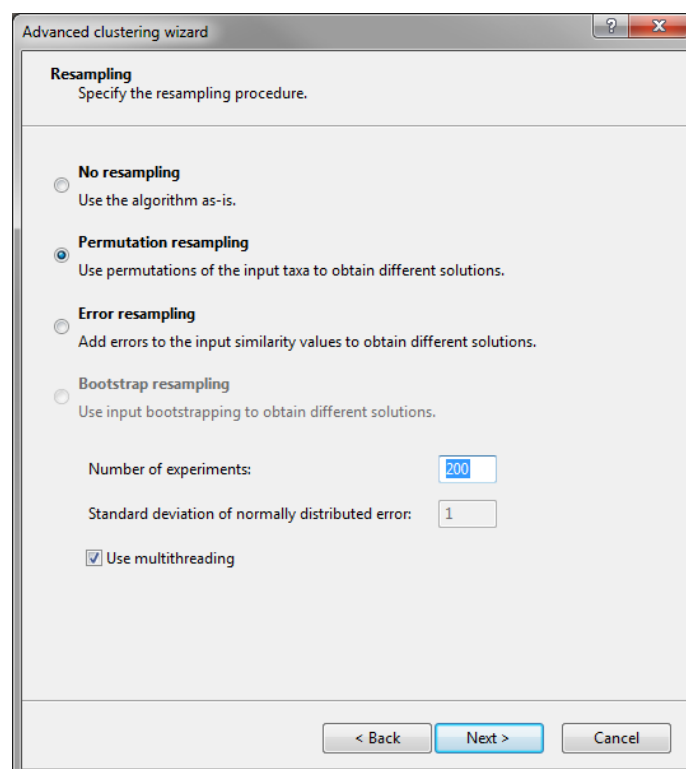


Figure 16.2.10: Specify the resampling procedure.

resampling is disabled.

For all resampling methods, the *Number of experiments* can be chosen. However, the default value of

200 resampling experiments, is more than satisfactory to obtain stable and accurate resampling support information.

With **Error resampling** enabled, you can enter the *Standard deviation of normally distributed error*. The program assumes that the error on the similarity values is normally distributed and uses the standard deviation of that error as a basis for generating resampled similarity matrices.

The check box **Use multi-threading** allows a multi-core CPU to use all cores simultaneously.

16.2.2.9 Summary creation

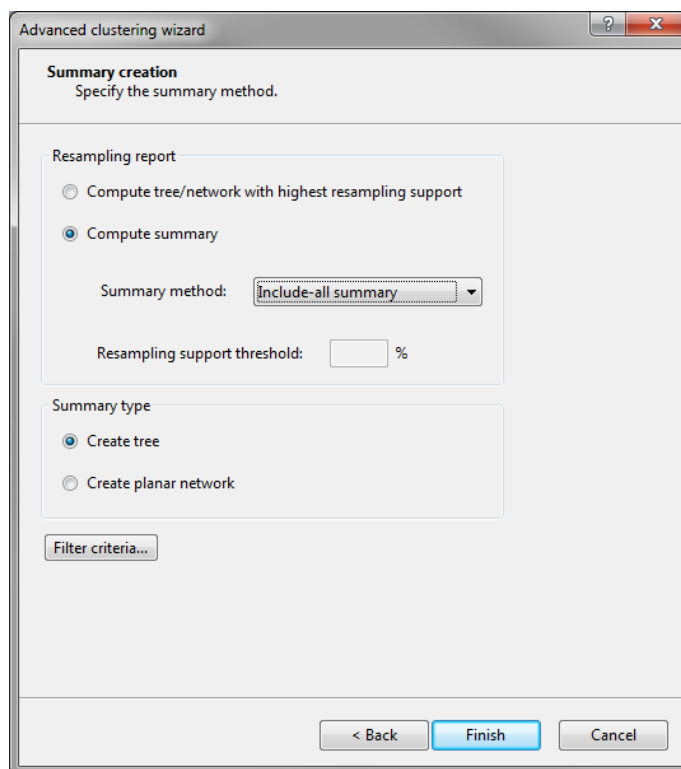


Figure 16.2.11: Specify the summary method.

This step in the clustering wizard allows the method for tree summarizing to be specified. Obviously, this step only applies if a resampling method is selected; if **No resampling** was chosen in the previous step, this step is skipped by the wizard.

The option **Compute network with highest resampling support** calculates a *top score tree* as defined in 16.1.5. This is one of the tree solutions from the resampling procedure that has the highest overall resampling support. The advantage is that this tree is interpretable as an outcome of the chosen clustering algorithm.

The option **Compute summary network** corresponds to the *consensus tree* as explained in 16.1.5. A consensus summary can either be an **Include-all summary**, a **Majority summary** or a **Strict summary**, having respectively 0%, 50% and 100% resampling support threshold (see 16.1.5). The user can also specify a different threshold by selecting **Threshold summary**. In that case, a percentage value can be entered in the **Resampling support threshold** box.

Under **Summary type**, you can specify to calculate a tree (**Create tree**) or a planar network (**Create planar network**) (see 16.1.6 for an explanation).

The advanced option **<Filter criteria>** allows the user to create a summary only from resampling solutions that fulfill certain criteria (Figure 16.2.12).

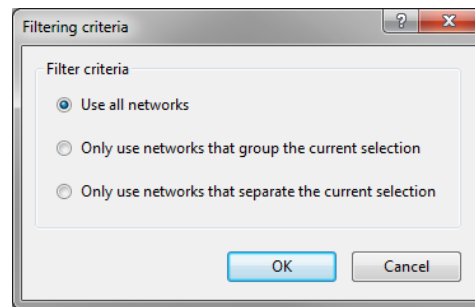



Figure 16.2.12: *Filtering criteria dialog box in the summary creation step for separation type networks.*

Besides *Use all networks* (no filtering), it is possible to summarize only those networks that have the current selection of entries grouped together in one cluster (*Only use networks that group the current selection*). Conversely, you can also specify to summarize only those networks that have the current selection of entries in different clusters (*Only use networks that separate the current selection*).


Applying filtering criteria can be useful, for example, to check the resampling support of specific clusters of entries that do not appear in a specific summary. However, this advanced feature should be used with care and one should pay attention that:

1. The entries to be grouped or separated are selected in the database using *Edit > Select all entries in selected nodes* or .
2. It is possible that no network satisfies the filtering criteria. In that case, no network is produced.
3. The results of the filtering are displayed under *Resampling strategy* in the *Cluster analysis method panel*. You can evaluate the occurrence percentage of a cluster configuration from this report, e.g. "Filtering has been applied. 200 networks have been created, of which 62 were used".


16.2.3 An example

The *Advanced cluster analysis* window is launched from the *Comparison* window. To illustrate the different resampling options, we will use the experiment data available in **DemoBase Connected**. The **DemoBase Connected** can be downloaded directly from the *BioNumerics Startup* window or restored from the back-up file available on our website:

- To download the database directly from the *BioNumerics Startup* window, click the **Download example databases** link, located in the lower right corner of the *BioNumerics Startup* window. Select **DemoBase Connected** from the list and select *Database > Download*.
- To restore the database from the back-up file, first download the file `DemoBase_Connected.bnbk` from <http://www.applied-maths.com/download/sample-data>, under 'DemoBase Connected'.

In the *BioNumerics Startup* window, press the  button, select **Restore database**, browse for the downloaded file and select **Create copy**. Specify a name and click **<OK>**.

3.1 Select all entries in the *Main* window except those labeled as STANDARD in the 'Genus' field.

3.2 Highlight the *Comparisons* panel in the *Main* window and select *Edit > Create new object...* () to create a new comparison for the selected entries.


3.3 In the *Comparison* window, right-click in the header of the "Genus" field and select **Create groups from database field** from the menu. Alternatively select **Groups** > **Create groups from database field**.

3.4 Press <Yes> to create three groups according to the genus names.

As an example we will cluster the **RFLP1** curves by means of a similarity matrix.

3.5 In the *Experiments* panel, select **RFLP1** and then perform a cluster analysis using the basic method: choose **Clustering** > **Calculate** > **Cluster analysis (similarity matrix)...** (see 4.2), select **Dice** as coefficient and specify 0.5% **Optimization** and 0.5% **Tolerance**, in the final step select **UPGMA** to obtain a clustering.

When the calculation is finished, the similarity matrix is displayed in the *Similarities* panel.

3.6 Select **Clustering** > **Calculate** > **Advanced cluster analysis...** or press the  button and select **Advanced cluster analysis** to launch the *Create network* wizard (see Figure 16.2.3).

To start, we will choose the template **Top score UPGMA**, which will calculate the UPGMA tree with the highest overall resampling support. The resulting tree can be considered as the UPGMA tree to be preferred among all equivalent solutions in terms of overall reliability.

3.7 Name sure **RFLP1** is selected, select **Top score UPGMA**, specify an analysis name (for example **RFLP1 Top score UPGMA**) and press <Next>.

The program starts calculating 200 resampled UPGMA trees and the tree with the highest overall branch support is produced (see Figure 16.3.1).

Chapter 16.3

The Advanced cluster analysis window

16.3.1 Introduction

The *Advanced cluster analysis* window is composed of seven panels: the *Network panel* on the left, which contains the tree or network object, the *Entry list panel* (upper right) containing the list of database entries the clustering is created from, and the *Cluster analysis method panel* which describes in detail the settings used to obtain the clustering. Behind the *Entry list panel*, two more panels are available as tabs: the *Selection entry list panel*, displaying the entries currently selected on the tree or network and the *Entry data panel*, displaying the character data for the selected entries. In case of similarity-based clustering, this panel remains empty. Behind the *Cluster analysis method panel*, there is a *Branch properties panel* and a *Statistics panel* available as tabs (see further).

Obviously, the positions of these panels are only valid for the default configuration, which can always be restored using **Window > Restore default configuration**.


The terminology used in the *Advanced cluster analysis* window is stricter than in the basic clustering application (see Figure 16.3.2):

- A *node* is every position on the tree or network where a (bi)furcation occurs. Nodes can be *internal nodes* or *leafs* (end-nodes).
- A *branch* is the line that connects two nodes (can also be a *tip-branch* when connecting a leaf).
- In a tree, a *cluster* can be seen as the set of all branches and nodes at one side of a branch.
- In a network, a *cluster* can be seen as the set of all branches and nodes at one side of a series of parallel branches.

In a rooted tree, only the horizontal part of a branch counts for the branch length.

16.3.2 Display settings

After calculation, the network is zoomed automatically to fit the size of the *Network panel*. For a rooted tree, this means that the width of the tree, including the entry labels fits the whole width of the panel. Vertically, the tree may extend beyond the panel so that you need to use the scroll bar. For unrooted trees and networks, the initial size is such that the entire network fits on the panel.

Zooming in or out on the network can be done using the  zoom slider at the left hand side of the *Network panel*.

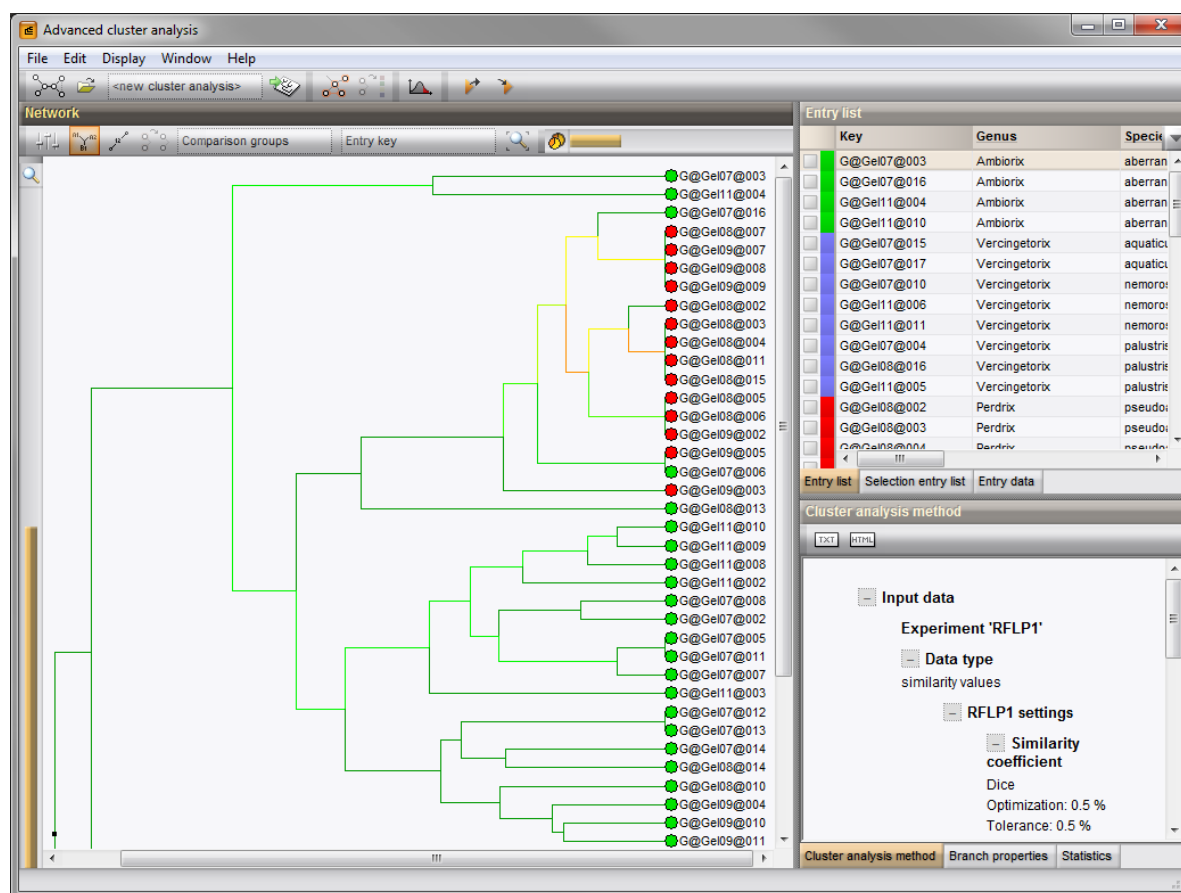


Figure 16.3.1: The *Advanced cluster analysis* window with *Top score UPGMA* calculated using the predefined template.

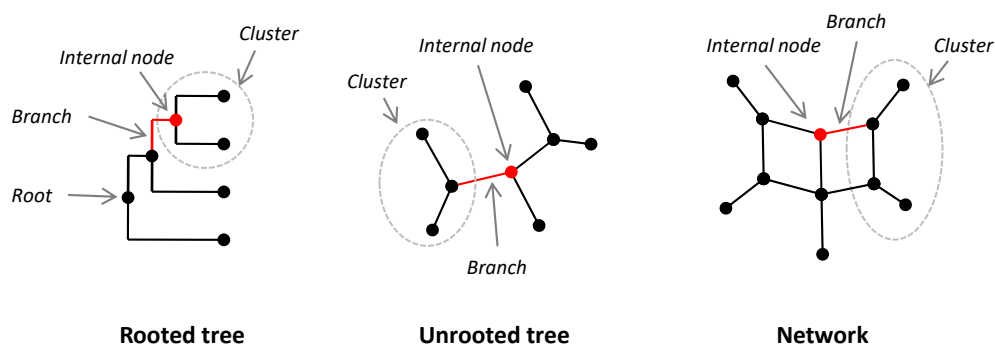




Figure 16.3.2: Illustration of branches, nodes and clusters in rooted trees, unrooted trees and networks.



After a zooming action, the initial fit can be restored by choosing *Display > Zoom to fit* or using the  button in the toolbar of the *Network* panel.

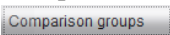
The node size can be increased and decreased using the  zoom slider.


Selecting *Display > Adjust node sizes* will rescale the nodes, so they will fit the *Network* panel.

By default, the *Advanced cluster analysis* window displays the network using specific layout options, which may depend on the template chosen.


Using the **Top score UPGMA** template for example, the branches are displayed in color according to the resampling support. The color ranges from red (very poor resampling support; unreliable branch) over orange, yellow, light green to dark green (excellent resampling support; very reliable branch).

By default, the entry keys are displayed as node labels. The labels can be changed using **Display > Select node labels** or with the right most drop-down menu ( in the toolbar of the *Network panel*. The display of the node labels can be switched on or off using **Display > Show node labels** or the  button. By default, the node labels are shown next to the nodes. To display the labels on top of the nodes choose **Display > Show node labels on top of the nodes**. To switch back to the default view, select **Display > Show node labels next to the nodes**. With **Display > Show node labels for selected nodes only** only the labels for the selected nodes will be retained.

If comparison groups were created in the *Comparison* window their colors are by default displayed on the dendrogram leafs. If *field states* (see 3.3.6) were defined for one or more information fields, these would also appear as possible options to color the dendrogram leafs in the **Display > Select node colors** menu and in the  drop-down menu. With **Display > Show color legend** information about the node colors is displayed next to the network image.

By default, no branch information is shown. To display labels with branch information, choose **Display > Show branch labels** or press the  button. With the branch labels enabled, the branch lengths are displayed by default.

Although the aforementioned functions and buttons provide an easy and quick tool to change the most basic layout settings, for full and advanced layout control, the *Display settings* dialog box should be used.

To open the *Display settings* dialog box, select **Display > Display settings** or press the  button (see Figure 16.3.3).

The *Display settings* dialog box allows you to customize numerous items, available as tabs in the dialog box.

16.3.2.1 Node labels and sizes

The *Node labels and sizes tab* allows the node labels to be displayed or not (**Show node labels**).

The labels can be chosen using the **Use label from** drop-down box.

A color can be specified for the labels using the **Use color from** drop-down box. You can choose the colors of the **Comparison groups** (if present) or any *Field state* or no color (**Nothing**).

In case multiple entries are contained in the same node, you can **Restrict number of labels to at most** a specific number (default 3).

In case statistics are calculated for the network, you can display the **cophenetic correlation**. In case of a non-resampled, rooted tree derived from a similarity matrix, you can display the **similarity/distance values** for internal nodes.

The **Minimum node size** can be entered as a percentage of the default node size whereas the **Maximum node increase** can be entered as a multiplication of the **Minimum node size**. These two parameters only make sense in case of unrooted trees or in case entries with zero distance are merged in a rooted tree (see further).

Finally, the option **Hide node disk** allows you to display the network without the node disks. In case you want to plot the *comparison groups* or *field states* as colors on the entries, these will not be shown if the node disks are hidden.

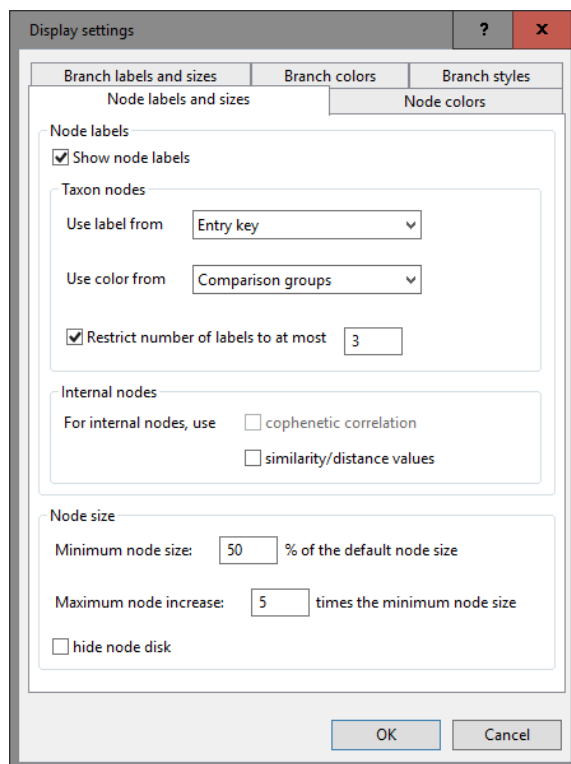


Figure 16.3.3: The *Display settings* dialog box with the *Node labels and sizes* tab.

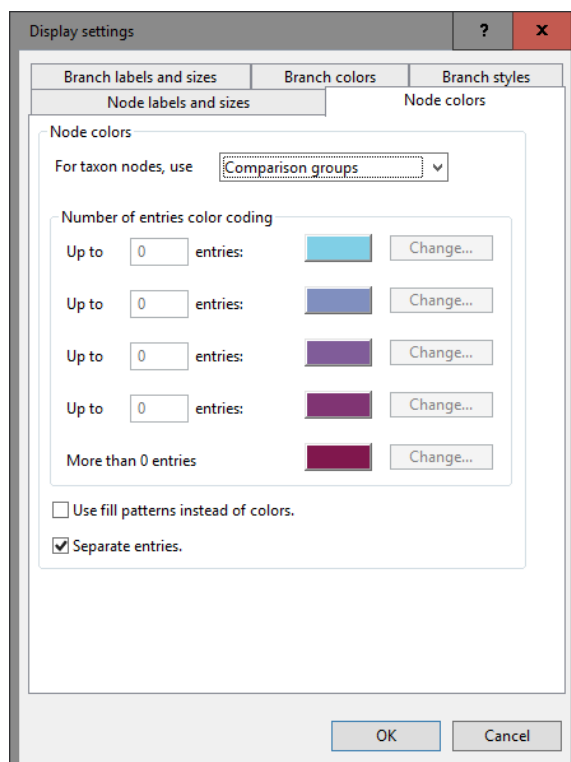


Figure 16.3.4: The *Display settings* dialog box with the *Node colors* tab shown.

16.3.2.2 Node colors

The *Node colors* tab allows colors to be displayed on the nodes according to various criteria.

With the *For taxon nodes use* option you can either display no color (*Nothing*), the *Number of entries*, the *Comparison groups* (if present), or any *Field state* which will appear if defined.

If *Number of entries* is selected, you can define your own color range and the number limits for each color. Alternatively, you can *Use fill patterns instead of colors*.

The option *Separate entries* allows segmented nodes consisting of multiple entries to be displayed as one single node when unchecked.

16.3.2.3 Branch labels and sizes

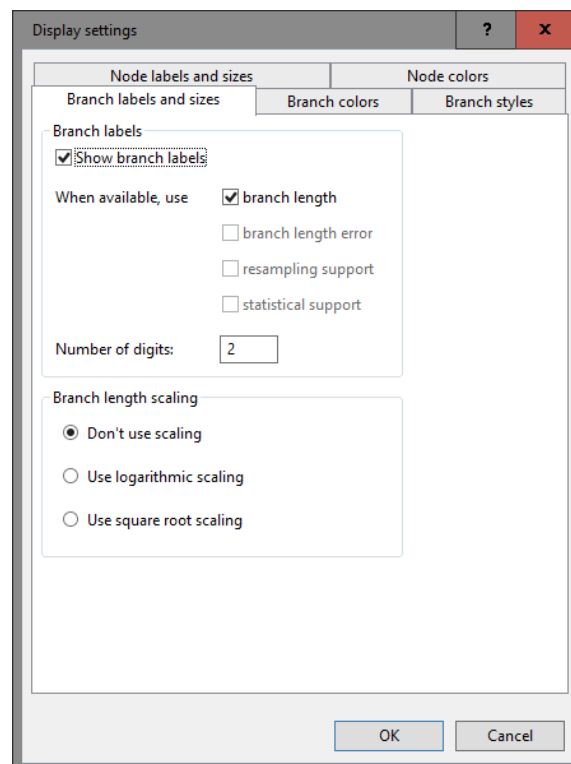


Figure 16.3.5: The *Display settings* dialog box with the *Branch labels and sizes* tab.

The *Branch labels and sizes* tab defines primarily what labels will be used for the network branches.

With *Show branch labels* you can show or hide the labels. If enabled, you can choose to display the *branch length*, the *branch length error* (if calculated), the *resampling support* (if calculated), and the *statistical support* (if calculated).

The accuracy of the label can be specified as *Number of digits*.

In case of clusterings with very small distances within the taxa and large distances between the taxa, it can be useful to apply a *Branch length scaling* by selecting *use logarithmic scaling* or *use square root scaling*. Default no scaling is applied (*Don't use scaling*).

16.3.2.4 Branch colors

From the *Branch colors* tab, branch colors can be either disabled (*nothing*), or can correspond to the *resampling support* (if calculated), *statistical support* (if calculated) or the *branch length*.

When *Use custom branch color coding* is checked, you can define your own color range and the limits for each color (percentage for resampling support or distance for branch length).

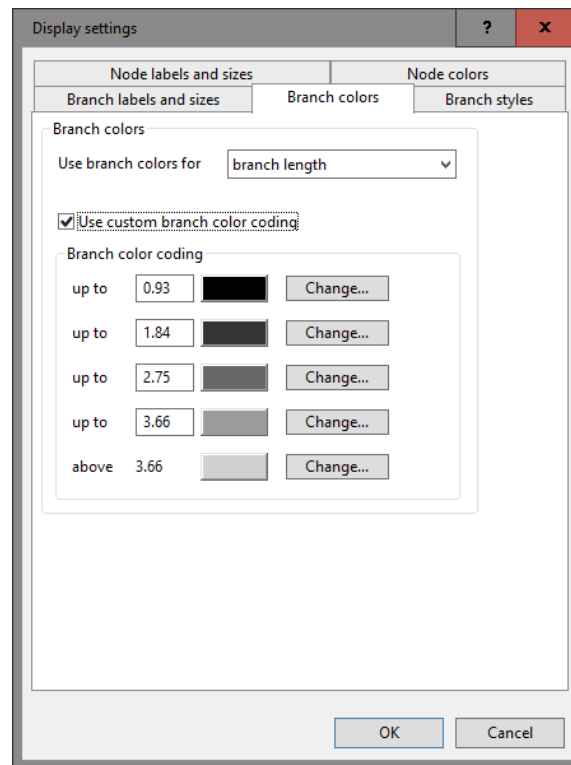


Figure 16.3.6: The *Display settings* dialog box, with the *Branch colors* tab.

16.3.2.5 Branch styles

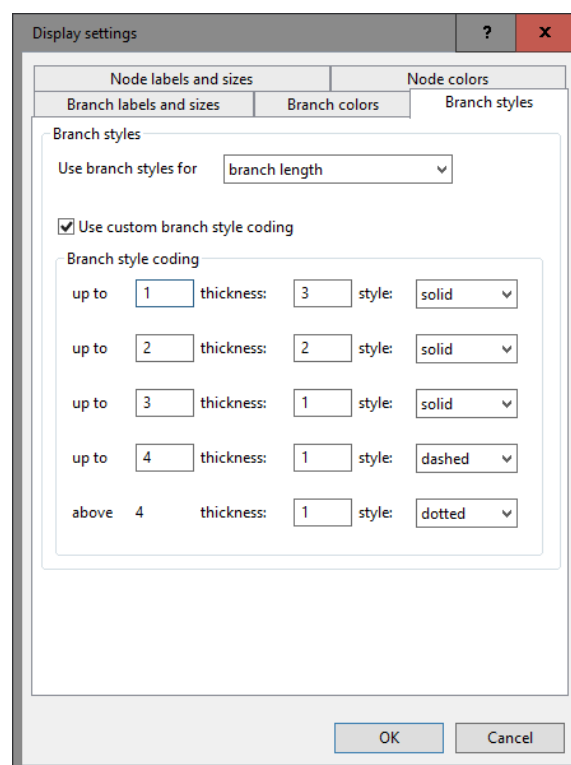


Figure 16.3.7: The *Display settings* dialog box with the *Branch styles* tab shown.

The *Branch styles* tab allows you to draw the network branches with different types of lines depending on

the *branch length*, the *resampling support*, or *statistical support*. The default choice is *nothing*.

With *Use custom branch style coding* checked, you can specify different line types and different line thicknesses for user-defined intervals (percentage for resampling support or distance for branch length).

16.3.3 Network editing functions

16.3.3.1 Selections on a network

As a basic edit function on a tree or network, you can select nodes and branches.


A branch can be selected by clicking somewhere on the branch line. This function can be used to obtain the properties of the selected branch in the *Branch properties panel*.


A node can be selected by simply clicking on the node. If the node is just one point (e.g. an internal node that does not contain entries), the node is selected as a yellow dot. If the node is a leaf containing one or more entries, the node is encircled in yellow. If the tree is rooted, the underlying cluster and all its leaf nodes are selected as well. In case of an unrooted tree, only the node is selected, as the program cannot know which is the "underlying" cluster.

Multiple nodes can be selected successively by holding down the **Shift** key and clicking on the nodes.

Groups of nodes and/or branches can also be selected by clicking and holding down the left mouse button and dragging the mouse pointer over the nodes to be selected.

With *Edit > Selection tools > Invert selection* the selection of nodes is inverted. Using the commands *Edit > Selection tools > Select singlets* and *Edit > Selection tools > Select doublets* the nodes containing one or two entries respectively are selected.

A selection of nodes can be exported to a selection of entries in BioNumerics using the function *Edit > Select all entries in selected nodes* or the  button.

Conversely, a selection of entries in BioNumerics can be imported as a selection of nodes using *Edit > Select nodes that contain selected entries* or the  button.

16.3.3.2 Creating an unrooted tree

Trees and networks can be displayed in different ways. For example, a rooted tree (e.g. UPGMA) can also be converted to an unrooted tree by removing the root. Conversely, an unrooted tree (e.g. Neighbor Joining) can also be converted to a rooted tree by appointing a root branch. The latter function is commonly used, particularly to make the interpretation of large unrooted trees easier. Only bifurcating unrooted trees can be rooted; MSTs, having star-like furcations, and true networks such as NeighborNet, cannot be rooted.

An example of an unrooted tree is displayed in Figure 16.3.8. Note that the option to place identical entries on a zero-distance branch as in a rooted tree is not possible in an unrooted tree. Therefore, identical entries are placed in one node, with individual entries displayed as exploded pie chart. Multi-entry nodes grow with the number of entries they contain and the size can be adjusted as explained in 16.3.2.

Note that the function to remove the root is not just a display function but is physically removing the root node from the tree. To restore a root position, specific criteria will need to be used (see below). If you want to display a tree with unrooted layout, without physically removing the root, we refer to the function *Show in weighted unrooted layout* (see Figure 16.3.12).

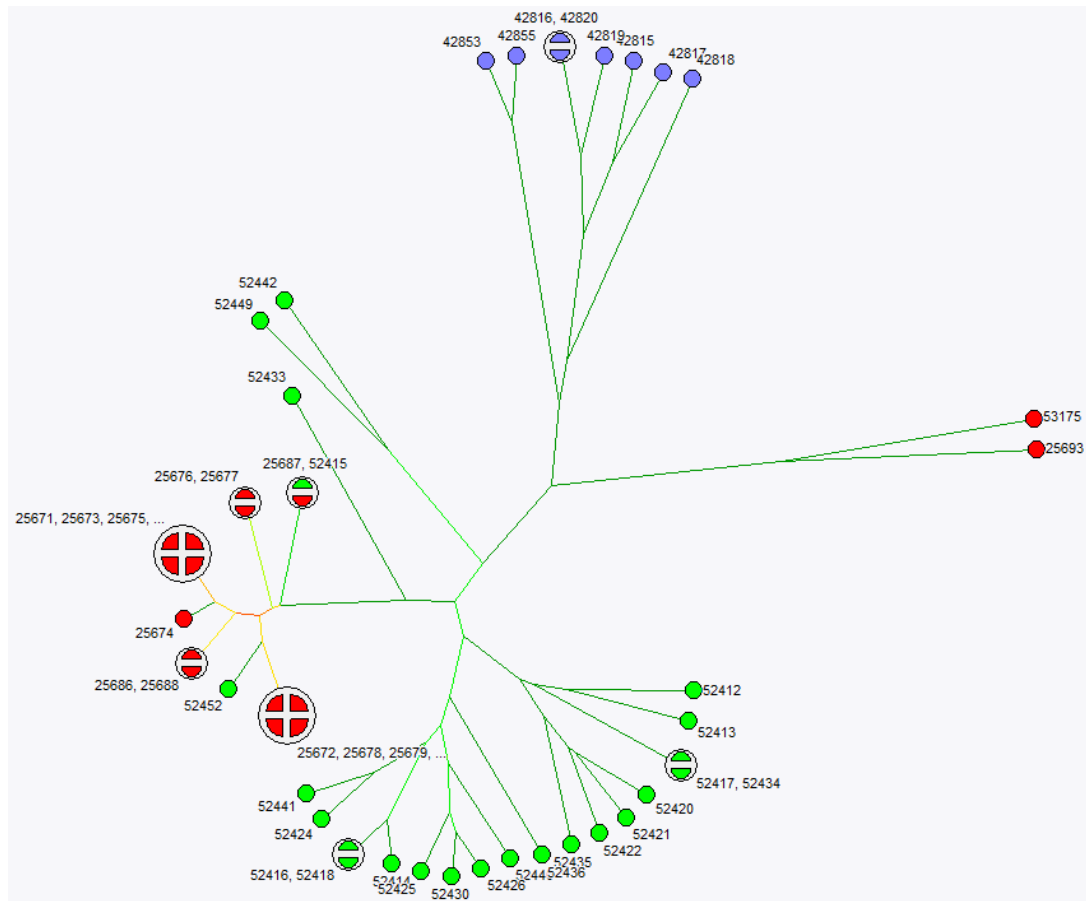


Figure 16.3.8: Same UPGMA tree as in Figure 16.3.1 after removing the root.

16.3.3.3 Creating a rooted tree

An unrooted tree can be rooted using *Edit > Determine root*.

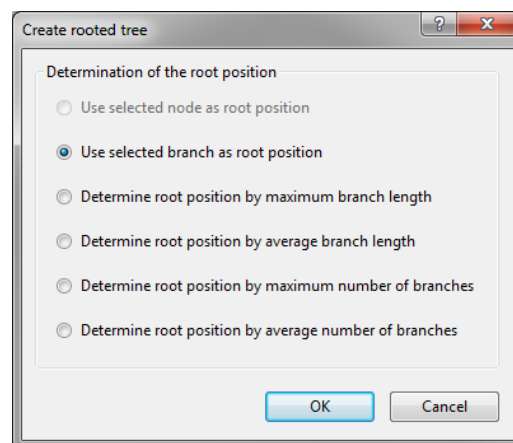


Figure 16.3.9: The *Create rooted tree* dialog box.

The *Create rooted tree* dialog box provides up to 6 possibilities to determine the root:

- **Use selected node as root position:** This option is only enabled if one single internal node is selected. However, with an internal node as root, three (or more) branches might depart from the root position, which is not favorable. Therefore, the next option is most commonly used.

- **Use selected branch as root position:** This option is only available if one single branch is selected. The selected branch will be split right in the middle and become two root branches.
- **Determine root position by maximum branch length:** The algorithm determines a point on a branch where the total length from that point to the most distant leaf node at one end is the same as the total length to the most distant leaf node at the other end.
- **Determine root position by average branch length:** The algorithm determines a point on a branch where the average length from that point to all leaf nodes at one end is the same as the average length to all leaf nodes at the other end.
- **Determine root position by maximum number of branches:** The algorithm determines a point on a branch where the total number of branches from that point to the furthest leaf node in terms of branch number at one end is the same as the total number of branches to the furthest leaf node at the other end.
- **Determine root position by average number of branches:** The algorithm determines a point on a branch where the average number of branches from that point to all leaf nodes at one end is the same as the average number of branches to all leaf nodes at the other end.

Starting from a rooted tree by nature (e.g. UPGMA), options 3 and 4 (maximum and average branch length) will create the same (original) root.

16.3.3.4 Grouping and ungrouping entries with zero distance

Identical entries in terms of character data or similarity/distance value usually occupy the same position on a tree or network and are thus displayed as zero-distance nodes. We have mentioned previously that zero-distance entries are displayed differently in rooted and unrooted trees. Whereas rooted trees can display identical entries as separate leaf nodes on a zero-distance branch, unrooted trees have to display zero-distance entries in the same node. To indicate the fact that a node contains multiple entries, the size of the node increases with the number of entries, a parameter that can be adjusted as explained in 16.3.2. In addition, each entry is displayed as an exploded pie chart within the node (Figure 16.3.10). This makes it possible to count the number of entries and to visualize the different comparison groups or field states contained in the group.

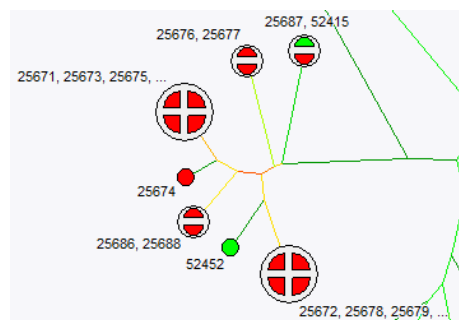


Figure 16.3.10: Multi-entry nodes in an unrooted tree.

For rooted trees, the program allows leaf node entries that have zero distance to be merged into one node using the function **Display > Group entries**.

The result looks as in Figure 16.3.11.

Grouped zero-distance entries can be restored as separate leaf nodes using **Display > Ungroup entries**. Obviously, this function only applies to rooted trees.

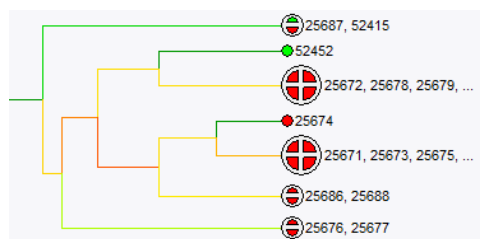


Figure 16.3.11: Grouped identical entries in a rooted tree.

For unrooted trees, segmented zero-distance entries can be unsegmented in the display settings: select **Display > Display settings**, click the *Node colors* tab and uncheck **Separate entries**.

Note that zero-distance entries in a network are not necessarily identical in terms of character data or similarity value. Zero distance on a tree can be a consequence of the clustering method used; for example, the Single Linkage method will group all entries in a zero-distance group that are identical to at least one other member of that group. Bootstrap resampling or similarity binning can also cause non-identical entries to appear as zero-distance leaf nodes.

As explained in 16.2.2.2, the software also allows entries having zero distance based upon their character data to be merged prior to clustering. The program then keeps the node size proportional to the number of entries contained, but the exploded pie charts are not shown.

16.3.3.5 Optimizing the network layout

To optimize the network layout, choose **Display > Layout**. This calls the *Change layout* dialog box (see Figure 16.3.12).

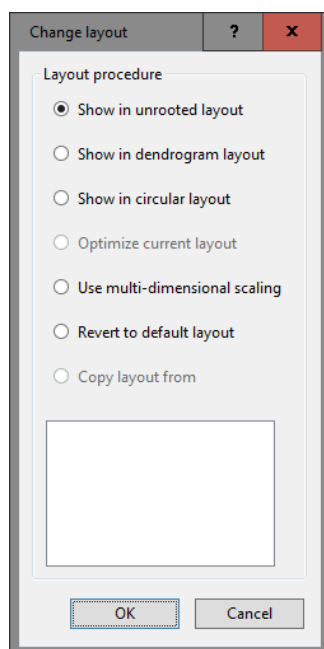


Figure 16.3.12: The *Change layout* dialog box.

Following options can be selected:

- **Show in unrooted layout:** For a rooted tree, this will display the tree with the same layout as an unrooted tree. However, the root node will be preserved so that it is possible at any time to revert the

tree to the rooted layout. As such, this function is different from the command **Edit > Remove root**, (see 16.3.3) which physically removes the root.

- **Show in dendrogram layout:** For an unrooted or circular tree, this will display the tree as a rooted tree.
- **Show in circular layout:** For an unrooted or rooted tree, this option will display the tree as a circular dendrogram.
- **Optimize current layout:** Tries to optimize the size of the nodes and the spread of the branches.
- **Use multi-dimensional scaling:** Uses a multi-dimensional scaling to position the entry nodes so that they occupy the best possible distance to each other to reflect the distances in the similarity/distance matrix. Depending on the type of data and analysis, this may result in many overlapping branches.
- **Revert to default layout:** Reverts the current layout to the default layout that appeared after the calculations.
- **Copy layout from:** Copies the layout from another network with the same entries. When checked, you can select a tree from a list of other networks loaded in the present window. This is a very useful feature to compare trees. Note that the layout can only be copied between unrooted trees.

16.3.3.6 Rotating a network

Another useful feature that is present in the *Advanced cluster analysis* window is the rotate function. The *Rotate network* dialog box is called with **Display > Rotate** (see Figure 16.3.13).

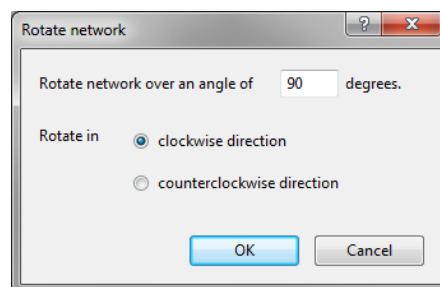


Figure 16.3.13: The *Rotate network* dialog box.

The *Rotate network* dialog box allows you to specify a rotation angle in degrees and a direction (clockwise or counterclockwise).

16.3.3.7 Hypothetical nodes

A parsimony tree is a dichotomic tree in which the entries usually occur on the leaf nodes. Although entries may sometimes occur on internal nodes, these are usually hypothetical nodes that can be considered as ancestor types that are not existing anymore or at least, not present in the entry set. Such nodes also correspond to a (hypothetical) character set, which can be inferred from the tree and displayed.

To display the character data for hypothetical nodes in a parsimony tree, select **Edit > Infer hypothetical node data**.

Click on an internal hypothetical node (i.e. containing no entries) to display the character data for that node in the *Entry data panel*. The character data for hypothetical nodes can be shown together with data from other nodes using the multi-select function (see 16.3.3). The result is shown in Figure 16.3.14. Hypothetical nodes are displayed as synthetic entries: "_synthetic_entry_" followed by an index.

	196	197	198	199	201	202	203	204	206	212	213	214	215	219	220	221
G@Gel07@002	A	A	T	A	C	A	C	A	T	A	C	A	C	C	C	A
G@Gel07@003	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A
G@Gel07@016	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A
_synthetic_entry_6	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A
G@Gel09@004	A	A	T	A	T	A	A	A	T	C	C	G	C	C	C	A
_synthetic_entry_17	A	A	T	A	T	A	A	A	T	C	C	G	C	C	C	A
G@Gel11@004	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A
G@Gel11@009	A	A	T	A	T	A	A	A	T	C	C	G	C	C	C	A
G@Gel11@003	A	A	T	A	T	A	A	A	T	C	C	G	C	C	C	A
G@Gel11@010	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A
_synthetic_entry_21	A	A	T	A	T	A	C	G	T	A	C	G	C	C	C	A

Figure 16.3.14: Entry data panel containing data for hypothetical nodes in a parsimony tree.

Note that the function to infer character data for hypothetical nodes can also be used on maximum likelihood trees.

16.3.3.8 Parsimonize/Likelihoodize a tree

For any Global Closest Pair and Neighbor Joining method, the resulting tree can be parsimonized, which means that a maximum parsimony tree is calculated with the current tree as starting point. In case of large data sets, this feature can be interesting as it can reduce the calculation time drastically. If used in combination with Neighbor Joining (most recommended), the result is not inferior to a native parsimony tree. In case a rooted tree was calculated (Global Closest Pair methods), the root will have to be removed first with *Edit > Remove root*.

An unrooted tree can be parsimonized using *Edit > Parsimonize*.

The parsimonization can be calculated using simulated annealing as well by selecting *Edit > Parsimonize (simulated annealing)*.

Note that a similar function can also be applied to obtain a maximum likelihood tree, which is even more useful, since the calculation of a maximum likelihood tree can be extremely time-consuming in case of large data sets. The corresponding functions are *Edit > Likelihoodize* and *Edit > Likelihoodize (simulated annealing)*.

16.3.3.9 Cross links in networks

Although a MST optimizes the connections between the entries so that related entries are connected as much as possible, sometimes, the algorithm has to make choices between equivalent solutions, thereby excluding possible connections. For example, consider the following case where 3 entries A, B and C each have a distance of 2 to one another. Figure 16.3.15 illustrates that a MST can be constructed in three ways, which are equivalent according to the algorithm. Obviously, priority rules might favor one particular solution, but still, close relationships between entries might go undetected.

In a MST, cross-links can be added between any two entries that have a shorter distance than predicted by the tree. In the example above, the second solution predicts a distance of 4 between B and C. The actual distance is 2, which is shorter and hence a cross-link will be shown. Obviously, since one only wants to see close relationships, it is useful to set a maximum distance to be displayed as cross-link. This is done as

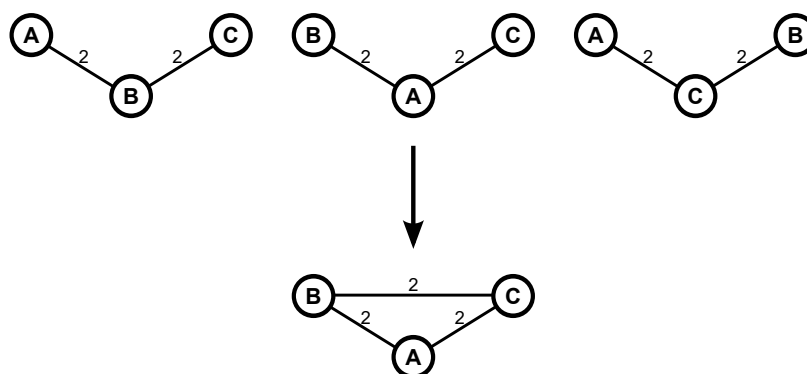


Figure 16.3.15: Displaying cross-links on a minimum spanning tree.

follows:

Select **Edit > Add cross links** to call the *Crosslinks* dialog box.

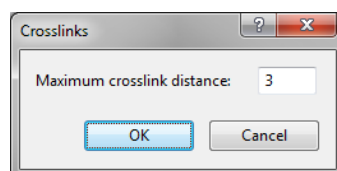


Figure 16.3.16: Specify the maximum cross link distance.

The dialog box prompts you to enter a **Maximum cross link distance**. When specify for example a maximum of 2, all entries with maximally 2 different bands and which the algorithm connected with a longer distance than 2, will be displayed with a cross link.

The result is shown in Figure 16.3.17, the cross-links indicated with an orange arrow. For convenience, the distances were displayed on the branches. The two cross-links in this MST reflect exactly the theoretical example in Figure 16.3.15.

Cross-links can be removed at any time using **Edit > Remove cross links**.

16.3.3.10 Hiding long branches

Besides adding cross-links for closely related entries, it can sometimes be useful to hide branches on a MST of which the distances are too long to be meaningful in the context of the analysis.

Selecting **Display > Hide long branches** calls the *Hide long branches* dialog box (see Figure 16.3.18).

This dialog ask to **Hide all branches longer than** a certain value. When entering 4 for example, all branches of 5 (or longer) will be hidden.

Hidden long branches can be shown again with **Display > Unhide long branches**.

16.3.3.11 Creating partitions

Partitions can be created for all rooted trees and for MSTs. Given a maximum distance value entered by the user, the partitioning algorithm will group entries in partitions (complexes) that have a distance lower than or equal to the entered value. However, the algorithm has to interpret the distances differently for rooted trees and MSTs.

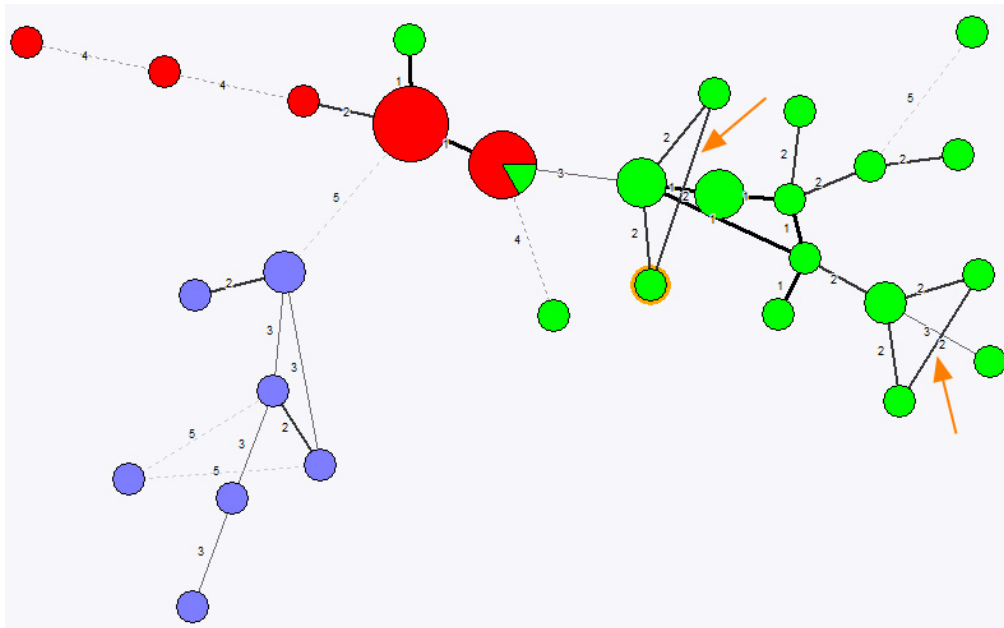


Figure 16.3.17: MST with cross-links added.

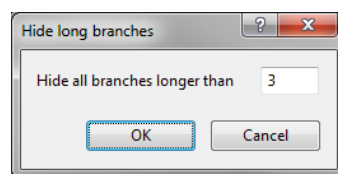



Figure 16.3.18: Hide branches.

- In a rooted tree, all leaf nodes within a cluster are grouped into a partition if the maximum distance from a leaf node to the root node of that cluster is within the entered distance boundary. In Figure 16.3.19, the left image is a rooted tree that was partitioned using 20 as a maximum distance. Within the pink partition, the maximum distance is 18 ($7+7+4$). Within the cyan partition, the maximum distance is 7 ($2+5$). The distances to the next node, however are 32 and 20, respectively. Although 20 is just within the boundary, 32 is not, so these two groups are separate partitions.
- In a MST, a partition is extended as long as the distance between connected nodes is less than or equal to the maximum distance. In Figure 16.3.19, the right image is a MST that was partitioned with 2 as maximum node distance. The partition extends for all nodes that are interconnected with distances of 2 or less. As soon as a connection has a longer distance (3 on the left and 5 on the right), the partition ends. Note that in a MST, individual nodes within a partition can have a greater distance to each other than the specified maximum, but they are connected through other nodes that have a smaller distance. The definition of a partitioning in a MST corresponds to the clonal complexes as defined for MLST, MLVA and similar allele-based bacterial typing techniques.

A partitioning can be created in the *Advanced cluster analysis* window with **Edit > Create partitioning** or using the  button. This calls the *Partitioning* dialog box (see Figure 16.3.20).

This dialog prompts to enter a **Maximum distance between nodes in the same partition**. The dialog box also allows you to specify the **Minimum number of entries in a partition** and the **Minimum number of nodes in a partition**. These options can be useful to avoid single-entry and single-node complexes, respectively.

Collapse partitions into single node is a function that will collapse the complexes into single nodes. The size of the nodes will be proportional to the number of entries they contain. This collapsing function cannot

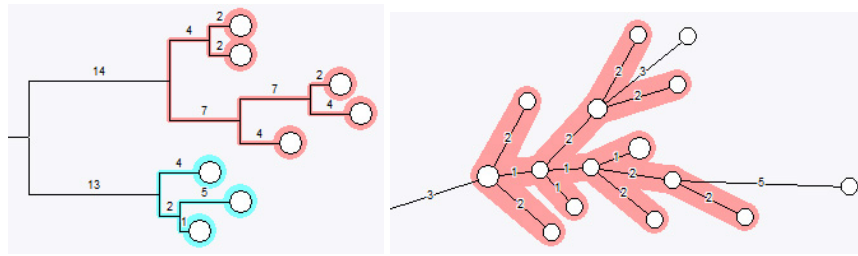


Figure 16.3.19: Partitioning in a rooted tree (maximum distance = 20) and a MST (maximum node distance = 2).

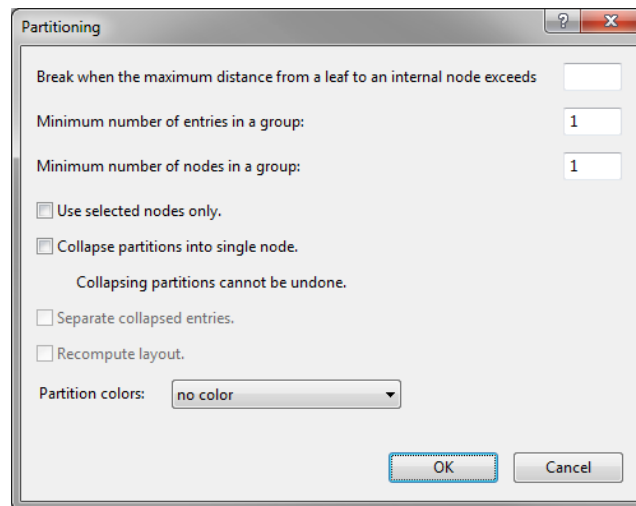


Figure 16.3.20: The *Partitioning* dialog box.

be undone. To restore the non-collapsed branches you will have to recalculate the network. If collapsing is applied, you can optionally display the entries separately within the nodes by checking *Separate collapsed entries* (exploded pie chart display mode) and have the layout optimized again with *Recompute layout*.



Since *Collapse partitions into single node* cannot be undone, a useful option in this respect is to first create a duplicate of the analysis with **File > Duplicate analysis** and to perform the command on the copy.

Although complexes are displayed in gray by default, they can optionally be displayed in color (*Partition colors*). The color can be adopted from the entries if comparison groups or field states were used to color the nodes. In case a complex consists of different entry colors, the color from the majority is taken (*Color from majority*). Alternatively, if no color was used for the entries, colors can also be assigned by the program (*Randomly chosen color*).


Changing the partition coloring can be done with the option **Edit > Colorize partitioning** (see Figure 16.3.21).

Different coloring options are possible:

Color from majority: The color from the majority of the entry colors is taken.

Randomly chosen color: Colors are randomly assigned by the program.

Color from range: Colors are taken ranging between the *Start color* and *End color*.

A partitioning can be exported as comparison groups or to a database information field. To do do, select **File > Transfer partitioning** or press the  button. A new dialog is displayed (see Figure 16.3.22).

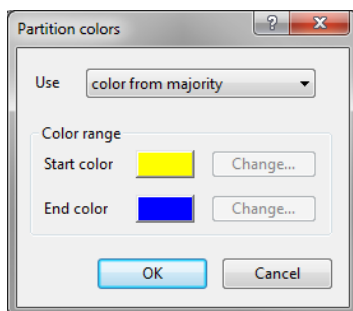


Figure 16.3.21: The *Partition colors* dialog box.

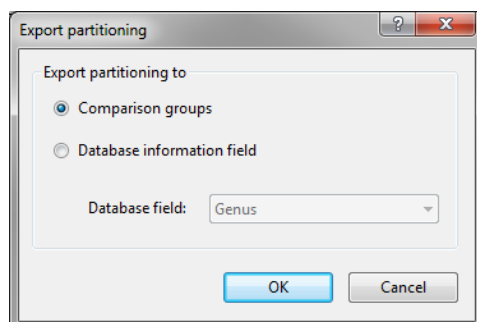


Figure 16.3.22: The *Export partitioning* dialog box.

This dialog allows the partitioning to be exported to **Comparison groups** or to a **Database information field**, which can be selected from the corresponding drop-down list. In the latter case, the information field will be filled with the complex information named as "Cluster" followed by an index.

16.3.3.12 Creating subnetworks

Subnetworks can be created from an existing cluster analysis. A subnetwork retains the same layout of the original network, but only a subset of the nodes are plotted.

To create a subnetwork or a set of subnetworks, select **File > Tools > Create subnetworks...** in the *Advanced cluster analysis* window. The *Create subnetwork(s)* dialog box pops up (see Figure 16.3.23).

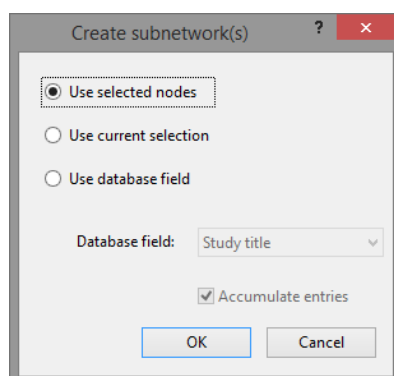


Figure 16.3.23: The *Create subnetwork(s)* dialog box.

Three options are available for creating subnetwork(s):


- **Use selected nodes:** A subnetwork will be created that contains the selected nodes from the original

network.

- **Use current selection:** A subnetwork will be created that contains the currently selected entries in the comparison.
- **Use database field:** In contrast to the previous options, a set of subnetworks will be created, i.e. a single subnetwork for each unique text that is found in the entry information field specified via the **Database field** drop-down list. If **Accumulate entries** is checked, nodes are cumulatively added to the subnetworks.

Subnetworks have a very illustrative application in epidemiology. For example, with a minimum spanning tree as the original network and subnetworks cumulatively created based on year, month or day of isolation, an "animation" that shows the progression of an outbreak can be created by scrolling through the list of cluster analyses.

16.3.4 Statistics

Statistics can be calculated based on the tree that is displayed in the *Advanced cluster analysis* window. Selecting **Edit > Compute statistics** or pressing the  button calls the *Statistics* dialog box.

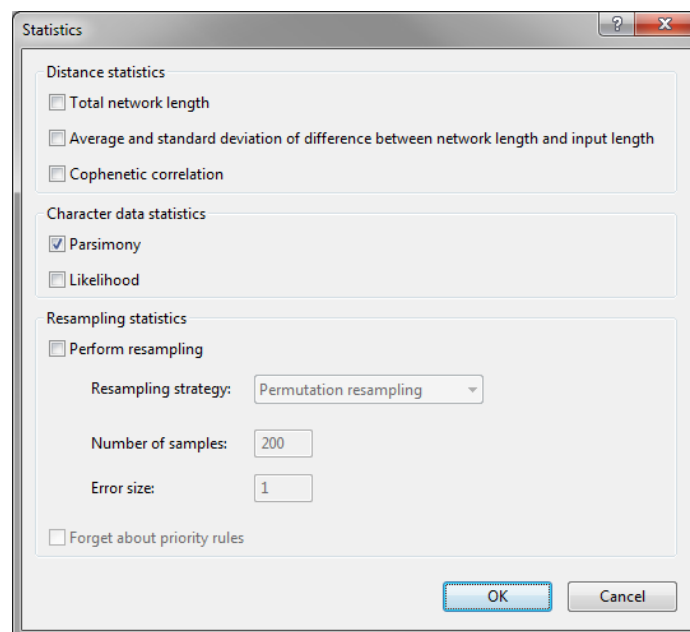


Figure 16.3.24: The *Statistics* dialog box.

This dialog box contains three types of statistics:

Distance statistics: These data will appear in the *Statistics panel* if calculated.

- **Total network length:** Calculates the total length of the network. This is particularly useful for parsimony trees, e.g. to compare the parsimony of different tree solutions.
- **Average and standard deviation of difference between network length and input length:** Calculates the average of differences between input distance and network length as well as the standard deviation. This measure is somewhat comparable to the cophenetic correlation but is suitable for unrooted tree methods, e.g. Neighbor Joining.

- **Cophenetic correlation:** Calculates the cophenetic correlation of the whole network. Only possible for rooted trees.

Character data statistics: These data will appear in the *Statistics panel* if calculated.

- **Parsimony:** Displays the total parsimony of a tree. Selected by default if parsimony is chosen.
- **Likelihood:** Displays the total likelihood of a tree. Selected by default if maximum likelihood is chosen.

Resampling statistics:

- **Perform resampling:** Allows a resampling calculation to be performed on the current network. The difference with the resampling framework as discussed in the introduction is that here, the resampling is done *after* a network has been calculated. As such, the user can make sure that a tree is calculated using specific parameters and settings and then perform a resampling statistics afterwards. The resampling possibilities depend on the type of data: for similarity-based data one can perform permutation resampling and error resampling, whereas on character data one can perform permutation resampling and bootstrap resampling.
- **Forget about priority rules:** In case of a MST, the resampling can be calculated using the minimum spanning tree algorithm but without taking the priority rules into account. This can be interesting to evaluate how much a specific tree topology relies on the priority rules used.

16.3.5 Analysis templates

The advanced clustering itself is saved when the underlying parent comparison is saved. The name of the clustering will appear in the *Analyses* panel of the *Comparison* window, so that the clustering can be opened again at any time by double-clicking on the name.

All the analysis and layout settings can be saved in a new template using **File > Save analysis template**. This call a new dialog (see Figure 16.3.25).

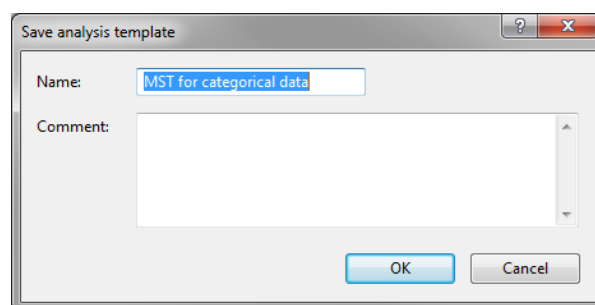


Figure 16.3.25: Save analysis template

You can enter a name and, optionally, a description of the template in the **Comment** field (e.g. what the template does, the settings, the data it should be used for...). This description will appear if you select the template in the *Create network* wizard.

A saved analysis template can be removed from the list of templates with **File > Remove analysis template**. This calls the *Remove analysis templates* dialog box (see Figure 16.3.26).

Select the template you want to remove from the template list.

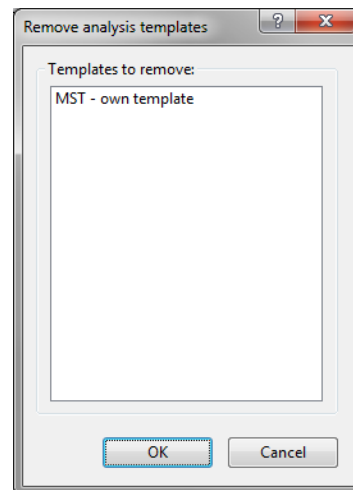




Figure 16.3.26: Remove an analysis template from the list of templates.

16.3.6 Analysis functions

The current window can have multiple cluster analyses loaded. This makes it easy to compare the results of different methods with each other.

Once a cluster analysis has been calculated, a new cluster analysis can be calculated with **File > New analysis** or the  button.

Analyses that were previously saved within the comparison can also be loaded using **File > Open analysis** or the  button (see Figure 16.3.27).

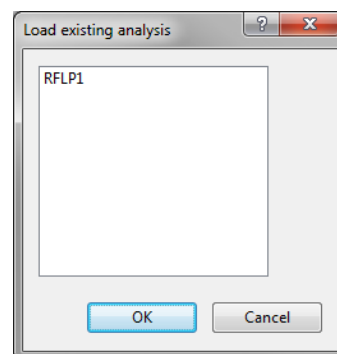
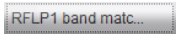


Figure 16.3.27: The *Load existing analysis* dialog box to load an existing analysis.

Select the analysis you want to load from the list.

If multiple analyses are loaded in the window, you can easily switch between them using the  drop-down menu.

Renaming the analysis is done with **File > Rename analysis**. This calls the *Rename* dialog box (see Figure 16.3.28).

A new name needs to be specified.

Creating a duplicate analysis with the same analysis and layout settings is done with **File > Duplicate analysis**. This calls the *Duplicate current network* dialog box (see Figure 16.3.29).

A new name needs to be specified.

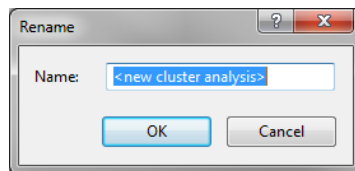


Figure 16.3.28: Specify a new name.

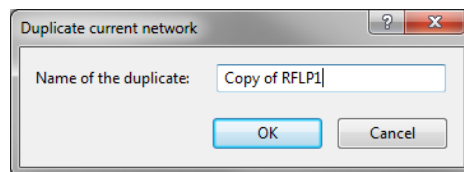




Figure 16.3.29: Specify a name for the duplicate analysis.

16.3.7 Exporting and printing the analysis

The tree, branch and statistics properties, which are displayed in the *Cluster analysis method*, *Branch properties* and *Statistics panels*, can be exported either as text file or as HTML file using the corresponding  and  buttons. The same functions are also available from the menu as **File > Export report**. The text or HTML files are saved in a `export.txt` or `export.htm` in the database folder and are opened in Notepad or the default web browser.

The network can be printed directly to a printer using **File > Print**. The dialog box that appears is the standard Windows Print dialog box, allowing you to choose a printer and change the properties.

The network image can also be exported with **File > Save as**.

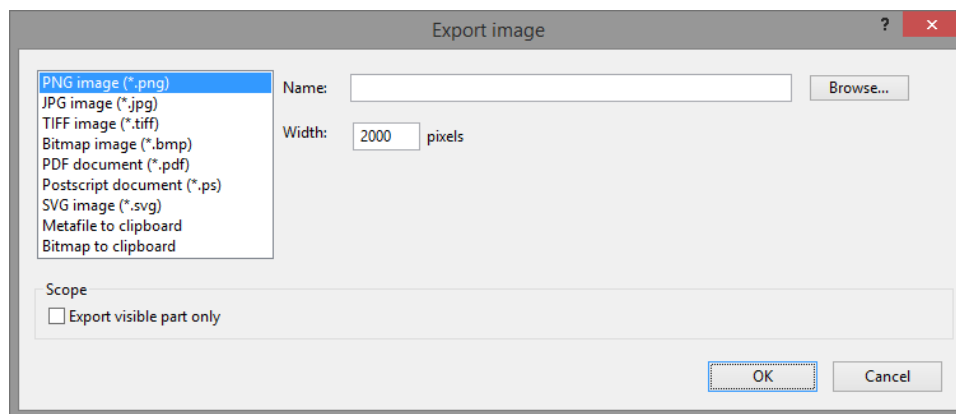


Figure 16.3.30: The *Export image* dialog box.


This dialog box allows you to export graphical information to a file or to the Windows clipboard in one of several available formats. In case a file is exported, a file **Name** should always be entered or browsed for via the **<Browse>** button. Exported files will open in their default editor. Information on the Windows clipboard can be pasted into other applications. Following export options are available:

- **PNG image (*.png)**: exports to a Portable Network Graphics (PNG) file. PNG is a raster graphics file format that supports lossless data compression. A **Name** and **Width** (in pixels) should be specified; the height will be determined automatically.
- **JPG image (*.jpg)**: exports to a Joint Photographic Experts Group (JPEG) file. JPEG or JPG is a

raster graphics file format that uses a lossy data compression. A **Name** and **Width** (in pixels) should be specified, as well as a **Quality** parameter. With the latter, a tradeoff can be obtained between storage size and image quality.

- **TIFF image (*.tiff)**: exports to a Tagged Image File Format (TIFF) file. TIFF is a raster graphics file format with optional lossless data compression. A **Name** and **Width** (in pixels) should be specified.
- **Bitmap image (*.bmp)**: exports to a BMP bitmap image or device independent bitmap (DIB) file. BMP is a raster graphics image file format used to store bitmap digital images, independently of the display device. A **Name** and **Width** (in pixels) should be specified.
- **PDF document (*.pdf)**: exports to a Portable Document Format (PDF) file. PDF is a file format used to present documents in a manner independent of application software, hardware, and operating systems. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **Postscript document (*.ps)**: exports to a PostScript (PS) file. PostScript is a computer language for creating vector graphics. A **Name** and the **Orientation** (either Landscape or Portrait) should be specified.
- **SVG image (*.svg)**: exports to a Scalable Vector Graphics (SVG) file. SVG is an XML-based vector image format for two-dimensional graphics. A **Name** should be specified.
- **Metafile to clipboard**: copies the graphics as Windows enhanced metafile to the clipboard. Enhanced metafile is the standard clipboard exchange format between native Windows applications.
- **Bitmap to clipboard**: copies the graphics as a bitmap to the Windows clipboard. The **Width** (in pixels) should be specified.

If character data is used as input, the character data for the selected entries is displayed in the *Entry data panel*. This table can be exported as tab-delimited text file with **File > Export character data**. A file `export.txt` is created in the Temp subdirectory of the database folder, and is opened in Notepad.

A rooted tree (dendrogram) can also be exported to the parent *Comparison* window with **File > Show dendrogram in comparison** or . Once copied in the *Comparison* window, all editing and layout functions available in that window can be used on the tree.

Chapter 16.4

Tutorials

16.4.1 Introduction

The *Create network* wizard contains a number of default templates for the most commonly performed types of cluster analyses. However, depending on the type of data and the analysis needs, it can be necessary to define your own analysis settings and save these as a new template.

To illustrate the power and versatility of the *Advanced cluster analysis* window we will use the wizard to calculate a few examples in **DemoBase Connected**. These will include:

- UPGMA with error resampling using quantitative character data;
- Maximum parsimony with bootstrap resampling of DNA sequence data;
- Minimum spanning tree of fingerprint band matching table.


The **DemoBase Connected** can be downloaded directly from the *BioNumerics Startup* window or restored from the back-up file available on our website:

- To download the database directly from the *BioNumerics Startup* window, click the **Download example databases** link, located in the lower right corner of the *BioNumerics Startup* window. Select **DemoBase Connected** from the list and select **Database > Download**.
- To restore the database from the back-up file, first download the file `DemoBase_Connected.bnbk` from <http://www.applied-maths.com/download/sample-data>, under 'DemoBase Connected'.

In the *BioNumerics Startup* window, press the  button, select **Restore database**, browse for the downloaded file and select **Create copy**. Specify a name and click **<OK>**.

16.4.2 UPGMA with error resampling

2.1 Select all entries in the *Main* window except those labeled as STANDARD in the 'Genus' field.

2.2 Highlight the *Comparisons* panel in the *Main* window and select **Edit > Create new object...** () to create a new comparison for the selected entries.


2.3 In the *Comparison* window, right-click in the header of the "Genus" field and select **Create groups from database field** from the menu. Alternatively select **Groups > Create groups from database field**.

2.4 Press **<Yes>** to create three groups according to the genus names.

In this example we will use the character experiment **FAME** (fatty acid methyl esters) as data source. Since a similarity matrix is required for error resampling, we first create a UPGMA dendrogram from experiment type **FAME**.

2.5 In the *Experiments* panel, select **FAME** and then perform a cluster analysis using the basic method: choose **Clustering > Calculate > Cluster analysis (similarity matrix)...** (see 4.2), select **Euclidean distance** as coefficient and use **UPGMA** to obtain a clustering.

When the calculation is finished, the similarity matrix is displayed in the *Similarities* panel.

2.6 Select **Clustering > Calculate > Advanced cluster analysis...** or press the  button and select **Advanced cluster analysis** to launch the *Create network wizard*.

2.7 Enter "UPGMA+error" as **Name** of the cluster analysis. Make sure **FAME** is selected under **Choose an experiment**.

2.8 Under **Analysis template**, choose **No template** and press **<Next>**.

2.9 Select **Similarity matrix** and press **<Next>** to move on to the next step.

2.10 Select **UPGMA** as clustering method and leave **Check preconditions** unchecked. Press **<Next>** to proceed to the next step.

2.11 Select **Error resampling**.

2.12 With **FAME** data, the error is relatively small so enter 0.5 as value for the **Standard deviation** of the error.

2.13 Press **<Next>** to proceed to the next step.

2.14 For this example, select **Compute summary network** with **Majority summary** as summary method and select **Create tree** as summary type.

2.15 Press **<Finish>** to start the calculation.

The result looks as in Figure 16.4.1. In this figure, the error resampling support is shown as branch labels and branch colors, and the strain numbers are used as entry labels.

Since we used a majority summary tree, every branch has a resampling support of at least 50%.

16.4.3 Maximum parsimony with bootstrap resampling

To illustrate the calculation of a maximum parsimony tree with bootstrap resampling, we will use the comparison in **DemoBase Connected**, for which comparison groups were already assigned (see 16.4.2), and use experiment **16S rDNA** as data source. Since a multiple alignment is required for sequence clustering with bootstrap analysis, we will first create a multiple alignment from experiment type **16S rDNA**.

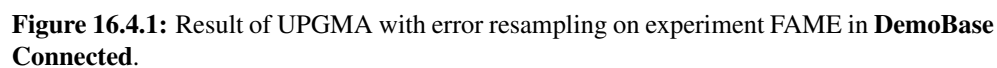
3.1 In the *Comparison* window, select experiment type **16S rDNA** in the *Experiments* panel.

3.2 Select **Sequence > Multiple alignment...** (.

3.3 In the *Multiple alignment* dialog box, uncheck **Use fast algorithm** and press **<OK>**.

3.4 When the multiple alignment is displayed, select **Clustering > Calculate > Advanced cluster analysis...**

3.5 In the *Create network wizard*, enter **16S rDNA parsimony** as **Name of the cluster analysis**. Make sure **16S rDNA** is selected under **Choose an experiment**.



- 3.6 Under *Analysis template*, choose *Maximum parsimony tree* and check *Modify template settings for new analysis*. By doing this, we will be able to run through the wizard with the default settings displayed for this template, allowing us to modify settings where needed.
- 3.7 Press <Next> twice.
- 3.8 In step 3, *Basic maximum parsimony tree* is selected as a default. As an example, we can change this setting into *Optimized maximum parsimony tree (Simulated Annealing)*. This variant finds the highest parsimony using simulated annealing, which is often a better heuristic approach.
- 3.9 Press <Next> to move to the next step.

3.10 Leave the settings unaltered and press **<Next>**.

The result looks as in Figure 16.4.2. In this figure, the branch length were displayed on the branches. Since

branch lengths in parsimony are expressed as numbers of mutations, the values were clipped to zero decimal digits in this example.

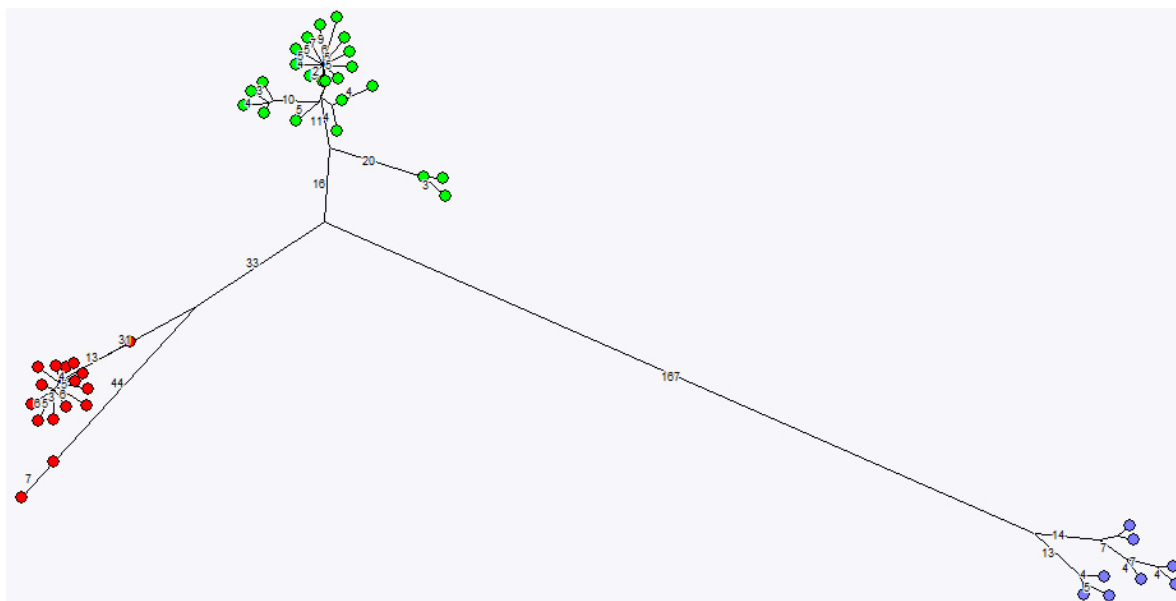



Figure 16.4.2: Result of parsimony clustering on experiment 16S rDNA in DemoBase Connected.

To calculate the bootstrap values on this tree, we will call the *Statistics* dialog box.

3.11 Select **Edit > Compute statistics** or press the  button.

3.12 In this example, we will perform a bootstrap resampling by checking **Perform resampling** and selecting **Bootstrap resampling** under **Resampling strategy**.

After the calculations, the program displays the bootstrap values as a percentage next to the parsimony values (Figure 16.4.3). For example, "33, 98%" means that the branch has a distance of 33 mutations (measured by the cost matrix) and a bootstrap value of 98%. In Figure 16.4.3, the parsimony values were displayed with zero decimal digits, a feature which can be specified in the display settings (**Display > Display settings, Branch labels and sizes tab**).

Note that an unrooted tree or network does not always have the desired layout for a given window size or for specific printing or exporting purposes. In Figure 16.4.3, for example, the tree layout was optimized for the available window size using the layout settings.

3.13 To optimize the network layout, select **Display > Layout**.

Another useful feature that has been applied to Figure 16.4.3 is the rotate function.

3.14 To rotate a network, select **Display > Rotate**.

In case a clustering consists of very homogeneous and well distinguished taxa, the resulting tree may have large between-cluster distances and very short within-cluster distances, so that the within-group relationships cannot be examined easily. This can be solved by showing the distances on a logarithmic scale as follows:

3.15 Select **Display > Display settings** or press the  button and click on the **Branch labels and sizes tab**. Under **Branch length scaling**, select **Use logarithmic scaling**.

As a result, the within-taxon distances are larger compared to the between-taxon distances.

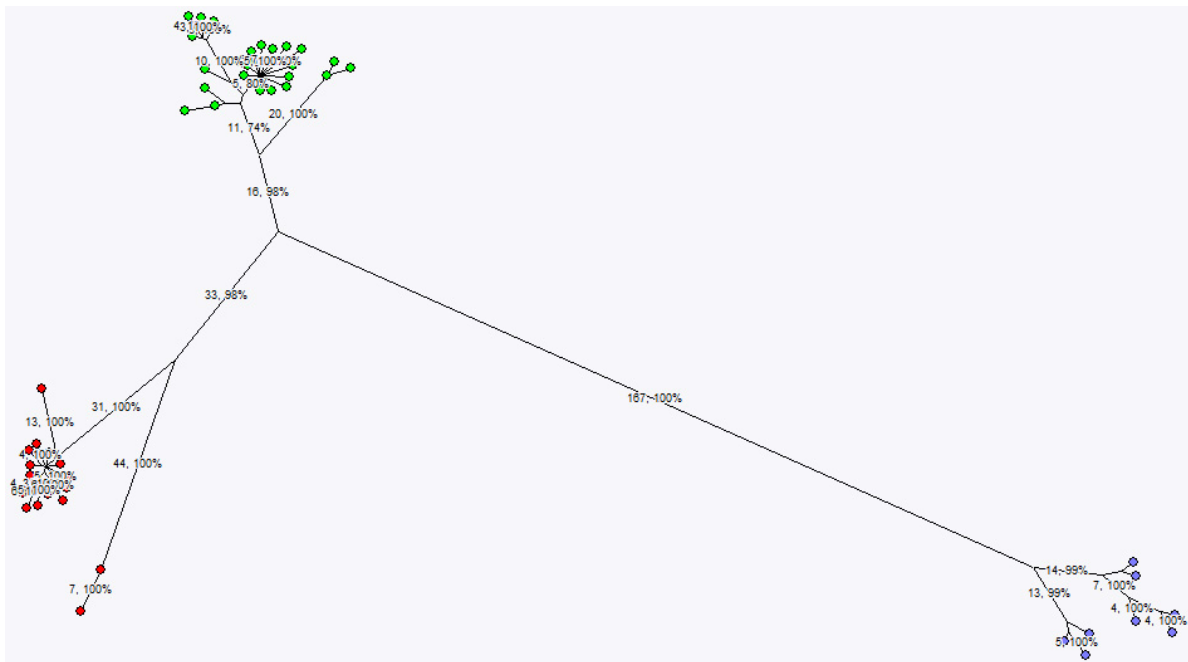


Figure 16.4.3: Result of bootstrap analysis on parsimony clustering on experiment **16S rDNA** in **DemoBase Connected**.

16.4.4 Minimum spanning tree with permutation resampling

Minimum spanning trees (MSTs) are known for a long time in the context of graph theory. When a set of distances is given between n samples, a minimum spanning tree is a tree that connects all samples in such a way that the summed distance of all branches of the tree is minimized.


In a biological context, the MST principle and the maximum parsimony (MP) principle share the idea that evolution should be explained with as little events as possible. However, there are major differences between MP and MST. The MP method allows the introduction of hypothetical samples, i.e. samples that are not part of the data set. Such hypothetical samples are created to construct the internal branches of the tree, whereas the real samples from the data set occupy the branch tips. The phylogenetic interpretation of the internal branches is that they are supposed to be common ancestors of current samples, which do not exist anymore but which are likely to have existed in the past, under the criterion of parsimony.

The MST principle, in contrast, requires that all samples are present in the data set to construct the tree. Internal branches are also based upon existing samples. This means that, when a MST is calculated for evolutionary studies, there are two important conditions that have to be met: (1) the study must focus on a very short time-frame, assuming that all forms or states are still present, and (2) the sampled data set must be complete enough to enable the method to construct a valid tree, i.e. representing the full biodiversity of forms or states as closely as possible. Through these restricting conditions, the method of MST is only applicable for specific purposes, of which population modeling (micro-evolution) and epidemiology are good examples.

To illustrate the calculation of a minimum spanning tree with permutation resampling, we will use the comparison in **DemoBase Connected**, for which comparison groups were already assigned (see 16.4.2), and use experiment **RFLP1** as data source.

As the most suitable data type for MST is binary or categorical character data, we will first create a band matching for fingerprint type **RFLP1** in the *Comparison* window.

4.1 In the *Comparison* window, select experiment type **RFLP1** in the *Experiments* panel.

4.2 Select **Fingerprints > Perform band matching...** (). Make sure **Find classes on all entries** is checked and press <OK>.

4.3 When the band matching is displayed, select **Clustering > Calculate > Advanced cluster analysis...**

4.4 In the *Create network* wizard, enter **RFLP1 band matching MST** as **Name of the cluster analysis**. Make sure **RFLP1** is selected under **Choose an experiment**.

We will again use a default template to start with and modify the settings where desired.

4.5 Under **Analysis template**, select **MST for categorical data** and check **Modify template settings for new analysis**.

4.6 Press <Next> to move to the next step.

The default setting for **Input data** is **Character data**, which in this case is a band matching table. Note that for character-based input data, the default setting for **Merging policy** is to **Merge taxa when distance is zero**. Using this setting, identical entries are displayed in one node without segmentation. Especially in combination with a resampling analysis (see below), this is the recommended setting.

4.7 Make sure **Character data** is checked and press <Next> to move to the next step.

4.8 **Minimum Spanning Tree** is selected as a default. Leave this setting unaltered and press <Next> to move to the next step.

Two priority rules are applied by default, designed for MLST analysis. These can be used for RFLP band matching data as well, translating "N-locus variants" into "N-bands difference".

4.9 Press <Next> to move to the next step.

In this step, **No resampling** is checked by default. In case of binary band matching data, which is subject to a high degree of degeneracy caused by the input order, it is useful to check **Permutation resampling**.

4.10 Check **Permutation resampling** and press <Next> to move to the next step.

4.11 In this last step, select **Threshold summary** from the **Summary method** drop-down list and enter 40% as a threshold value. This means that all branches with a percentage resampling support of 40% and higher will be displayed.

The resulting graph (Figure 16.4.4) is actually not a tree but a network because it is a consensus summary of all solutions with more than 40% resampling support.

4.12 Select **Edit > Add cross links**.

A dialog box prompts you to enter a **Maximum cross link distance**.

4.13 In the present example, enter for example "2". This means that all entries with maximally 2 different bands and which the algorithm connected with a longer distance than 2, will be displayed with a cross-link.

The result is shown in Figure 16.4.5, the cross-links indicated with an orange arrow. For convenience, the distances were displayed on the branches. The two cross-links in this MST reflect exactly the theoretical example in Figure 16.3.15.

4.14 Cross-links can be removed at any time using **Edit > Remove cross links**.

4.15 Select **Display > Hide long branches**.

A dialog box pops up, asking to **Hide all branches longer than** a certain value.

4.16 Enter 4, so that all branches of 5 (or longer) will be hidden, and press <OK>.

Part 17

Statistics and dimensioning

Chapter 17.1

Statistics on charts

17.1.1 Introduction

BioNumerics offers the possibility to perform some basic statistic analysis on the entries and variables used in a chart. For most chart types, one or more standard statistical tests are available. The next paragraphs are intended to provide some information on the terminology and the mathematical background of these tests. The use of the chart tools is described in [14.1](#) to [14.5](#). Please note that this functionality requires the Dimensioning and Statistics module (DI) to be present in your BioNumerics configuration.

This manual is not aimed to be an introduction to basic statistics. For more detailed literature, we refer to the following handbooks:

- J.D. Jobson. Applied Multivariate Data Analysis (2nd Vol.). Springer Verlag. ISBN 0 387 97804 6.
- Press W., Teukolsky S.A., Vetterling W.T., Flannery B.P., "Numerical recipes in C", Cambridge University Press, Cambridge.
- Sheskin D.J., "Handbook of parametric and nonparametric statistical procedures", CRC Press, Boca Raton.
- Zwillinger D., Kokoska S., "Standard probability and statistics tables and formulae", Chapman & Hall/CRC, Boca Raton.

17.1.2 Basic terminology

17.1.2.1 The use of statistical tests

In general terms, the application of a statistical test can be outlined as follows:

- Make a proposition that will be referred to as the *null hypothesis*. Statistical tests cannot be employed for proving that a certain hypothesis is true, but only for proving that alternative hypotheses can be rejected. Therefore, the null-hypothesis is what one wants to reject.
- Determine what *statistic* will be used. A statistic is a value calculated from the data set by means of some formula and which is sensitive to the null-hypothesis that will be tested for.
- If the null-hypothesis is true, the *probability distribution* of the statistic is known.

- If the statistic is located on an unfavorable position in the probability distribution, i.e. if its probability is very small, the null-hypothesis can be *rejected*. The opposite is not true: the null-hypothesis cannot be accepted as fulfilled if the statistic has a favorable location in the probability distribution.

Note that not all tests are applicable in all situations. There may be restrictions to e.g. the amount of data in the sample, or to some basic properties of the data set. These restrictions are mentioned where the tests are described.

17.1.2.2 Parametric or non-parametric tests

Parametric tests basically suppose that the data are distributed normally; they generally make use of the values for the mean and the standard deviation.

Non-parametric tests are commonly based on a ranking of the data. These ranks are distributed uniformly, hence these tests are independent of any underlying distribution. The toll to pay is that an estimate of the significance is more complicated and often relies on approximations. These methods also generally lose some strength because they lose some information about the data. In comparison with parametric tests they require more data to come to an equally significant result. For non-parametric tests the values of the data points are usually replaced by their rank among the sample. The data points are ordered, the lowest in order is assigned rank one and the highest in order is assigned the rank that equals the total sample size. If some of the data points originally have the same values, they can be assigned the mean of the ranks (called "tie rank") they would have had if they were different.

17.1.2.3 Categorical or quantitative data

Within the chart tool, a distinction between three types of variables is made.

- **Categorical variable:** This type of variable divides a sample into separate categories or classes. In the chart and statistics tools in BioNumerics, categories are derived from string properties. Examples are database fields like e.g. genus, species, etc..
- **Quantitative variable:** This type of variable can take either continuous numerical values or binary values. Character data are a typical example for this type of variable. Continuous numerical values can be converted into interval data if necessary. If this option is chosen, an interval size can be specified.

In addition, character data can be treated as numerical values, binary (present/absent) data or - in case a mapping was defined (see 6.1.2 for more information) - the mapped values can be used.

With combinations of these variables various types of charts can be created. Table 17.1.1 displays all charts for which tests are available, classified on the basis of the variable types used.

17.1.3 Description of tests

17.1.3.1 Kolmogorov–Smirnov test for normality

For a sample containing a single quantitative variable, an often recurring question is if it is normally distributed or not. In this case the null-hypothesis is that the sample is drawn from a normal distribution). The *mean value* $\langle x \rangle$ and *corrected standard deviation*

$$SD_{cor} = \sqrt{\frac{\sum_{i=1}^n (x_i - \langle x \rangle)^2}{n - 1}}$$

Variables	Chart type	Test(s)
1 numerical	Profile chart	Kolmogorov-Smirnov test for normality
	Bar graph	
	Value histogram	
	Box & Whiskers chart	
2 numerical	Scatter chart	Pearson correlation test (P)
		Spearman rank order test (NP)
		T-test for paired samples (P)
		Wilcoxon signed ranks (NP)
1 categorical	Frequency bar graph	Chi square test for category sizes Diversity index
	Frequency bar graph (colored)	
	Pie chart histogram table	
2 categorical	Contingency table	Chi square test for contingency tables
	3-D contingency chart	
	Pie chart histogram table	
1 categorical & 1 numerical	ANOVA chart	T-test for 2 independent groups (P)
		Mann-Whitney test for 2 groups (NP)
		Analysis of variance (F-test) (P)
		Kruskal-Wallis test for ANOVA (NP)

Table 17.1.1: Chart types and associated tests based on variable types used. P: Parametric, NP: Non-Parametric.

with x_i the observations and n the sample size) are calculated from the sample and are used to determine a normal distribution that can be used as a model (further referred to as model normal distribution) for the underlying distribution of the sample if the null-hypothesis holds.

The *Kolmogorov-Smirnov test for normality* is applied to test how different the cumulative distribution of the sample is from the cumulative distribution of the model normal distribution (see Figure 17.1.1). For a sample where each observation is associated with a single numerical value, the *sample cumulative distribution* $F(x_i)$ gives for each observation (x_i) the total number of values associated to all observations in the sample that are smaller or equal to the observation (x_i). The *model cumulative distribution* gives at each observation the probability of obtaining that observation or a lower one.

The test statistic is the *maximum difference* in absolute value between the cumulative distribution of the sample and the cumulative distribution of the model normal distribution. In case the null-hypothesis is true and under certain conditions (see note below), the distribution function for this statistic can be calculated approximately. The p -value gives the probability that the statistic obtains a higher value than the observed one. If the p -value is low, the null hypothesis can be rejected. The *significance* of the test can be calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.2). How such a report can be created is explained in 17.1.4.



The Kolmogorov-Smirnov test for normality should not be used for small sample sizes. The test becomes more accurate if more data points are used.



This test cannot be used to prove that a sample follows a normal distribution, since its aim is only to reject the null-hypothesis with a certain level of significance.

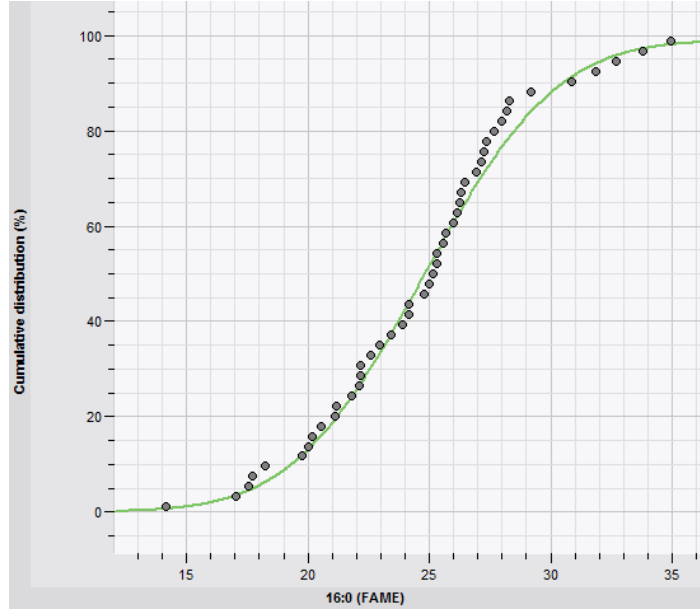


Figure 17.1.1: Example of a 1-D numerical distribution and model normal distribution.

Kolmogorov-Smirnov test

Mean: 93.446809
 Corrected standard deviation: 50.740948
 Maximum difference: 0.1676
 P value= 0.128416
 Significance= 87.1584%

Figure 17.1.2: Example of a test report for the Kolmogorov-Smirnov test for normality applied to a 1-D numerical distribution.

17.1.3.2 Parametric test for correlations: Pearson correlation test

Pearson correlation measures the linear correlation between two normally distributed quantitative variables. The null hypothesis is that there is no linear relationship between the sample variables. Assume the observations in the sample are x_i and y_i ($i = 1, \dots, n$), with $\langle x \rangle$ and $\langle y \rangle$ the *mean values*,

$$s_x = \frac{\sum_{i=1}^n (x_i - \langle x \rangle)^2}{n}$$

and

$$s_y = \frac{\sum_{i=1}^n (y_i - \langle y \rangle)^2}{n}$$

the *variances* and

$$\text{Cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{n}$$

the *covariance* of the sample. *Pearson's correlation* r is calculated as

$$r = \frac{\text{Cov}(x, y)}{\sqrt{s_x s_y}}$$

If the null-hypothesis holds and under certain conditions (see note below) the statistic defined as

$$|r| \frac{\sqrt{n-2}}{1-r^2}$$

approximately follows a t distribution with $n - 2$ degrees of freedom. The absolute value $|r|$ is used in order to be able to test negative correlations as well. The p -value gives the probability that the statistic obtains a value at least as high as the observed one. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.3). How such a report can be created is explained in 17.1.4.

```
Pearson correlation test

Mean values:
 15:0      0.5928
 14:0      3.2485
Variances:
 15:0      1.1745
 14:0      4.3859
Covariance= 0.6332

Pearson correlation= 27.901%
P value (single tail)= 0.028769 (T test approximation)
Significance= 97.1231%
```

Figure 17.1.3: Example of a test report for the Pearson correlation test applied to a scatter chart.

In case there is a significant linear correlation, Pearson's r can be used to indicate its strength. A positive value for Pearson's r is associated with a positive correlation and would result in a regression line with positive slope. A negative value for Pearson's r is associated with a negative correlation and would result in a regression line with negative slope.

If the samples contain less than 30 observations, an alternative way for testing the null-hypothesis is offered by *Monte-Carlo simulations*. To do this, 10,000 samples with n pairs of randomly distributed observations are created. For each of these samples, a value for the statistic is obtained and is compared to the observed value. The p -value from the simulations is determined by the number of times the simulations give a larger value for the statistic than the value observed in the real sample. Also here, the *significance* is calculated as $s = 100(1 - p)$. The results for the simulated p -value and significance also appear in the test report.



This test should not be used if the distributions of x_i or y_i are not normally distributed, e.g. if they have strong wings.

17.1.3.3 Non-parametric test for correlations: Spearman rank-order correlation test

The Spearman rank-order test measures the linear correlation between two quantitative variables for which a normal distribution is not required. The null-hypothesis is that there is no linear correlation between the sample rank variables, or equivalently that there is no monotonic relation between the sample variables. The sample observations x_i and y_i ($i = 1, \dots, n$) are replaced by their rank after ordering them from smallest to largest. This results in a sample of ranks R_i and S_i ($i = 1, \dots, n$). The *Spearman rank-order correlation coefficient* is defined as

$$r_s = \frac{Cov(R, S)}{\sqrt{S_R S_S}}$$

with

$$S_R = \frac{\sum_{i=1}^n (R_i - \langle R \rangle)^2}{n}$$

and

$$S_S = \frac{\sum_{i=1}^n (S_i - \langle S \rangle)^2}{n}$$

the *rank variances*,

$$\text{Cov}(R, S) = \frac{\sum_{i=1}^n (R_i - \langle R \rangle)(S_i - \langle S \rangle)}{n}$$

the *rank covariance* and $\langle R \rangle$ and $\langle S \rangle$ the *rank mean values* of the rank variables R_i and S_i respectively.

The null-hypothesis can be tested using the statistic

$$|r_s| \frac{\sqrt{n-2}}{\sqrt{1-r_s^2}}$$

If the null-hypothesis holds, this statistic approximately follows a t distribution with $n - 2$ *degrees of freedom*. Since $|r_s|$ is used to calculate the statistic, the *p*-value can be calculated using a single tail of the t distribution. The *p*-value gives the probability that the statistic obtains a value at least as high as the observed one. In this case, a single tail test is performed. The *significance* of the test is calculated as the complement of the *p*-value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.4). How such a report can be created is explained in 17.1.4.

```
Spearman rank order correlation test

Rank mean values:
15:0      24.0000
14:0      24.0000
Rank variances:
15:0      173.6596
14:0      183.9468
Rank covariance= -23.0851

Spearman rank-order correlation= -12.916%
P value (single tail)= 0.193442 (T test approximation)
Significance= 80.6558%
```

Figure 17.1.4: Example of a test report for the Spearman rank-order correlation test applied to a 2-D scatter chart.

If the samples contain less than 30 observations, an alternative way for testing the null-hypothesis is offered by *Monte-Carlo simulations*. To do this, 10,000 samples with n pairs randomly distributed observations are created. For each of these samples, a value for the statistic is obtained and is compared to the observed value. The *p*-value from the simulations is determined by the number of times the simulations give a larger value for the statistic than the value observed in the real sample. Also here, the *significance* is calculated as $s = 100(1 - p)$. The results for the simulated *p*-value and significance also appear in the test report.

17.1.3.4 T test for paired samples

This parametric test assesses whether the means of samples from two variables that are paired are statistically different from each other. In BioNumerics, an obvious example of *paired samples* is comparing two different characters within a given set of entries. The null-hypothesis is that the two paired samples have the same mean values. Assume the sample observations are x_i and y_i ($i = 1, \dots, n$), with $\langle x \rangle$ and $\langle y \rangle$ the respective *mean values* and

$$s_x = \frac{\sum_{i=1}^n (x_i - \langle x \rangle)^2}{n - 1}$$

and

$$s_y = \frac{\sum_{i=1}^n (y_i - \langle y \rangle)^2}{n - 1}$$

the *corrected variances*.

For paired data, it is generally not guaranteed that all entries have a completely independent pair of observations. The test statistic should be corrected for the influences this may have on the variance of the observations. Therefore, the *corrected covariance* of the sample,

$$\text{Cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \langle x \rangle)(y_i - \langle y \rangle)}{n - 1}$$

is taken into account. The sample variance can be expressed by means of the *pooled corrected standard deviation* s_d . In this case, s_d can be calculated as

$$s_d = \sqrt{\frac{s_x + s_y - 2\text{Cov}(x, y)}{n}}$$

A statistic is defined as $T = \frac{\langle x \rangle - \langle y \rangle}{s_d}$. If the null-hypothesis holds and under certain conditions (see note below) this statistic follows a T distribution with $n - 1$ *degrees of freedom*. The p -value gives the probability that the statistic indeed has the observed value or higher. If the p -value is small, the null-hypothesis can be rejected. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.5). How such a chart and report can be created is explained in 17.1.4.

```
T test for mean value (paired samples)

Mean values:
15:0      0.5928
14:0      3.2485
Corrected variances:
15:0      1.2000
14:0      4.4812
Corrected covariance = 0.6470
Pooled corrected standard deviation = 0.3055

T = -8.692 (46 degrees of freedom)
P value= 0.000000
Significance= 100.0000%
```

Figure 17.1.5: Example of a test report for the T test applied to a scatter chart.



This test should not be used if the data points are not normally distributed. In this case the Wilcoxon signed-rank test can be used.



This test should not be used if the variances of the two samples are not the same.

17.1.3.5 Non-parametric test for means: Wilcoxon signed ranks test

The Wilcoxon signed ranks test assesses whether the means of samples from two variables that are paired are statistically different from each other. In BioNumerics, an obvious example of *paired samples* is comparing two different characters within a given set of entries. This test does not require that the variables are normally distributed. The null-hypothesis is that the two samples have the same mean values. Assume the sample observations are x_i and y_i ($i = 1, \dots, n$). The absolute values of the differences of these observations $|d_i| = |x_i - y_i|$ are ranked (zero values are eliminated from the analysis). As a first step, these ranks are assigned to rank variables R_i . Afterwards, these R_i get the sign of corresponding d_i . These two steps turn the R_i into ranks of positive or negative differences. The *sum of ranks of positive differences* (sum of all positive R_i) and the *sum of ranks of negative differences* (absolute value of the sum of all negative R_i) are determined and the smallest of these sums is called the *Wilcoxon T test statistic*.

If the null-hypothesis holds, the expected value for T is

$$\frac{n(n-1)}{4}$$

(with n the number of pairs of observations), while the expected standard deviation on T is

$$\sqrt{\frac{n(n-1)(2n+1)}{24}}$$

Hence, if the null-hypothesis holds and under certain conditions (see note below) the statistic defined as

$$\frac{T - \frac{n(n-1)}{4}}{\sqrt{\frac{n(n-1)(2n+1)}{24}}}$$

approximately follows a normal distribution. The p -value gives the probability that the statistic is at least as high as the observed one. If the p -value is low, the null hypothesis can be rejected. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

$$s_x = \frac{\sum_{i=1}^n (x_i - \langle x \rangle)^2}{n}$$

The values for the parameters can be found in the test report (see Figure 17.1.6). How such a chart and report can be created is explained in 17.1.4.

Wilcoxon signed ranks (paired samples)

Sum of ranks of positive differences= 26.0

Sum of ranks of negative differences= 1102.0

P value= 0.000000 (Normal approximation)

Significance= 100.0000%

Figure 17.1.6: Example of a test report for the Wilcoxon signed ranks test applied to a scatter chart.



This test should not be used if the population distribution is not symmetric.



The approximation by using a normal distribution is only valid if the sample contains more than 20 observations.

17.1.3.6 T test for two independent groups

This parametric test assesses the difference of the mean values of a variable between two independent groups. The null-hypothesis is that the two groups have the same mean values. Assume the sample group observations are x_i ($i = 1, \dots, n$) and y_j , ($j = 1, \dots, m$), with $\langle x \rangle$ and $\langle y \rangle$ the respective *mean values* for the groups. The *pooled corrected standard deviation* is defined as

$$S_d = \sqrt{\frac{(\sum_{i=1}^n (x_i - \langle x \rangle)^2 + \sum_{j=1}^m (y_j - \langle y \rangle)^2) (\frac{1}{n} + \frac{1}{m})}{n + m + 2}}$$

A statistic is defined as

$$T = \frac{\langle x \rangle - \langle y \rangle}{S_d}$$

If the null-hypothesis is true and under certain conditions (see note below) this statistic follows a t distribution with $n + m - 2$ *degrees of freedom*. The *p*-value gives the probability that the statistic indeed has the observed value or higher. If the *p*-value is small, the null-hypothesis can be rejected. The *significance* of the test is calculated as the complement of the *p*-value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.7). How such a report can be created is explained in 17.1.4.

```
T test for mean value (independent samples)

Mean values:
  Ambiorix    90.500
  Perdrix   114.500
Pooled corrected standard deviation = 27.85804

T = -0.862 (16 degrees of freedom)
P value= 0.401684
Significance= 59.8316%
```

Figure 17.1.7: Example of a test report for a T test applied to an ANOVA plot with two categorical variables.



This test should not be used if the data points are not normally distributed.



This test should not be used if the variances of the two samples are not the same.

17.1.3.7 Mann–Whitney test

This non-parametric test assesses the difference of the median values of a variable between two independent groups. The null-hypothesis is that the two groups have the same median values. Assume the observations in the sample groups are x_i ($i = 1, \dots, n$) and y_j ($j = 1, \dots, m$). All observations are combined into one sample and are ranked. For each group, the *sum of ranks* is determined and the smallest of those sums is taken as the U statistic. If the null-hypothesis holds and under certain conditions (see note below) this statistic approximately follows a normal distribution with mean $nm/2$ and variance $nm(m + n + 1)/12$. The *p*-value

gives the probability that the statistic indeed has the observed value or higher. If the p -value is small, the null-hypothesis can be rejected. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.8). How such a report can be created is explained in 17.1.4.

```
Mann-Whitney test

Sum of ranks:
  Ambiorix    49.0
  Perdrix    122.0

P value= 0.453695 (Normal approximation)
Significance= 54.6305%
P value= 0.497000 (simulated)
Significance= 50.3000%
```

Figure 17.1.8: Example of a test report for a Mann-Whitney test applied to an ANOVA plot with two categorical variables.



This test should not be used if one of the groups has a small member size.

If the sample contains less than 30 observations, an alternative way for testing the null-hypothesis is offered by *Monte-Carlo simulations*. To do this, 10,000 samples with two groups of n and m randomly distributed observations are created. For each of these samples, a value for the statistic is obtained and is compared to the observed value. The p -value from the simulations is determined by the number of times the simulations give a larger value for the statistic than the value observed in the real sample. Also here, the *significance* is calculated as $s = 100(1 - p)$. The results for the simulated p -value and significance also appear in the test report.

17.1.3.8 Chi square test for equal category sizes

For a given set of categories, this test assesses whether there are significant differences in the category sizes (i.e. number of members per category). The null-hypothesis is that all categories have an equal member size.

If this null-hypothesis holds, the *expected average count per category* (N_e) can be calculated as the total number of entries divided by the number of categories, $N_e = \frac{N}{n}$, with N the total number of entries and n the number of categories. The χ^2 (*chi square*) statistic is calculated from the values for the expected average count (N_e) and the *observed entries per category* (N_{oi}),

$$\chi^2 = \sum_{i=1}^n \frac{(N_{oi} - N_e)^2}{N_e}$$

with n the number of categories.

If the null-hypothesis is true and under certain conditions (see the note below) this statistic approximately follows a χ^2 distribution with $n - 1$ *degrees of freedom*. The p -value that is returned gives the probability that the statistic is at least as high as the observed one. If the p -value is low, the null-hypothesis can be rejected. The *significance* s of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for these parameters can be found in the test report (Figure 17.1.9). How such a report can be created is explained in 17.1.4.



This test should not be used if the expected average count per category is too small. If this is the case, consider combining categories in order to increase the expected average count.

```

Chi square test for categorical data

Chi square: 7.383 (3 degrees of freedom)
P value= 0.060643
Significance= 93.9357%

Number of categories: 4
Expected average count per category: 11.75

```

Figure 17.1.9: Example of a test report for the chi square test for equal categorical sizes applied to a bar graph.

17.1.3.9 Simpson and Shannon–Weiner indices of diversity

A commonly asked question about a number of entries occurring in different categories, is how they are distributed. Two widely used coefficients to measure the diversity are the *Shannon-Weiner index of diversity* and *Simpson's index of diversity*. Both coefficients take into account the *diversity*, i.e. the number of categories present in the sampled population, as well as the *equitability*, i.e. the evenness of the distribution of entries over the different categories.

Simpson's index of diversity is defined as the probability that two consecutive entries will belong to different categories. Given K categories present in a sampled population of N entries, the probability of sampling category i twice consecutively is as follows (n_i is the number of entries in category i):

$$P_i = \frac{n_i(n_i - 1)}{N(N - 1)}.$$

The probability of sampling any two samples of the same category is given by $P = \sum_{i=1}^K P_i$. Hence, the probability D of sampling two different categories is $D = 1 - P$, which is Simpson's index of diversity.

For a sampled population of N entries belonging to K categories, the Shannon-Weiner index of diversity is calculated as follows (n_i is the number of entries in category i): $H = -\sum_{i=1}^K \frac{n_i}{N} \cdot \ln\left(\frac{n_i}{N}\right)$

```

Diversity indices

Simpson's index of diversity: 72.62%
Standard deviation: 2.88%
Shannon-Weiner index of diversity: 1.3013

```

Figure 17.1.10: Example of a test report for diversity indices.

17.1.3.10 Chi square test for contingency tables

A *contingency table* contains information on the association between two categorical variables. Each cell contains the number of members for a specific combination of row and column categories. For this kind of representation of the data, the obvious question is usually if the information contained in the rows and columns is correlated or not. The null-hypothesis is that there is no association between the rows and columns.

If the null-hypothesis is true, the expected count per cell can be calculated. Therefore, we need to know the total number of cells n in the table, $n = n_i n_j$ with n_i the number of rows and n_j the number of columns. The summed numbers of counts in each row and column are called the *marginal row counts* (e.g. $N_{\text{row } i}$ stands for the marginal row count of row i) and *marginal column counts* ($N_{\text{col } j}$). If there is no association between rows and columns, the expected cell count N_{ij} for a cell on row i and column j can be calculated as

$$n_{ij} = \frac{N_{\text{row}i} N_{\text{col}j}}{N}$$

with N the total number of entries.

Using these expected cell counts (n_{ij}) and the observed counts per cell (N_{oij}), a *chi square* statistic is calculated,

$$\chi^2 = \sum_{i=1, j=1}^{n_i, n_j} \frac{(N_{oij} - n_{ij})^2}{n_{ij}}$$

with n_i the number of rows and n_j the number of columns.

If the null-hypothesis is true and under certain conditions (see note below), this statistic approximately follows a chi square distribution with $N - n_i - n_j + 1$ *degrees of freedom*. The p -value that is returned gives the probability that the statistic is at least as high as the observed one. If the p -value is low, the null-hypothesis can be rejected. The *significance* s of the test can be calculated as the complement of the p -value, $s = 100(1 - p)$.

In case there is a significant association, its strength can be expressed using *Cramer's V*. The formula is

$$V = \sqrt{\frac{\chi^2}{N \min(n_i - 1, n_j - 1)}}$$

with χ^2 the value for the statistic, N the total number of entries, n_i the number of rows and n_j the number of columns. This gives a value between 0%, in case there is no association, and 100%, in case there is a perfect association. Cramer's V can be used to compare the strengths of different associations.

Values for the various parameters can be found in the test report (see Figure 17.1.11). The marginal column and row counts are expressed in absolute counts and relative to the total number of counts in the table.

```
Chi square contingency table test

Chi square: 8.336 (6 degrees of freedom)
P value= 0.214520
Significance= 78.5480%

Cramer's V: 29.78%

Total count: 47
Average cell count: 3.92

Marginal column counts:
    Ambiorix    23  48.94%
    Vercingetorix  8  17.02%
    Perdrix    16  34.04%

Marginal row counts:
    France    13  27.66%
    Belgium   11  23.40%
    Netherlands 18  38.30%
    Denmark    5  10.64%
```

Figure 17.1.11: Example of a test report for the chi square test for contingency tables.

The contingency table can be displayed showing the residuals for the cells. The *residual* r is a measure for the deviation from the expected number of counts in that cell and is calculated as

$$r = \frac{N_{oij} - n_{ij}}{\sqrt{n_{ij}}}$$

with N_{oij} the observed cell count and n_{ij} the expected cell count.



This test should not be used if the expected average count per category is less than 5. If this is the case, consider combining categories in order to increase the expected average count. In practice, this also means that there should be no empty rows or columns in the contingency table.

17.1.3.11 Parametric test for more than two groups: F test

The F test compares the means of a normally distributed quantitative variable over g groups (categories). The null-hypothesis is that all groups have the same mean. The group sizes are given by n_1, n_2, \dots, n_g , in total n observations for the complete sample. The j th observation in the i th group is denoted as x_{ij} . The sample *group means* are

$$\langle x \rangle_{\text{group } i} = \sum_{j=1}^{n_i} \frac{x_{ij}}{n_i}$$

with x_{ij} all observations within group i . The mean of all observations is

$$\langle x \rangle = \sum_{i=1}^g \frac{\langle x \rangle_{\text{group } i}}{g}$$

The *total sum of squares*,

$$SST = \sum_{i=1}^g \sum_{j=1}^{n_i} (x_{ij} - \langle x \rangle)^2$$

is a measure for the variation in the sample around the mean of all observations. The *sum of squares among groups*

$$SSA = \sum_{i=1}^g n_i \left(\langle x \rangle_{\text{group } i} - \langle x \rangle \right)^2$$

measures the variation among the group means. The *total within-group sum of squares*

$$SSW = \sum_{i=1}^g \sum_{j=1}^{n_i} \left(x_{ij} - \langle x \rangle_{\text{group } i} \right)^2$$

gives the variation in the sample within the groups. From the definitions it is clear that

$$SST = SSA + SSW$$

If the null-hypothesis holds and under certain conditions (see note below) the statistic

$$F = \frac{SSA(n-g)}{SSW(g-1)}$$

approximately follows an F-distribution with $g-1$ and $n-g$ *degrees of freedom*.

The p -value gives the probability that the statistic obtains a value at least as high as the observed one. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.12). How such a report can be created is explained in 17.1.4.

```
ANOVA test

SST=  905.748
SSA=  356.821
SSW=  548.927

F= 14.301 (2;44 degrees of freedom)
P value= 0.000016 (F approximation)
Significance= 99.9984%
P value= 0.000000 (Simulated)
Significance= 100.0000%

Group means:
  Ambiorix  21.930
 Vercingetorix 28.624
   Perdrix  26.592
```

Figure 17.1.12: Example of a test report for an F test applied to an ANOVA plot with more than two categorical variables.

In case the sample contains less than 30 observations, an alternative way for testing the null-hypothesis is offered by *Monte-Carlo simulations*. To do this, 10,000 samples with g groups and n_1, n_2, \dots, n_g randomly distributed observations in the groups are created. For each of these samples, a value for the F statistic is obtained and is compared to the observed value. The p -value from the simulations is determined by the number of times the simulations give a larger value for the F statistic than the value observed in the real sample. Also here, the *significance* is calculated as $s = 100(1 - p)$. The results for the simulated p -value and significance also appear in the test report.

17.1.3.12 Non-parametric test for more than two groups: Kruskal–Wallis test

The Kruskal-Wallis test compares the means of a quantitative variable over g groups (categories). The variable is not assumed to be normally distributed. The null-hypothesis is that all groups have the same median. The number of observations in the groups are given by n_1, n_2, \dots, n_g , with n the total number of observations. All observations are ranked, the rank for the j th observation in the i th group is denoted by R_{ij} and R_i stands for the *group rank sum* of group i .

A statistic is defined as:

$$H = \left[\frac{12}{n(n-1)} \sum_{i=1}^g \frac{R_i}{n_i} \right] - 3(n-1)$$

If the null-hypothesis holds and under certain conditions (see below) the statistic approximately follows a χ^2 distribution with $g - 1$ *degrees of freedom*.

The p -value gives the probability that the statistic obtains a value at least as high as the observed one. The *significance* of the test is calculated as the complement of the p -value, $s = 100(1 - p)$.

The values for the parameters can be found in the test report (see Figure 17.1.13). How such a report can be created is explained in 17.1.4.

If the sample contains less than 30 observations, an alternative way for testing the null-hypothesis is offered by *Monte-Carlo simulations*. To do this, 10,000 samples with g groups and n_1, n_2, \dots, n_g randomly distributed

```

Kruskal-Wallis test

H= 17.246 (2 degrees of freedom)
P value= 0.000180 (Chi square approximation)
  Significance= 99.9820%
P value= 0.000000 (simulated)
  Significance= 100.0000%

Group rank sums:
    Ambiorix 357.0
    Vercingetorix 262.0
    Perdrix 509.0

```

Figure 17.1.13: Example of a test report for the Kruskal-Wallis test applied to an ANOVA plot with more than two categorical variables.

observations in the groups are created. For each of these samples, a value for the H statistic is obtained and is compared to the observed value. The p -value from the simulations is determined by the number of times the simulations give a larger value for the H statistic than the one observed in the real sample. Also here, the *significance* is calculated as $s = 100(1 - p)$. The results for the simulated p -value and significance also appear in the test report.

17.1.4 Calculating a statistic on a chart

Statistics can be calculated on a chart in the *Charts and statistics* window. To calculate a statistic, select the required chart in the *Chart list* panel and select **Plot > Plot statistics...** (⚙️). This opens the *Create chart statistic* wizard (Figure 17.1.14).

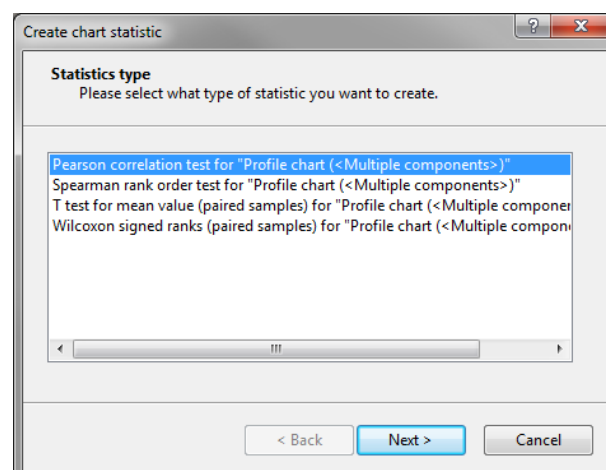


Figure 17.1.14: The *Statistics type* wizard page: Choose a chart statistic from the list.

The software automatically determines which test(s) are applicable to a given chart type, as outlined in Table 17.1.1. On the first page of the dialog box, the available statistics are listed, from which the required one should be selected.



In case the *Charts and statistics* window contains multiple compatible charts (i.e., charts that can be displayed together in the Chart area panel, see [14.5.6](#)), some statistics may apply to the combination of the charts. For example, when two profile charts are present, irrespective of which profile chart being selected in the Chart list, the following statistics are available:

- T test for mean value (paired samples) for first vs. second profile chart
- Pearson correlation test for first vs. second profile chart
- Spearman rank order test for first vs. second profile chart
- Wilcoxon signed ranks (paired samples) for first vs. second profile chart
- T test for mean value (independent samples) for first vs. second profile chart
- Mann-Whitney test for first vs. second profile chart
- Kolmogorov-Smirnov test for first profile chart
- Kolmogorov-Smirnov test for second profile chart

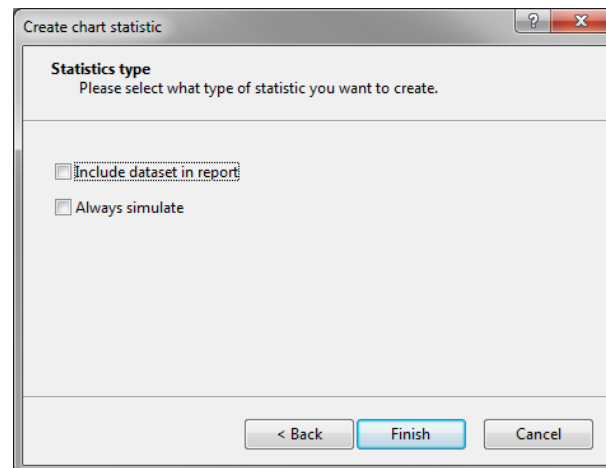


Figure 17.1.15: The *Parameters* wizard page.

The second page offers the additional choice ***Include dataset in report***. When this option is checked, the numerical data from which the chart and the statistic were calculated will be included in the *Statistics report* dialog box.

In case of non-parametric tests the option ***Always simulate*** is also shown. If this option is not checked, the software determines automatically how to calculate the p -value(s) from the data: in case of large data sets, the original data is used whereas in case of small data sets, a distribution is obtained using a kind of Monte Carlo simulation, for which the error is independent of the sample size. As such, the simulation method is more accurate for small sample sizes but takes longer to calculate. With the option ***Always simulate*** checked, the software is forced to always use a simulation on the data.

The resulting *Statistics report* dialog box contains the results of the statistical analysis in Rich Text Format. The background of the statistics and the resulting reports are described in [17.1.3](#).

The button **<Copy to clipboard>** in the *Statistics report* dialog box allows the report to be copied to the clipboard in editable Rich Text Format.

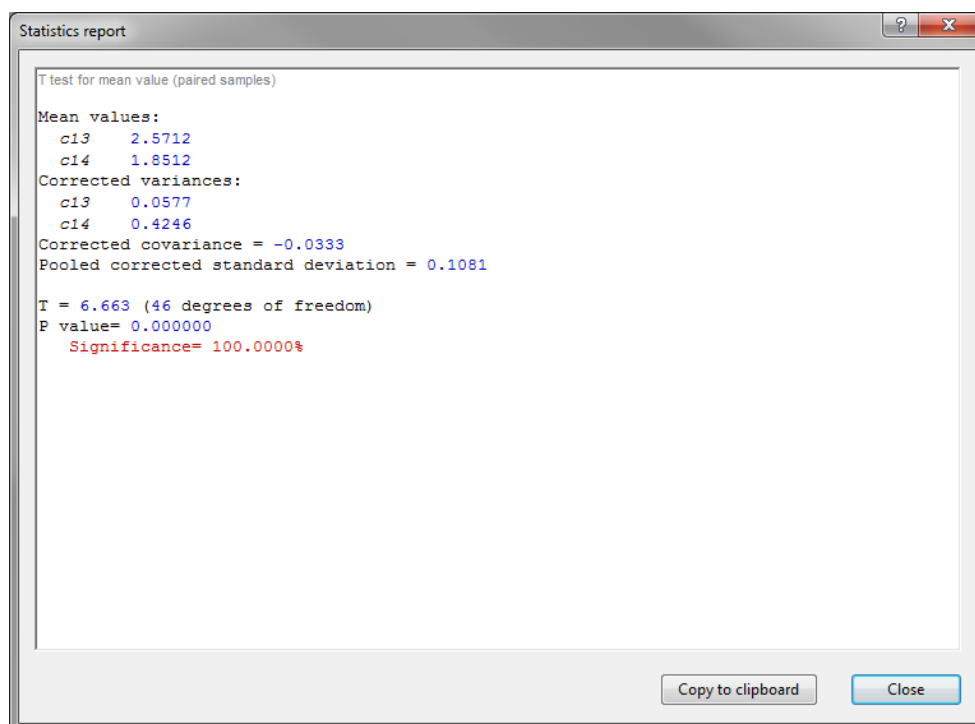


Figure 17.1.16: The *Statistics report* dialog box.

Chapter 17.2

Multivariate analysis of variance (MANOVA)

17.2.1 What is (M)ANOVA?

The central goal of an *analysis of variance* (ANOVA) is to investigate the differences between the means of a (set of) quantitative variable(s) across a number of groups. The main ingredients of an ANOVA are:

- **Explanatory variables:** one or more qualitative variables that determine the group membership of an entry. Therefore, explanatory variables sometimes are called *grouping variables*. An explanatory variable takes values from a finite set of possibilities, i.e. is categorical or binary.
- **Response variables:** one or more quantitative variables. A response variable is treated as a real number.

Depending on the number of explanatory variables, an ANOVA is called *one-way* if there is only one explanatory variable, *two-way* for two explanatory variables, and so on. In case the ANOVA is multi-way, groups are defined by all explanatory variables simultaneously, as in the example below.

Consider 2 explanatory variables:

- Explanatory variable E_1 with values A, B
- Explanatory variable E_2 with values 1, 2, 3

This yields 6 different groups:

- group 1: $E_1=A$ and $E_2=1$
- group 2: $E_1=A$ and $E_2=2$
- group 3: $E_1=A$ and $E_2=3$
- group 4: $E_1=B$ and $E_2=1$
- group 5: $E_1=B$ and $E_2=2$
- group 6: $E_1=B$ and $E_2=3$

With only one response variable, the ANOVA is called *univariate*, whereas for more than one response variable the ANOVA is called *multivariate* (or MANOVA). In the univariate situation, the group mean is a real number, whereas in the multivariate situation, the group mean is a vector of real numbers.

The statistical setting for ANOVA is the hypothesis testing context. For p groups, denoting the mean of group i by μ_i , the null hypothesis H_0 of equal means

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_p$$

is tested against the alternative hypothesis H_a that at least one pair of groups has different means

$$H_a : \exists i, j \in \{1, \dots, p\} : \mu_i \neq \mu_j$$

This pair of hypotheses is tested using a sample of observations of both the group membership (the explanatory variables) and the response variables. The result of an ANOVA is a p -value expressing the probability that the null hypothesis is true given the observed sample.

Of course, there is more to an analysis of the differences in means between groups than just a p -value. For instance, which pairs of groups have different means, and which pairs do not? How big are the differences between the groups? In a multivariate setting, what is the relationship between the multivariate test result and the individual response variables? Which response variables cause the difference in means, and which ones are just obscuring the result? Which combination of response variables yields the largest variability between the groups? Can groups be visually separated? In a multi-way setting, what is the strength of the interaction between the two explanatory variables, on all response variables simultaneously or individually? What is the size of this interaction? Can ANOVA be applied blindly to any set of explanatory and response variables? How can the test results be validated?

17.2.2 ANOVA prerequisites

17.2.2.1 Introduction

To be able to perform an ANOVA, the sample needs to contain enough variability. In a univariate setting, this means that the sample variance of the response variable should be nonzero. A zero variance can be detected easily: it simply means that every observation has the same value for the response variable. The multivariate equivalent is much more complicated. To be precise, the sample covariance matrix must be nonsingular. This means that the variance of any linear combination of response variables should be nonzero. Geometrically, any direction in the multidimensional space of the response variables needs to contain variability. Unfortunately, this is usually not easy to see from the sample data.

Consider the following sample of 3 response variables X , Y and Z

X	3	2	3	5	7
Y	6	7	5	8	5
Z	10	9	9	16	17

Clearly, all variables have nonzero variance. However, the sample covariance matrix

$$\begin{pmatrix} 3.2 & -0.4 & 6 \\ -0.4 & 1.36 & 0.56 \\ 6 & 0.56 & 12.56 \end{pmatrix}$$

is singular. A closer look at the data reveals that the combination $2X + Y - Z$ yields the following table of observations

$$2X + Y - Z \mid \begin{matrix} 2 & 2 & 2 & 2 & 2 \end{matrix}$$

and thus a direction with zero variance. When confronted with this lack of variability in the sample data, two courses of action can be taken.

17.2.2.2 Use variances only

In case all individual variables have nonzero variance, the covariance matrix can be made nonsingular by putting all covariances *between* variables to zero. This implies that combinations of variables are no longer considered, and that interactions between response variables are taken out of the analysis. In geometric terms, the MANOVA is restricted to what happens on the axes of the space of response variables. This is a very radical procedure that reduces the MANOVA to a simultaneous ANOVA of all the response variables individually. This approach allows the construction of canonical discriminants, but also there only the variances of the response variables are taken into account. In the above example, the covariance matrix reduces to

$$\begin{pmatrix} 3.2 & 0 & 0 \\ 0 & 1.36 & 0 \\ 0 & 0 & 12.56 \end{pmatrix},$$

which is nonsingular.

17.2.2.3 Use variable directions only

A much less radical solution to the lack of variability problem, is to use only that part of the data that actually contains enough variability. A principal component analysis of the original covariance matrix yields a set of components and associated eigenvalues (or variances). By selecting the largest components until a certain threshold of accumulated variance is reached, components with zero variance are excluded. The space of response variables is thus reduced to a subspace containing the bulk part of the variability. This subspace is then used to perform the ANOVA, and the results are translated back to the original space of response variables. Note that, in case the covariance matrix is nonsingular, applying this procedure with a variance threshold of 100% yields exactly the same results as using the untransformed data would. In this sense, if a true MANOVA is the goal, using only the variable directions always yields the best results that can be obtained from the data. In particular, this approach allows the construction of canonical discriminants using as much of the data as possible, avoiding the singularity problem of the covariance matrix.

17.2.3 Validating ANOVA test results

17.2.3.1 Introduction

The validity of the p -value produced by an ANOVA is based on two assumptions:

- **Normality within each group:** The sample data needs to be drawn from a normal distribution.
- **Homoscedasticity among groups:** The covariance structure needs to be the same.

There is only one way of knowing that these assumptions are satisfied, that is, when it is assured by the nature of the data. There are some statistical techniques to test these assumptions, but the p -values produced by these tests should be interpreted carefully. A high p -value does not prove the hypothesis of normality

or homoscedasticity, but simply states that, at some significance level, this hypothesis cannot be rejected. Although a lack of normality or homoscedasticity can lead to false conclusions in the ANOVA, a normality or homoscedasticity test merely serves as a indication of how trustworthy the p -values produced by the ANOVA are. Therefore, the importance of the tests discussed in this section should not be overestimated. Moreover, these tests only indicate the trustworthiness of the p -values themselves, and do not question the quantification of the differences between groups, nor the quantification of interactions, the canonical discriminants or their eigenvalues.

17.2.3.2 Testing normality

17.2.3.2.1 Available tests

For a single response variable, testing whether the samples are drawn from a normal distribution is a well-established problem, and numerous tests have proven their usefulness. Two tests stand out for their precision, graphical presentation and ease of interpretation: the *quantile-quantile plot* (or QQ-plot for short) and the *Kolmogorov-Smirnov test* (or KS-test for short).

17.2.3.2.2 Univariate QQ-plot

If the sample observations of a response variable are drawn from a certain theoretical distribution, the quantiles of the sample distribution estimate the corresponding quantiles of the theoretical distribution. When using the sample to estimate the parameters of the theoretical distribution, the pairs of points formed by a sample distribution quantile as first coordinate and the corresponding theoretical distribution quantile as second coordinate should all lie on a straight line through the origin at an angle of 45° (see Figure 17.2.1).

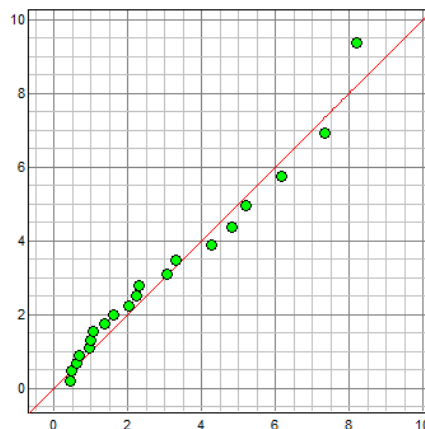


Figure 17.2.1: Example of a quantile-quantile plot (QQ-plot).

The correlation coefficient between the two sets of quantiles can be used to quantify collinearity. If the response variable is normally distributed, the points in the QQ-plot are scattered randomly around the straight line. Observations that lie extremely far from the straight line possibly are outliers. Note that removing an outlier observation from the sample is a very fundamental operation, and therefore the hint given by the QQ-plot should be verified thoroughly at the level of the data itself, or even of the experiment that led to the data.

17.2.3.2.3 Univariate KS-test

The KS-test tests the hypothesis that a response variable follows a particular statistical distribution. The resulting p -value is the probability of the sample data assuming that the response variable indeed follows

this distribution. Plotting the cumulative sample distribution against the theoretical distribution yields a graphical check of the hypothesis. As in the QQ-plot, the data points are expected to be scattered randomly around the cumulative distribution function (Figure 17.2.2). Outliers and deviations from normality will show on this plot as patterns in the scattered points.

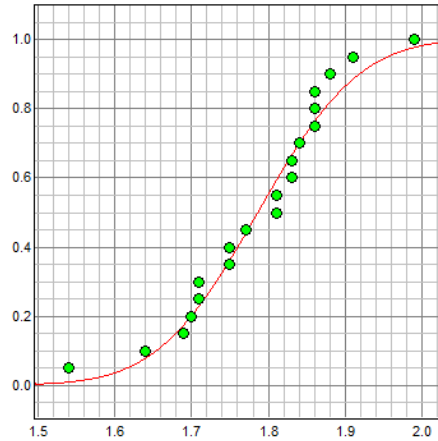


Figure 17.2.2: Example of a cumulative sample distribution as graphical verification of the Kolmogorov-Smirnov test (KS-test).

Normality testing in a multivariate setting is a much more complicated story. As multivariate (co)variance requires the data to contain variance in every direction of the space of response variables, multivariate normality requires the data to be normal in every direction of the space of response variables. As such, this requirement cannot be tested directly, since an infinite amount of individual tests would compromise the power of the overall test. Many tests have been designed to overcome this problem. The multivariate QQ-plot is a straightforward generalization of the univariate case. The best known of the multivariate normality tests are probably Mardia's skewness and kurtosis tests. Unfortunately, they are very error-prone in small-sized samples, have no graphical presentation and give no hints concerning outliers or structural deviations from normality. Since the univariate tests do have these interesting properties, a framework has been devised to lift the univariate techniques to a multivariate level.

17.2.3.2.4 Multivariate QQ-plot

A multivariate QQ-plot looks just the same and can be interpreted the same way as a univariate QQ-plot. The only difference is in the computation of the quantiles. Since quantiles are based on ordered statistics, and multivariate data cannot be ordered, a multivariate distribution has no quantiles.

The solution to this problem is to use the *Mahalanobis distance* M of each sample observation to the mean. This distance measure is nothing but an Euclidean distance that takes into account the covariance structure S of the data. In the univariate case, taking the Mahalanobis distance boils down to transforming the one response variable to a standard normal variable.

$$\text{Multivariate: } M(x) = \sqrt{(x - \bar{x})\Sigma^{-1}(x - \bar{x})^t}$$

$$\text{Univariate: } M(x) = \sqrt{(x - \bar{x})\sigma^{-1}(x - \bar{x})} \text{ or: } M(x) = \left| \frac{x - \bar{x}}{\sigma} \right|$$

For a vector of response variables that follows a multivariate normal distribution, the squared Mahalanobis distances follow a χ^2 distribution. A multivariate QQ-plot shows how well the ordered squared Mahalanobis distances of the sample align with the quantiles of the χ^2 distribution. As a χ^2 distribution in the univariate case is nothing but the square of a normal distribution, the multivariate QQ-plot is a straightforward generalization of the univariate case.

17.2.3.2.5 Multivariate KS-test

Although multivariate normality of a vector of response variables implies that each response variable individually is normally distributed, the converse is not true. Therefore, when testing for multivariate normality, it is not enough to test all response variables individually. As testing all combinations of response variables is statistically powerless, there is some sense in testing a limited number of meaningful directions. First, the individual response variables should be tested. Second, for a particular direction exhibiting non-normality to have some influence on the ANOVA, there should be a substantial amount of variance in this direction. Therefore, the individual principal components corresponding to nonzero eigenvalues are a second batch of directions to be tested for non-normality.

For a vector of k response variables, testing all response variable directions and all principal component directions with a univariate KS-test yields at most $2k$ p -values p_1, \dots, p_r . A conservative bound on the true p -value of the multivariate test would be $1 - (1 - \min(p_1, \dots, p_r))^r$.

However, this bound does not make full use of the situation at hand. Since p -values are what is left when all the distribution information is taken out of the test statistic, a p -value can be seen as a random number drawn from a uniform distribution between 0 and 1. When drawing 100 random numbers from this uniform distribution, it is to be expected that one of them will be smaller than 0.01. In other words, when performing k independent tests, some low p -values are to be expected. What is needed here to reject the hypothesis of multivariate normality, is an abnormally large number of low p -values. For instance, for $k = 100$, finding 7 p -values below 0.001 would be alarming, but so would be finding 37 p -values below 0.05. Formalizing, for any bound $b \in [0, 1]$, the number of p -values smaller than b is expected to have a binomial distribution with k repeats and chance of success equal to b . A more elaborate bound on the true p -value of the multivariate normality test is then given by

$$\min_{b \in [0,1]} P(X \geq \#\{l | l \in \{l_1, \dots, l_k\} \text{ and } l < b\})$$

where $X \sim \text{Binomial}(k, b)$.

In the MANOVA window, this p -value is referred to as the *Family-wise corrected p-value*.

17.2.3.3 Testing homoscedasticity

The problem of testing whether the groups in an ANOVA have the same covariance structure is less complicated than the normality tests. Bartlett's variance test - both in a univariate and a multivariate setting - directly yields a p -value assessing the probability of the sample data assuming that all groups have the same covariance structure. The multivariate version of Bartlett's variance test is acceptably reliable. As a more detailed analysis of the variance issue, the same procedure can be followed as with normality tests, replacing the KS-test with Bartlett's univariate variance test. This yields a more detailed image of which response variables or principal components have different variances across the groups, and what their influence is on the overall p -value.

17.2.4 Interpreting ANOVA test results

17.2.4.1 Introduction

An analysis of variance comes in many flavors, and the questions that need to be answered are often very different. This section lists the most important ones and shows how the answer should be interpreted at a theoretical level. All conclusions drawn from the results of the statistical tests, and from their p -values in particular, are based on the normality and homoscedasticity assumptions, and should be validated accordingly. On the other hand, only the statistical tests are based on these assumptions, and therefore conclusions

based on values such as mean vectors and differences between groups or canonical discriminants remain valid even when the assumptions are not met.

17.2.4.2 Do the groups have different means?

This is the main question in an analysis of variance, and the starting point for further exploration.

- **Null hypothesis:** All groups have the same mean.
- **Alternative hypothesis:** There exists a pair of groups with different mean.
- **Test statistic:** Fisher's F -test (2 degrees of freedom, approximate)
- **p -value:** A low value rejects the hypothesis of equal means.

Upon rejection of the null hypothesis, the question of which groups cause this null hypothesis to fail immediately pops up. This calls for a new ANOVA based on the same explanatory and response variables, but with observations from two groups only.

To know which individual response variables cause the difference in mean, an ANOVA can be performed with the same explanatory variables and on the same set of observations, but with only one single response variable. In the case of one response variable, the F -test used for the ANOVA is exact. The differences between the means of the groups can be computed relative to the overall mean of the sample. This quantification of the differences is independent of the normality and homoscedasticity assumptions.

17.2.4.3 In a two-way ANOVA, how important are the two explanatory variables?

Two aspects of the two-way setting can be tested. First, the importance of each explanatory variable alone is tested by performing an ANOVA with the same response variables and observations as before, but with only one explanatory variable. This assesses the importance of the single explanatory variable as a grouping variable. Second, the *interaction* of the two variables can be tested by comparing the full ANOVA model with a model without the interaction.

This importance testing of the explanatory variables and of their interaction can be repeated for each response variable individually.

17.2.4.4 What does the separation of groups look like?

A *canonical discriminant* analysis determines those orthogonal directions in the multidimensional space of response variables that optimally separate the groups. Discriminants are ordered relative to their importance, which can be tested as well. Whereas the canonical discriminants themselves do not require the response variables to be normally distributed and homoscedastic across the groups, their importance tests do. The observations can be visualized in *pairwise scatter plots*.


17.2.5 Example data set

17.2.5.1 Sample data

To illustrate the full possibilities of the *MANOVA* window, a separate data set is made available via the Applied Maths website (<http://www.applied-maths.com/download/sample-data>, click on "MANOVA sample data").

The sample data set describes an experiment in which the optimal conditions for growth and product formation were determined for a bacterial strain in a broth with a certain carbon source. Two different nitrogen sources were evaluated (yeast extract and ammonium chloride) and three different incubation temperatures (30, 35 and 37°C). These represent the explanatory variables (or grouping variables) in the MANOVA. The experiments were done in 24-well micro titer plates of which four wells were not inoculated. Therefore, 20 replicates are available for each condition. Bacterial growth was evaluated after 24 hours using dry cell weight (in mg/ml) and optical density at 600 nm. The yield of a desired fermentation product was determined using gas chromatography and expressed in mM. The data are available as a tab-delimited text file, designated MANOVA.TXT. It is recommended to create a new, separate database.

17.2.5.2 Creating a new database

5.1 Press the  button in the BioNumerics *BioNumerics Startup* window to enter the *New database* wizard.

5.2 Enter a name for the database, and press <Next>.

A new dialog box pops up, prompting for the type of database (see Figure 17.2.3).

5.3 Since we want to create a new database to demonstrate the features of the plugin, leave the default option selected and press <Next>.

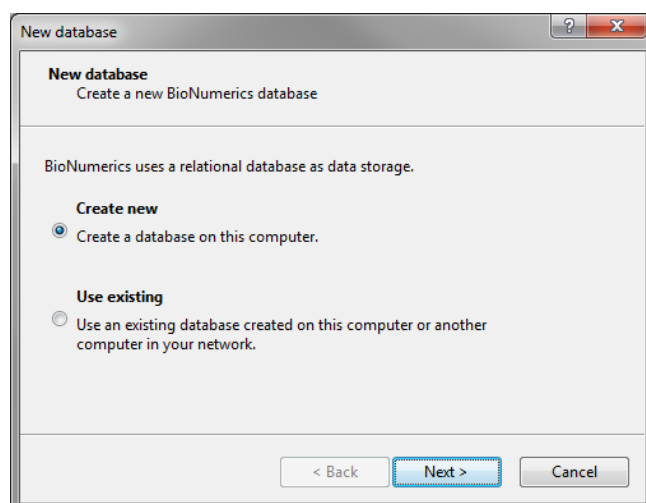


Figure 17.2.3: The *New database* wizard page.

A new dialog box pops up, prompting for the database engine (see Figure 17.2.4).

5.4 Leave the default option selected and press <Next>.


5.5 Press <Finish> to complete the setup of the new database.

The *Plugins* dialog box appears.

5.6 Press <Proceed> to close the *Plugins* dialog box and to continue to the *Main* window.

17.2.5.3 Creating a new character type experiment

In the new empty database, we will first create a new character type experiment:

5.7 In the *Main* window, highlight the *Experiment types* panel and select *Edit > Create new object...* (.

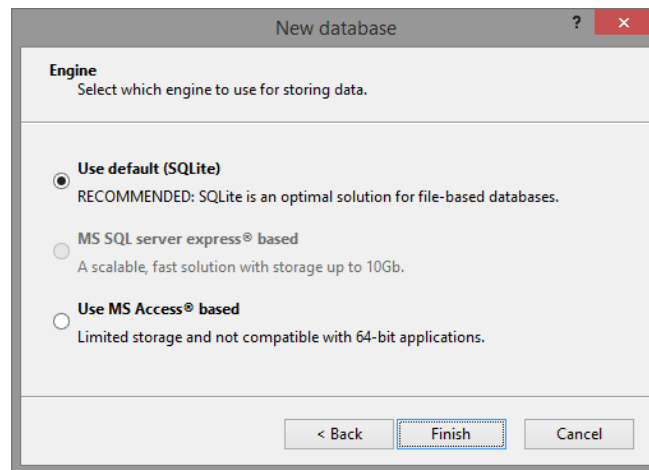


Figure 17.2.4: The *Database engine* wizard page.

5.8 Highlight "Character type" and press **<OK>**.

5.9 Enter a name for the character type, e.g. "Fermentations" and press **<Next>**.

5.10 Check *Numerical values*, set two decimal digits and press **<Next>**.

5.11 Set 100 as *Max value* and press **<Finish>** to complete the creation of the new character type.

The new character type is added to the *Experiment types* panel.

17.2.5.4 Importing data

5.12 In the *Main* window, select **File > Import...** (📁, **Ctrl+I**).

5.13 In the *Import* dialog box, expand *Character type data*, highlight *Import fields and characters (text file)* and press **<Import>**.

5.14 Browse for the MANOVA.TXT file and press **<Next>**.

5.15 Highlight the row that corresponds to "Experiment" and press **<Edit destination>**.

5.16 In the *Edit data destination* dialog box, highlight "Key" and press **<OK>**.

5.17 Highlight the rows that correspond to "Temperature", "N-source" and "Replica" using the **Shift**-key, and press **<Edit destination>**.

5.18 In the *Edit data destination* dialog box, highlight "Entry info field" and press **<OK>**.

5.19 In the *Create new* dialog box that appears, leave the default names unaltered and press **<OK>**. Confirm the action.

5.20 Highlight the three remaining file fields and press **<Edit destination>** again.

5.21 In the *Edit data destination* dialog box, highlight "Fermentations" (located under "Character value") and press **<OK>**.

5.22 In the *Create new* dialog box that appears, leave the default names unaltered and press **<OK>**. Confirm the action.

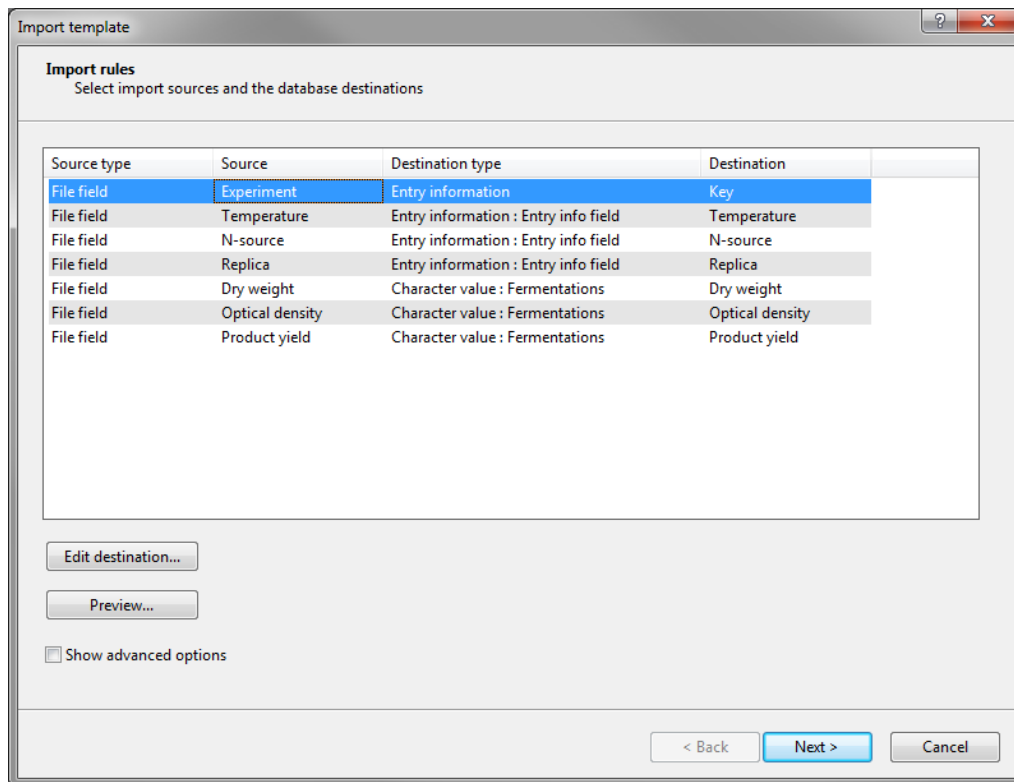


Figure 17.2.5: The *Import rules* dialog box after setting up the template for import of the example "MANOVA.TXT" file.

The *Import rules* dialog box should now look like in Figure 17.2.5.

5.23 Press <*Next*> and <*Finish*>.

5.24 Specify a template name and press <*OK*>.

The template is automatically selected.

5.25 Press <*Next*> and <*Finish*> to import the information in the database.

The example data are imported in the database (see Figure 17.2.6), with the growth conditions (= explanatory variables) as information fields and the growth parameters (= response variables) as character values.

5.26 Double-click on the **Fermentations** experiment type in the *Experiment types* panel.

The *Character type* window opens (see Figure 17.2.7).

5.27 Close the *Character type* window.

17.2.6 Performing a MANOVA

The *MANOVA* window will be illustrated using the bacterial growth data as described in previous paragraph.

6.1 In the *Main* window with the database created in 17.2.5.2 loaded, select all entries (e.g. using the **Ctrl+A** keyboard shortcut).

6.2 Create a new comparison by highlighting the *Comparisons* panel in the *Main* window and selecting *Edit* > *Create new object...* (+).

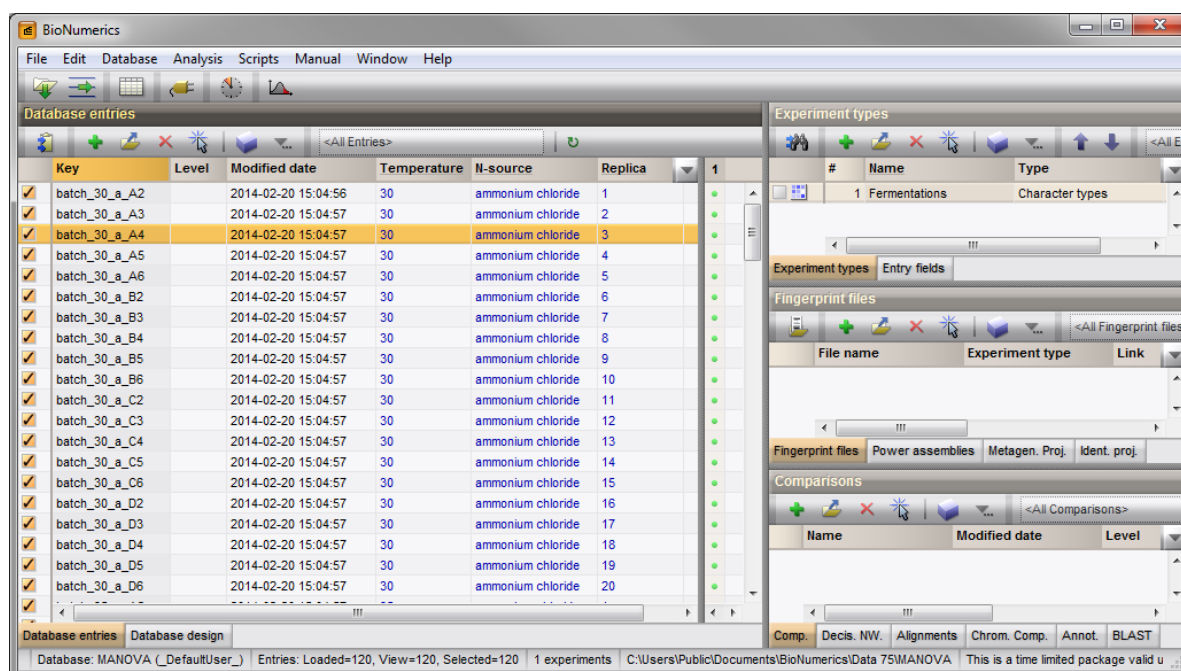


Figure 17.2.6: The *Main* window after import of the data.

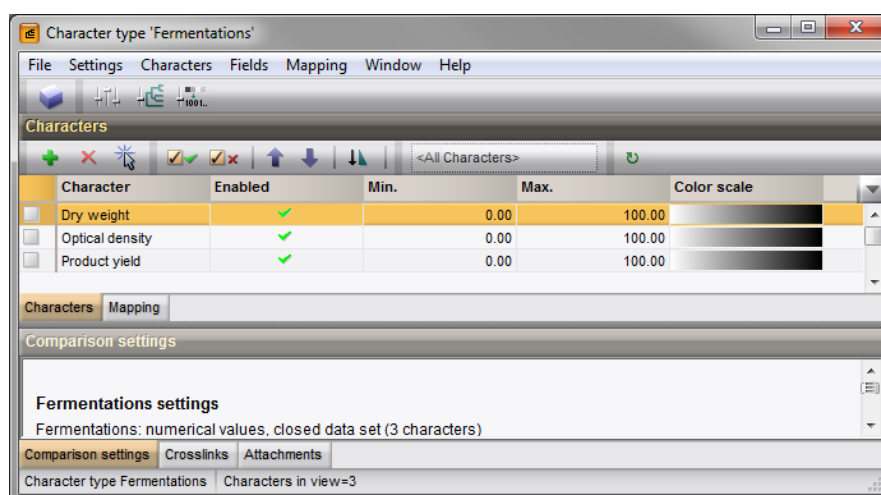





Figure 17.2.7: The character type experiment with three characters.

6.3 Select **File** > **Save** (, **Ctrl+S**). Enter e.g. *All fermentations* as comparison name and press <OK>.

6.4 Click on the  next to the experiment name **Fermentations** in the *Experiments* panel to display the fermentation data in the *Experiment data* panel.

Initially, the character values are displayed as colors according to the color scale defined for each character.

6.5 Select **Characters** > **Show values** () to show the corresponding character values for all entries in the comparison.

6.6 To start a MANOVA, select **Statistics** > **MANOVA...** or press the  button and select **MANOVA** from the menu that appears.

The *Manova analysis* dialog box pops up (Figure 17.2.8).

From the *Manova analysis* dialog box, the components of a (multivariate) analysis of variance can be se-

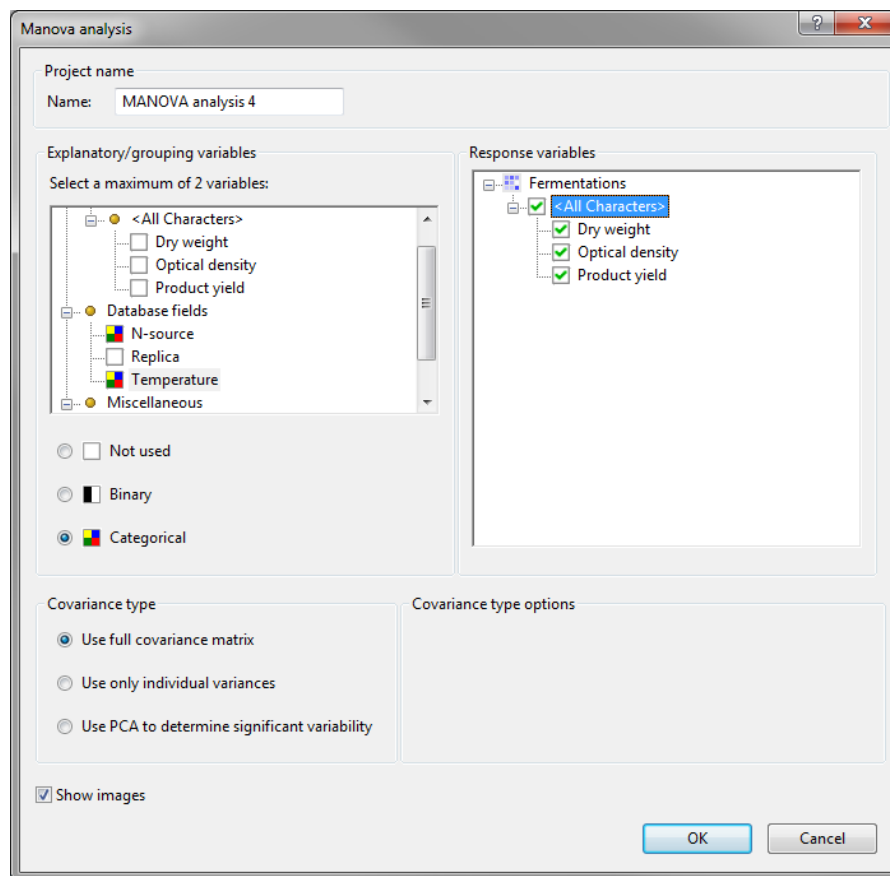


Figure 17.2.8: The *Manova analysis* dialog box.

lected.

A **Project name** can be entered in the corresponding text box. The default name suggested is "MANOVA analysis", followed by a serial number. The MANOVA analysis will be listed under this name in the *Analyses* panel of the *Comparison* window.



If a comparison is closed and re-opened, a MANOVA analysis is recalculated when it is called from the *Analyses* panel. Therefore, if meanwhile changes were made to the underlying data or to the list of entries in the comparison, these changes will be reflected in the MANOVA analysis.

The hierarchical representation in the *Explanatory/grouping variables panel* lists all character types and composite data sets for which characters are available, all user-defined **Database fields** and a **Miscellaneous** category containing only **Comparison groups**. From each of these categories, **Explanatory variables** can be selected by highlighting the variable in the tree and checking the **Binary** (■) or **Categorical** (■) option below the tree. The check box next to the highlighted variable is updated according to the selected option. A highlighted explanatory variable can be removed again by checking the **Not used** option. Note that maximum two explanatory variables can be selected and only the characters that are enabled in the *Character type* window (see 6.1.2) are listed and can be selected.

Since **Response variables** should be quantitative values only the characters types available in the database are listed in the *Response variables panel*. Only characters that are enabled in the *Character type* window are listed and can be selected. **Response variables** can be selected by clicking the ballot box next to the character name. Selected characters are marked by a checked ballot box (☑) and can be unselected in the same way. To select all characters from an experiment aspect, enable the option **<Aspect name>**.

When more than one **Response variable** is used, the **Covariance type** can be selected. By default, **Use full covariance matrix** is selected, which performs a "true" MANOVA that takes any linear combination

of response variables into account. *Use only individual variances* reduces the MANOVA analysis to a simultaneous ANOVA of each of the response variables. With *Use PCA to determine significant variability*, principal component analysis is used to reduce the space of response variables to a subspace with the bulk of variance. When the latter option is checked, the *Significance threshold (relative to total variance)* can be entered. For more information regarding when to select these options, see [17.2.2](#).

To save calculation time in case of large data sets, the option *Show images* can be unchecked. In this case, images such as histograms, bar graphs and plots will not be displayed.

For the example data, we will calculate a *two-way* (with two explanatory variables) *multivariate* analysis of variance (MANOVA):

6.7 In the *Manova analysis* dialog box, select "N-source" and "Temperature" as categorical *Explanatory variables* and select all characters of the **Fermentations** character type as *Response variables* (see [Figure 17.2.8](#)).

6.8 Leave *Use full covariance matrix* checked and press <OK> to calculate a MANOVA analysis with the specified components.



The MANOVA window pops up ([Figure 17.2.9](#)).

17.2.7 The MANOVA window



17.2.7.1 General features


The MANOVA window ([Figure 17.2.9](#)) consists of four different pages, displayed in tabbed view:



- *Exploratory data analysis*
- *Testing model assumptions*
- *Analysis of variance*
- *Canonical discriminants*


One can navigate from one page to the other with the  and  buttons (menu commands *Edit > Go to next page* or *Edit > Go to previous page*) or by clicking on the corresponding tab.

Each page in the MANOVA window contains a number of sections, represented in a hierarchical view. Sections can be collapsed and their content hidden by clicking on the small "-" (minus) sign that precedes the section name.

The currently displayed page in the MANOVA window can be exported with *File > Export current page* or by pressing the  button. All pages can be exported at once with *File > Export all pages* or . In both cases, the *Save report* dialog box pops up ([Figure 17.2.10](#)).

A report (without images) can be exported as a flat text file or an HTML-formatted text. A path and file name can be entered directly or browsed for by pressing the  button. When <OK> is pressed, the report pops up in Notepad (in case of a flat text file) or in the default browser (in case of an HTML file).

The currently displayed page in the MANOVA window can be printed with *File > Print current page* or by pressing the  button. All pages can be printed at once with *File > Print all pages* or . A printer can be selected and print settings specified from the dialog box that appears.

The display settings for the MANOVA window can be accessed with *Edit > Display settings* or the  button. The *Display settings* dialog box pops up ([Figure 17.2.11](#)).

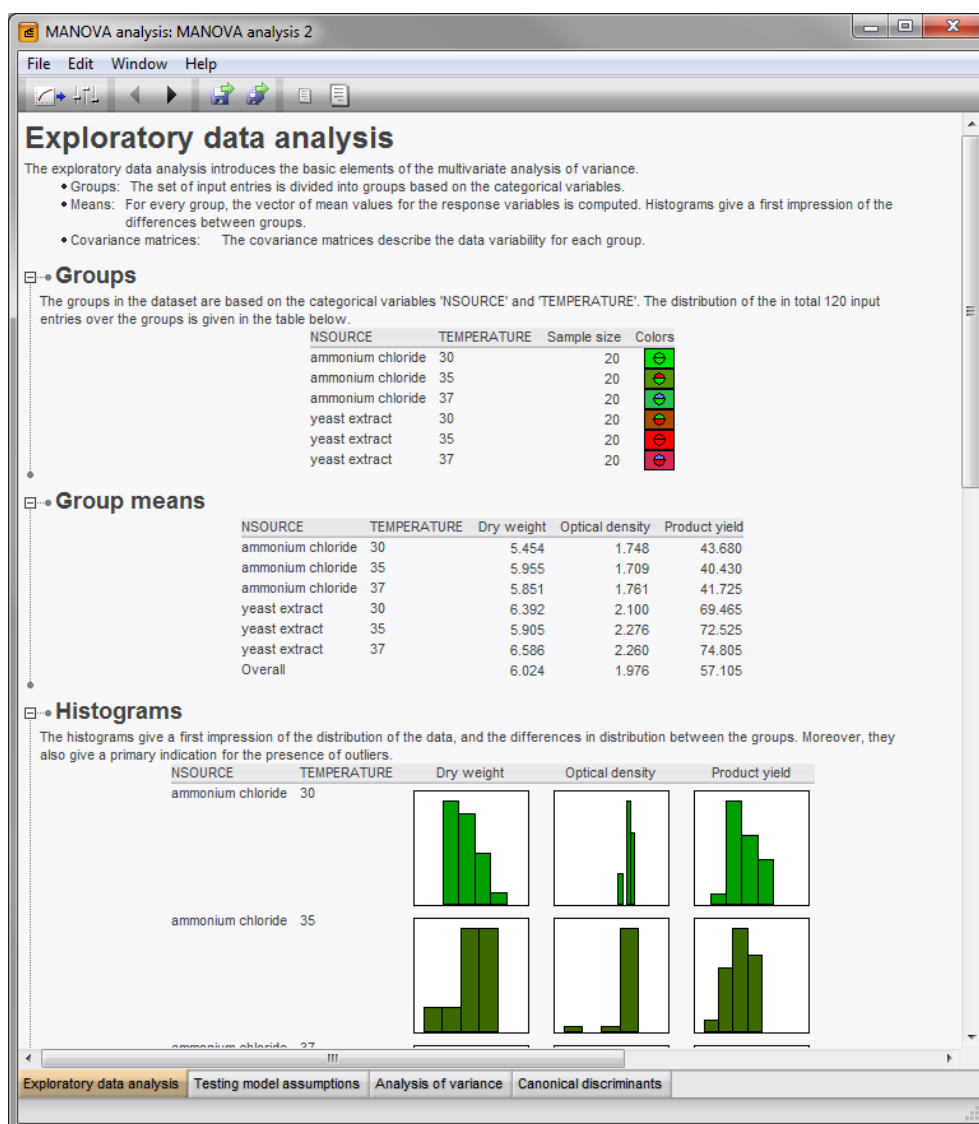


Figure 17.2.9: The MANOVA window, *Exploratory data analysis* page.

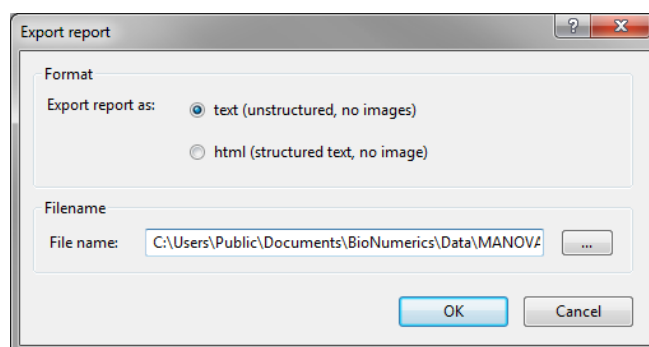


Figure 17.2.10: The *Save report* dialog box.

Under *Significance levels*, the *Significance level for model assessment tests* and *Significance level for ANOVA tests* can be entered as a percentage. This affects the color coding of reported *p*-values in the MANOVA window (see below). The default value for both significance levels is 5%. The smaller the value is set, the sooner *p*-values will be reported in a color different from green.

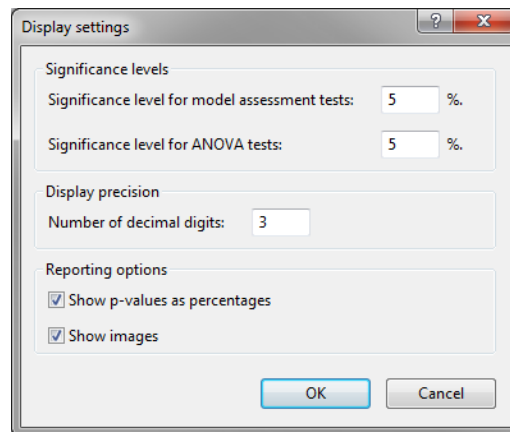


Figure 17.2.11: The *Display settings* dialog box for the *MANOVA* window.

The *Number of decimal digits* for all reported values can be set (default number is 3).

By default, *Show p-values as percentages* is checked. If this option is unchecked, *p*-values are expressed as fractions ($0 \leq p \leq 1$).

When *Show images* (checked by default) is unchecked, images such as histograms, bar graphs, and plots are not displayed in the *MANOVA* window.

In the *Exploratory data analysis page* of the *MANOVA* window (Figure 17.2.9), the basic elements of a *MANOVA* are introduced.

17.2.7.2 Groups

Groups in an ANOVA analysis are based on the explanatory variables used. Remember that, in a multi-way setting, groups are defined by all explanatory variables simultaneously (see 17.2.1). An overview of the groups in the data set, the group sizes and corresponding colors used in all graphical representations throughout the *MANOVA* window is presented in a table.

Right-click the groups table or any other table in the *MANOVA* window to pop up a floating menu with options to copy the table's content to the clipboard (as tab-delimited text or HTML table) and to sort the table in ascending or descending order according to a certain information field.

Click within the groups table or any other table in the *MANOVA* window; as a result, the table pops up in its own window (Figure 17.2.12).

NSOURCE	TEMPERATURE	Sample size	Colors
ammonium chloride	30	20	(image)
ammonium chloride	35	20	(image)
ammonium chloride	37	20	(image)
yeast extract	30	20	(image)
yeast extract	35	20	(image)
yeast extract	37	20	(image)

Figure 17.2.12: The *MANOVA Table* window, showing the groups in the ANOVA analysis in a grid view.

From this window, the table can be copied as flat text with *Edit > Copy table (text)* or in HTML format with

Edit > Copy table (HTML).

By pressing the column properties button (📄) in the table header, options become available to search the table, copy the table content to the Windows clipboard, set the active fields and to change field positions (see 3.2.7 for more information).

Select **File > Exit** in the *MANOVA Table* window to close it.

17.2.7.3 Group means

For each of the groups in the ANOVA analysis, the mean value for each of the response variables is shown in a table. This gives a first indication how different or similar the group means are. If you click within the table, the table pops up in its own window.

17.2.7.4 Histograms

The group means as such do not give an impression of the spread of the data. Therefore, histograms are drawn for each group - response variable combination. Per response variable, the same X-axis scale is used (global scale). Therefore, the distribution of the values can be visually compared over the different groups.

If you click on a histogram, a *MANOVA Image* window pops up for the corresponding group (Figure 17.2.13).

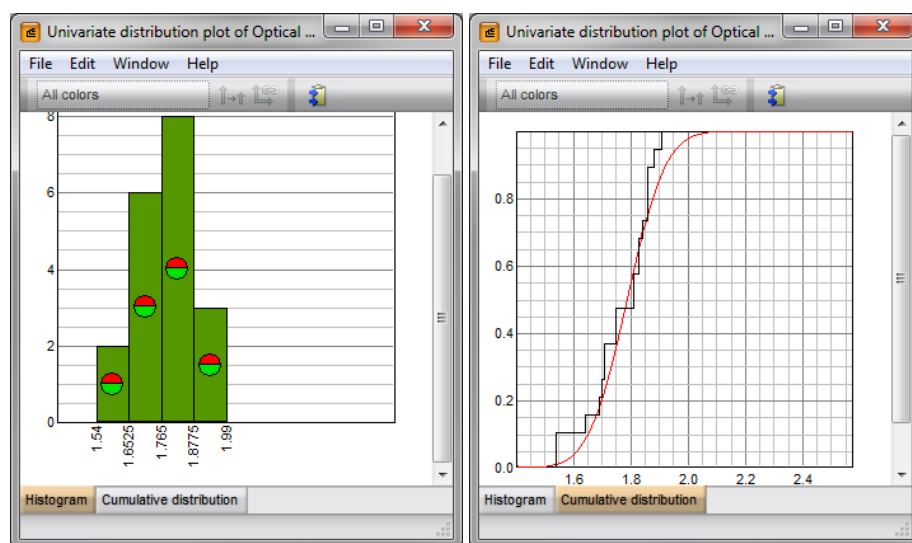


Figure 17.2.13: The *MANOVA Image* window for an ANOVA group, with the *Histogram* panel and the *Cumulative distribution* panel activated.

The *MANOVA Image* window contains a *Histogram* panel and a *Cumulative distribution* panel, which can be activated by clicking on the corresponding tab.

The colors used in the histogram can be set:


- 7.1 Select **Edit > Color scheme** and select the desired group color to be displayed from the menu. Alternatively, use the pull-down menu in the toolbar (all colors) to set the colors.

By default, the same global X-axis scale is used in the *MANOVA Image* window as used for the histograms in the *Exploratory data analysis* page. Alternatively, a scale optimized for the values in that specific ANOVA group can be used:

- 7.2 Select **Edit > Use global scale** or press the (📏) button to toggle between the global scale and the scale optimized for the group.

A bar in the *Histogram panel* can be selected by holding the **Shift** key and drawing a rectangle with the mouse. Entries that fall in the category represented by that bar are selected in the database. Conversely, a selection of entries in the database is reflected in the histogram; selected entries are displayed in a darker shade (see Figure 17.2.13). This feature can be used, e.g. to identify possible outliers.

The histogram or cumulative distribution plot can be exported:

- 7.3 Select **Edit > Copy image** or press the  button. This pops up the *Copy image* dialog box (Figure 17.2.14).

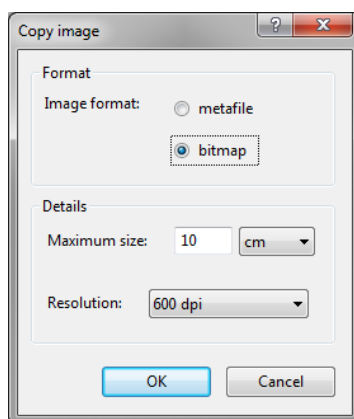



Figure 17.2.14: The *Copy image* dialog box.

For the *Image format*, the choice is offered between *metafile* (for import in Windows applications) and *bitmap* (for other applications). In case *bitmap* is checked, the *Maximum size* and the *Resolution* can be set.

Press <OK> to export the image in the specified format to the Windows clipboard. It can now be pasted from the clipboard into other applications.



A number of options and buttons are inactive (grayed out), since they are not applicable in the *MANOVA Image* window when histograms are plotted: **Edit > Show labels** to display the entry keys and **Edit > Maintain aspect ratio** or the  button. These functions are relevant in other plots, such as the Mahalanobis distance QQ-plot and the discriminants plots (see below).

- 7.4 Select **File > Exit** to close the *MANOVA Image* window.

17.2.7.5 Covariance matrices

For each of the groups in the ANOVA analysis, a matrix of covariances is shown. Clicking on any of the matrices will pop up the covariance matrix in its own window.


Following observations can be made for the example data on the *Exploratory data analysis page* (see Figure 17.2.9):

In the **Groups** section, it can be seen that the example data set contains in total six groups: three temperature groups (30, 35, and 37°C), multiplied with the two N-source groups (ammonium chloride and yeast extract). Each group contains 20 samples.

The **Group means** are reported for each of the response variables Dry weight, Optical density and Product yield.

From the **Histograms**, we can learn that the within-group variability of the Dry weight measurements is high in relation to the differences between the group means. Therefore, it will be hard to obtain significant

conclusions from this variable. Further examination of the histograms reveals a few possible outliers, for example the low value in the Optical density histogram for the 35°C - ammonium chloride group.

7.5 In the *MANOVA* window, press the  button to go to the next page: the *Testing model assumptions* page.

In the *Testing model assumptions* page of the *MANOVA* window (Figure 17.2.15), the concordance of the data with the model assumptions is verified. Two assumptions are made: **normality** and **homoscedasticity** (see 17.2.3).

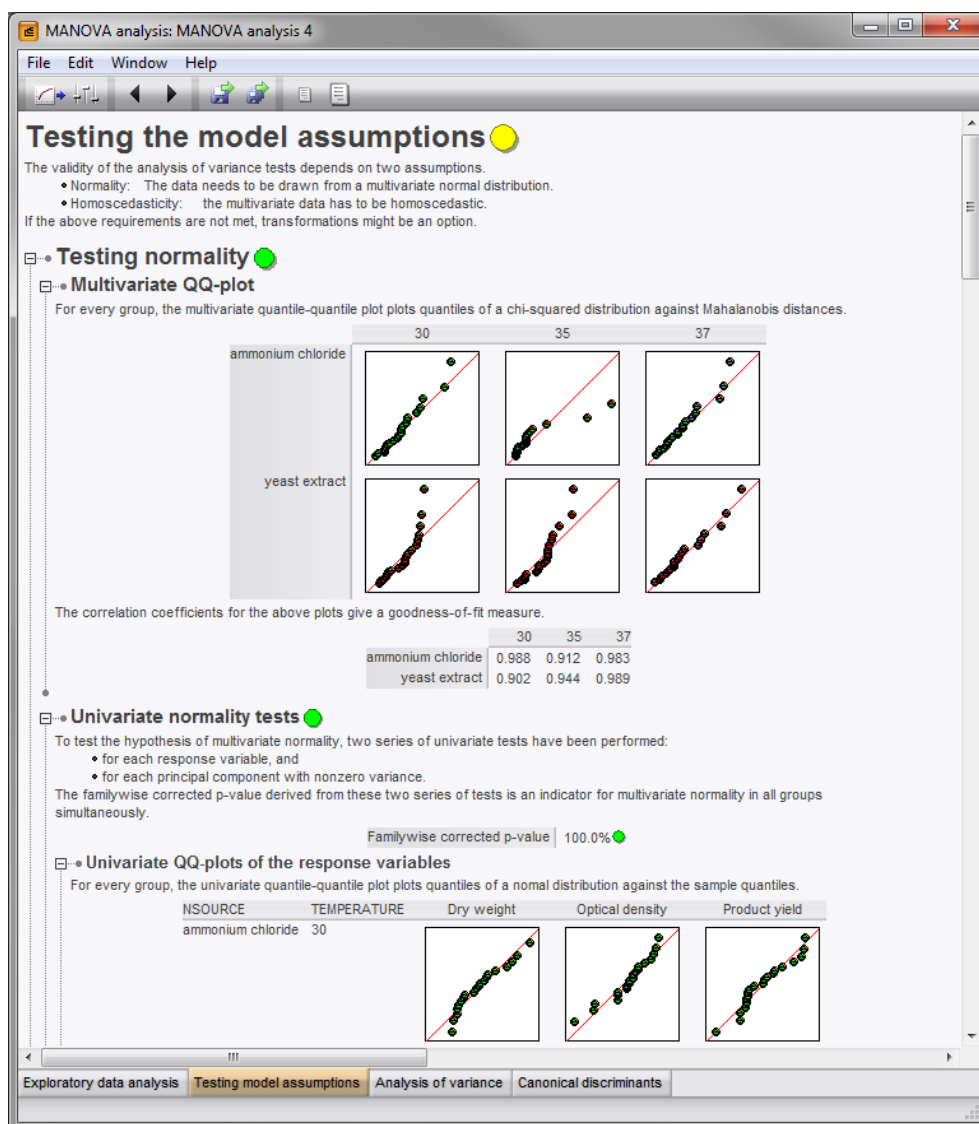


Figure 17.2.15: The *MANOVA* window, *Testing model assumptions* page.

For the test of the model assumptions as a whole and for each of the individual tests, a *p*-value is displayed as a colored dot to the right of the test name. The color of the dot ranges from green (high probability that the assumption is correct) over yellow to red (low probability that the assumption is correct). The color is determined according to the significance levels entered in the *Display settings* dialog box (see Figure 17.2.11). When one hovers over the dot with the mouse, the actual *p*-value is shown.

17.2.7.6 Testing normality

In a *multivariate* MANOVA, this section contains two sub-sections: *Multivariate QQ-plot* and *Univariate normality tests*.

In the **Multivariate QQ-plot**, quantiles of a χ^2 distribution are plotted against the Mahalanobis distance for each group in the ANOVA (see 17.2.3.2 for more information). Optimally, the points should all lie on a straight line through the origin at an angle of 45° .

Click on any plot to display the Mahalanobis distance QQ-plot in the *MANOVA Image* window. Entries can be selected in the plot by holding the **Shift**-key and dragging a rectangle with the mouse. A selected dot (entry) is encircled in orange. The selection is synchronized with the database: any selection made in the *MANOVA Image* window will be reflected in other BioNumerics windows and vice versa.

The correlation coefficients for the Mahalanobis distance QQ-plots are a measure for the collinearity of the data and are reported in a table. Clicking within the table will open the table in its own window.

In the **Univariate normality tests** section, two series of univariate tests (univariate QQ-plots and Kolmogorov-Smirnov tests) are performed: on each of the response variables and on each of the principal components with nonzero variance.

In the univariate QQ-plots, the quantiles of a normal distribution are plotted against the sample quantiles for each of the groups. Clicking on any of the plots opens the univariate QQ-plot in its own window. The correlation coefficients for the univariate QQ-plots are a measure for the collinearity of the data and are reported in a table. Clicking within the table will open the table in its own window.

In the Kolmogorov-Smirnov tests, the data points in the univariate quantile plots are expected to be scattered randomly around the cumulative distribution function. Clicking on any of the plots opens the univariate quantile plot in its own window. The associated *p*-values are reported in a table, which can be opened in its own window by clicking on it.

Obviously, in a *univariate* MANOVA, univariate QQ-plots and Kolmogorov-Smirnov tests are calculated on the response variable only.

17.2.7.7 Testing homoscedasticity

Homoscedasticity (or homogeneity of variance) is verified using Bartlett's variance test. In this test, the null hypothesis, that all population variances are equal, is tested against the alternative hypothesis that at least two are different. The *p*-value produced by this test is therefore the probability that all groups have the same covariance structure.

To test *multivariate* homoscedasticity, two types of tests are performed:

Multivariate equal variance test: the degrees of freedom, F-statistic and *p*-value are reported for Bartlett's variance test on the complete covariance matrix.

Univariate equal variance tests are performed on each of the response variables and on each of the principal components with nonzero variance. In both cases, the *p*-value is the probability that the groups have the same covariance structure. In addition, a family-wise corrected *p*-value (see 17.2.3) is reported as an overall measure of the homoscedasticity assumption.

In a *univariate* MANOVA, Bartlett's equal variance test is performed on the response variable directly.



Following observations can be made for the example data on the *Testing model assumptions page* (see Figure 17.2.15):

Outliers can be detected in the 35°C - ammonium chloride group, by looking at e.g. the Univariate QQ-plot of Optical density or at the corresponding correlation coefficient in the table. The same observation can be made in the Univariate quantile plot of Optical density and the *p*-value of the corresponding KS-test.


Before we can decide to omit or include this outlier, we should have a closer look at the actual data:

7.6 Open the Univariate quantile plot of Optical density for the 35°C - ammonium chloride group by clicking on this plot in the *Testing model assumptions page*. The plot opens in its own window.

7.7 In the univariate quantile plot, shown in the *MANOVA Image* window, select the outlier by holding the **Shift** key and dragging a rectangle around it with the mouse. The entry is encircled in orange.

7.8 Go to the underlying *Comparison* window and click on the eye button () next to the **fermentations** character type in the *Experiments* panel to display the character data in the *Experiment data* panel. Show the character values by clicking the  button.


By looking at the actual character values for the selected entry, it becomes clear that all three values (Dry weight, Optical density, and Product yield) are abnormally low for that specific reaction. Most probably, something went wrong during inoculation of that well, so the reaction can safely be omitted from the analysis.

7.9 Press the  button to remove the selected entry from the comparison.


7.10 Save and close the *Comparison* window.

7.11 Open the **All fermentations** comparison again and call the MANOVA analysis from the *Analyses* panel.

In the **Groups** section on the *Exploratory data analysis page*, it can be seen that the 35°C - ammonium chloride group now contains only 19 samples.

7.12 Press the  button to go to the *Testing model assumptions page*.

It can be seen that the *p*-values for the tests of the model assumptions are now much better.

7.13 Press the  button to go to the next page: the *Analysis of variance page*.

In the *Analysis of variance page* of the *MANOVA* window (Figure 17.2.16), the actual analysis of variance is done.

17.2.7.8 Test hypothesis

The null hypothesis H_0 that all groups have the same mean is tested against the alternative hypothesis H_a : there exists a pair of groups with a different mean.

17.2.7.9 Analysis of variance

In a *multivariate* MANOVA, the null hypothesis is assessed using *Wilk's lambda likelihood ratio F-test*. The degrees of freedom, *F*-statistic, and *p*-value are reported. The sum of squares matrices are reported in a separate section (see below).

In a *univariate* MANOVA, the null hypothesis is assessed using *Fisher's F-test*. In addition to the degrees of freedom, *F*-statistic and *p*-value, the among-groups and within-groups sum of squares are reported.

17.2.7.10 Variable and interaction significance

To test the significance of the explanatory variables, i.e. to know which groups cause the null hypothesis to fail, each of the explanatory variables individually is tested with a one-way ANOVA. Again, the degrees of freedom, *F*-statistic, and *p*-value are reported.

The significance of the interaction of the two explanatory variables is investigated by comparing models with and without interaction (see 17.2.4). A low *p*-value means that the explanatory variables are likely to behave independent of each other.

In the example data set, interaction would occur e.g. if at a given temperature one nitrogen source would be preferably metabolized.

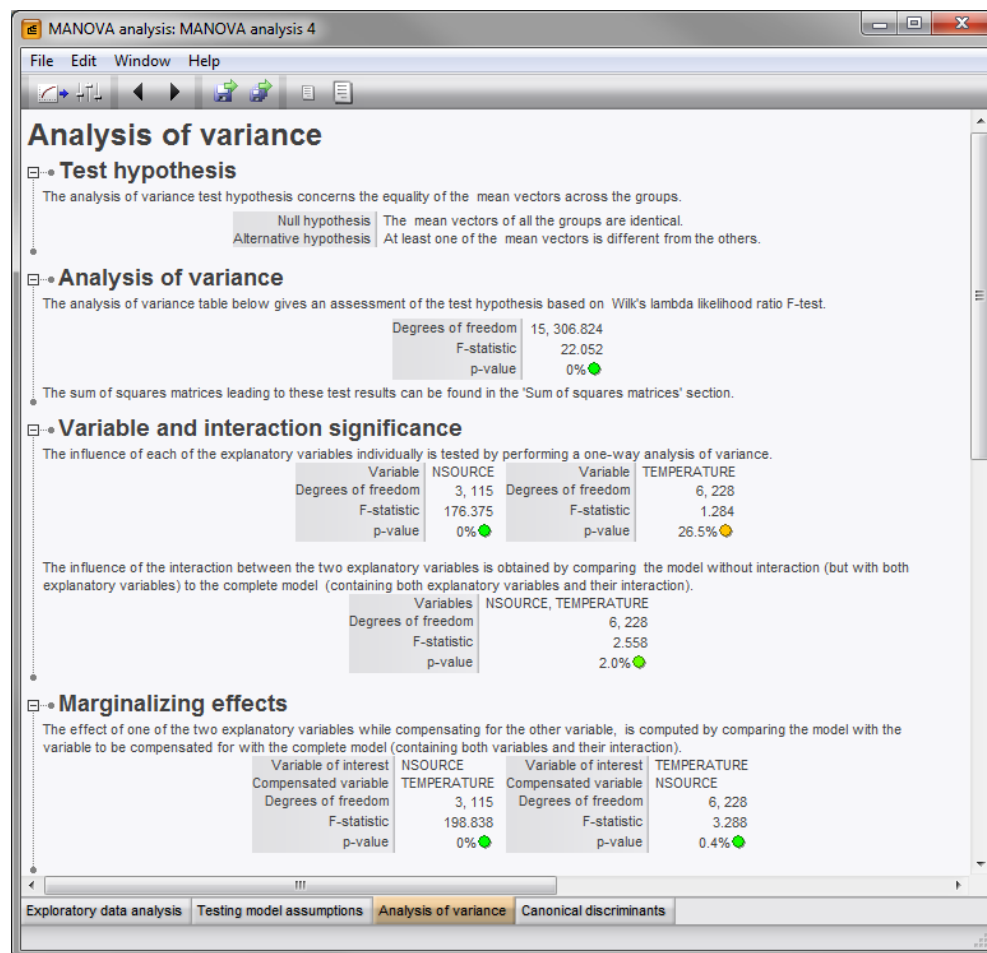


Figure 17.2.16: The MANOVA window, Analysis of variance page.

17.2.7.11 Marginalizing effects

The effect of the first explanatory variable (the variable of interest) is evaluated, while the effect of the second (the compensated variable) is filtered out. The reciprocal test (second explanatory variable as variable of interest and first one compensated for) is performed as well. A low p -value in this case means that the variable of interest is still significant after filtering out the effect of the other variable.

17.2.7.12 Univariate analyses

In a *multivariate* MANOVA, the same tests as described in 17.2.7.9 and 17.2.7.10 are repeated in a univariate setting, so for each of the response variables individually.

17.2.7.13 Sum of squares matrices

The sums of squares correspond to the variability within and between groups and are used to calculate the multivariate test statistics. In a *multivariate* MANOVA, two matrices are reported: the **Among groups sum of squares matrix** and the **Within-groups sum of squares matrix**.


The following observations can be made for the example data on the *Analysis of variance* page (see Figure 17.2.16):

The low p -value in the *Analysis of variance* section indicates that the null hypothesis can be rejected: there

is at least one pair of groups with a different mean, for at least one of the response variables.

From the one-way ANOVA done in the **Variable and interaction significance** section, it can be concluded that the nitrogen source is more significant than the temperature. Furthermore, both variables are likely to behave independently of each other.

From the **Univariate analyses**, it can be seen that the N-source is a significant explanatory variable according to the Optical density and Product yield measurements. A possible relation exists between Temperature and Optical density and between N-source and Dry weight. Most probably no relation exists between Temperature and Product yield. Furthermore, it can be concluded that Dry weight does not have much predictive value. This is in concordance with the observation of high within-group variability and the relative small differences between group means made earlier in the *Exploratory data analysis page*.

7.14 In the *MANOVA* window, press the  button to go to the next page: the *Canonical discriminants page*.

In the *Canonical discriminants page* of the *MANOVA* window (Figure 17.2.17), a canonical discriminant analysis is performed. Discriminant analysis is very similar to PCA, in a sense that it tries to maximize the difference between groups by making linear combinations of the original directions. However, while PCA calculates the best discriminating components without reference to groups, discriminant analysis calculates the best discriminating components for user-defined groups, i.e. formed by the explanatory variables. The best discriminating components are called discriminants.

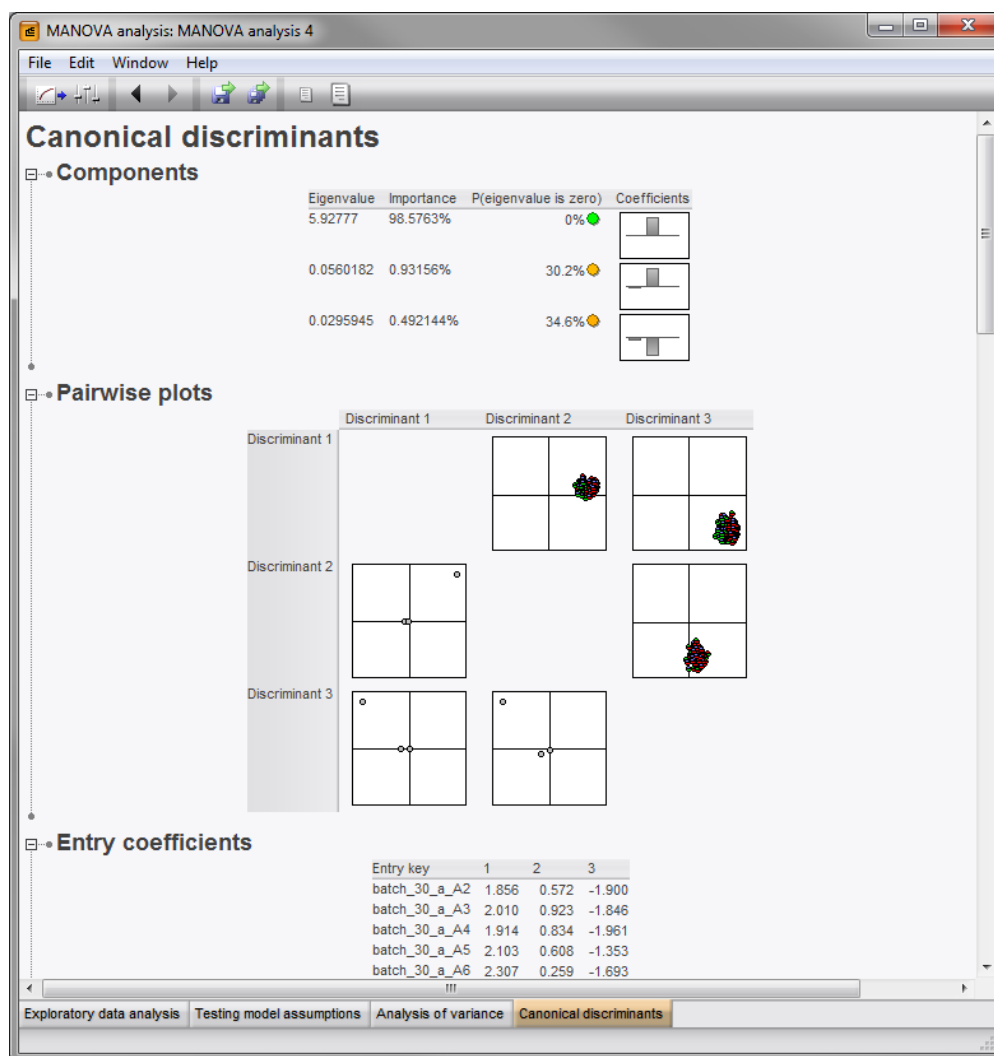


Figure 17.2.17: The *MANOVA* window, *Canonical discriminants page*.

17.2.7.14 Components

All components (or discriminants) with an eigenvalue greater than zero are listed. Components are listed in order of importance: the first discriminant is always the most important, i.e. it accounts for most of the discrimination; the second one is the second most important, etc.. The p -value is the probability that the eigenvalue of the component is zero, i.e. that a random subdivision in groups would yield the same degree of discrimination. The coefficients for each of the components are shown in bar graphs (the actual numbers are displayed under **Component coefficients**; see 17.2.7.17). Clicking on any of them opens the bar graph in its own window (see Figure 17.2.18).

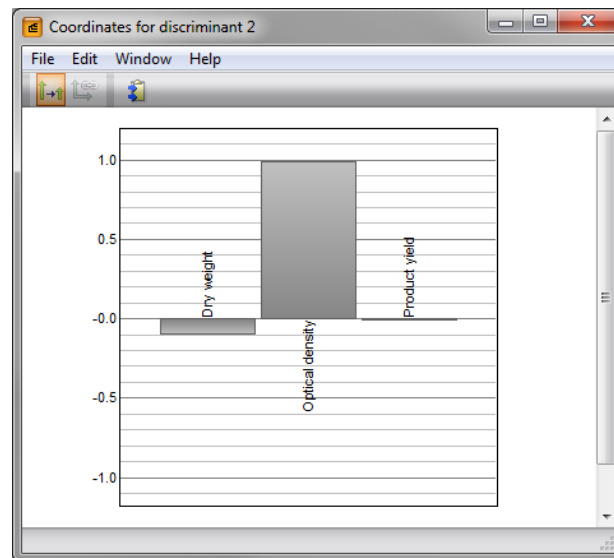


Figure 17.2.18: Discriminant coefficients plotted in a bar graph.

17.2.7.15 Pairwise plots

The entry plots (top right part of the matrix) in the *MANOVA* window map the entries on the corresponding discriminants. Note that the X-axis always accounts for most of the discrimination. Clicking on an entry plot opens it in its own *MANOVA Image* window (see Figure 17.2.19).

In this window, entries can be selected by holding the **Shift**-key and dragging a rectangle with the mouse. A selected dot (entry) is encircled in orange. The selection is synchronized with the database: any selection made in the *MANOVA Image* window will be reflected in other BioNumerics windows and vice versa. The other functionality of the *MANOVA Image* window is discussed under 17.2.7.4.

The character plots in the *MANOVA* window (left lower part of the matrix) map the characters on the corresponding discriminants. The more distant a character occurs from the center along a discriminant axis, the more it contributes to that discriminant. Clicking on a character plot opens it in its own *MANOVA Image* window (see Figure 17.2.20).

In this window, characters can be selected by holding the **Shift**-key and dragging a rectangle with the mouse. A selected dot (character) is encircled in orange. The selection is synchronized with the character type: any selection made in the *MANOVA Image* window will be reflected in the *Character type* window and vice versa.

17.2.7.16 Entry coefficients

The entry coefficients are listed for all discriminants with an eigenvalue greater than zero.

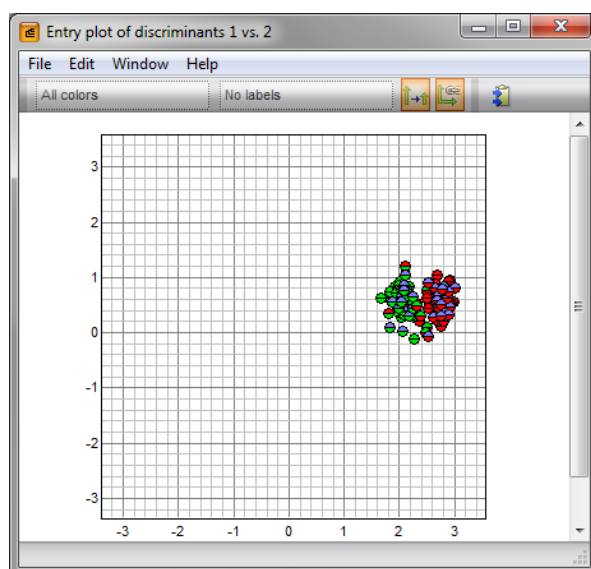


Figure 17.2.19: Entry plot in the *MANOVA Image* window.

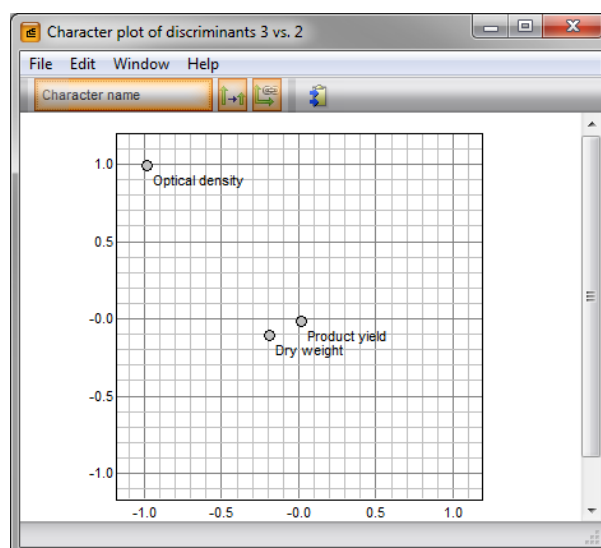


Figure 17.2.20: Character plot in the *MANOVA Image* window.

17.2.7.17 Component coefficients

The component (discriminant) coefficients are listed for each response variable.

Following observations can be made for the example data on the *Canonical discriminants page* (see Figure 17.2.17):

In the **Components** section, it can be seen that the most important component (or discriminant), i.e. the one that discriminates best between the groups formed by the explanatory variables, relies most on the Optical density response variable.

In the **Pairwise plots** where Discriminant 1 is included (1 vs. 2 and 1 vs. 3), a clear separation of the entries according to the N-source is obtained (see Figure 17.2.21). In the plot of Discriminant 2 vs. 3, no separation is obtained.

The MANOVA analysis performed so far can be the starting point for further (M)ANOVA analyses to explore the example data set, e.g. given a certain N-source (either ammonium chloride or yeast extract),

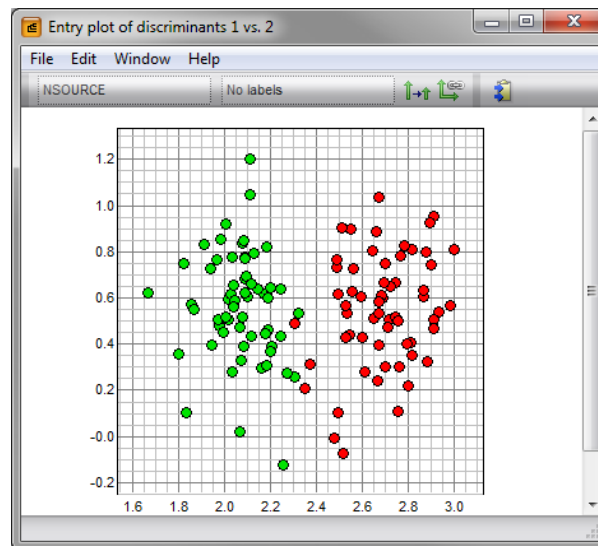


Figure 17.2.21: Discriminant 1 plotted versus Discriminant 2 in the example data, colored by N-source.

is there an effect of the temperature on the optical density? What happens if the incubation temperatures are categorized differently, e.g. low (30°C) and high (35 and 37°C) temperature? Can we actually learn something from the dry weight or should the dry weight measurements just be omitted from the setup of future experiments?

Chapter 17.3

Partition mapping

17.3.1 Introduction and definitions

Classification is very important in (micro)biology. By an organism's membership of a class or group, we can deduce some of its characteristics. Organisms can be classified according to different criteria, such as exposure of surface antigens (serotyping), susceptibility to bacteriophages (phage typing), banding patterns obtained with various fingerprint techniques (PFGE, RFLP, ...), etc. When a set of organisms is classified by more than one system, the question arises as to how the classification systems relate to each other. This is what the partition mapping tool is designed for: to compare the outcome of two classification systems. A *partition* is thereby defined as a division of a collection of entries into a number of *classes* (groups or types). In Figure 17.3.1, two partitions are compared: a first partition with classes I, II and III and a second partition with classes A, B, C and D.

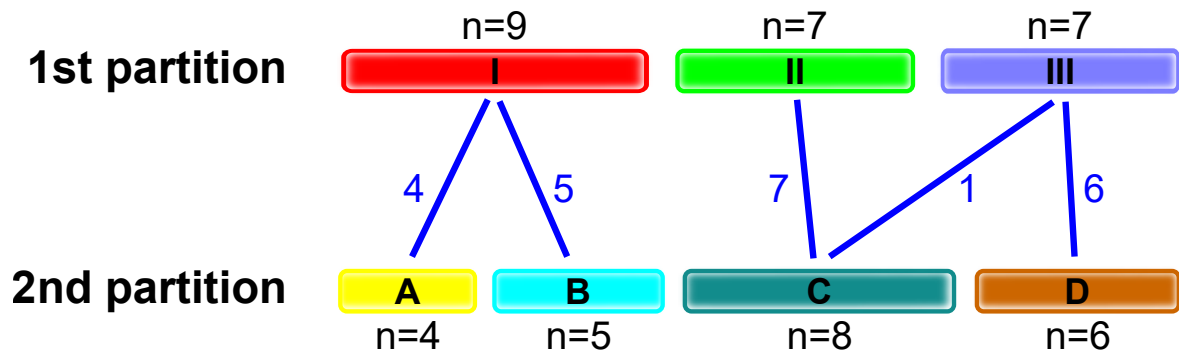


Figure 17.3.1: Example partitions. The first partition contains classes I, II and III and the second partition classes A, B, C and D. The number of entries (n) is indicated for each class.

The class membership frequencies can be represented conveniently in a *contingency table*. For the example partition mapping of Figure 17.3.1, the contingency table is given in Figure 17.3.2.

	A	B	C	D
I	4	5	0	0
II	0	0	7	0
III	0	0	1	6

Figure 17.3.2: Contingency table for the two example partitions in Figure 17.3.1.

A *mapping rule* between two partitions is a relation between a number of classes from the first partition and a number of classes from the second partition. In the contingency table, mapping rules can be visualized as non-overlapping rectangles, in the sense that the rectangles should have neither rows or columns in common. A *partition mapping* is a set of mapping rules that are mutually exclusive (no class from the first or the second partition appears in two different rules) and that completely covers both partitions (every class in either the first or the second partition appears in at least one mapping rule). It forms a *model* that tries to predict the actually observed contingency table as truthfully as possible. When based on a sufficiently high number of observations, the partition mapping can have *predictive power*. A partition mapping can also be interpreted as a new partition or classification of the entries, which summarizes the two input partitions. It is therefore also indicated as *result partition*. Because of violations (see below), the *forward* mapping, i.e. the mapping from the first partition to the result partition is different from the *reverse* mapping, i.e. the mapping from the second partition to the result partition.

It is important to realize that a partition mapping can manifest two kinds of *mapping errors*, since a set of rules can either be too precise or too general. When the set of rules is too precise, *violations* exist. In the contingency table, violations are represented by non-empty cells that are not covered by any rule. This situation is illustrated for the two example partitions in Figure 17.3.3, with the corresponding contingency table shown in Figure 17.3.4. In this partition mapping, class I from the first partition is mapped on classes A and B from the second partition (Rule1), class II is mapped on class C (Rule2) and class III on class D (Rule3). This leaves III;C as a violation, i.e. not covered by any rule.

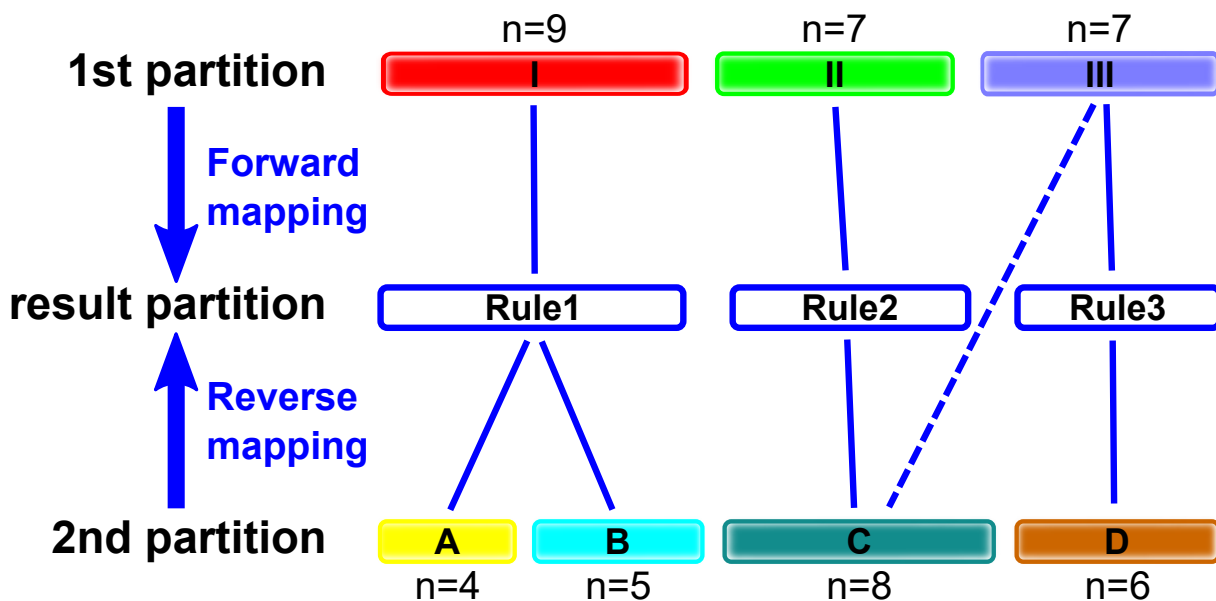


Figure 17.3.3: Possible mapping for the two example partitions. The violation III;C is shown as a dashed line.

	A	B	C	D
I	4	5	0	0
II	0	0	7	0
III	0	0	1	6

Figure 17.3.4: Contingency table for the partition mapping in Figure 17.3.3. Rules are indicated as continuous rectangles. Cells that confirm the mapping rules are green, while the violation is indicated in red.

A second kind of mapping error occurs when the set of rules is too general and results in *missing entries*. In this case, the model (i.e. the partition mapping) predicts some entries, while they are not actually observed. In the contingency table, missing entries are seen as one or more empty cells within the rectangles formed by the rules. This situation is illustrated for the two example partitions in Figure 17.3.6, with the corresponding contingency table shown in Figure 17.3.5. In this partition mapping, class I from the first partition is mapped on classes A and B from the second partition (Rule1) and classes II and III are mapped on classes C and D (Rule2). This means that cell II;D - although empty - is covered by Rule2.

	A	B	C	D
I	4	5	0	0
II	0	0	7	0
III	0	0	1	6

Figure 17.3.5: Contingency table for the partition mapping in Figure 17.3.6. Rules are indicated as continuous rectangles. Cells that confirm the mapping rules are green, while missing entries are displayed in yellow.

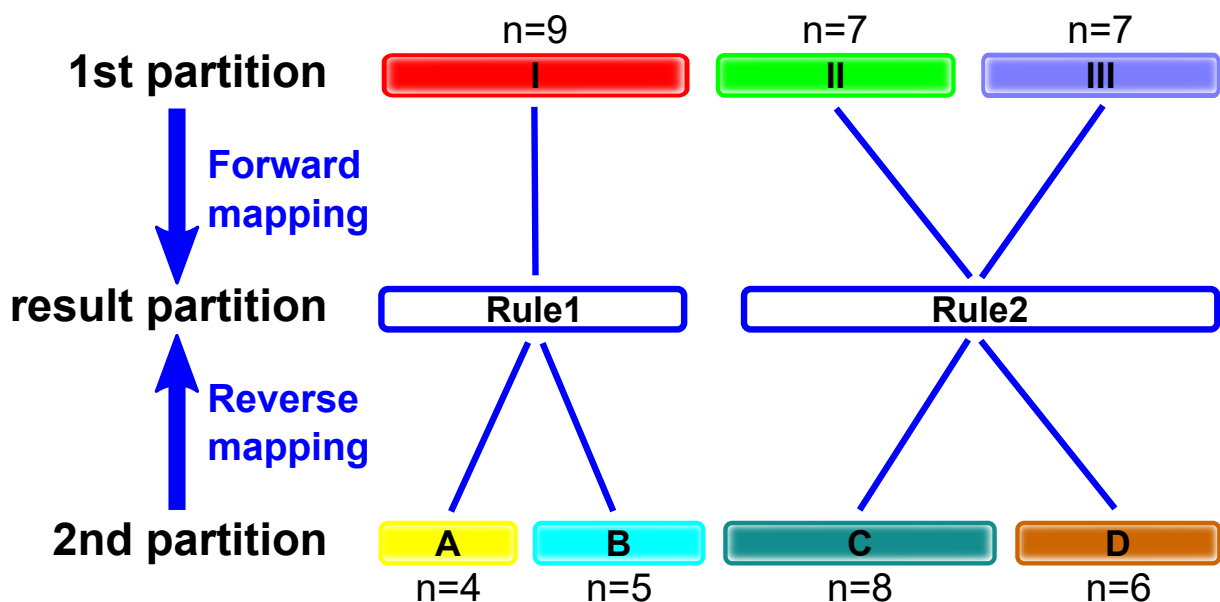


Figure 17.3.6: Another possible mapping for the two example partitions.

Different *methods* are available for determining mapping rules. Mappings can either be *asymmetric* or *symmetric*. Asymmetric mappings are the easiest to understand: they assign classes from the second partition to classes from the first partition. Asymmetric mappings are very useful, e.g. when one wants to use an older, more established classification system (the first partition) as a reference frame or "gold standard" to which a newer typing technique (the second partition) is compared. Symmetric mappings map classes in the forward and reverse direction and are needed to obtain a consistent nomenclature when using two different classifications.

17.3.2 Example data set

The partition mapping tool requires the class information of both partitions to be stored as an information field or as comparison groups. To illustrate the partition mapping tools, classification data is available on


download page of the website (<http://www.applied-maths.com/download/sample-data>, click on "Partition sample field data").

The sample data set contains classification results obtained via PFGE and serotyping. For each method, a letter code (type) is assigned to each typical PFGE profile or serological behavior, respectively. The data are available as a tab-delimited text file, designated `Partition_data.txt`.

17.3.3 Preparing the database

The **DemoBase Connected** will be used to illustrate the partition mapping tool available in BioNumerics and can be downloaded directly from the *BioNumerics Startup* window or restored from the back-up file available on our website:

- To download the database directly from the *BioNumerics Startup* window, click the **Download example databases** link, located in the lower right corner of the *BioNumerics Startup* window. Select **DemoBase Connected** from the list and select **Database > Download**.
- To restore the database from the back-up file, first download the file `DemoBase_Connected.bnbk` from <http://www.applied-maths.com/download/sample-data>, under 'DemoBase Connected'.

In the *BioNumerics Startup* window, press the  button, select **Restore database**, browse for the downloaded file and select **Create copy**. Specify a name and click **<OK>**.

3.1 Open the database **DemoBase Connected**.

3.2 In the *Main* window, select **File > Import...** (, **Ctrl+I**).

3.3 In the *Import* dialog box, expand **Entry information data**, highlight **Import fields (text file)** and press **<Import>**.

3.4 Browse for the `Partition_data.txt` file and press **<Next>**.

3.5 Highlight the row that corresponds to "Key" and press **<Edit destination>**.

3.6 In the *Edit data destination* dialog box, highlight "Key" and press **<OK>**.

3.7 Highlight the two remaining file fields and press **<Edit destination>** again.

3.8 Highlight "Entry info field" and press **<OK>**.

3.9 Leave the default names unaltered and press **<OK>**. Confirm the action.

The *Import rules* dialog box should now look like in Figure 17.3.7.

3.10 Press **<Next>** and **<Finish>**.

3.11 Specify a template name and press **<OK>**.

3.12 Press **<Next>** and **<Finish>** to import the information in the database.

The data is now imported as information fields in the **DemoBase Connected** demonstration database. Optionally, we can set information field properties and a color coding for "PFGE type".

3.13 Right-click the information field header of "PFGE type" and select **Field properties** from the floating menu (see Figure 17.3.8).

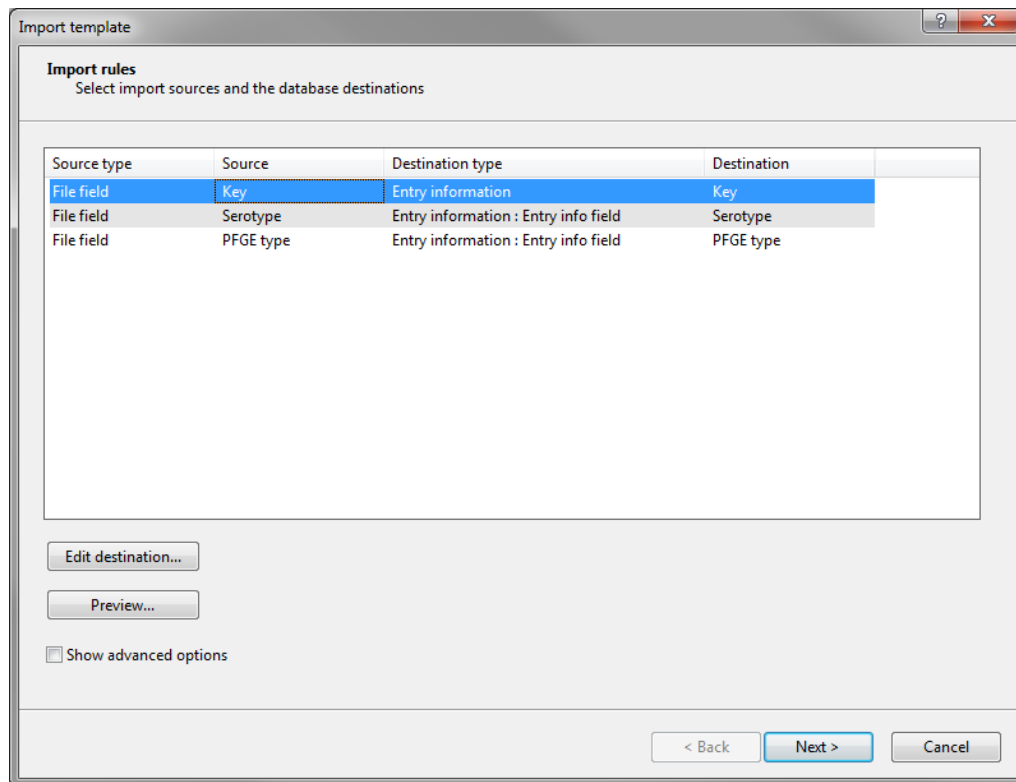


Figure 17.3.7: The *Import rules* dialog box after setting up the template for import of the example data.

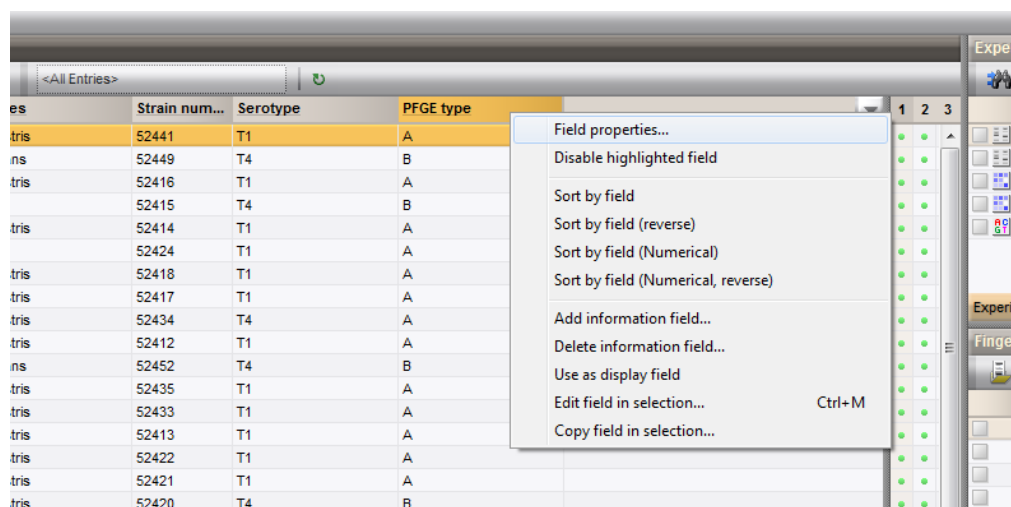


Figure 17.3.8: Set the field properties for the "PFGE type" field.

3.14 In the dialog box that appears, press **<Add all>** to add all current field states and confirm. Check **Use colors** and close the dialog.

Each PFGE type (A, B, C or D) is now displayed in its own color.



Furthermore, we will create a new information field to store mapping information in.

3.15 Right-click the information fields header and select **Add information field** from the floating menu.

3.16 Enter a name for the new information field (e.g. "Mapping"), leave the other settings unaltered and press **<OK>**.

17.3.4 Performing a partition mapping

The steps involved in creating a partition mapping will be illustrated using the **DemoBase Connected** database, with the additional classification information based on PFGE and serotype added (see 17.3.2).

- 4.1 Open the database **DemoBase Connected**.
- 4.2 If the "PFGE type" and "Serotype" information is not yet present, import the `Partition_data.txt` file first (see 17.3.2).
- 4.3 In the *Main* window select all entries in the database except the Standards, e.g. use **Ctrl+A** to select all entries and unselect the standards with **Ctrl+click**.
- 4.4 Click on the  button in the *Comparisons* panel to create a new comparison for the selected entries.
- 4.5 In the *Comparison* window, select **Statistics > Partition mapping** or press the  button and select **Partition mapping** from the drop-down menu that appears.

The *Partition mapping* dialog box pops up (Figure 17.3.9).

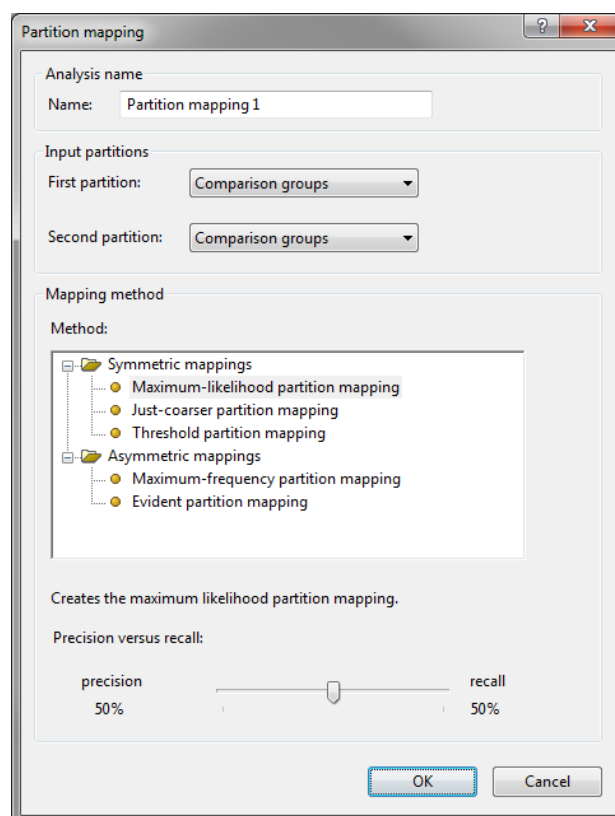


Figure 17.3.9: The *Partition mapping* dialog box.

An *Analysis name* can be entered in the corresponding text box. The default name suggested is "Partition mapping", followed by a serial number. The partition mapping will be listed under this name in the *Analyses* panel of the *Comparison* window.

Under *Input partitions*, two drop-down lists let you select any user-defined information field or comparison groups as *First partition* and *Second partition*.

The hierarchical representation under *Mapping method* provides an overview of the available methods. Depending on the selected method, a brief explanation and – if required by the algorithm – additional input parameters are displayed below. The mapping methods are subdivided in two categories: **Symmetric**

mappings and **Asymmetric mappings**. A category can be collapsed by clicking on the small "-" (minus) sign that precedes its name. **Asymmetric mappings** start from the classes defined in the **Second partition** and assign them to classes in the **First partition**. For **Symmetric mappings**, by nature, the order in which the partitions are defined does not matter.

Symmetric mappings:

- In the **Maximum-likelihood partition mapping**, initially mapping rules are defined based on the most frequent mappings. Then a maximum likelihood model is made that predicts the contingency table as closely as possible (best possible fit). A trade-off needs to be made between **precision** and **recall**. A high value for **precision** will create more rules and predict the classes as good as possible, at the expense of creating some violations. Using high value for **recall** will result in more classes taken together in fewer rules and therefore will propose rules where there are no actual observations in the contingency table to support these rules (missing entries).
- A **Just-coarser partition mapping** creates a mapping without violations; two classes map as soon as there is a non-empty cell in the contingency table that connects both classes.
- In the **Threshold partition mapping**, a **Threshold** is first applied to the contingency table and any frequency lower than this threshold is removed. Then, a just-coarser partition mapping (see above) is applied to the modified contingency table. The threshold is expressed as a percentage, relative to the marginal frequencies.

Asymmetric mappings:

- **Maximum-frequency partition mapping**: each class of the second partition is mapped on the class of the first partition that has the highest frequency.
- **Evident partition mapping**: a **Threshold** is first applied to the contingency table and any observation with a frequency lower than this threshold is removed. Only the evident mappings are made (one-on-one relation). As a consequence, orphans are possible in both directions.

4.6 For the example data, select "PFGE type" as **First partition** from the corresponding drop-down list and "Serotype" as **Second partition**.

4.7 Under **Mapping method**, highlight **Maximum-likelihood partition mapping** and leave the **Precision** versus **Recall** slider at its default position.

4.8 Press <OK> to calculate the partition mapping.

The **Partition mapping** window appears (Figure 17.3.10).

17.3.5 The Partition mapping window

The **Partition mapping** window (Figure 17.3.10) consists of seven panels: *Graphical representation*, *Global mapping*, *Forward mapping*, *Reverse mapping*, *Contingency table*, *Entry selection* and *Partition comparison panel*. All panels are dockable, which enables the user to customize the layout of the **Partition mapping** window according to personal preference and/or the type of analysis to be performed. See 2.3.4 for detailed information on the display options of dockable panels.

In the **Contingency table panel**, the contingency table of the first partition (rows) and second partition (columns) is displayed. All mapping rules are calculated based on this table. The classes from the first and second partition can be rearranged by selecting the corresponding row or column and holding the **Ctrl**

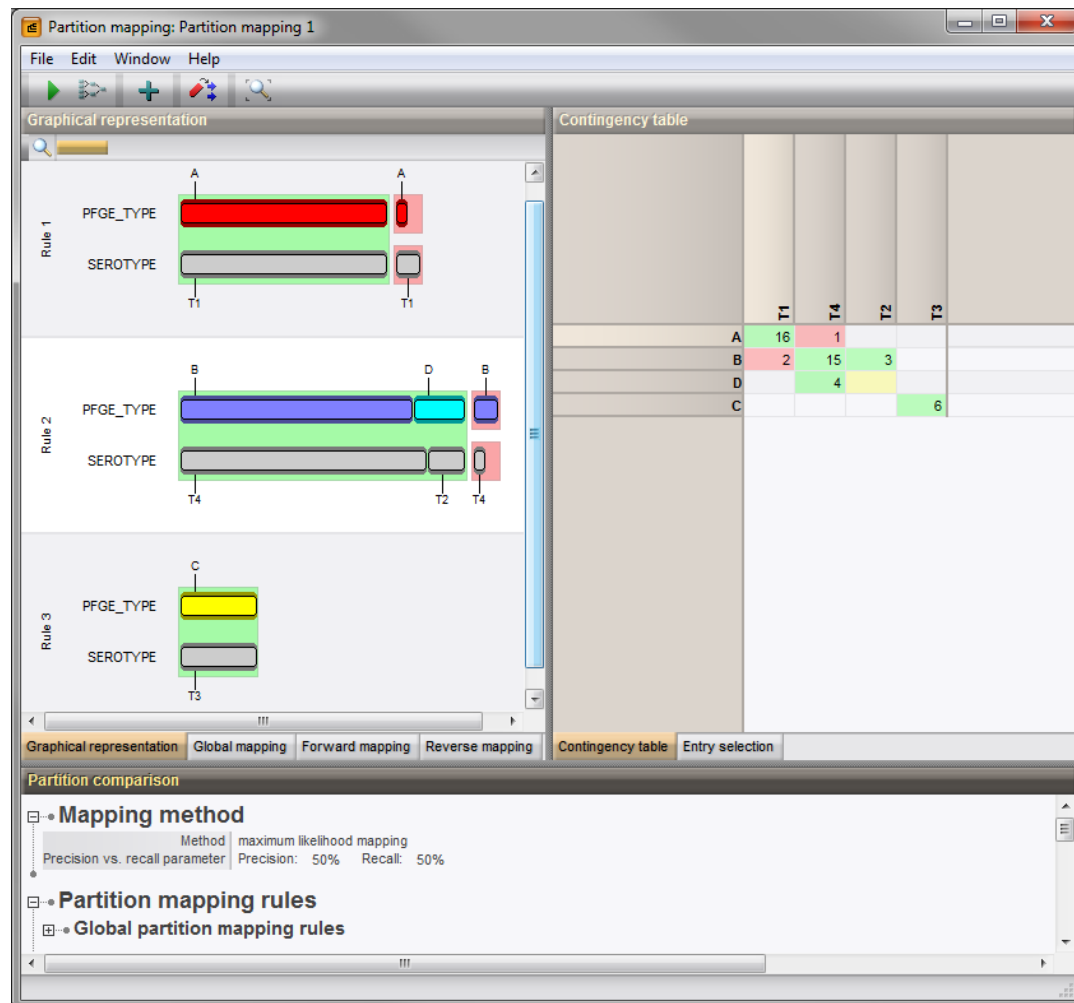


Figure 17.3.10: The *Partition mapping* window.

key while pressing the arrow keys to move the class in the desired direction: **Ctrl+Up arrow** or **Ctrl+Down arrow** to move a class from the first partition up or down, respectively and **Ctrl+left arrow** or **Ctrl+right arrow** to move a class from the second partition left or right, respectively. The cells in the contingency table are colored according to their compliance with the mapping rules: green is a confirmation of the mapping rules, red means a violation and yellow means missing entries (cells that are predicted to contain entries according to the mapping rules, but that are empty in reality). Individual cells in the contingency table can be selected by **Ctrl+click** or a complete range at once by clicking the first cell in the desired range and holding the **Shift** key while clicking the last cell. Selected cells in the contingency table are shown in a darker color than unselected cells.



When the preference *Use color background for complete field* in the *Preferences* window (see 2.3.3) is set to "No", the contingency table cells will only display a small colored strip on the left edge of the cells.

The complete contingency table or only the selected cells can be copied to the clipboard as tab-delimited text with **Edit > Copy complete contingency table** or **Edit > Copy selection from contingency table**, respectively. From the clipboard, the information can be pasted in other applications.

The *Global mapping panel* is a grid view panel that lists all mapping rules that were calculated based on the contingency table and the selected mapping method (see 17.3.4). Each mapping rule has a name (shown in "Rule name"), which can be modified: just click the cell twice to edit the rule name in spreadsheet mode. For each rule, the classes from the first and the second partition that are mapped on that rule, are shown. Other information listed is "Forward size" (the total number of entries in those classes from the first partition that


the rule applies on, i.e. the sum of the corresponding rows in the contingency table), "Forward violations" (the percentage of entries violating the forward rule, relative to the forward size), the "Reverse size" (the total number of entries in those classes from the second partition that the rule applies to, i.e. the sum of the corresponding columns in the contingency table), and "Reverse violations" (the percentage of entries violating the reverse rule, relative to the reverse size). An individual rule can be selected by holding the **Ctrl** key while clicking the rule (**Ctrl+click**). A range of mapping rules can be selected at once by clicking the first rule, holding the **Shift** button and clicking the last rule to be selected. When the above actions are repeated, the rule(s) become unselected again.

The *Forward mapping panel* lists how each class from the first partition is mapped, according to the mapping rules, on the second partition. A class of the first partition can be selected using **Ctrl+click**. Any other class, contained in the same mapping rule, will automatically be selected as well. A range of mappings can be selected at once by clicking the first mapping, holding the **Shift** button and clicking the last mapping to be selected. Likewise, if classes exist outside the selection that are connected by the same mapping rules, they will be selected as well. When the above actions are repeated, the class(es) become unselected again.

The *Reverse mapping panel* lists how each class of the second partition is mapped on the first partition. Classes from the second partition can be selected in the same way as described for the *Forward mapping panel*.

As the name suggests, the *Graphical representation panel* visualizes the mapping rules in a graphical fashion. Classes from the first and second partition are represented by bars, which are sized proportionally with the number of entries they contain. For each rule, the classes from the first partition are shown above the classes from the second partition. Classes are colored according to the colors defined in the information field properties (see 3.3.6); if no colors are defined, the bars appear in gray. Classes from the first and second partition that confirm the mapping rule are enclosed in a green rectangle. Violations of the mapping rule are shown in two, vertically separated, red rectangles.

You can zoom in or out on the graphical representation using the zoom slider. See 2.3.7 for a detailed description of zoom slider functionality.

To scale the graphical representation so it fits the *Graphical representation panel* exactly, select **Edit > Zoom to fit** or press the  button.

In the *Graphical representation panel*, all entries from a class can be selected by pressing the **Ctrl** key while clicking the class. Selected entries are indicated with a dark shade. When only a part of the entries in a class is selected (e.g. by making a manual selection in the *Comparison* window), a proportional part of the bar is shaded. Note that the same entries also appear shaded in the classes of the other partition. To select more than one class at once, hold the **Shift**-key and drag a rectangle with the mouse. Entries can be unselected by repeating the above actions. All selected entries can be unselected at once by pressing the **F4** key on the keyboard.

Rules can be selected as well: **Ctrl+click** anywhere in a rule (not on a bar) to select it. The rule becomes highlighted in dark yellow. In the *Contingency table panel*, the cells corresponding to that rule are selected as well. **Ctrl+click** again to unselect the rule.

Selected database entries are displayed in the *Entry selection panel*. The entry information is preceded by a green or red rectangle, for entries that confirm or violate a mapping rule, respectively. Entry selections can be made in many parts of the software, e.g. in the *Main* window, the *Comparison* window, or in the *Graphical representation panel* (see above). When mapping rules or a part of the contingency table is selected, this selection can be transferred to a selection of entries as well.

5.1 For example, **Ctrl+click** the first mapping rule to select it.

5.2 Select **Edit > Transfer selection** or press the  button.

The *Export selection* dialog box appears (see Figure 17.3.11).

The *Selection source* can be either the *Selected partition mapping rules* or the *Selected contingency table*

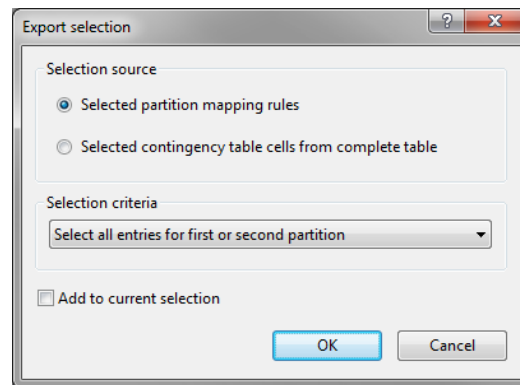


Figure 17.3.11: The *Export selection* dialog box, used to transfer a selection of mapping rules or contingency table cells into a selection of entries.

cells. Under **Selection criteria**, a criterion that will be used to transfer the selection can be chosen from the drop-down list. If **Add to current selection** is checked, newly selected entries will be added to the already selected ones. When the option is unchecked, the current selection is replaced by the newly selected entries.

5.3 Leave **Selected partition mapping rules** checked, choose **Select all violations for first partition** from the drop-down list and press <OK>.

In the *Entry selection panel*, a single entry is selected: entry G@Gel07@013, the only entry with PFGE type A that does not map on Serotype T1.

The *Partition comparison panel* provides a complete report of the partition mapping, which is divided in a number of sections:

In the **Mapping method** section, the applied mapping method used is shown, together with relevant parameters.

The **Optimization** section lists any optimization(s) applied (see below).

The **User-defined rules** section lists any user-defined rule(s). See 17.3.6 for instructions on how to define mapping rules manually.

In the **Partition mapping rules** section, the global partition mapping rules, the partition mapping rules from the first to the second partition and the partition mapping rules from the second to the first partition are summarized.

The **Statistics** section contains a table, which displays a number of indices that can be used to estimate the quality of the mappings. All indices are expressed as percentages. A diamond is displayed next to each index, of which the color ranges from very dark (0%) to very pale blue (100%). On top of the table, the **Mapping rules likelihood** is shown. The closer the value is to 1, the better the mapping rules predict the contingency table. A brief description of each index is given below, for a more detailed description, we refer to the relevant literature [26].

Purity is calculated by assigning each class from one partition to the class of another partition with the highest frequency, i.e. having the largest overlap. Next, the accuracy of the assignment is measured by counting the number of correctly assigned entries and dividing by the total number of entries in the comparison (N). Since purity can be calculated in two directions (from the first into the second partition and from the second into the first), this translates in a contingency table as the sum over the rows (columns) of the column (row) maxima, divided by N . Note that purity will be maximized (100%) when each entry gets its own class.

In the example data, with "PFGE type" the **first partition** and "Serotype" the **second partition**, the contingency table is the one displayed in the *Contingency table panel* in Figure 17.3.10. The **Purity (into 1st)**, i.e. the second partition mapped into the first, is the sum of the column maxima divided by N :

$$\frac{16 + 15 + 3 + 6}{47} = \frac{40}{47} \approx 0.851$$

or 85.1% when expressed as a percentage. The **Purity (into 2nd)**, i.e. the first partition mapped into the second, is the sum of the row maxima divided by N :

$$\frac{16 + 15 + 4 + 6}{47} = \frac{41}{47} \approx 0.872$$

or 87.2%.

With "PFGE type" the **first partition** and the forward mapped partition the **second partition**, the contingency table would be as shown in Table 17.3.1. The **Purity (into 1st)**, i.e. the second partition mapped into the first, is

$$\frac{17 + 20 + 6}{47} = \frac{43}{47} \approx 0.915$$

or 91.5%. The **Purity (into 2nd)**, i.e. the first partition mapped into the second, is then

$$\frac{17 + 20 + 6 + 4}{47} = \frac{47}{47} = 1$$

or 100%.

	A	B&D	C
A	17	0	0
B	0	20	0
C	0	0	6
D	0	4	0

Table 17.3.1: Contingency table for the example data with "PFGE type" as first partition (classes A, B, C and D) and the forward mapped partition as second partition (classes A, B&D and C).

With "PFGE type" the **first partition** and the reverse mapped partition the **second partition**, the contingency table would be as shown in Table 17.3.2. The **Purity (into 1st)**, i.e. the second partition mapped into the first, is

$$\frac{16 + 18 + 6}{47} = \frac{40}{47} \approx 0.851$$

or 85.1%. The **Purity (into 2nd)**, i.e. the first partition mapped into the second, is then

$$\frac{16 + 18 + 4 + 6}{47} = \frac{44}{47} \approx 0.936$$

or 93.6%.

Similar calculations can be made for "Serotype" (containing classes T1, T2, T3 and T4) as **first partition** and the forward mapped partition (classes A, B&D, and C) or the reverse mapped partition (classes T1, T2&T4 and T3) as **second partition**. The purity for the forward mapped partition as **first partition** and the reverse mapped partition as **second partition** will always be symmetric.

	T1	T2&T4	T3
A	16	1	0
B	2	18	0
C	0	0	6
D	0	4	0

Table 17.3.2: Contingency table for the example data with "PFGE type" as first partition (classes A, B, C and D) and the reverse mapped partition as second partition (classes T1, T2&T4 and T3).

The **Normalized mutual information** is a weighted version of the forward and reverse purity and has predictive power in both directions. A trade-off is made between precision and recall. In contrast to purity, normalized mutual information can be used to compare partition mappings with different numbers of classes.

In the calculation of the **Rand index** RI , the partition mapping is regarded as a series of decisions, one for each of the $N(N-1)/2$ pairs of entries in the comparison. A true positive (TP) decision assigns two similar entries to the same class, a true negative (TN) decision assigns two dissimilar entries to two different classes. Two types of error exist: A false positive (FP) decision assigns two dissimilar entries to the same class and a false negative (FN) decision assigns two similar entries to different classes. The **Rand index** is the percentage of decisions that are correct:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

The **Rand index** will always result in a value higher than zero, even for two completely random partitions. For the **Adjusted rand index**, an expectation value is subtracted. This index will therefore result in zero in case of two random partitions.

The **Recall (Wallace I)**, sometimes referred to as "reverse Wallace", is defined as:

$$W_I = \frac{TP}{TP + FN}$$

The **Precision (Wallace II)**, sometimes referred to as "forward Wallace", is defined as:

$$W_{II} = \frac{TP}{TP + FP}$$

The complete report, displayed in the *Partition comparison panel*, can be copied to the clipboard as tab-delimited text with **Edit > Copy report**. From the clipboard, the information can be pasted in other applications.

17.3.6 Refining a partition mapping

A partition mapping, calculated by any of the available mapping methods (see 17.3.4), can be modified e.g. by manually adding (a) mapping rule(s) and/or by performing an optimization according to the maximum likelihood criteria. To see the results of different adjustments side-by-side, the active partition mapping can be cloned.

- 6.1 Select **File > Clone** in the *Partition mapping* window. A copy of the active partition mapping appears in its own window.

In some cases, it can occur that one or more of the mapping rules are already known. These rules can be added manually.

6.2 Select *Edit > Add mapping rule*.

The *Create new rule* dialog box appears (Figure 17.3.12).

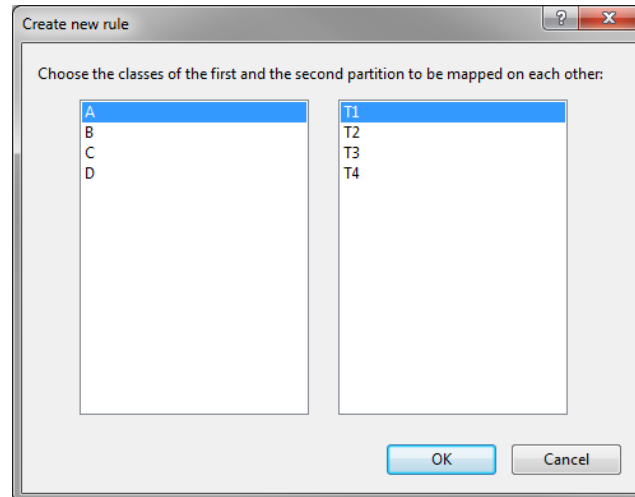


Figure 17.3.12: The *Create new rule* dialog box to manually create a mapping rule.

To add a mapping rule, select the class(es) from the first partition in the list on the left, the class(es) of the second partition on which they should map in the list on the right and press **<OK>**.

A partition mapping, calculated with any of the available methods, can be further optimized according to the maximum likelihood criterion.

6.3 Select *Edit > Optimize mapping rules* to call the *Optimize mapping rules* dialog box (Figure 17.3.13).

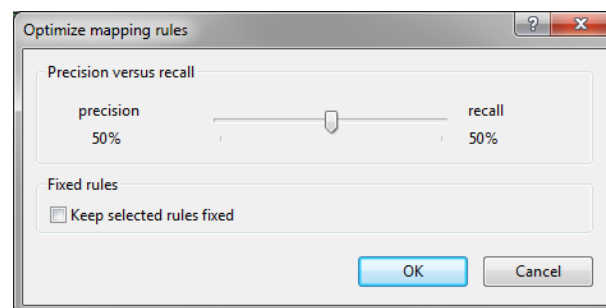


Figure 17.3.13: The *Optimize mapping rules* dialog box.

The *Precision versus recall* can be set using the slider, similar as in the *Partition mapping* dialog box (see Figure 17.3.9).


When one or more rules are selected (see 17.3.5), the option *Keep selected rules fixed* becomes available. Fixed rules will be left unaltered by the maximum likelihood optimization.

17.3.7 Applying a partition mapping

One reason to formulate mapping rules is because of their predictive power: if no information is available for a number of entries for one of the partitions, we can predict the classes by means of the mapping rules. This

information can either be stored in an information field or a comparison group. Alternatively, the "recipe" for the mapping can be stored as a decision network, which can be applied to any selection of entries later on.

A partition mapping can be applied and the result stored in an information field or as comparison groups.

- 7.1 Select **File > Apply mapping** or press the  button. The *Apply partition mapping* dialog box appears (see Figure 17.3.14).

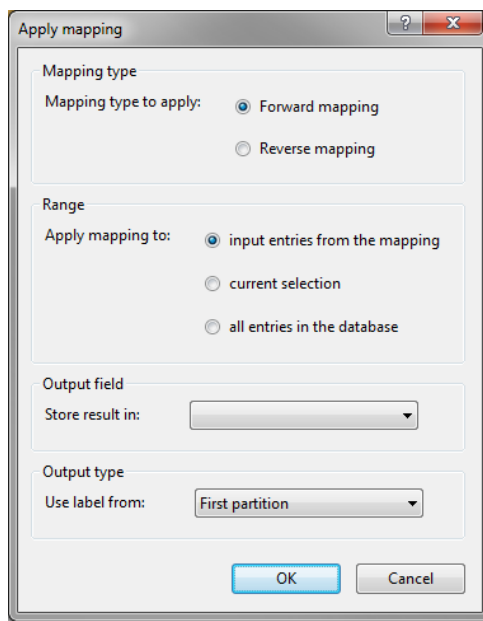


Figure 17.3.14: The *Apply partition mapping* dialog box.

Under **Mapping type**, either the **Forward mapping** (from the second to the first partition) or **Reverse mapping** (from the first to the second partition) can be applied.

The **Range** of entries to which to apply the mapping can be set: either the **input entries from the mapping**, i.e. all entries in the comparison on which the partition mapping is based, the **current selection** of entries or **all entries in the database**.

The **Output field** to store the result in can be any of the user-defined information fields or the comparison groups. Comparison groups can only be selected when **input entries from the mapping** is selected under **Range**.

Under **Output type**, the label from either the first partition, second partition or the mapping rule name can be selected. The drop-down list is inactivated when the comparison groups are checked as **Output field**.


- 7.2 For example, check **Forward mapping** as mapping to apply and apply the mapping to **all entries in the database**. Select the previously defined "Mapping" information field (see 17.3.2) as **Output field** and use the label from the **Second partition**.

- 7.3 Press <OK> and confirm the action.

The "Mapping" information field now contains the mapping information from the second partition. For entries where the information in the "PFGE type" field cannot be mapped (e.g. for the STANDARD entries in the **DemoBase Connected**), the "Mapping" field will be left unaltered.

Instead of directly writing mapping information into an information field, a decision network (see 15.6) can automatically be generated based on the partition mapping. Decision networks generated this way can either

be used to write mapping information to an information field or to select database entries.

- 7.4 Select **File > Create decision network** or press the  button. The *Export rules to decision network* dialog box pops up (see Figure 17.3.15).

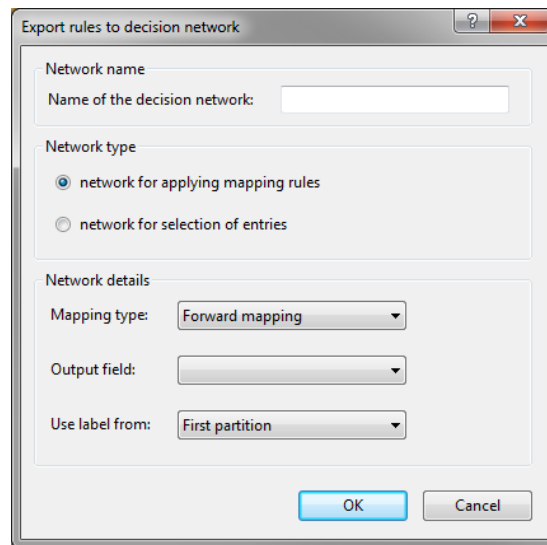


Figure 17.3.15: The *Export rules to decision network* dialog box when no mapping rule is selected.

A **Network name** can be entered in the corresponding text box. The decision network will be listed under this name in the *Decision Networks panel* in the *Main window*.

Under **Network type**, the option **network for applying mapping rules** is selected by default. In this case, the **Network details** settings are similar as discussed earlier for the *Apply mapping dialog box* (see Figure 17.3.14): **Mapping type**, **Output field** and **Use label from**.

In case a mapping rule is selected prior to calling the *Export rules to decision network dialog box*, the option **network for selection of entries** becomes available under **Network type**. With this option checked, the **Selection policy** can be set under **Network details**, similar to the selection criteria in the *Export selection dialog box* discussed earlier (see Figure 17.3.11).

- 7.5 Enter "Forward Mapping" as decision network name and leave **network for applying mapping rules** selected.

- 7.6 Next, select **Forward mapping** as **Mapping type**, "Mapping" as **Output field** and use the label from the **Second partition**.

- 7.7 Press <**OK**> to create the decision network.

Navigate to the *Decision Networks panel* in the *Main window* and double-click **Forward Mapping** to open the decision network (see Figure 17.3.16).

This decision network performs the same action as we did earlier via the *Apply partition mapping dialog box*, but has the advantage that it can be applied in a flexible way to any selection of entries.

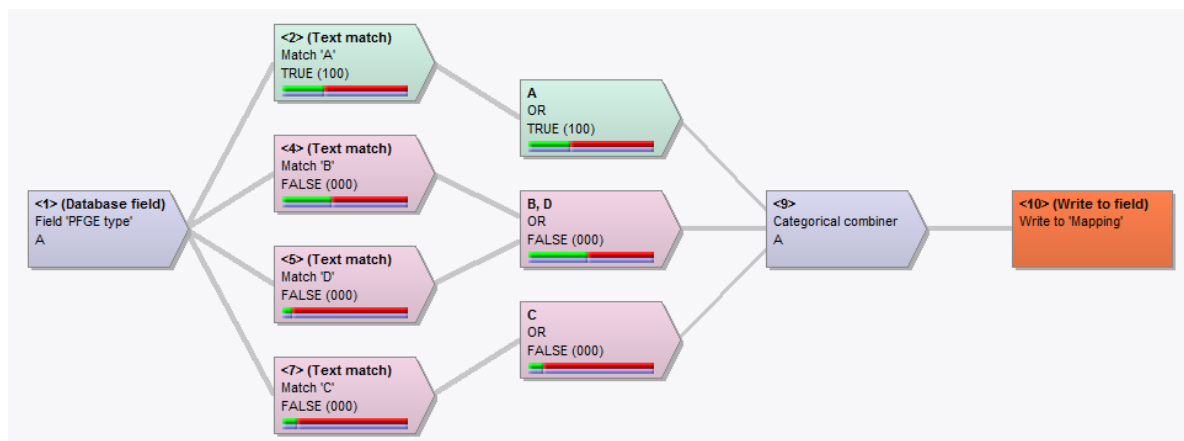


Figure 17.3.16: Decision network to apply the mapping rules of the example data.

Chapter 17.4


Dimensioning techniques

17.4.1 Introduction

Principal Components Analysis (**PCA**) and Multi-Dimensional Scaling (**MDS**) are two alternative grouping techniques that can both be classified as *dimensioning techniques*. In contrast to dendrogram inferring methods, they do not produce hierarchical structures like dendrograms. Instead, these techniques produce two-dimensional or three-dimensional plots in which the entries are spread according to their relatedness. Unlike a dendrogram, a PCA or MDS plot does not provide "clusters". The interpretation of the obtained comparison is, more than in cluster analysis, left to the user.

PCA assumes a data set with a known number of characters and analyzes the characters directly. PCA is applicable to all kinds of character data, but not directly to fingerprint data. Fingerprints can only be analyzed when converted into a band matching table (see [4.3](#)).

MDS does not analyze the original character set, but the matrix of similarities obtained using a similarity coefficient. Rather than being a separate grouping technique, MDS just replaces the clustering step in the sequence *Characters > Similarity matrix > Cluster analysis*. However, it is a valuable alternative to the dendrogram methods, which often oversimplify the data available in a similarity matrix, and tend to produce overestimated hierarchies.

Please note that calculating a PCA or MDS requires the Dimensioning techniques and statistics module () to be present in your BioNumerics configuration.


17.4.2 Calculating an MDS

Any experiment type for which a complete similarity matrix is available can be analyzed by MDS. Matrix types are not suitable for MDS clustering if the matrices are incomplete.

In the *Experiments* panel of the *Comparison* window, select the experiment type on which you want to calculate an MDS.

Check whether a similarity matrix is available for this experiment type. If no similarity matrix is available, perform a cluster analysis as described in [13.2.6](#).

Optionally, create comparison groups, e.g. based on one of the information fields (see [13.3.4](#)).

Select **Statistics > Multi-dimensional scaling...** () to call the *Perform multi-dimensional scaling* dialog box (see Figure [17.4.1](#)).

When performing a MDS using the **Metric algorithm**, a PCA is calculated based on the similarity matrix that was calculated for the selected experiment type.

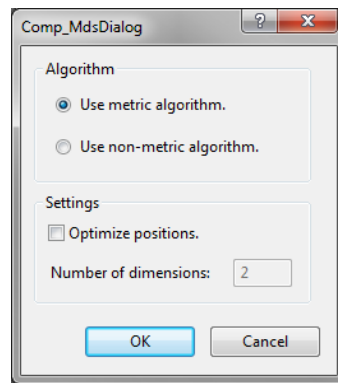


Figure 17.4.1: MDS settings.

When performing a MDS using the *Non-metric algorithm*, a non-metric multidimensional scaling is calculated using the `Nmbs` command in `mothur` (see <http://www.mothur.org/wiki/Nmbs>) (for more information, see [11])

When checking the option *Optimize positions* BioNumerics will iteratively recalculate the MDS, each time again optimizing the positions of the entries in the space to resemble the similarity matrix as closely as possible. When enabling this optimization, the calculations will take longer.

The MDS is calculated and the *Coordinate space* window is shown (see Figure 17.4.2).

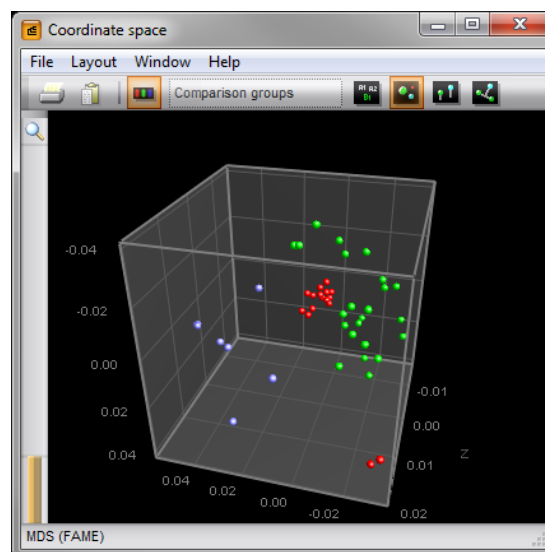


Figure 17.4.2: The *Coordinate space* window, resulting from a PCA or MDS analysis.

The *Coordinate space* window shows the entries as dots in a cubic coordinate system. By default, the entries are represented as 3D spheres in a realistic perspective. They appear in the colors as defined for the groups in the comparison.

The type of analysis (PCA or MDS) and the experiment type are displayed in the status bar of the *Coordinate space* window.


To zoom in and zoom out on the image, use **Layout > Zoom in (Pge Down)** and **Layout > Zoom out (Pge Up)**, respectively. Alternatively, the zoom slider can be used.

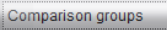
The image can be rotated in real time by clicking on the image and dragging in the desired direction with the mouse.

With **Layout > Show keys** () , database keys of the entries are displayed.

However, the entry keys may be long and uninformative for the user, so the entry keys can be replaced by a *group code*. The program assigns a letter to each defined comparison group, and within a comparison group, each entry receives a number. The group codes are displayed by selecting **Layout > Use group numbers as key** in the parent *Comparison* window.

Alternatively, a selected information field can be displayed instead of the key: in the parent *Comparison* window, click on the information field which you would like to see displayed and select **Layout > Use field as key**. The information of the selected field is now displayed in the *Coordinate space* window.


With **Layout > Show group colors** () , you can toggle between the color representation and the non-color representation, in which the entry groups are represented (and printed) as symbols instead of colored dots. On the screen, it is generally easier to evaluate the groups using colors.


If field states with corresponding color coding were defined (see 3.3.6), the drop-down list  allows you to select a coloring based on groups or any of the available field states.

An individual entry in the coordinate system is selected using **Ctrl+click**. To select several entries at a time, hold down the **Shift**-key while dragging the mouse in the coordinate system. All entries included in the rectangle will become selected. Selected entries are contained in orange cubes.


Selections made in the *Coordinate space* window are shown in the *Information fields* panel in the *Comparison* window, and in the *Database entries* panel of the *Main* window.


By double-clicking on an entry in the *Coordinate space* window, its *Entry* window pops up.

With **Layout > Show construction lines** () , the entries are displayed on vertical lines starting from the bottom of the cube. This may facilitate the three-dimensional perception.

With **Layout > Show rendered image** () , you can toggle between the realistic three-dimensional perspective with entries represented by spheres, and a simple mode where entries are represented as dots.

With **Layout > Preserve aspect ratio** enabled, the relative contributions of the three components are respected, which means that the coordinate system is no longer shown as a cube.

Another very interesting display option is **Layout > Show dendrogram** () . When this option is enabled, the entries in the coordinate system are connected by the branches of the active (i.e., currently displayed) dendrogram from the parent *Comparison* window. The dendrogram name will be displayed in the status bar of the *Coordinate space* window. This is an ideal combination to co-evaluate a dendrogram and a coordinate system (PCA or MDS).

To copy the coordinate space image to the clipboard, select **File > Copy image to clipboard...** () . This calls a new dialog (see Figure 17.4.4).

The software prompts for the resolution of the bitmap to be exported.

The image can be printed with **File > Print image...** () . The image will print in color if the colors are shown on the screen.

Select **File > Export coordinates** to export all entry coordinates.

The *Coordinate space* window is closed with **File > Exit**.

17.4.3 Calculating a PCA

PCA is typically executed on complete character data. It does not work on sequence types. Fingerprints can only be analyzed by PCA if a band matching table is first generated (see 4.3).

In the *Experiments* panel of the *Comparison* window, select the character type or fingerprint type on which you want to calculate a PCA. In case of a fingerprint type, a band matching needs to be performed first (see

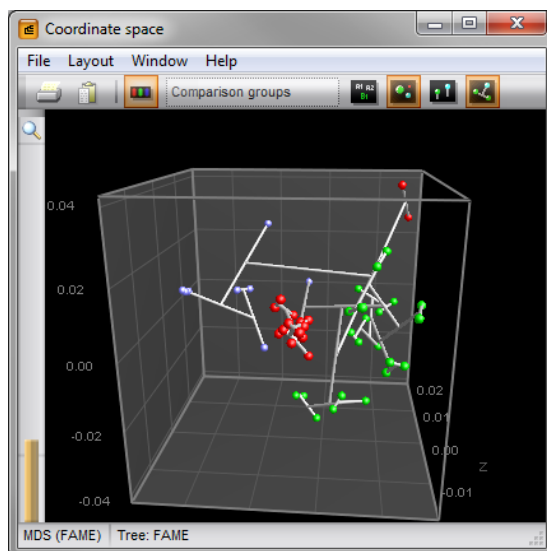


Figure 17.4.3: Coordinate system with a dendrogram displayed.

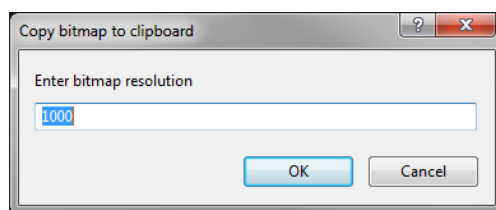


Figure 17.4.4: Copy bitmap to clipboard.

4.3.2).

Optionally, create comparison groups, e.g. based on one of the information fields (see 13.3.4).

Select **Statistics > Principal Components Analysis...** () to call the *Principal Components Analysis* dialog box (Figure 17.4.5).

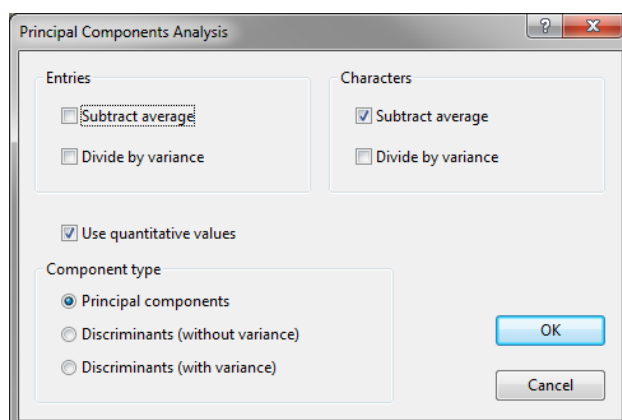


Figure 17.4.5: The *Principal Components Analysis* dialog box.

By default, *Use quantitative values* is checked, and if the technique provides quantitative information (not just absent/present), one will normally want to use this information for the PCA calculation. If this option is unchecked, the character values will be converted to binary as specified in the *Conversion to binary settings* (see 6.1.2).

More sophisticated options are the possibilities to **Subtract average** character value over the *Entries*, and to **Subtract average** character value over the *Characters*. Figure 17.4.6 explains how the averaging works.

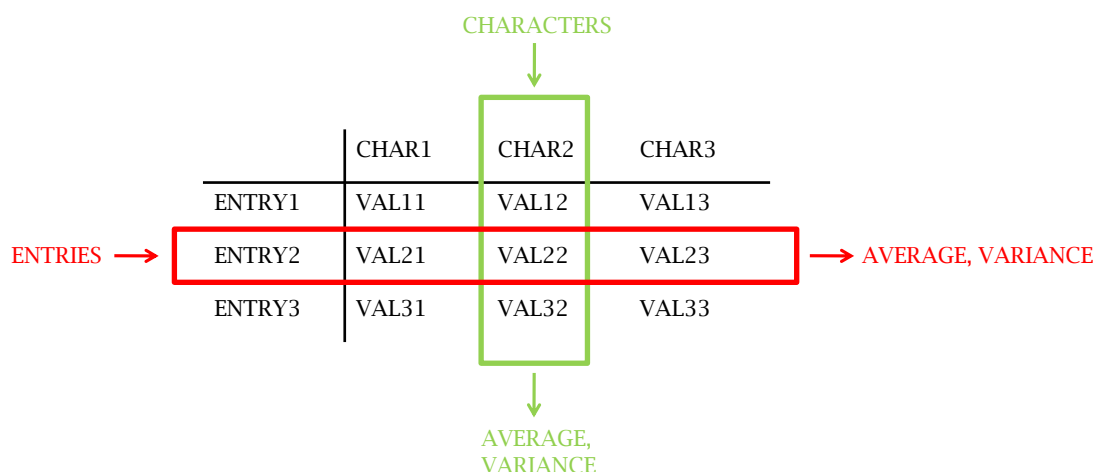


Figure 17.4.6: Character table showing the meaning of average and variance correction at the entries and characters level.

- Subtraction of the *averages* over the *characters* (green in the figure) results in a PCA plot arranged around the origin, and therefore, it is recommended for general purposes.
- Division by the *variances* over the *characters* (green in the figure) results in an analysis in which each character is equally important. Enabling this option can be interesting in a study containing characters of unequal occurrence. For example, if fatty acid extractions are analyzed for a set of bacteria, some fatty acids may be present in abundant amounts, whereas others may occur only in very small amounts. It is well possible that the "minor" fatty acids are as informative or even more informative than the abundant ones, taxonomically seen. If no correction is applied, those minor fatty acids will be completely masked by differences in the abundant fatty acids. Dividing by the variance for each character normalizes for such range differences, making each character contributing equally to the total separation of the system.
- Subtraction of the *averages* over the *entries* (red in the figure) results in character sets of which the sum of characters equals zero for each entry. This feature has little meaning for general purposes.
- Division by the *variances* over the *entries* (red in the figure) results in character sets for which the intensity is normalized for all entries. For example, suppose that you have scanned phenotypic test panels for a number of bacterial strains and want to calculate a PCA. If some strains are less grown than others, the overall reaction in the wells will be less developed. Without correction, well developed and less developed panels will fall apart in the study. Dividing by the variances normalizes the character sets for such irrelevant differences, making character sets with different overall character developments fall together as long as the relative reactions of the characters are the same.



The two latter features are exactly what is done by the Pearson product-moment correlation coefficient. This coefficient subtracts each character set by its average, and divides the characters by the variance of the character set. The feature **Divide by variance** under *Entries* should not be used in character sets where the characters are already expressed as percentages (for example, fatty acid methyl esters).

The lower panel of the dialog box displays the *Component type*. The *Principal components* option is to calculate a principal components analysis.

Pressing <OK> will start the calculation of the PCA.

When the calculations are finished, the PCA analysis is added to the *Analyses* panel in the *Comparison* window, preceded by a "PCA analysis" icon (📊). The name of the analysis is the name of the experiment type that was used for the calculation of the PCA.



The previously stored PCA analysis for the experiment type - if present - is removed.

The resulting window, the *Principal Components Analysis* window, is shown in Figure 17.4.7.

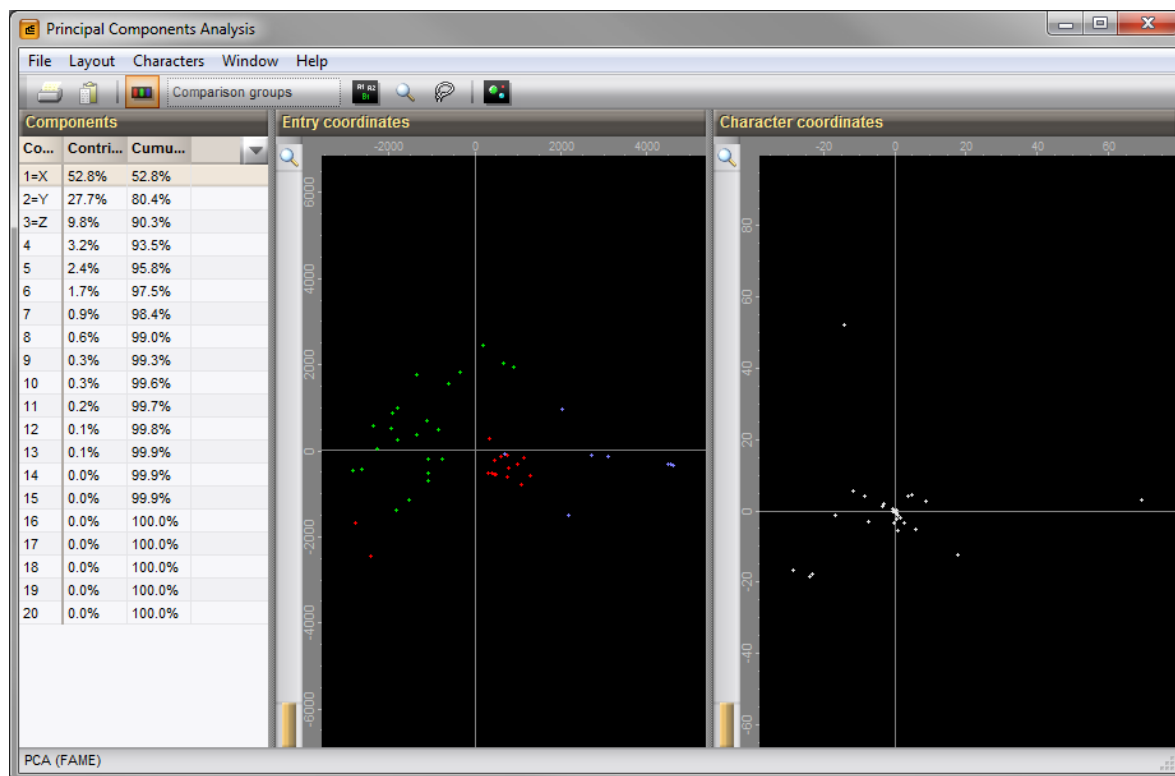


Figure 17.4.7: The *Principal Components Analysis* window.


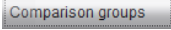
The *Principal Components Analysis* window is divided in three dockable panels (for display options of dockable panels, see 2.3.4).

- In the *Components panel* (the left panel in default configuration), the first 20 components are shown in the **Component** column, with their relative contribution (**Contribution** column) and the cumulative contribution displayed (**Cumulative** column). Also, the components used as X-, Y- and Z-axes are indicated.
- The *Entry coordinates panel* shows the *entries* plotted in an X-Y diagram corresponding to the first two components.
- The *Character coordinates panel* shows the *characters* plotted in the same X-Y diagram. From the *Character coordinates panel*, one can see the contribution each character has to the two displayed components, and hence, what contribution it has to the separation of the groups along the same components. For example, if a group of entries appears left along the X-axis whereas the other entries appear right, those characters occurring left on the X-axis are positive for the left entries and negative for the right entries, and *vice versa*.

By default, the first component is used for the X-axis, the second component is used for the Y-axis, and the

third component is used for the Z-axis. The Z-axis is not shown here, but can be shown in the *Coordinate space* window.


If you want to assign another component as one of the axes, select the component in the *Components panel*, and **Layout > Use component as X axis**, **Layout > Use component as Y axis**, or **Layout > Use component as Z axis**.

Switching from color indication for the groups to symbol indication can be done with **Layout > Show color coding** (). Select a coloring based on groups or any of the available field states from the drop-down list . See 3.3.6 on how to define states for an information field.

Show the keys or a unique label based upon the groups for the entries with **Layout > Show keys** ().



In case the keys are not very informative, one could use **Layout > Use group numbers as key** or click on an information field and use **Layout > Use field as key** in the parent *Comparison* window to display group numbers, respectively the content of the information field, instead of the keys.

The option **Layout > Preserve aspect ratio** allows you to either preserve the aspect ratio of the components, i.e. the relative discrimination of the component on the Y-axis with respect to the component on the X-axis, or to stretch the components on the axes so that they fill the image optimally.

With **Layout > Zoom in / zoom out** () , you can zoom in on any part of the *Entry coordinates* or *Character coordinates panel* of the PCA plot: drag the mouse pointer to create a rectangle; the area within the rectangle will be zoomed to cover the whole panel. In order to restore the original size of the image, simply left-click within the panel. Disable the zoom-mode afterwards. Alternatively, the zoom sliders of the *Entry coordinates* and *Character coordinates panel* can be used to zoom in or out on the plots.


If you move the mouse pointer over the *Character coordinates panel* (characters), the name of the pointed character is shown.

Entries can be selected in a *Principal Components Analysis* window by holding the **Shift**-key down and selecting the entries in a rectangle using the left mouse button. You can also hold down the **Ctrl**-key while clicking on an entry. Selected entries are encircled in blue. Entry selections made in the *Principal Components Analysis* window are also shown in the *Information fields* panel in the *Comparison* window and in the *Database entries* panel of the *Main* window.

An even more flexible way of selecting entries is using the lasso selection tool. To activate the lasso selection tool, use **Layout > Lasso selection tool** (). With the lasso selection tool enabled, selections of any shape can be drawn on the plot. The lasso selection tool menu item is flagged and the button shown as  when the tool is enabled. To stop using the lasso selection tool, you have to click the button a second time, or disable it from the menu.

Plotted characters of a character type experiment can be selected in the *Character coordinates panel* by holding the **Shift**-key down and selecting the characters in a rectangle using the left mouse button. You can also hold down the **Ctrl**-key while clicking on a character. Selected characters are encircled in blue. Character selections made in the *Principal Components Analysis* window are shown in the *Experiment data* panel in the *Comparison* window (if the image of the character type is displayed), and in the *Character type* window of the character type (see 6.1.2).



It is possible to add entries to an existing PCA or remove entries from it. The feature to add entries to an existing PCA is an interesting alternative way of identifying new entries. They can be placed in a frame of known database entries, and in this way, identifying is just looking at the groups they are closest to. Note that, since the components are not recalculated when entries are added to an existing PCA, the PCA does not reflect the full data matrix anymore!

If you want to add entries to an existing PCA, you can select new entries in the *Main* window and copy them to the clipboard using **Edit > Views > Copy selection** (**Ctrl+C**). In the *Comparison* window, select **Edit > Paste selection** (, **Ctrl+V**). The new entries are placed in the *Comparison* window and in the *Principal Components Analysis* window.

To delete entries from a PCA, select them and use **Edit > Cut selection** (, **Ctrl+X**) in the *Comparison* window.


If you started the PCA from a composite data set, you can order the characters according to the selected component in the underlying *Comparison* window. This is an interesting feature to locate characters that separate groups you are interested in. The feature works as follows (only for composite data sets):

1. In the *Principal Components Analysis* window, first determine the component that best separates the groups.
2. Select that component in the *Components panel* and select **Characters > Order characters by component**. The characters are now ordered by the selected component in the parent *Comparison* window.


The entry plot can be printed with **File > Print image (entries)...** () and the character plot can be printed with **File > Print image (characters)...** Alternatively, the entry plot can be copied to the clipboard with **File > Copy image to clipboard (entries)...** () and the character plot can be copied to the clipboard with **File > Copy image to clipboard (characters)**.

If you want to reconstruct or analyze the PCA system in another software package, it is possible to export the coordinates of the **entries** along a selected component (for example the X-axis): select a component in the *Components panel* and **File > Export selected entry coordinates**. If you want to reconstruct the PCA with the first two components, you should also export the second component (Y-axis), by selecting that component in the *Components panel* and **File > Export selected entry coordinates** again. It is also possible to export all entry coordinates at once in a tab-delimited format using **File > Export all entry coordinates**.

Similarly, one can export the coordinates for the **characters** for a certain component: select a component in the *Components panel* and **File > Export selected character coordinates**. To export all character coordinates at once, use **File > Export all character coordinates**.

The command **Layout > Show 3D plot** () allows you to display three components at the same time, by plotting the entries in a *Coordinate space* window. See 17.4.2 to edit a PCA in 3-D representation mode.


The *Principal Components Analysis* window is closed with **File > Exit**.

A PCA that was calculated earlier can be opened again from the *Comparison* window by selecting the experiment type in the *Experiments panel* and **Statistics > Show coordinate space (2D)**. Alternatively, select the PCA analysis in the *Analyses panel* in the *Comparison* window, and use **File > Analysis components > Open** ()

17.4.4 Calculating a discriminant analysis

Discriminant analysis is very similar to PCA. The major difference is that PCA calculates the best discriminating components for the character table as a whole, without foreknowledge about groups, whereas discriminant analysis calculates the best discriminating components *for groups that are defined by the user*. In case of discriminant analysis, these principal components are then called *discriminants*. Like PCA, discriminant analysis is executed on complete character data. It does not work on sequence types. Fingerprints can only be analyzed by discriminant analysis if a band matching table is generated first (see 4.3).

Since discriminant analysis works on user-delineated groups, the comparison should contain groups (see 13.3.4 on how to create comparison groups).

In the *Experiments panel* of the *Comparison* window, highlight the experiment type on which you want to calculate a discriminant analysis and select **Statistics > Principal Components Analysis...** ()

The *Principal Components Analysis* dialog box (Figure 17.4.5) allows a number of choices to be made under **Entries** and **Characters**, which are described in 17.4.3.

The choices under *Entries* and *Characters* also apply for discriminant analysis. However, the *Divide by variance* option under *Characters* makes no difference whether it is enabled or disabled for discriminant analysis.

The following two options are available for discriminant analysis: *Discriminants (without variance)*, and *Discriminants (with variance)*. If you select *Discriminants with variance*, each character is divided by its variance. In order to understand what this implies, consider this figure:

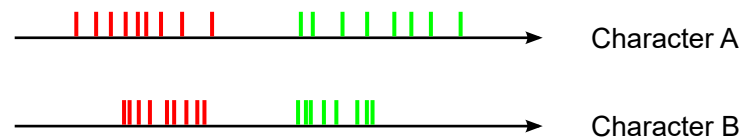


Figure 17.4.8: The influence of character spread on discriminant analysis.

This example shows two groups, 1 (red) and 2 (green), that are separated by two characters, A and B. On the average, group 1 is less positive both for characters A and B. Character A seems to be better at discriminating between the two groups than character B, because the centers of the groups are lying further from each other in case of character A. However, if the internal spread of groups are considered, then the groups are found much more coherent for character B, which may render this character at least as much value for discriminating as character A. In a non-corrected discriminant analysis, character A will account for most of the discrimination, just by the fact that the centers of the groups are more distant. This is the case in option *Discriminants (without variance)*. When the characters are divided by the variances of the groups, the internal spread is compensated for, and character B will become at least as important as character A. This is achieved with option *Discriminants (with variance)*.

When the calculations are finished, the analysis is added to the *Analyses* panel in the *Comparison* window, preceded by a "PCA analysis" icon (📊). The name of the analysis is the name of the experiment type that was used for the calculations. The resulting window is identical to the *Principal Components Analysis* window described before (Figure 17.4.7), and the same features apply. As for a PCA (see 17.4.3), if you started the discriminant analysis from a composite data set, you can order the characters according to the selected discriminant in the underlying *Comparison* window:

In the *Principal Components Analysis* window, first determine the discriminant that best separates two groups you have in mind. You can examine the discriminants by selecting them in the *Components panel* and selecting *Layout > Use component as Y axis* (or *Layout > Use component as X axis*).

Select that discriminant in the *Components panel* and select *Characters > Order characters by component*. The characters are now ordered by the selected discriminant in the *Comparison* window.

The *Principal Components Analysis* window is closed with *File > Exit*.

A discriminant analysis that was calculated earlier can be opened again from the *Comparison* window by selecting the experiment type in the *Experiments* panel and *Statistics > Show coordinate space (2D)*. Alternatively, select the discriminant analysis in the *Analyses* panel in the *Comparison* window, and use *File > Analysis components > Open* (📂).

17.4.5 Self-organizing maps

A self-organizing map (SOM, also called Kohonen map) is a neural network that classifies entries in a two-dimensional space (map) according to their likeliness. The technique which is used for grouping, i.e. the training of a neural network, is completely different from all previously described methods. SOMs therefore provide an interesting addition to conventional grouping methods such as cluster analysis, principal component analysis and related techniques. Also, similar as in PCA, a SOM can start from the characters as

input, thus avoiding the choice of one or another similarity coefficient. Unlike PCA, the distance between entries on the map is not in proportion to the taxonomic distance between the entries. Rather, a SOM contains areas of high distance and areas of high similarity. Such areas can be visualized by different shading, for example when a darker shading is used in proportion to the distance in the SOM.

When the similarity values with all of the other entries of a comparison are considered as the character set, a SOM can also be applied to similarity matrices, which makes the technique also suitable for grouping of electrophoresis patterns that are compared pair by pair using a band matching coefficient such as Dice.

To calculate a SOM on an experiment type, highlight the experiment type in the *Experiments* panel of the *Comparison* window.

Optionally, create comparison groups, e.g. based on one of the information fields (see 13.3.4).

Use *Statistics > Self-organizing map...* to calculate a SOM based on the character data or *Statistics > Self-organizing map (similarities)...* to calculate the SOM from the similarity matrix.

In the latter case, the result of the SOM is based on similarity values of the entries with each other and hence is dependent on the similarity coefficient used, and the tolerance and optimization settings in case of fingerprint types. Obviously, a similarity matrix should be available for the experiment type. If this is not the case, first perform a cluster analysis as described in 13.2.6.



When a SOM is calculated on fingerprint type data, the densitometric curves are used as character data sets for training of the SOM.

A dialog box prompts to *Enter map size* (see Figure 17.4.9).

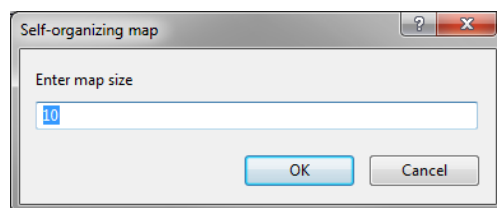


Figure 17.4.9: Specify the map size.

The map size is the number of nodes of the neural network in each direction. For the default size 10, a neural network containing 10×6 nodes is generated. The larger the map is taken, the longer the training takes. Note that the optimal size of the map depends on the number of entries compared. For a small number of entries, a small map size will usually provide better results.

Enter a map size and press <OK> to calculate the SOM. When finished, the SOM analysis is added to the *Analyses* panel in the *Comparison* window, preceded by a "SOM analysis" icon (■). The name of the analysis is the name of the experiment type that was used for the calculation of the SOM.

The *Self-Organizing Map* window is shown in Figure 17.4.10.

This window displays the calculated SOM. The entries are located on the nodes. The map is differentially shaded according similarity: areas of high similarity are black, areas of lower similarity are lighter. Selected entries in the parent *Comparison* window are also selected on the map.

To show the information of a particular entry in the SOM, right-click on the entry and select *Edit database fields*. The *Entry* window pops up.

You can (un)select entries on the SOM by left-clicking on an entry while pressing the **Ctrl**-key, or groups of entries by left-clicking and moving the mouse while pressing the **Shift**-key.

It is possible to add entries to an existing SOM or remove entries from it. The feature to add entries to an existing SOM is an interesting alternative way of identifying new entries. Added entries are placed in a frame of known database entries in the SOM, and in this way, identifying is just looking at the groups they

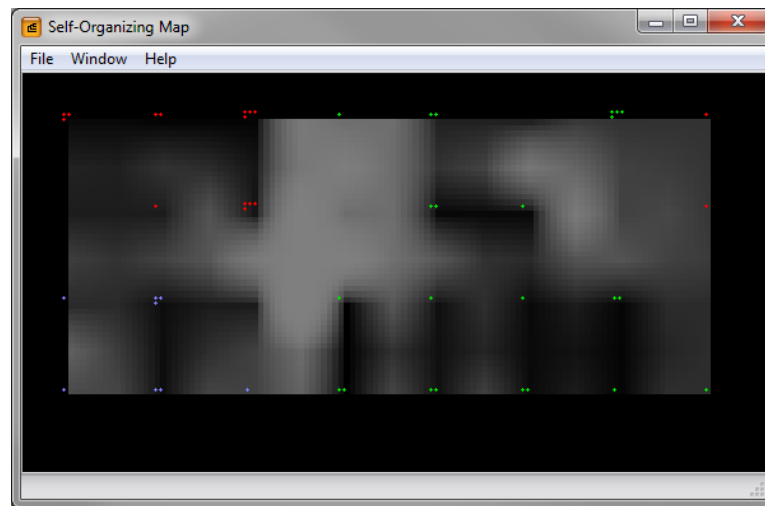


Figure 17.4.10: Self-organizing map.

are joining.

To add entries to an existing SOM, select new entries in the *Main* window and copy them to the clipboard using **Edit > Views > Copy selection** (**Ctrl+C**). In the *Comparison* window, select **Edit > Paste selection** (📄, **Ctrl+V**). The new entries are placed in the *Comparison* window and in the *Self-Organizing Map* window.



An identification based upon a self-organizing map is only reliable if the new entries belong to one of the groups the SOM is based upon. A SOM will always produce a "positive" identification: an unknown profile will **always** find a place in the SOM, i.e. the cell having the highest similarity with the new entry. If, after adding a new entry to a SOM, the entry falls next to a known entry of that SOM, this means only that the new entry has the highest similarity with that particular cell compared to the other cells; it does **not** mean that it is highly related to that entry. Hence, identification based upon a SOM is only recommended if you are sure the unknown entries belong to one of the groups composing the SOM.



Since no new cells can be created in a SOM, one should never add new entries which are known to constitute a group that is not represented in the SOM.

To delete entries from a SOM, select the entries and in the underlying *Comparison* window, select **Edit > Cut selection** (✂, **Ctrl+X**).

The SOM can be printed with the **File > Print...** command, or exported via the clipboard as enhanced metafile using **File > Copy to clipboard**. In these cases, the map colors are inverted, i.e. white corresponds with areas of high similarity, whereas darker shading corresponds with areas of low similarity.

The *Self-Organizing Map* window can be closed with **File > Exit**.

The SOM that was last calculated for an experiment type can be opened again from the *Comparison* window by selecting the experiment type in the *Experiments* panel and selecting **Statistics > Show map**.

Alternatively, click on the SOM analysis in the *Analyses* panel in the *Comparison* window, and use **File > Analysis components > Open** (📄).

Part 18

High-throughput sequence analysis

Chapter 18.1

An introduction to the Power Assembler

The goal of the Power Assembler is to provide a tool within the BioNumerics software that can manage and analyze high-throughput sequencing data. This functionality has been developed to face the computationally challenging problem of the assembly of thousands or millions of short sequences -further called reads- of typically ± 30 -800 nucleotides in length, generated by high-throughput sequencing techniques. This tool targets *de novo* assembly of bacterial genomes, whole bacterial genome resequencing projects, partial targeted resequencing of e.g. human, plant or yeast genomes, high-throughput amplicon sequencing, etc. For the *de novo* assembly, two open-source tools, Velvet and Ray, are made available in the Power Assembler.

- Velvet, a widely adopted de Bruijn graph-based assembly program, uses both single and paired-end reads, and uses coverage information to resolve repeat regions. The final output from Velvet is the assembly file that is directly visualized in the *Assembly panel* of the Power Assembler. Parameter settings and algorithmic information behind the Velvet program are documented in [18.7.5.5.1](#), and have been described in detail by the authors of the software [[43](#)].
- Ray is a parallel short-read assembly program developed to assemble both single and paired-end reads obtained from a combination of sequencing platforms. Parameter settings of Ray are described in [18.7.5.5.2](#). For algorithmic information on Ray reference is made to the original article of the software [[9](#)].

The Power Assembler is fully integrated in the BioNumerics software, which makes it possible to use sequence information already present in the underlying BioNumerics database. This is particularly useful to import e.g. one or more reference genome(s) used for mapping the reads or the start the power assembly analysis directly from the imported sequence read sets in the database. Also, the resulting contig sequences or scaffolds from the power assembly projects can be easily transferred to the database, where the entries can be selected for further analysis. For the follow-up analysis, BioNumerics offers the Chromosome Comparison tools including mutation analysis (SNPs/indels) (see [8.6](#) and [8.7](#)), chromosome annotation and feature searches (see [8.8](#)).

In order to understand the functionality, the descriptions and the settings in the following sections, we first consider the workflow used by the Power Assembler and introduce some terms which will be used further on.

A *power assembly project* is essentially a series of *actions*. The sequence of these actions is defined in the *project pipeline* of the project (Figure [18.1.1](#)).

The project pipeline of a project can be run, stored and changed at any time. Project pipelines can also be exported as *project templates*, which on their turn can be imported into new projects. In this way, a project outline can be applied to multiple data sets, or the same project can be run as a separate "clonal" project using different parameter values. See [18.3.2](#) for more information on the project pipeline.

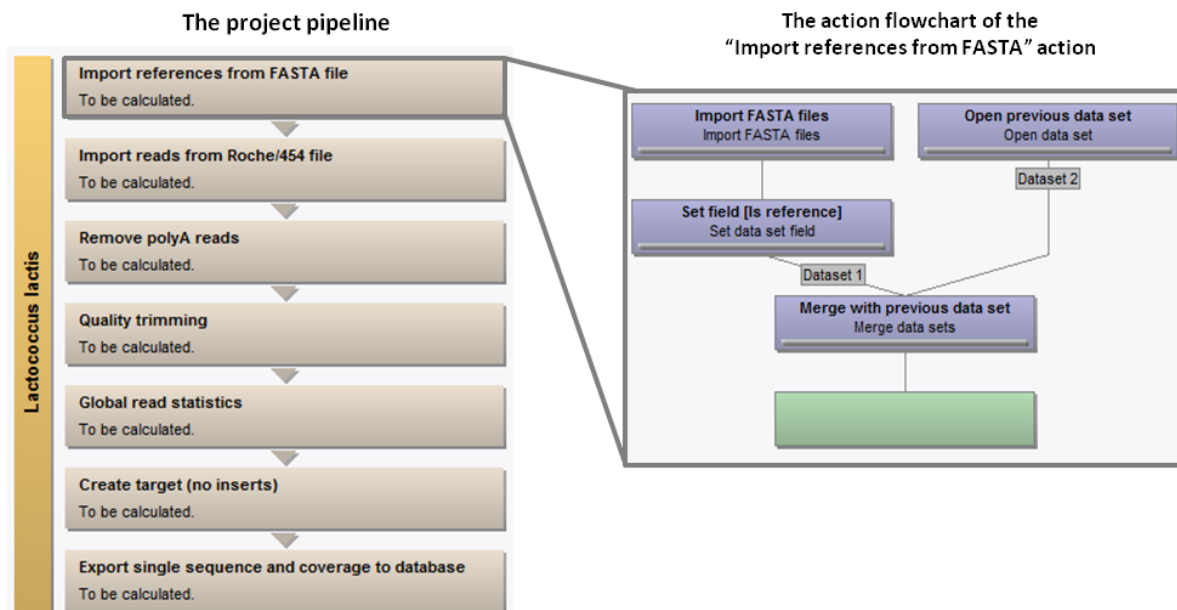


Figure 18.1.1: The project pipeline of the project "Lactococcus lactis" and the action flowchart of the **Import FASTA** action.

Actions that are frequently used in many projects (e.g. the import of read sequences, quality trimming, *de novo* assembly, mapping of reads to one or multiple reference sequence(s), creating contigs based on coverage) are present as *predefined actions* that come with the software (see 18.5). A project is created by adding the necessary predefined actions to the project pipeline (see 18.3.2). Subsequently, the project pipeline can be executed. Besides the use of predefined actions which are present upon installation of the software, there is also the possibility to create *user-defined actions* (18.6). In this case, the project pipeline can be built by a combination of predefined and user-defined actions.

An action is built from a series of processes -further called *operators*-, organized as displayed in the action flowchart of the action (see Figure 18.1.1). The operators are considered the atomic blocks of the Power Assembler.

More than hundred operators are defined which act on sequences, sequence properties, sample properties, data set properties and project properties. The operators are organized in different operator categories (e.g. import & export, trimming, preprocessing, *de novo* assembly, mapping, statistics), and are used to define the action flowcharts. Detailed information on the various operators can be found in 18.7. Many default parameters of the operators are chosen design time, whereas other parameters are very specific, and therefore need to be prompted at the user at runtime. More information on runtime parameter questioning can be found in 18.7.4.2.

The action flowchart displays how the output data set of one operator is used as input for the next operator. The output data set of the last operator in the action flowchart is also saved as the resulting data set of the action (Figure 18.1.1, displayed as the green box). A *data set* consists of multiple sequence records, managed as rows. The field types, the default fields of the sequence record data set and the default fields of the sample set are documented in 18.7.2.4.

After the execution of a project, an overview of the project results is provided by the *summarizing reports* (Figure 18.1.2), displayed for each of the different levels (operator, action and project).

In addition to these reports, the project results can be presented in the following ways:

- The *sequence curves* are based on sequence record properties present in the data set. Typical applications include plotting the sequencing depth over the genome (Figure 18.1.3), or plotting the GC

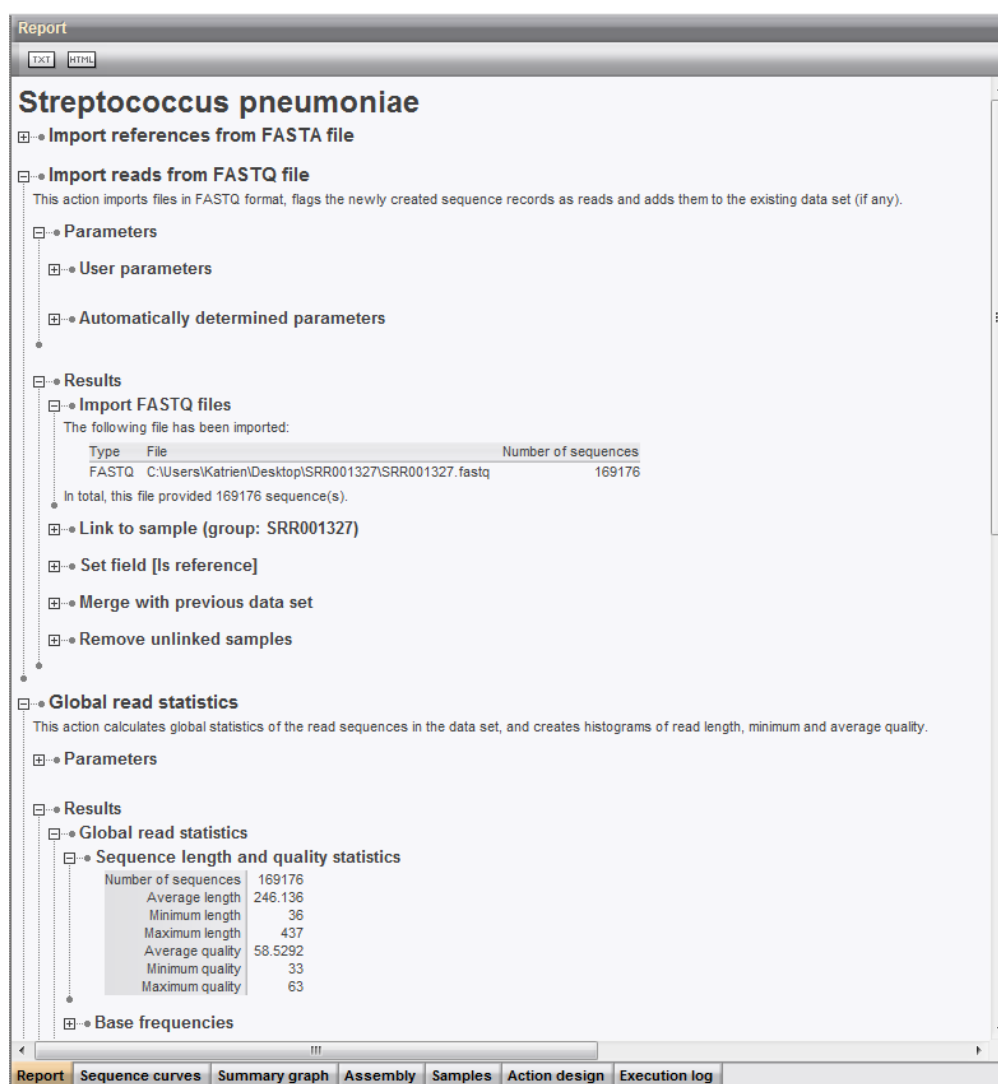


Figure 18.1.2: A project report.

content over the genome. See 18.3.5 for more information on the sequence curves.

- *Summary graphs* (Figure 18.1.4) are used to visualize and provide insight in the properties of the millions of reads produced by high-throughput sequencing techniques. Linked to these summary graphs, parameter values for e.g. trimming can be defined graphically. More information on summary graphs and graphically linked parameter values e.g. trimming thresholds can be found in 18.3.6 and 18.7.4.3, respectively.
- The *assembly view* (Figure 18.1.5) displays the target sequence, the coverage (if calculated) and the mapping of the reads onto the reference. In this assembly view, one can easily scroll through the genome or the areas of interest. See 18.3.7 for more information on the assembly view.

The information flow from the Power Assembler to other BioNumerics functionalities is effected by different export operators that allow the export of sequences, sequence quality information, coverage information and information fields to the underlying BioNumerics database.

The Power Assembler supports the import of raw data obtained by the 454[®] sequencing technology from 454 Life Sciences technology, Roche[®] (<http://www.roche.com>) and the technology from Illumina[®]

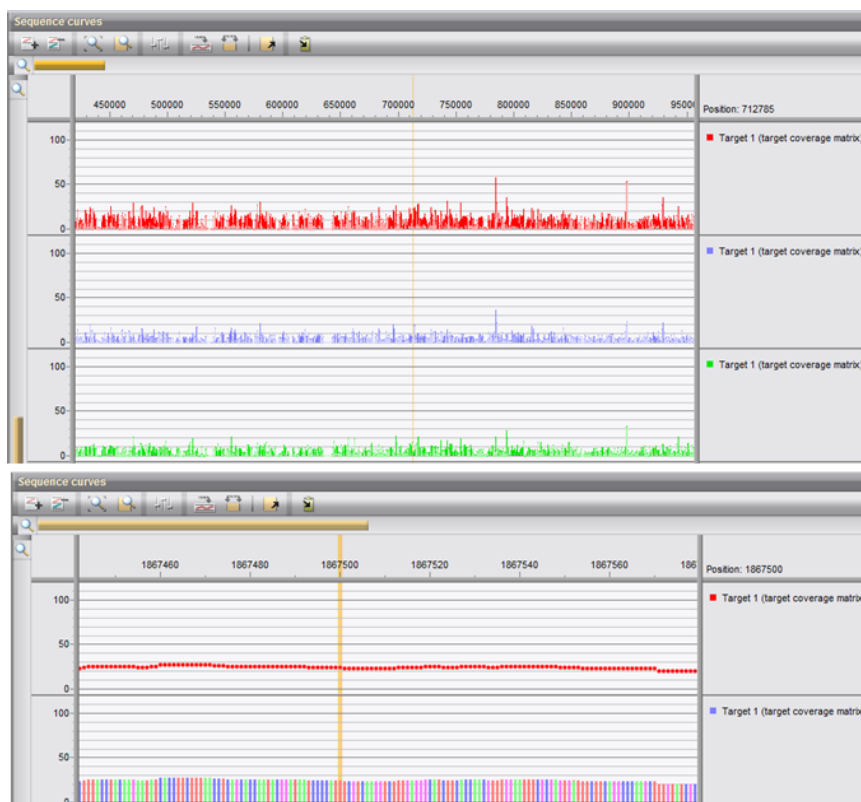


Figure 18.1.3: Upper panel: sequence curves displaying the total coverage, the forward and reverse coverage separately. Lower panel: sequence curves displaying the total coverage and the individual base coverage.

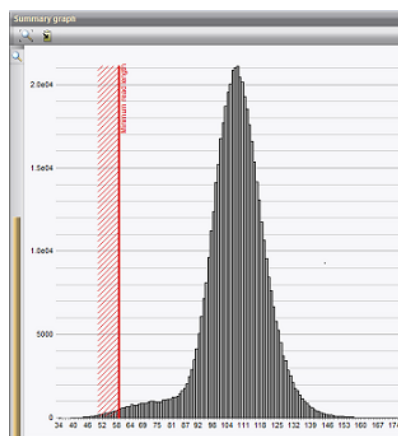


Figure 18.1.4: A summary graph with linked parameter value.

(<http://www.illumina.com>). Data obtained by any other technology (e.g. AB SOLiD data) can be imported as FASTA or FASTQ files.

The different data types need different treatments. Illumina[®] data has relatively low error rates ($\pm 1\%$) and less than 400 bp reads, with almost all the sequencing errors being substitutions, while insertion and deletion errors are much less common. On the other hand, 454 data of typically 400 bp reads has low homopolymer resolution and high indel rates near the ends of the reads. The Power Assembler offers predefined actions specifically tailored for these different short-read sequencing technologies.

In the following sections, the functionality of the Power Assembly will be explained in full detail. For a

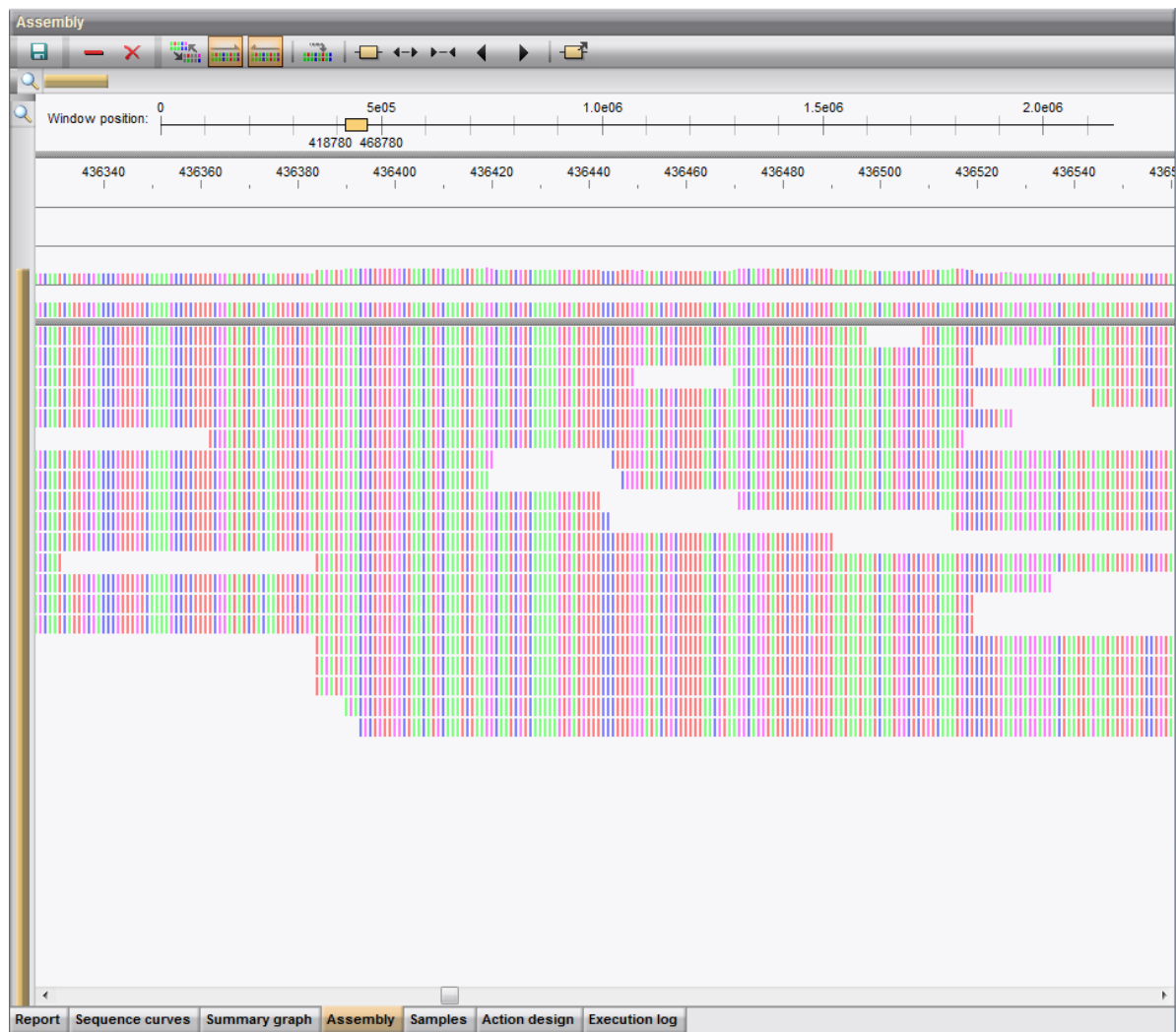


Figure 18.1.5: The assembly view.

quick introduction to the Power Assembler, one can continue directly with the tutorial, where the use of the Power Assembler is illustrated for a resequencing analysis, using a publicly available data set.

Chapter 18.2

Creating a new power assembly

A power assembly can be created from the *Power assemblies* panel in the *Main* window (see Figure 18.2.1).

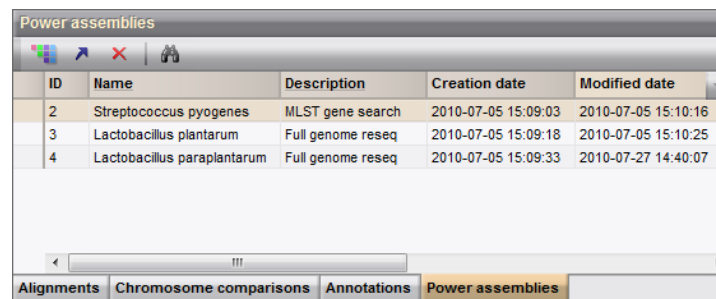


Figure 18.2.1: The *Power assemblies* panel in the *Main* window.

Starting from the *Power assemblies* panel in the *Main* window, a new power assembly is created by selecting *Edit > Create new object...* (+). The *Create new power assembly* dialog box appears (see Figure 18.2.2).

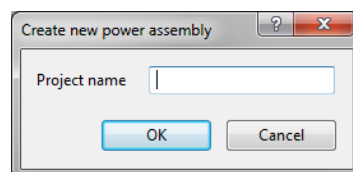


Figure 18.2.2: The *Create new power assembly* dialog box.


In this dialog box, the project name of the new power assembly can be entered.

Once created, the power assembly project is automatically added to the list of power assemblies. This list contains the name and the description of the project, the creation and modification date as well as some other properties (owner, shared, locked) that are automatically monitored. As power assembly projects are database objects (see 3.2), they can be locked, custom fields can be created to store additional project information, all fields can be managed and searched by the user, etc..

Chapter 18.3

The Power assembly window

18.3.1 Panel structure

To open an existing power assembly, double-click the power assembly, or alternatively, highlight the power assembly in the *Power assemblies* panel and select **Edit > Open highlighted object...** (, **Enter**). Multiple power assemblies may be open at the same time.

The *Power assembly* window (see Figure [18.3.1](#)) contains nine dockable panels:

- The *Project pipeline panel* (at the left in the default configuration) contains the consecutive actions which form a power assembly project.
- The *Action data panel* lists *sequence curves*, *summary graphs* and *assemblies* generated by the actions.
- The *Report panel* displays the analysis report, including a summary of the results obtained by the different executed actions and the created summary graphs (see [18.3.3](#)).
- The *Sequence curves panel* displays the sequences or the sequence properties of these sequences (see [18.3.5](#)).
- The *Summary graph panel* is used for visualization and manipulation of the summary graphs, displaying characteristic properties of the reads (e.g. read length distribution, read quality distribution) (see [18.3.6](#)).
- The *Assembly panel* displays the assembly result, where each mapped read is displayed against the reference sequence (see [18.3.7](#)).
- The *Samples panel* is used to define the sample specific multiplex identifiers (see [18.3.8](#)).
- The *Action design panel* is used for display of the action flowchart and the herein existing operators, their descriptions and parameter settings (see [18.3.9](#)).
- The *Execution log panel* displays the information about the executed action and its operators in real time.

The default configuration of the *Power assembly* window for an empty project is presented in Figure [18.3.1](#). One can alter the configuration of the different panels by re-docking. In this way, a personalized configuration can be obtained and stored. The default configuration can be restored at any time (see [2.3.4](#)).

In the following sections, the different panels will be discussed.

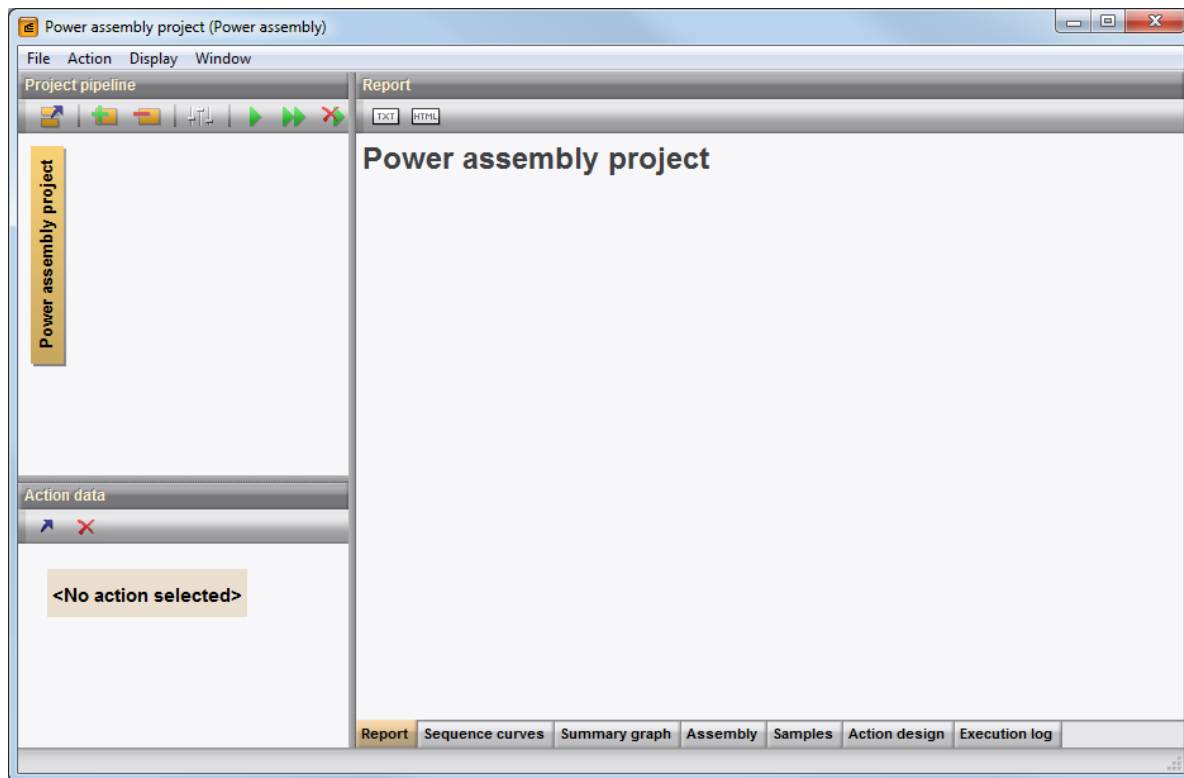


Figure 18.3.1: The *Power assembly* window.

18.3.2 The Project pipeline panel

18.3.2.1 Introduction

The *Project pipeline panel* is used to construct the action pipeline of the project. From this panel, the project is also modified, executed and can be saved at any time. A project can be constructed in various ways, by

- Loading a predefined project template, with or without making further changes to the imported project pipeline
- Loading the project pipeline from another project in the database and altering some of the actions
- Loading the project pipeline from an xml template file
- Starting an empty project and adding or removing different actions to/from the project pipeline ...

18.3.2.2 Working with project templates

Project templates contain a specific analysis workflow. A project template includes the order of the actions to be executed, the operators present in each action, the defined parameters, the settings of the runtime parameters, the settings for the summary graph boundaries ... Project templates are particularly useful when repeatedly performing the same analysis on different data sets.

Unlike the functionality discussed in 18.3.2.6, which offers a quick and easy access to all project pipelines present in the database and is typically used for fine tuning analyses, project templates offer a more controlled environment for the exchange and the use of project pipelines. Project templates contain the same functionality as other objects in the BioNumerics database, including object privileges that can be assigned, e.g. a specific project template can be locked or can be made editable for a specific user group. This makes

the project templates very suitable for use within a production environment where an analysis, after a phase of development, should be considered as a validated analysis for which further changes are not desirable.

By storing a project template, the project pipeline can be loaded permanently, even after the project on which the template is based, was deleted from the database. In contrast to this, the command **File > Load pipeline from project...** only allows to load the pipeline from projects present in the database (see 18.3.2.6 for more information on this command).

A project pipeline can be stored as a template in the database by selecting **File > Store pipeline as template...**

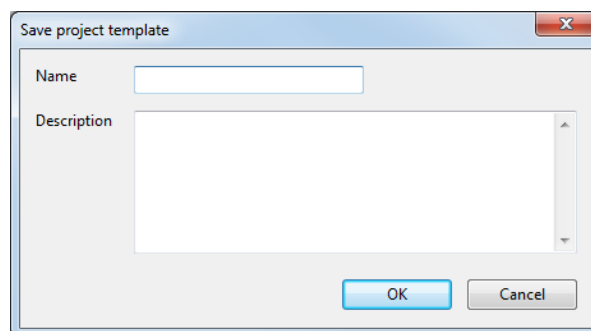



Figure 18.3.2: The *Save project template* dialog box.

In the *Save project template* dialog box, the **Name** of the template and a **Description** can be entered.

A project pipeline that is stored as a template in the database can be loaded into a power assembly by selecting **File > Load pipeline from template...** ().

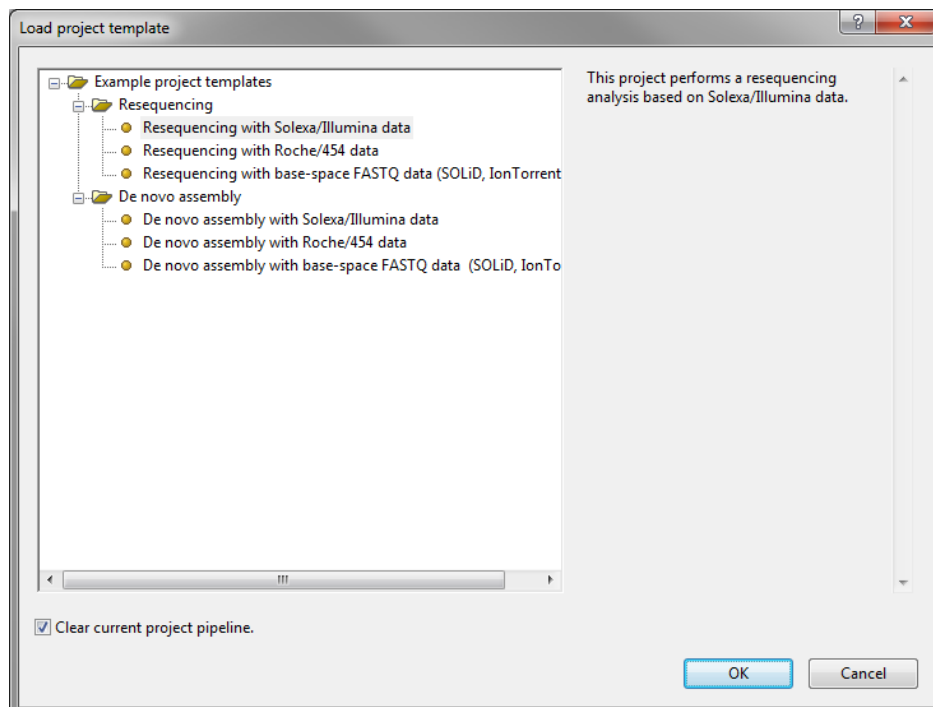


Figure 18.3.3: The *Load project template* dialog box.

In the *Load project template* dialog box, one has the choice to load one of the predefined example project templates (see 18.4), or a user-defined project template that was previously saved to the database. The different items can be collapsed or expanded by pressing the “-” or “+” signs. To load a project template into the power assembly, select the template from the list and press **<OK>** to confirm.

Default, the option **Clear current project pipeline** is checked, which means that the existing project will be cleared and the project template will be loaded in the empty power assembly. With the option **Clear current project pipeline** unchecked, the project pipeline is added to the rear of the existing pipeline.

The user-defined project templates can be managed from the *Remove project templates dialog box*. Select **File > Remove pipeline templates...** to open this dialog.

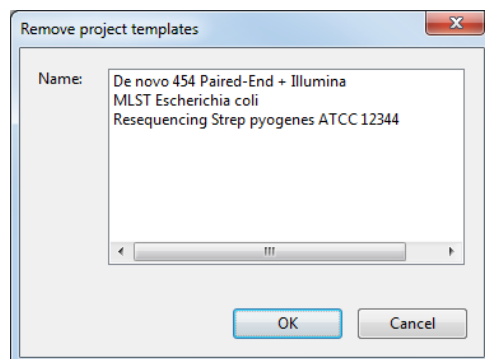


Figure 18.3.4: The *Remove project templates* dialog box.

From the *Remove project templates* dialog box, select the project template to be removed and confirm by pressing **<OK>**.

The functionality of the project templates listed above is valid within one database. However, to exchange project pipelines between different databases or different users, the same XML project templates can be used.

To export a project pipeline, select **File > Export pipeline...**. The *Export project template as XML dialog box* is now displayed.

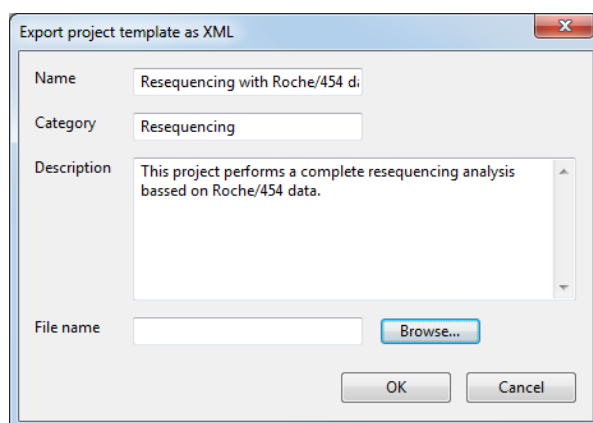


Figure 18.3.5: The *Export project template as XML* dialog box.

In this dialog box, the template **Name**, **Category** and **Description** can be entered. Press **<Browse>** to navigate to the path where the XML file is stored, and enter the **File name**.

To import a project template from XML file, select **File > Import pipeline...**. This opens the *Import project template from XML* dialog box.

In this dialog, browse for the correct XML file that contains the template, and confirm the import.

Default, the option **Clear current project pipeline** is checked, which will clear the existing project and load the project template in the empty power assembly. With the option **Clear current project pipeline** unchecked, the project pipeline is added to the rear of the existing pipeline.

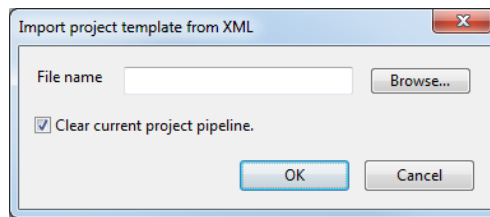


Figure 18.3.6: The *Import project template from XML* dialog box.

18.3.2.3 Adding actions to the project pipeline

To add an action, select **Action > Add action...** (🔧). This opens the *Load action* dialog box (see Figure 18.3.7).

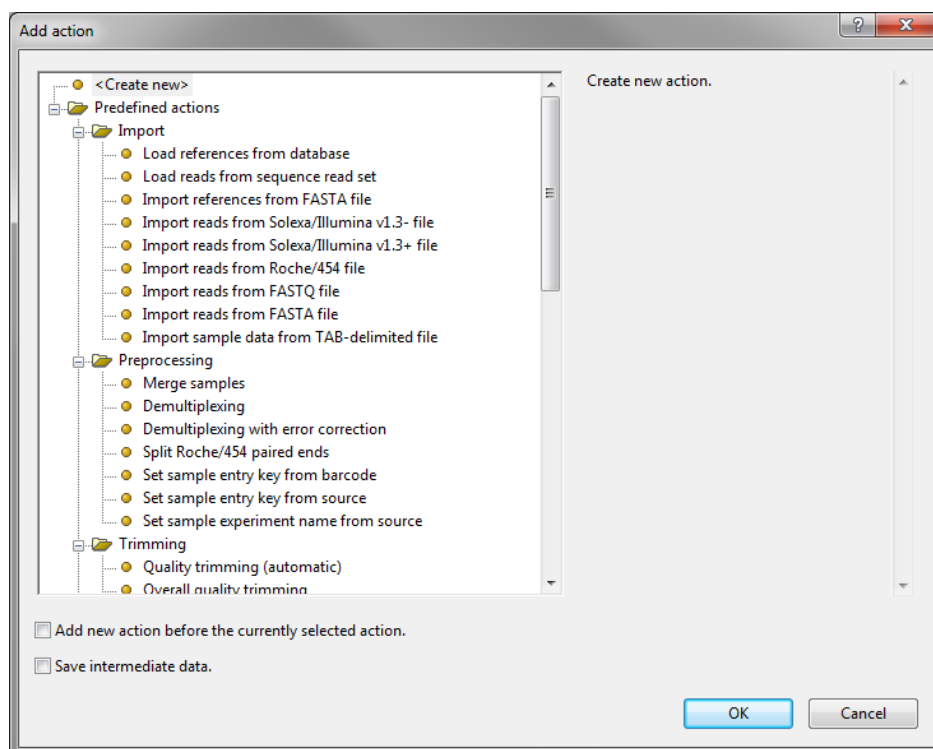


Figure 18.3.7: The *Load action* dialog box.

In the *Load action* dialog box, one can select the action to be added to the project pipeline. Three types of actions can be added.

- The option **Create new** will add a new action with an empty action flowchart to the project pipeline. This command can be used to start the construction of a user-defined action. More information on the modification of actions can be found in [18.6](#).
- The folder **Predefined actions** contains all the factory-installed actions. The predefined actions are organized in a tree structure reflecting the different functional categories. Selecting one of these actions will add a new action with the action flowchart from a factory-installed template. An overview of the predefined actions can be found in [18.5](#).
- The folder **User-defined actions** contains all the user-defined actions that were created and stored as template within the current database. Selecting one of these actions will add a new action with the

operator flowchart from a template created by the user (18.6). The folder only contains the action templates defined within the current database.

To display all predefined actions, expand the tree control by pressing the "+" sign. Clicking on one of the actions updates the description displayed at the right of this dialog box. Any selected action from the tree control can be added to the project pipeline.

By default, an action is added as the last action in the project. However, actions can be inserted at any place in the project pipeline. Therefore, check the option **Add new action before the currently selected action** from the *Load action* dialog box. This will insert the action just before the currently selected action in the project pipeline.

The option **Save intermediate data** defines whether or not the data set resulting from calculation of the action is saved. Default this option is unchecked, which means that no intermediate data will be saved. If an action is added to the pipeline with the option **Save intermediate data** checked, a disk will be displayed in the upper right corner of that action block. It is important to realize that each time an action is (re-)calculated, the calculation starts from the last saved data set in the project pipeline. If no data set is saved in the complete analysis pipeline, each recalculation starts from the first action in the project.

18.3.2.4 Removing actions from the project pipeline

An action can be removed from the project pipeline by selecting **Action > Remove action** (🗑️). This command will automatically remove the highlighted action from the project pipeline, maintaining the order of the remaining actions. If the action to be removed contains sequence curves, summary graphs or assembly information, one can choose whether or not to remove these components.

18.3.2.5 Changing the action's name and description

The properties of a selected action can be changed in the *Action properties* dialog box (Figure 18.3.8) which is launched by selecting **Action > Properties...** (⚙️).

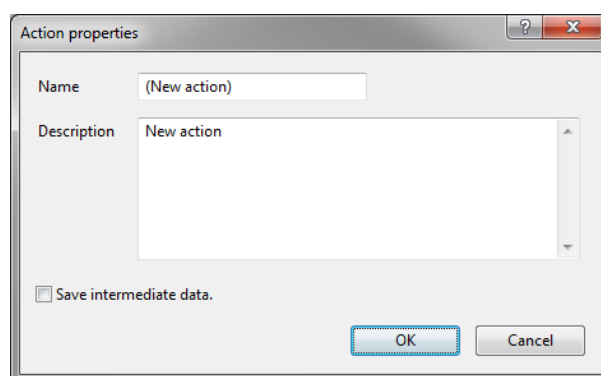


Figure 18.3.8: The *Action properties* dialog box.

In the *Action properties* dialog box the name and description of an action can be changed.

Additionally, one can define if the resulting data set calculated from the action should be saved or not. If the data set is saved, a disk icon will be displayed in the upper right corner of the action block. If no data set is saved, the action block remains empty. It is important to realize that each time an action is (re-)calculated, the calculation starts from the last saved data set in the project pipeline. If no data set is saved in the complete analysis pipeline, each recalculation starts from the first action of the project.

18.3.2.6 Loading an existing project structure into the project pipeline

Within the same BioNumerics database, an entire power assembly project can be copied to a new project. In this way, an identical project including the actions, operators and settings can be created, but e.g. another reference sequence can be used, other parameter values can be applied or other sequence read data sets can be used. An existing project is loaded into a (new) power assembly project by selecting **File > Load pipeline from project....** This launches the *Load pipeline from project* dialog box (see Figure 18.3.9).

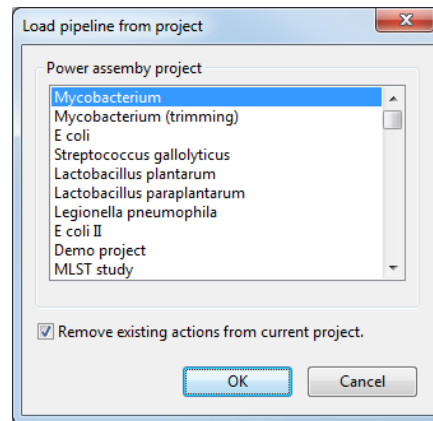


Figure 18.3.9: The *Load pipeline from project* dialog box.

The *Load pipeline from project* dialog box displays all power assembly projects from the current database. If projects are available, the project to copy can be selected from the alphabetical list. If the option **Remove existing actions from current project** is unchecked, actions already present in the current project are preserved, and the actions from the selected project are added in the rear of the current project. If the option is checked, all existing actions are removed from the current project, and the actions from the selected project are loaded to the empty project pipeline of the current project.

The import of a project structure only works within the same database. To import the structure of a project into a power assembly created in another database, project templates should be used (see 18.3.2.2).

18.3.2.7 Executing actions

To execute a single action, select the action from the project pipeline, and select **Action > Execute** (▶). This pops up the dialog box asking for the runtime parameters (18.7.4.2). The wizard pages that contain the parameters can be navigated by the <Next> and <Back> buttons. Pressing <Next> on the last parameter page automatically launches the execution of the action. Information on specific operator parameter settings can be found in 18.5 and 18.7. As an example, the dialog box asking for the runtime parameters for the import of 454[®] read sequences is displayed in Figure 18.3.10.

In addition to the execution of a single action, one can also run a set of actions at once. To this end, select the action where the execution has to start and select **Action > Execute from current** (▶▶). This will start the execution of the selected action and will continue the execution for all subsequent actions. The entire project can be executed by selecting the first action of the project, or no action at all, and selecting **Action > Execute from current** (▶▶).

Note that when executing an action, the software is only capable to execute the action if the data set from the previous action is available, thus saved. If the data set from the preceding action was not saved, the software will look for the latest data set available in the project, and will start the calculation from there. A disk icon displayed in the action block of the project pipeline indicates whether or not intermediate data sets are saved for a specific action.

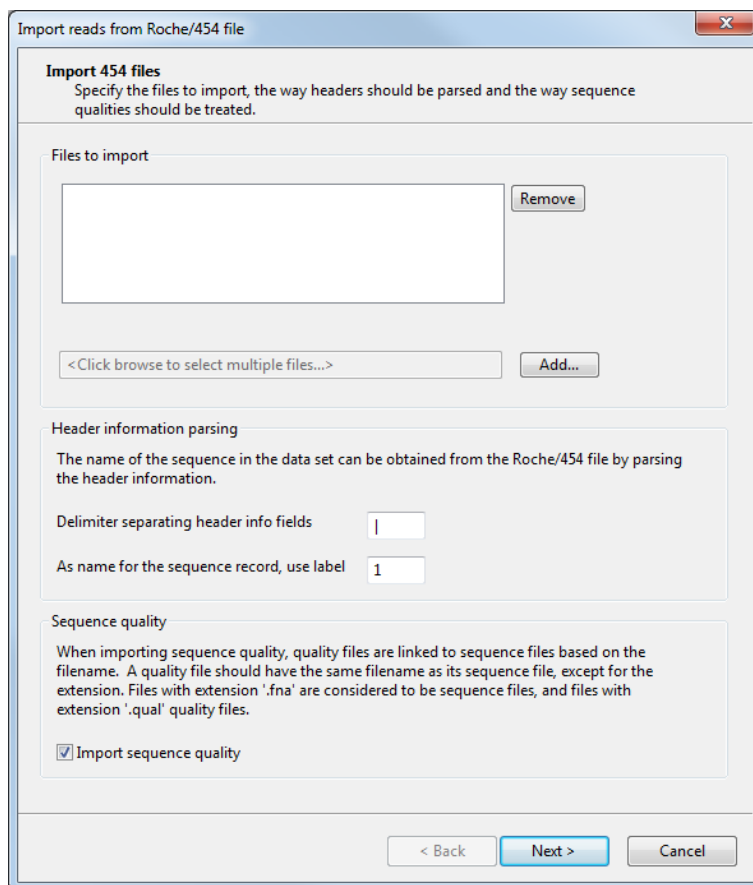


Figure 18.3.10: The *Import Roche/454* runtime parameter dialog box.

18.3.2.8 Aborting the execution of actions

Once the execution of an action of the power assembly project has started, this execution can be aborted at any time by selecting **Action > Cancel execution** (🛑). Please note that, when canceling the calculation, it can take a few seconds before the power assembly project is reactivated.

18.3.2.9 Saving the project pipeline

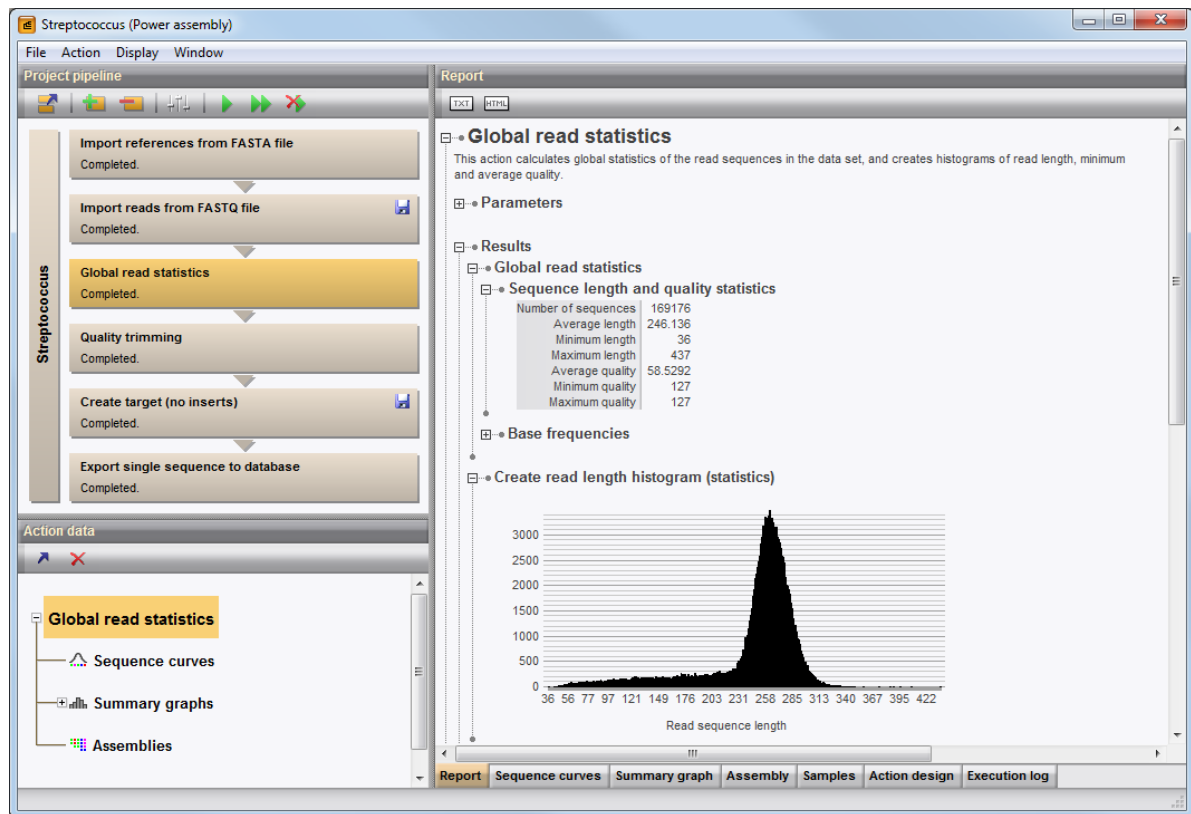
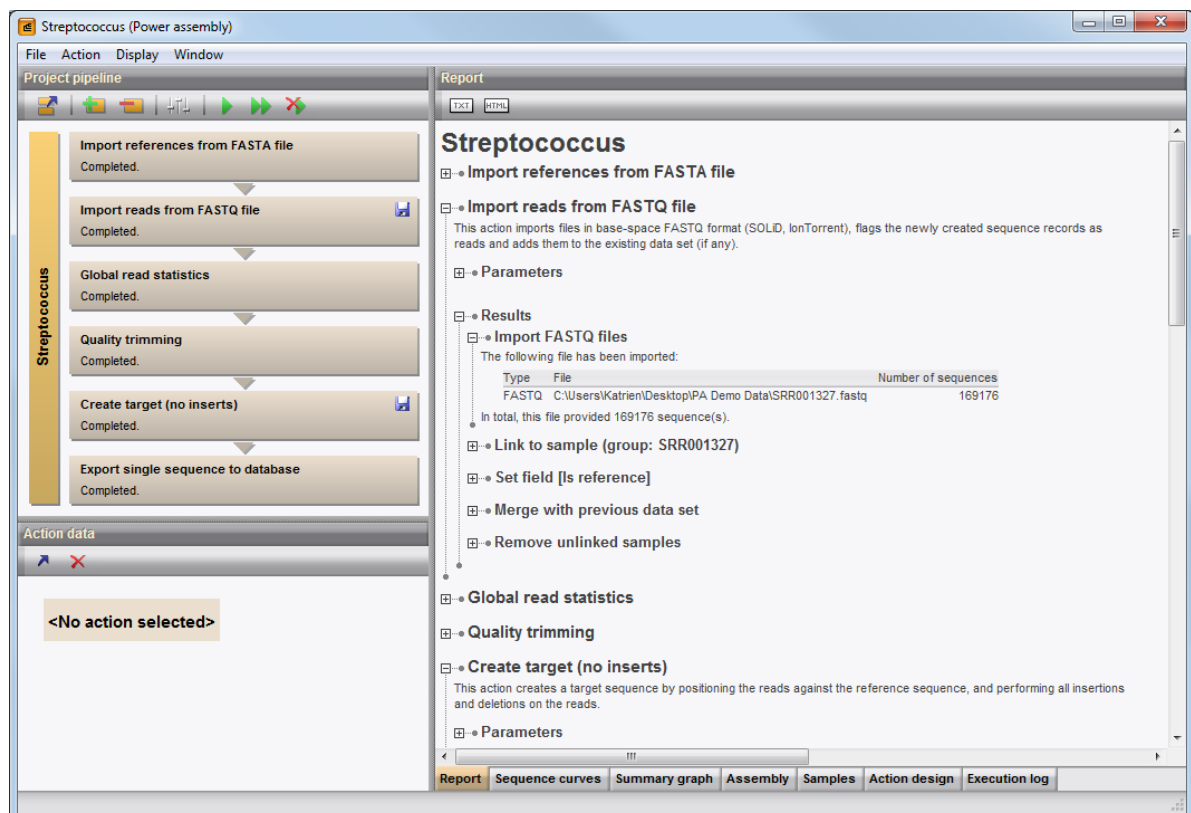
Along with the project pipeline of the power assembly project, also the reports, the summary graphs, the assembly settings and associated data are saved by selecting **File > Save assembly** (💾).

18.3.3 The Report panel

The *Report panel* is the main panel of the Power Assembler and presents an overview of the results generated by the actions. Within the report, detailed information on the items can be collapsed or expanded by pressing the "–" or "+" signs. When selecting an action in the *project pipeline*, this panel displays the results from the selected action (Figure 18.3.11).

To obtain a complete project report, click the project name stretched out vertically at the left side of the project pipeline (Figure 18.3.12). The project report is composed of all the different action reports.

The summary graphs presented in the report can be opened in the *Summary graph panel* by right-clicking the graph and selecting **Open in dedicated window**.

Figure 18.3.11: The *Report* panel: the action report.Figure 18.3.12: The *Report* panel: the project report.

Complete reports can be exported as text file by selecting **File** > **Export report as text** > **Report** (📄). The file `export.txt` is created in the database folder. An information window indicates the location of the saved text file. After confirmation of the export, the `export.txt` file automatically opens.

Alternatively, the report can be exported to HTML by selecting **File** > **Export report as html** > **Report** (📄) and an information window indicates the location of the HTML file. After confirmation of the export, the HTML file automatically pops up in the default internet browser of the system.

Tables from the report can be copied by right-clicking on the table. From the context menu, select **Export table (tab-delimited)** or **Export table (HTML)** to copy the table to the clipboard in the specified format.

18.3.4 The Action data panel

The *Action data panel* displays a tree containing *sequence curves*, *summary graphs* and *assemblies* generated by the highlighted action from the project pipeline. For the sequence curves and the assemblies, the name of the sequence curve or assembly, and the name of the sequence record are displayed in the tree. If no sequence record name is defined, the sequence record key is used. For the summary graphs, only the graph name is displayed (Figure 18.3.13).

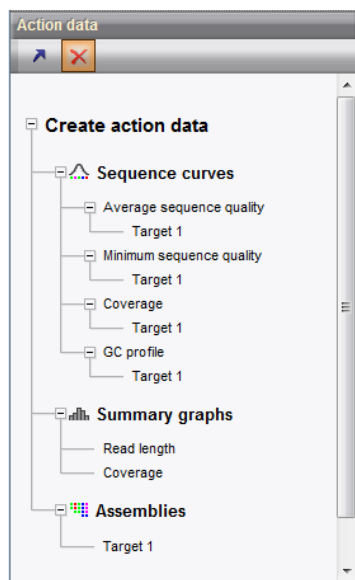


Figure 18.3.13: The Action data panel.

From this panel, sequence curves can be launched to the *Sequence curves panel* (see 18.3.5). Summary graphs can be opened in detail in the *Summary graph panel* (see 18.3.6) and similarly, assemblies can be displayed in the *Assembly panel* (see 18.3.7). To display one of the action data from the *Action data panel*, select the data source and select **Action** > **Show...** (🔍).

One can manually remove the action data present in the *Action data panel* by selecting the node that holds the data and selecting **Action** > **Remove** (✖).

18.3.5 The Sequence curves panel

The *Sequence curves panel* is used to display the sequence curves generated by specific actions. The *Sequence curves panel* combines different channels, each channel containing one or multiple sequence curves.

Following, some examples of sequence curve types.

- *Sequence curves*, displayed as nucleotides or color blocks, depending on the zoom level applied (see Figure 18.3.14).



Figure 18.3.14: Display of a sequence in the *Sequence curves* panel.

- *Sequence quality curves* (see Figure 18.3.15), displaying the sequence quality of the target sequence.

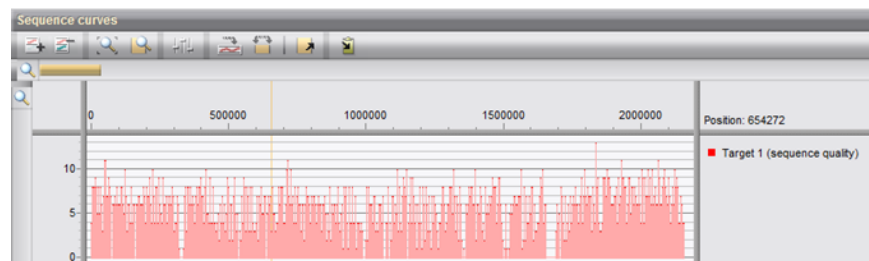


Figure 18.3.15: Display of a sequence quality curve in the *Sequence curves* panel.

- *Coverage curves* (see Figure 18.3.16), displaying the global coverage (red), the forward coverage (blue), as well as the reverse coverage (green). In this example, the coverage curves are combined in one channel, and the reverse coverage is mirrored over the X-axis.

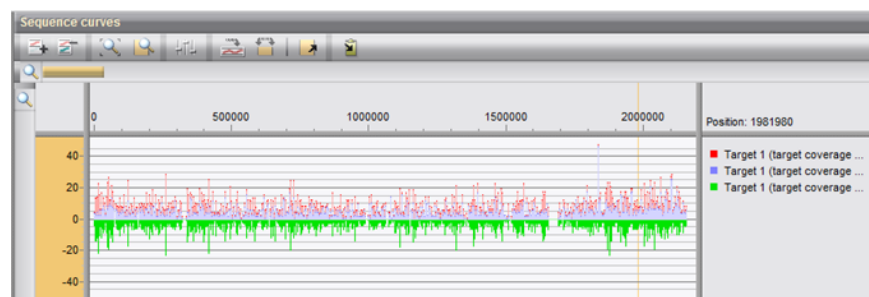


Figure 18.3.16: Display of coverage information in the *Sequence curves* panel.

- *Quality curves* (per base), displaying the base calling quality for each of the sequence positions (see Figure 18.3.17).
- *Regions on a curve* (see Figure 18.3.18, second channel) can be used to visualize specific regions from the sequence. In Figure 18.3.18, the first channel displays the total coverage displayed in red, and regions covered more than 30 times displayed in blue. The same regions, regions covered more than 30 times, are also displayed in the second channel.

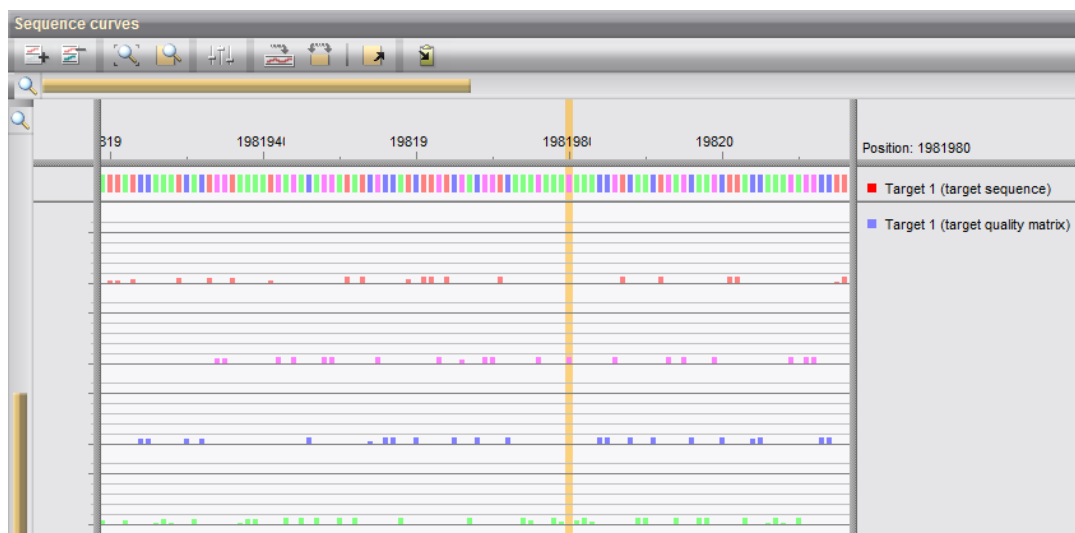


Figure 18.3.17: Display of the sequence and its quality curve (per base) in the *Sequence curves* panel.

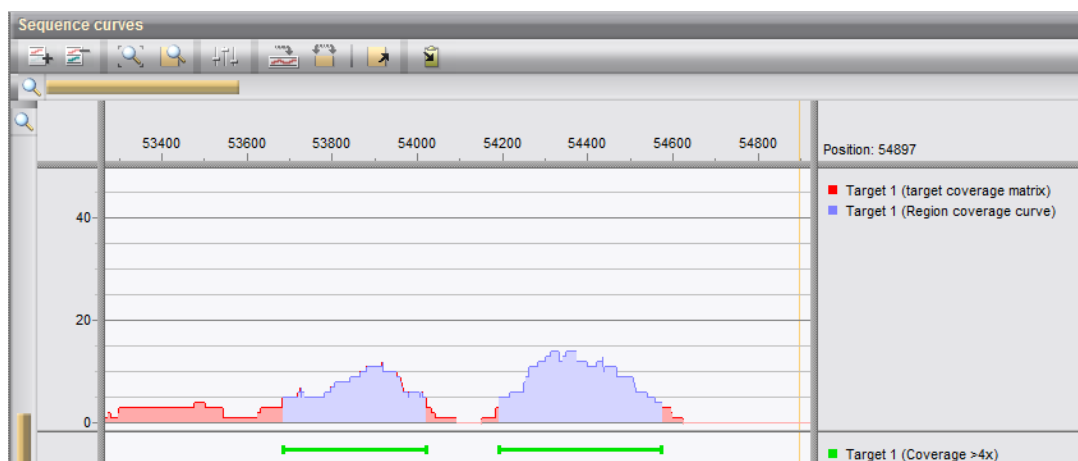


Figure 18.3.18: Display of the coverage and the regions on a curve in the *Sequence curves* panel.

Curves that are displayed in the *Sequence curves* panel are based on a specific profile (e.g. coverage profile, quality profile) over the sequence. The different types of profiles generated by the Power Assembler are listed.

- The *signature profile* is created by scanning a sequence for a specific sequence string. The start position of the defined sequence string is monitored in the profile field.
- The *GC profile* is created by scanning the sequence for base positions that contain the nucleotides G or C.
- Three types of *windowed profiles* exist: the minimum, the average and the maximum windowed profile. Given a profile and a fixed window size, a e.g. windowed average profile is obtained by first taking the average of the first subset of values within the window, and then, the window is shifted forward to create a new subset of numbers, which is averaged. This calculation is repeated over the entire profile range. The windowed profile is created by combining the set of calculated values (in this case, the average values as calculated for each subset). As an example, the calculation of the GC content of a genome is calculated as a windowed average profile from the GC profile.

- The *coverage profile* is used to calculate e.g. the sequencing depth over the genome. The total coverage, as well as the coverage of each of the individual nucleotides, both forward and reverse, can be used in the calculation of this coverage profile. A typical example is the calculation of the target coverage (see Figure 18.3.16).
- The *quality profile* can be calculated from an expression containing the individual nucleotide base calling quality values.

A sequence curve can only be displayed starting from the *Action data panel* of a selected action. The selected sequence curve is displayed by **Action > Show...** (🔍). This will automatically open the *Coverage curve parameter dialog (Create)* dialog box (see Figure 18.3.19).

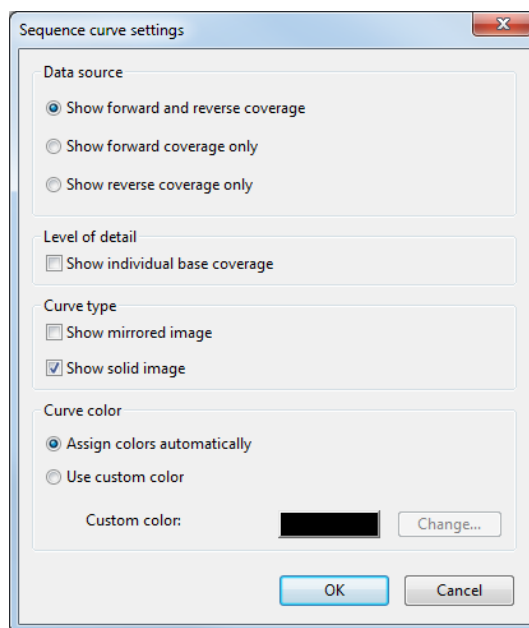


Figure 18.3.19: The *Coverage curve parameter dialog (Create)* dialog box.

The different options present in the *Sequence curves settings* dialog box are described below.

- The **Data source** for showing a target coverage profile can be specified as both forward and reverse data, or, alternatively, the coverage can be displayed based on only the forward or the reversely mapped reads.
- The **Level of detail** for the sequence coverage curve can be specified. The check box **Show individual base coverage** allows displaying the coverage of the reference sequence as a solid line (unchecked) (see upper channel Figure 18.3.20) or as a bar representation displaying the nucleotides of the reads mapped at a specific position (see lower channel Figure 18.3.20). Depending on the horizontal zoom level, the individual base coverage representation will be adjusted in size. In case the zoom level is too low, increase the zoom level to display the individual base coverage.
- The **Curve type** determines whether the image is shown as a curve over the X-axis, or whether the curve is shown mirrored over the X-axis. The combination of both curve types is useful for e.g. plotting the forward/reverse coverage against each other.
- Curve colors can be assigned automatically or can be user-defined. If custom colors are used, one can specify the color settings by pressing <Change>. Automatically, the *Color* dialog box pops up.

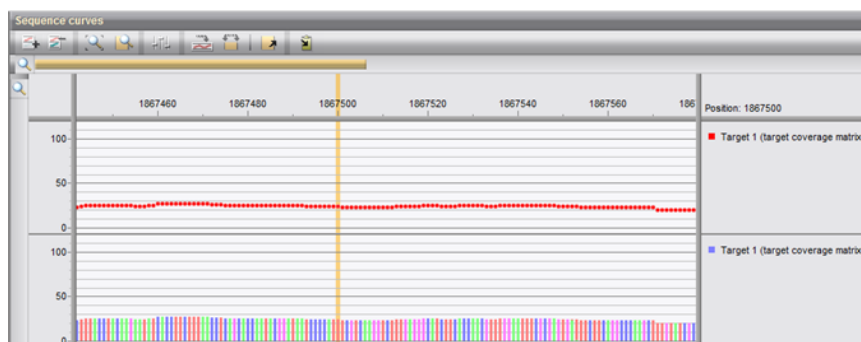


Figure 18.3.20: Individual base coverage

Depending on the sequence curve type to be displayed, some additional display features become available for the *Sequence curves settings* dialog box:

- To display *coverage information*, the **Data source**, the **Level of detail**, the **Curve type** and the **Curve color** can be specified (Figure 18.3.19).
- For the *sequence curve*, the **Curve type** and the **Curve color** can be set.
- To display *sequence regions*, only the **Curve color** can be specified.

To change the display options of a sequence curve after it was added to the *Sequence curves panel*, select the sequence channel by clicking in front of the sequence channel, and select **Display > Sequence curves > Channel properties...** (⚙️). This calls the *Sequence curve channel properties dialog box* (see Figure 18.3.21).

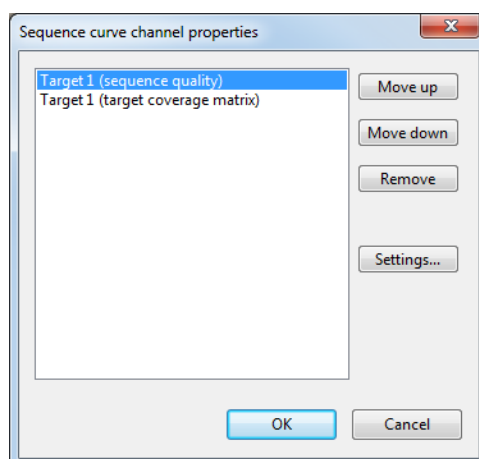


Figure 18.3.21: The *Sequence curve channel properties* dialog box.

All sequence curves displayed in the selected channel, are presented in this dialog box.

- To change the display order of the sequence curves, use **<Move up>** and **<Move down>**.
- To delete a curve from the channel, press **<Remove>**.
- To call the *Sequence curve settings* dialog box, press **<Settings>**. As stated before, depending on the sequence curve type that is displayed, different dialog boxes will pop up.

Other sequence curves can be added to the same channel. Therefore, first select a channel from the *Sequence curves panel*, then select the sequence curve to be added from the *Action data panel* and open this sequence curve. If two sequence curve types cannot be combined, an error is shown.

Once a sequence curve is displayed, the same display color will be used each time the sequence curve is loaded to a new or existing channel. The sequence record name is displayed in the legend at the right.

To create a new channel in the *Sequence curves panel*, select **Display** > **Sequence curves** > **Add a new channel** (🔍).

To remove one or multiple channels, first select all channels to be removed by holding the **Ctrl** key. Subsequently, the channels can be removed from the *Sequence curves panel* by selecting **Display** > **Sequence curves** > **Remove selected channels** (🗑️).

The full sequence curve can be fitted in the *Sequence curves panel* by selecting **Display** > **Sequence curves** > **Zoom to fit** (🔍). Alternatively, use the zoom slider on top of the *Sequence curves panel* to adjust the display region of the curves and use the zoom slider on the left to alter the height of the channel. If the zoom level is too low to display the specific curve, the channel is shown hatched.

A selected part of the sequence curve can be fitted in the *Sequence curves panel* as well: firstly, select a fragment of the sequence curve by selecting the start and stop position within the curve while holding the **Shift** key. Then, the selected sequence fragment can be stretched to fit in the *Sequence curves panel* by selecting **Display** > **Sequence curves** > **Zoom to selection** (🔍).

To jump to a specific position on the sequence curve, select **Display** > **Sequence curves** > **Go to...** (🔍). The *Go to curve position* dialog box then pops up (see Figure 18.3.22).

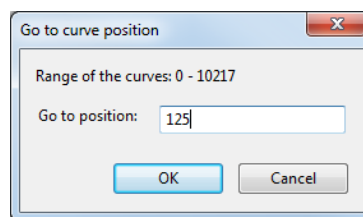


Figure 18.3.22: The *Go to curve position* dialog box.

In this dialog, the range of the selected sequence curve is indicated by the **Range of the curves**. The position to jump to can be entered in the **Go to position** field. After confirmation, the specified position is centered in the *Sequence curves panel*.

A specific part of the sequence curve can be fitted in the *Sequence curves panel* by selecting **Display** > **Sequence curves** > **Select...** (🔍).

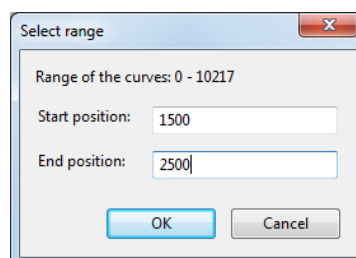


Figure 18.3.23: The *Select range* dialog box.

In the *Select range* dialog box (see Figure 18.3.23) the range of the selected sequence curve is indicated by the **Range of the curves**. The start and stop position of the range to be displayed should be entered in the **Start position** and **Stop position** fields, respectively. After confirmation, the specified region is fitted in

the *Sequence curves panel*. If both an assembly and sequence curves are created by the same action, the viewport can be transferred between the assembly and the displayed sequence curves, and vice versa. See 18.3.7 for more information on the assembly viewport. From within the *Sequence curves panel*, a selected part of the sequence can be transferred as assembly viewport to the *Assembly panel* by selecting **Display** > *Sequence curves* > **Transfer selected range to assembly** (📁).

To copy (a part of) the sequence curve(s) to the clipboard, select **File** > **Copy selected channel to clipboard...** (📄). To export a single sequence curve channel, first select this channel, and then select **File** > **Copy selected channel to clipboard...** (📄). Also multiple channels, selected by holding the **Ctrl** key, can be exported by this command. To create the image from all sequence curves, make sure no channel is selected.

In the *Copy image* dialog box that appears (see Figure 18.3.24), the export settings can be altered.

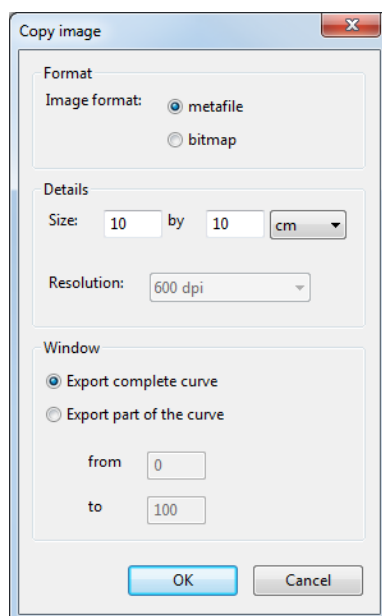


Figure 18.3.24: The *Copy image* dialog box.

The export function allows one to copy the sequence curves as a *metafile* or as a *bitmap*. To copy the image as metafile, define the size of the image (cm or inch) to be created (*Details*), and define the window size of the sequence curves to be exported (*Window*). **Export complete curve** will create the image for the full sequence length, whereas **Export part of the curve** allows the user to define a start and stop position for the image export.

To copy the image as bitmap, first select bitmap as image format type. Bitmap export details that were disabled for the metafile export then become active. Similar to the metafile export, define the size of the image and the window to be exported. Additionally define the resolution (150, 300, 600 or 1200 dpi).

18.3.6 The Summary graph panel

The size of the data sets used in high-throughput sequencing projects can be immense. The most efficient method to gain insight in the characteristics of millions of reads is to visualize the characteristic properties of the reads in the *Summary graph panel*. Different types of graphs can be created, for example

- *Read quality histogram*,
- *Read length histogram*,
- *Coverage histogram*,

- *Sequence identity histogram*,
- *Degeneracy histogram*.

A summary graph is created by an action of the power assembly project. When selecting an action in the *Project pipeline panel*, all graphs generated within the selected action are displayed in the *Action data panel* under **Summary graphs**. When executing the action, the summary graphs will be regenerated. Graphs are automatically saved along with the power assembly project.

To visualize a graph in the *Summary graph panel*, select a graph name from the *Action data panel*, and select **Action > Show...** (🔍).

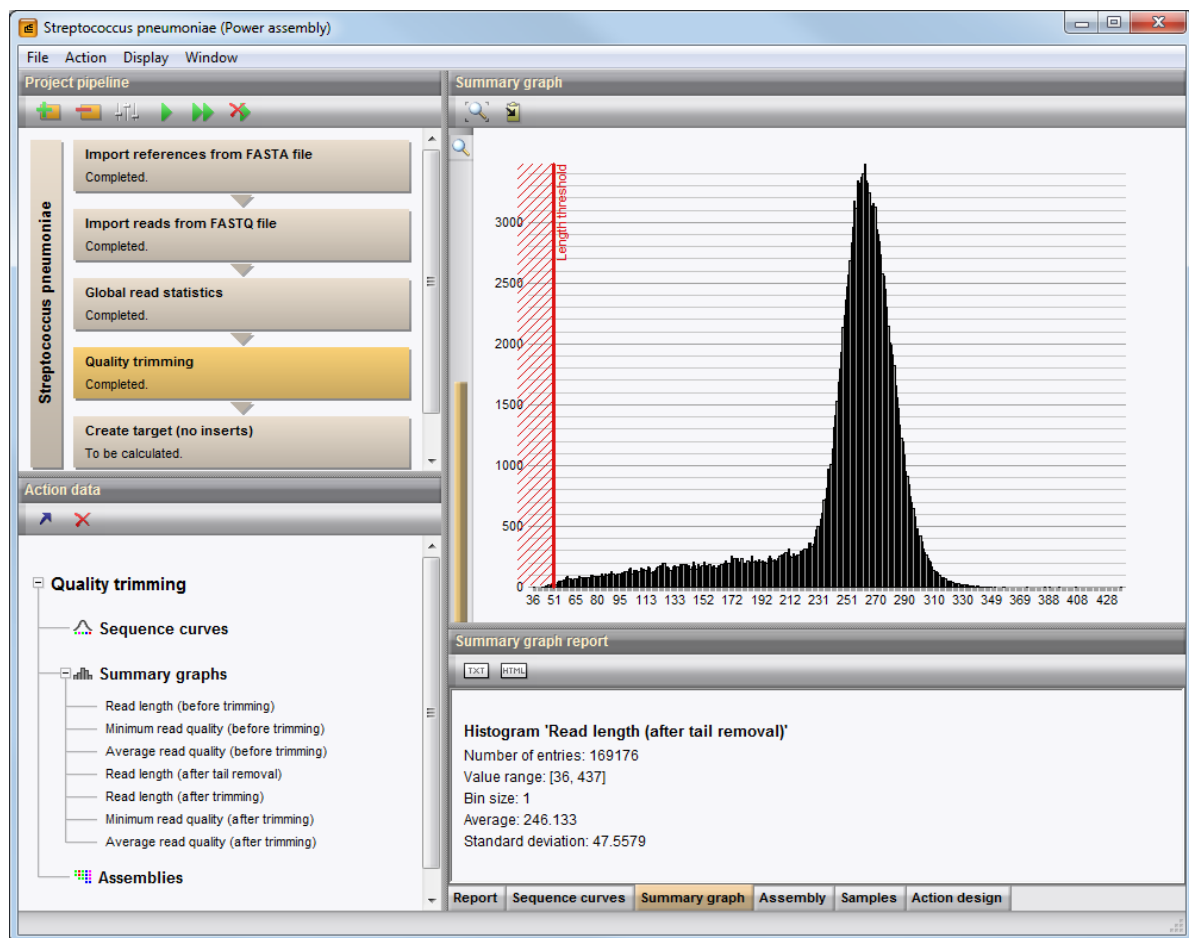


Figure 18.3.25: The *Summary graph panel* displaying the read length after tail removal, and the linked trimming operator parameter.

The *Summary graph panel* consists of two sub-panels (Figure 18.3.25): the *Summary graph panel*, which displays the plot, and the *Summary graph report*, summarizing some characteristics particular to the graph (e.g. the number of entries, the value range, the average value and the bin size).

Some of the summary graphs also display flexible thresholds. These thresholds are graphically linked operator parameter values which are indicated by a red solid line and are designed a descriptive name indicative to the type of threshold (see Figure 18.3.25, *Length threshold*). The region at the non-hatched area passes the filtering that runs in a particular action of the power assembly project. More information on defining graphically linked flexible thresholds (also called summary graph boundaries) can be found in 18.7.4.3.

The size of the graph is fit to the panel by selecting **Display > Summary graphs > Zoom to fit** (🔍). The zoom slider at the left of the *Summary plot panel* can be used to zoom proportionally in the horizontal direction.

To copy the graph to the clipboard, select **File > Copy summary plot to clipboard...** (📄). This calls the *Copy image dialog box* (see Figure 18.3.26).

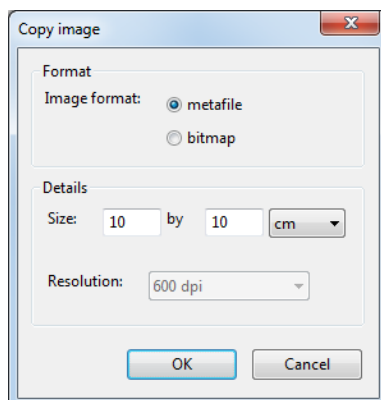


Figure 18.3.26: The *Copy image dialog box*.

This dialog box displays all the export settings. The graph can be exported as a bitmap or as a metafile. When copying the image as bitmap, the maximum size of the image (cm or inch), and the resolution (150, 300, 600 or 1200 dpi) can be defined. To copy the image as metafile, only the size can be adjusted.

Similar to the project and action reports, the summary graph reports as displayed within the *Summary plot report panel* can be exported as text or HTML file by selecting **File > Export report as text > Summary graph report** (📄) or **File > Export report as html > Summary graph report** (📄), respectively.

18.3.7 The Assembly panel

Each assembly is based on one (or multiple) reference sequence(s). Once calculated, the mapping of the reads onto this reference sequence can be visualized in the *Assembly panel*. Hereto, first select the action from the *Project pipeline* panel that created the assembly map (see 18.3.2). This updates the *Action data tree* where the name of the assembly is now displayed. To open the assembly in the *Assembly panel*, select the assembly and select **Action > Show...** (🔍).

The *Assembly panel* is divided in three horizontal parts (see Figure 18.3.27).

- The upper part displays the position of the assembly viewport on the reference sequence. The viewport, displayed as a yellow bar, defines a restricted part of the target sequence that is displayed in detail in the lower panels.
- The middle part shows that part of the target sequence inside the assembly viewport, and, if calculated, the coverage (represented as the height of the colored bars).
- The lower part of the *Assembly panel* displays the mapping for each individual read.

The numbering, displayed in the two upper parts of the panel, corresponds to the position on the target sequence (first position counted as zero). The assembly viewport parameters can be adjusted by using the buttons at the top of the *Assembly panel*. When moving the mouse pointer over the ends of the yellow viewport, a double arrow pops up, allowing the user to manually shrink or enlarge the assembly viewport. The viewport range is automatically updated in the other two panels.

It is possible to jump to a specific position in the assembly by selecting **Display > Assembly > Go to...** (🔍). In the *Go to assembly position* dialog box that pops up (see Figure 18.3.28), the position of interest can be entered.

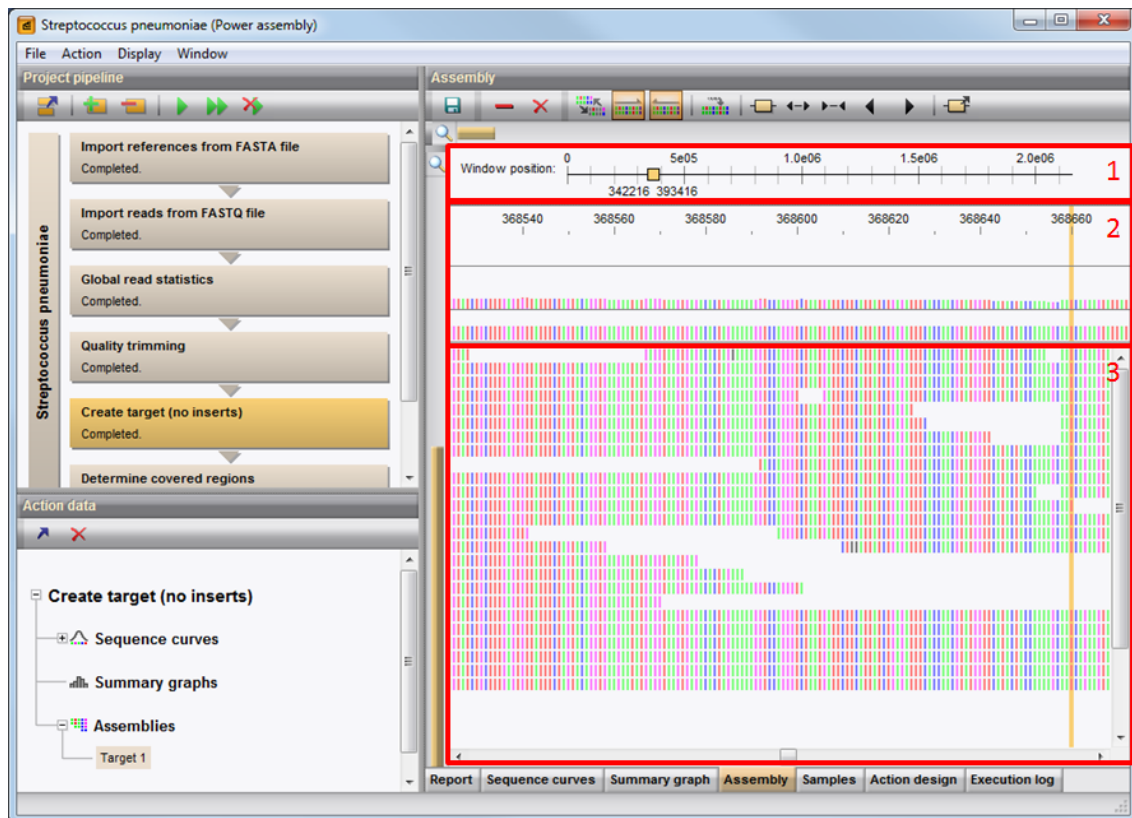
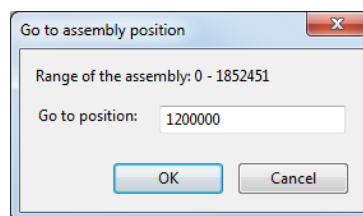
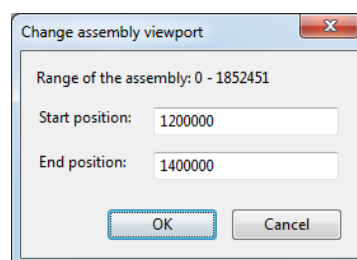


Figure 18.3.27: The Assembly panel.



Figure 18.3.28: The *Go to assembly position* dialog box.



In this dialog box, the range of the assembly is indicated and the position to jump to can be entered in the *Go to position* field. After confirmation, the specified position is centered in the *Assembly panel*. If the requested position lies outside the current viewport, the software will automatically adjust the viewport.


A specific part of the assembly can be set as viewport by selecting **Display > Assembly > Set viewport...** (🖥️). This opens the *Change assembly viewport* dialog box (see Figure 18.3.29).

Figure 18.3.29: The *Change assembly viewport* dialog box.




In the *Change assembly viewport* dialog box (see Figure 18.3.29) the range of the selected assembly is indicated. The start and stop position of the range to be displayed should be entered in the **Start position** and **Stop position** fields, respectively. After confirmation, the specified region is defined as viewport in the *Assembly panel*.

One can also select **Display > Assembly > Enlarge assembly viewport** () or **Display > Assembly > Shrink assembly viewport** (), to enlarge or shrink the assembly view window by half of the current window size, respectively.

To move the assembly viewport to the left or to the right, move the mouse cursor over the yellow bar and when the four-headed arrow appears, click and drag the assembly viewport to another position. Alternatively, move the assembly viewport window to the left or to right by selecting **Display > Assembly > Move assembly viewport to left** () or **Display > Assembly > Move assembly viewport to right** (), respectively. This operator will move the assembly viewport to adjacent positions defined by one quarter of the assembly viewport size.



If one action generates both an assembly and sequence curves based on the same reference sequence positions, the viewport can be transferred between the assembly from the *Assembly panel* and the displayed sequence curves from the *Sequence curves panel*. Selecting the assembly viewport (defined in the *Assembly panel*) on the sequence curves is achieved by **Display > Assembly > Select assembly viewport on curves** ()


By default, the zoom of the *Assembly panel* is set to nucleotide level. To zoom in or out on the assembly viewport, one can use the zoom slider on the left of the *Assembly panel* to zoom vertically (also **Ctrl+scroll**) and the zoom slider on top of the *Assembly panel* to zoom horizontally (also **Shift+scroll**). For low zoom levels, the nucleotides are replaced by colored squares having the same color as predefined in the sequence display settings (see 8.1.2.1).

Nucleotide differences in the reads compared to the reference sequence can be visualized by **Display > Assembly > Show differences only** () (Figure 18.3.30). In this way, all the nucleotides from the reads that are identical to the nucleotide in the reference sequence are shown in gray and only the nucleotide differences in the reads are highlighted. By default, both forward and reverse mapped reads are displayed in the *Assembly panel*. To display only the forward or reverse mapped reads, select **Display > Assembly > Show forward mapped reads** () or **Display > Assembly > Show reverse mapped reads** (), respectively.

Within a project, reads can be automatically excluded from the data set based on low average quality, low minimum quality, sequence length, etc. (18.5). These data preprocessing steps (usually called trimming) are executed by specific actions. However, when visually checking the assembly, one may want to manually exclude a selected sequence read based on its position or behavior in the assembly. Multiple reads can be selected by holding the **Ctrl** key.

Two different exclusion actions can be performed:

- Any selected read in the *Assembly panel* can be removed from the assembly by **Display > Assembly > Remove read** (, **Del**). If multiple reads are selected, they will all be removed without warning. Removing reads cannot be undone without recalculation of the project. Removed reads are still present in the data set, and therefore, they will be re-mapped when the project is recalculated.
- A selected read can also be permanently deleted from the data set. To delete the reads, select **Display > Assembly > Delete read** (, **Shift+Del**). If multiple reads are selected, they will all be deleted without warning. Deleted reads cannot be re-called later.

To save the settings of the assembly viewport for subsequent analysis, select **File > Save assembly** ()

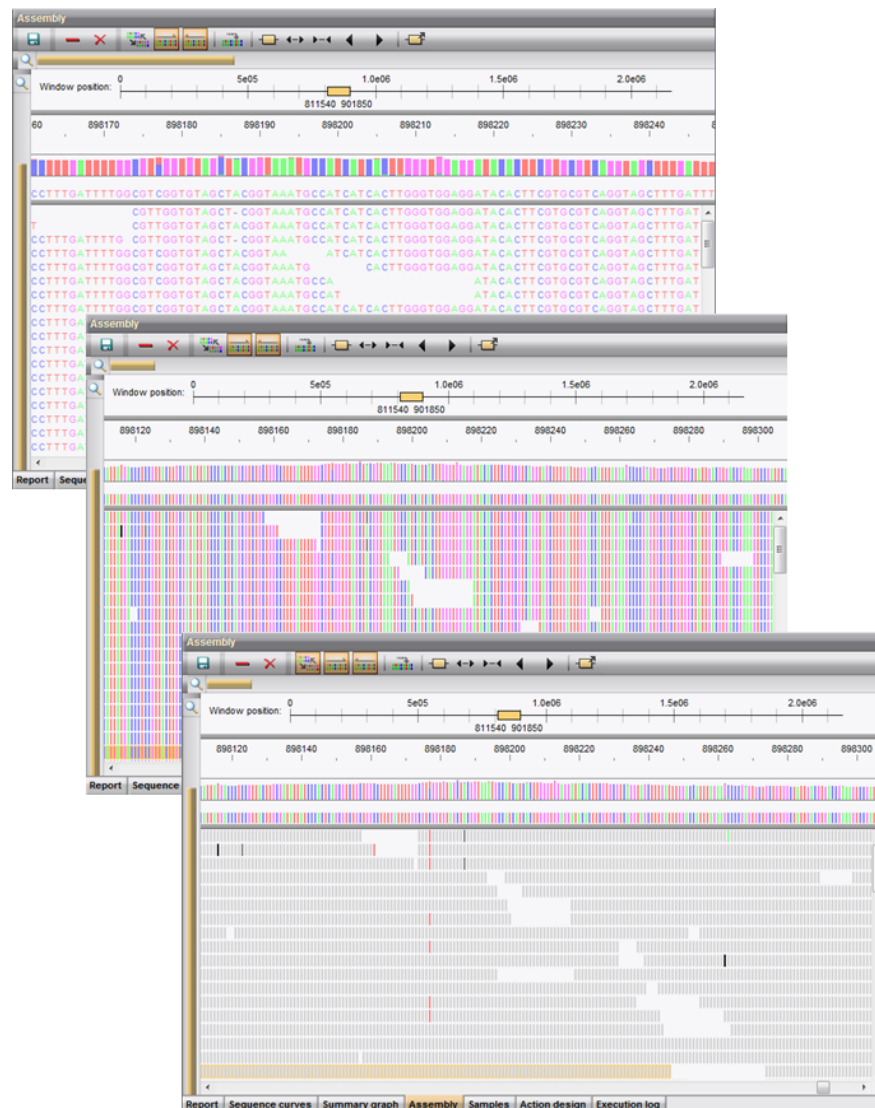


Figure 18.3.30: The different views of the *Assembly* panel.

18.3.8 The Samples panel

Present sequencing technologies allow simultaneous sequencing of multiple samples on a single plate or slide while preserving the unique identity of each sample. This unique sample identity is maintained by the use of multiplex identifiers (MIDs). In general, the reads can be recognized as belonging to a separate sample, based on a specific sample tag. Data processing based on these MIDs is supported by BioNumerics (see Figure 18.3.31).

If samples are used in a power assembly project, an overview of the defined samples is displayed in the *Samples panel* (Figure 18.3.32).

In the *Samples panel* (Figure 18.3.32), the link between a sequence record from the data set and a sample is stored. This panel will only be functional when multiple samples are processed within the same project. Using this information, the groups of sequence records belonging to a specific sample can be identified, sorted and processed in parallel. Each sample group serves as input data set for the actions to be executed. This way, the same pipeline of actions can be run for each sample separately. If desired, the results for each sample can be exported into separate BioNumerics database entries and/or experiments.

The *Samples panel* contains three default fields:

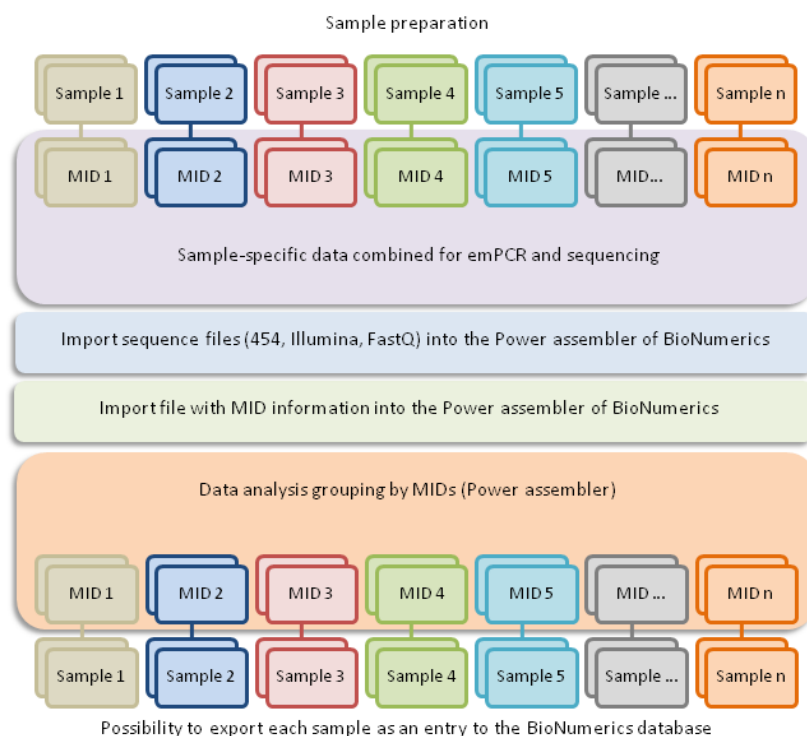


Figure 18.3.31: The use of MIDs.

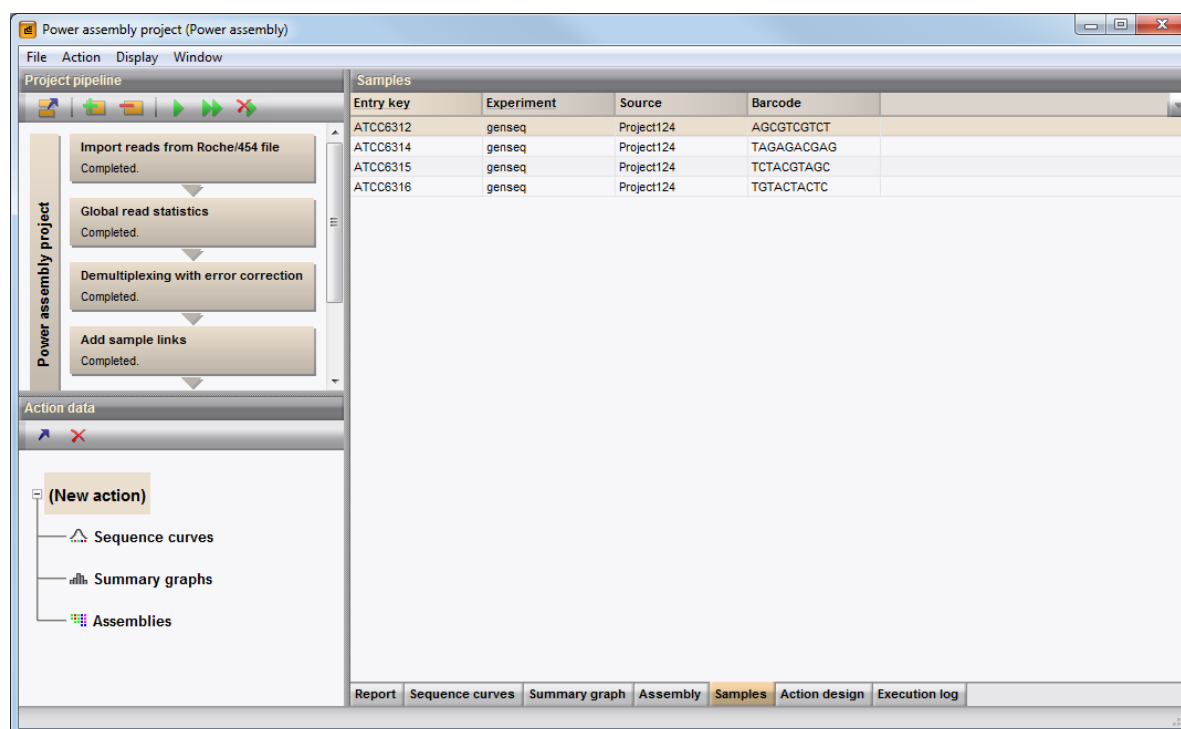


Figure 18.3.32: The *Samples* panel.

- The *Entry key* contains the entry key information used by BioNumerics when creating a new record in the database.
- The *Experiment* contains the name of the sequence experiment type, used by BioNumerics when creating an experiment in the database.

- The *Source* field is used for saving the file name of the import file.

Both the *Entry key* and the *Experiment* information are important for the export of sample information to the BioNumerics database as the export requires both fields defined in the *Samples panel*, to match the entry information and the sequence experiment type present in the database. For example, if multiple sequences need to be exported to multiple existing entries in the database, the entry keys defined in the *Samples panel* should be identical to the ones from the database. Similarly, to export e.g. multiple sequence loci to a single entry in the database, all samples defining the loci should contain the same *Entry key*, but different *Experiment* information.

Sample information can only be imported from a tab-delimited file.

The sample information present in the *Samples panel* from the example (Figure 18.3.32), displays data imported from the same data source (“Project124”), but from different strains (*Entry key*). In this example, the reads are separated according to the defined barcode. For each of the samples, a new entry, defined by the entry key, will be created in the database, and the target sequence will be exported to the “genseq” sequence experiment type.

18.3.9 The Action design panel

The *Action design panel* is the panel where actions are composed by combining different operators. This panel will only be used if user-specific actions need to be created, or if predefined actions need to be customized. As already mentioned in the introduction, power assembly projects are created by combining actions, and these actions consist of a combination of operators. The operators provide the basic functionalities to create dedicated actions.

The *Action design panel* contains three separate sub-panels (see Figure 18.3.33):

- The *Operators panel* gives an overview of the defined operators, organized in different operator categories (e.g. import & export, trimming, preprocessing, mapping, statistics).
- The *Action flowchart panel* displays a graphical overview of the successive operators to be executed in the action.
- The *Operator report panel* shows a detailed summary for the highlighted operator in the action flowchart.

In the *Operators panel*, the available operators are ranked into different functional categories. Selecting an operator from the tree structure automatically updates the lower part of this panel, where a description of the selected operator is displayed. The operator categories and their operators are extensively documented in 18.7.

In the *Action flowchart panel*, each operator is presented by a purple rectangle and the selected operator is highlighted by a red box. The first line shows the user-defined operator name, as set in the general parameters of the operator (see 18.7.2.2 for more information on the general operator parameters). This operator name is the name that is displayed in the reports, and that is used e.g. to refer to this operator when defining graphic parameter links. The second line shows the operator name, factory-defined within the Power Assembler. At the bottom of the operator block, a progress bar is displayed which becomes active when executing the operator.

Operator blocks are connected by lines which indicate the operation flow between the operators. A data set is imported by the first operator of the action, then this data set is passed on to the subsequent operator, this operator is executed, and its manipulated output data set is used as input for the next operator. In this way, the data set is passed on from one operator to the other, according to the action workflow. Please note that

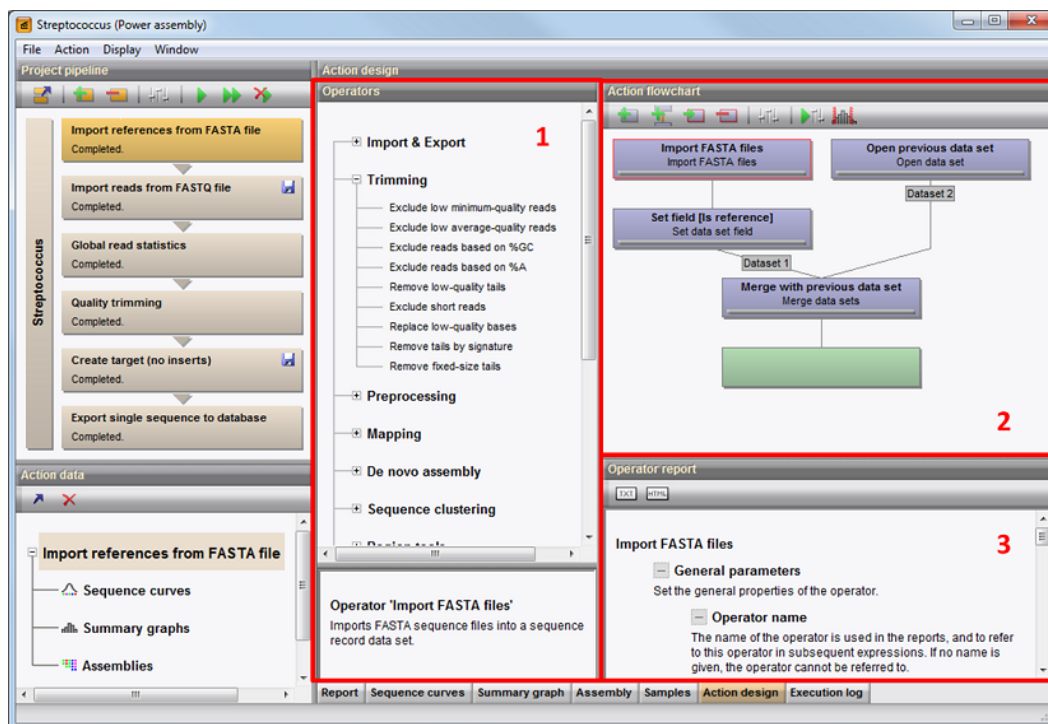


Figure 18.3.33: The *Action design* panel with its sub-panels. 1: *Operators panel*, 2: *Action flowchart panel* and 3: *Operator report panel*.

operators can only be added to the action flowchart if they are compatible, i.e. if the output data set of the preceding operator contains the necessary data set fields for the subsequent operator. More information on the default data set fields can be found in 18.7.2.4. At the end of the action workflow, one or multiple green rectangles represent the created data set(s). This data set is then further used as input data set for the first operator of the next action. This way, the complete project pipeline is executed.

Detailed operator settings for a selected operator can be called by **Action > Action design > Operator properties...** (🔧). A tabbed dialog box pops up with all the parameters related to that specific operator. The different parameters are organized in tab panels which typically contain the *General settings*, *Input fields*, *Operator parameters* and *Output fields* (see also 18.7 for more information on parameter settings).

To create an action, the different operators and their settings need to be defined in the *Action flowchart panel*. Before adding operators to the action flowchart, the action itself should be selected in the *Project pipeline panel*. Selecting an action automatically updates the *Action flowchart panel*. Below, an overview of the commands present in the *Action flowchart panel* are listed. Detailed instructions on how to create customized actions are provided in 18.6.

- Adding an operator to the action flowchart is done by **Action > Action design > Add operator...** (+). Default, the created operator is appended as the last operator to the action flowchart.
- To insert an operator into a branch of the action flowchart, select the part of the branch between two operators already present in the flowchart where the new operator needs to be inserted. The selected part of the branch is now highlighted in orange. Next, select the operator to be inserted in the *Operators panel*, and select **Action > Action design > Insert operator...** (+).
- To replace a selected operator from the action flowchart by a new operator, first select the operator to be replaced in the *Action flowchart panel*, second select the new operator from the *Operators panel*, and third select **Action > Action design > Replace currently selected operator...** (🔄). If the created operator is compatible with the subsequent operators, then the operator is replaced. If not, the existing operator remains and an error message is displayed.

- Removing operators from the action flowchart is done by **Action > Action design > Remove currently selected operators** (🗑️). To remove multiple operators at once, first select all the operators to be removed by **Ctrl-click**. The command will remove the selected operator(s) and will merge the remaining adjacent operators if they are compatible. If the operators that become adjacent are not compatible, the subsequent operators will be removed from the action flowchart as well.
- To set the action-specific questioned runtime parameters from the *Runtime parameters* dialog box (Figure 18.3.34), select **Action > Action design > Runtime parameters...** (🔧). From this dialog box, the different parameters for runtime questioning can be selected, and organized into pages and groups. Detailed information on setting runtime parameter questioning is given in 18.7.4.2.
- As mentioned in 18.3.6, summary graphs can be used to set flexible thresholds. These summary graph boundaries can be set from the *Action flowchart panel* by pressing **Action > Action design > Summary graph boundaries...** (📊). This opens the *Graphic parameter dialog box* (Figure 18.3.35) where the existing thresholds that can be linked to a graph are displayed. From herein, the parameters can be linked and modified. More information on setting flexible thresholds linked to summary graphs is given in 18.7.4.3.

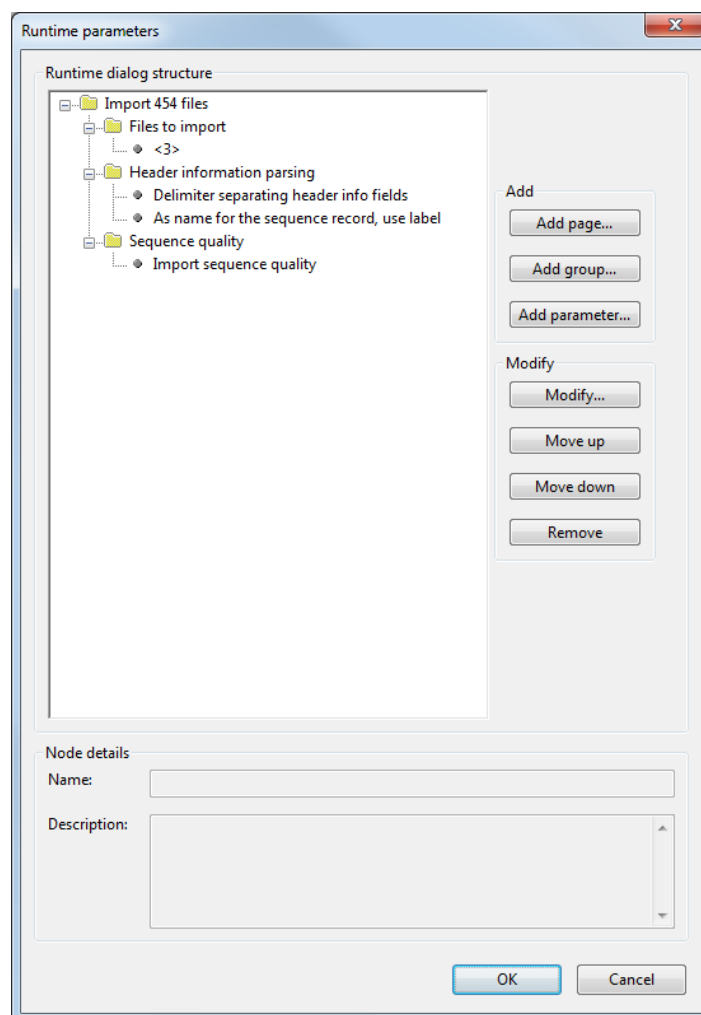


Figure 18.3.34: The *Runtime parameters* dialog box.

The full operator report can be exported to a text or HTML file by selecting **File > Export report as text > Operator report** (📄), or **File > Export report as html > Operator report** (📄), respectively. After confirmation, the exported file will open automatically.

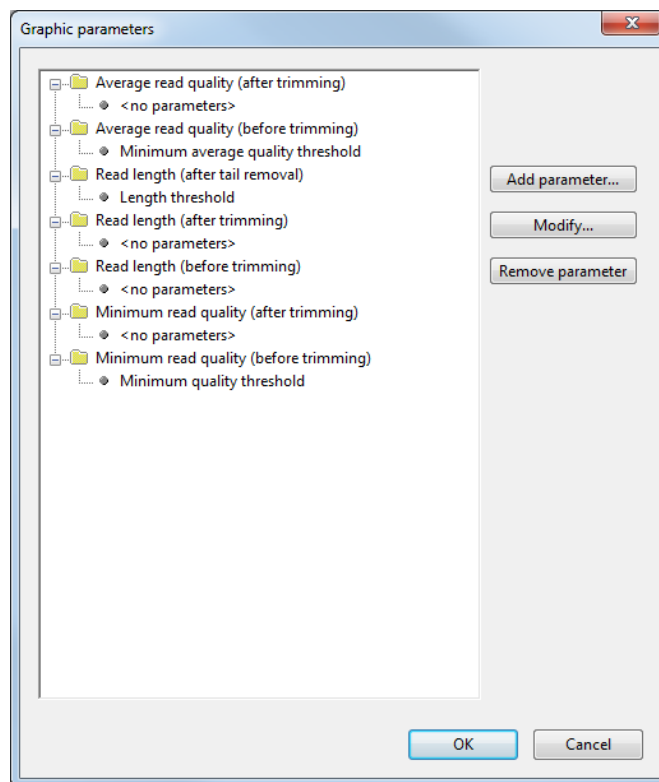


Figure 18.3.35: The *Graphic parameters* dialog box.

18.3.10 The Execution log panel

During the computation of an action from the project pipeline, the log file is displayed in real time in the *Execution log panel*. For each series of operators, detailed information including the time the computation has started and stopped, the time elapsed and a descriptive message are written to the log file. Actions that include third party tools, e.g. the Velvet program for *de novo* assembly, have the information provided by the software tool streamed to the execution log of the action. For every action in the project, the last log file is kept within one session.

18.3.11 Power Assembler general settings

Some general (i.e. database-wide) settings for the Power Assembler can be entered in the *Power Assembler settings* dialog box, which pops up after selecting **File > General settings...**

The **Working directory** is the directory where intermediate results, such as data sets and assembly data, are stored. During calculation of a Power Assembler project, these data sets are fetched only when required, without the need to store the data in memory. Although the use of in-memory data sets is discouraged, it remains possible to choose the type of data set upon creation. By default, the working directory is set to the `PAssemblies` folder of the Temp directory (`%USERPROFILE%\AppData\Local\Temp\PAssemblies`).



The working directory can be changed to any other directory, however, avoid to use the database directory when using in-memory data sets as this may cause the directory to increase in size on a very short time. One can choose an external drive or a network drive as working directory, however this can have a significant effect on the speed of the calculations if the read or write speed is too slow. It is therefore important to designate a well-considered working directory.

The **Calculation priority** for calculations performed by the Power Assembler can be set to one of five

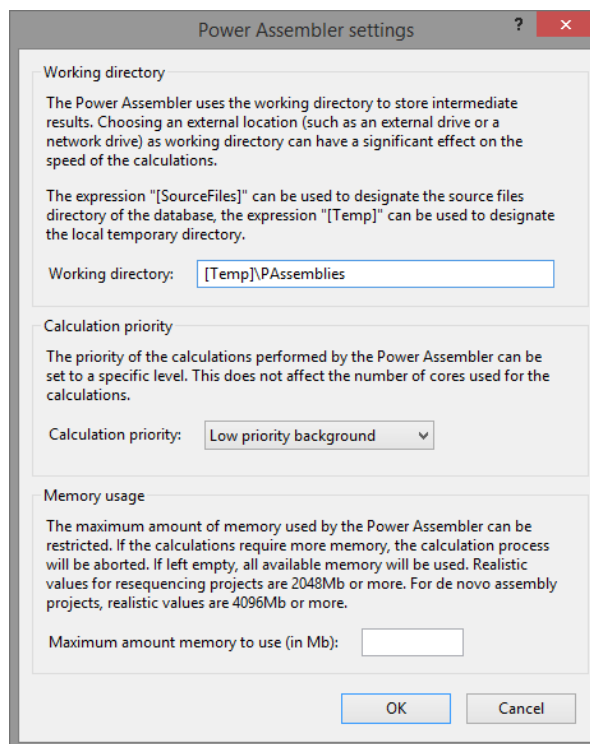


Figure 18.3.36: The *Power Assembler settings* dialog box.

available levels via a drop-down list, in a similar fashion as for comparison calculations in BioNumerics (see 2.3.3.5).

Since assembly of next-generation sequence data – particularly de novo sequence assembly – is memory-intensive, the **Memory usage** by the Power Assembler can be restricted by specifying a **Maximum amount of memory to use (in Mb)**. If left empty, all available RAM memory will be used if the calculations require it.



Note that these general settings not only apply to the power assembly project at hand, but for all projects in the same BioNumerics database.

18.3.12 Cleanup project data

To remove intermediate data from the working directory, and thus to reduce the amount of stored data per project, the cleanup function is used. To call the *Cleanup dialog box*, select **File > Cleanup...**

The *Cleanup* dialog box displays all data types that can be removed from the linked power assembly project. These data types include: reports, data sets, search data, sequence curves, summary graphs and assemblies. The data types already generated by the actions of the power assembly are active, whereas the other data types remain grayed. Check the boxes of the data types to be removed, and press <OK> to confirm.

If necessary, the data can easily be regenerated by re-running the project pipeline.

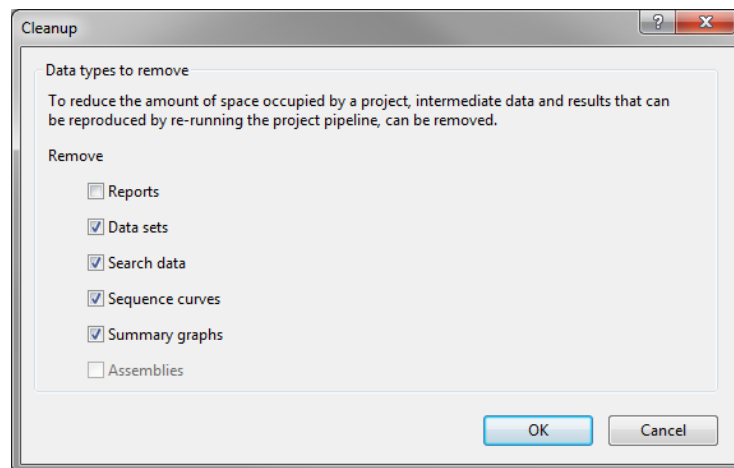


Figure 18.3.37: The *Cleanup* dialog box.

Chapter 18.4

Predefined projects

For a number of combinations between common project types and data formats, a predefined project template is made available. In such predefined project templates, the actions are specifically tuned to the application and data formats at hand.

Following predefined project templates are available for resequencing projects:

- Resequencing with Solexa/Illumina data
- Resequencing with Roche/454 data
- Resequencing with base-space FASTQ data (SOLiD, IonTorrent)

Following predefined project templates are available for de novo assembly:

- De novo assembly with Solexa/Illumina data
- De novo assembly with Roche/454 data
- De novo assembly with base-space FASTQ data (SOLiD, IonTorrent)

A predefined project template is loaded via **File > Load pipeline from template...** ( (see [18.3.2.2](#)). More information on how to calculate the project can be found in [18.3.2.7](#).

Chapter 18.5

Power assembler predefined actions

The Power Assembler is designed to combine a number of actions, predefined or user-defined, into a power assembly project. This section only covers detailed information on the predefined actions. The creation of user-defined actions is discussed in [18.6](#).

Below is an overview of the predefined actions, sorted into different functional categories.

18.5.1 Import

The predefined import actions are designed to make sequence data and metadata available to the Power Assembler.

The actions importing sequence data come in two kinds, depending on the role the imported sequences play: on the one hand, sequences that are considered to be *read sequences* (for instance, sequences coming from a sequencer), and, on the other hand, sequences that are considered to be *reference sequences* (for instance, full genome sequences used for mapping).

- The Power Assembler can import *read sequences* from files in Solexa/Illumina[®] format, Roche/454[®] format, FASTA or FASTQ format. When sequence qualities are provided, one can choose to import them along with the sequences. The predefined actions providing these import operations automatically flag the imported sequences as read sequences.
- *Reference sequences* can be imported from the BioNumerics database or from a FASTA file. The predefined actions providing these import operations automatically flag the imported sequences as reference sequences.

The import actions for reference and read sequences are discussed in [18.5.1.1](#) and [18.5.1.2](#).



Sequences are loaded in the field *[Sequence]*. All bases are converted to uppercase letters. All IUPAC codes are allowed, along with the characters “-” (for a gap) and “|” (for concatenated sequences). Any other character is converted to a gap.

The boolean field *[Is reference]* is used to distinguish between reference sequences (field is set to *True*) and read sequences (field is set to *False*).

The source of the sequence is written in the field *[Source]*.

Sample information (see [18.3.8](#)) can be very useful to document the experiment that led to a particular power assembly, especially when performing multiplexed experiments. This meta data can be made available to a power assembly project by importing it as sample information from a TAB-delimited file. The import action for sample data is discussed in [18.5.1.3](#).

18.5.1.1 Importing reference sequences

Reference sequences can be read from the BioNumerics database, or can be imported from a FASTA file.

An important property of a sequence record containing a reference sequence is the *name* of the record. This name is used to address the sequence record from a user perspective (for instance, when creating sequence curves over a sequence, or creating an assembly with respect to a sequence). Therefore, the name of the sequence record is set during import.



The name of a sequence record is stored in the field *[Name]*.

18.5.1.1.1 Load references from database

This action loads the sequences from the BioNumerics database corresponding to the entry and experiment type specified, and flags them as reference sequences. Within one experiment type, more than one entry can be imported at the same time.

The name of the sequence is set to the entry key.

Parameters

A sequence in the BioNumerics database is determined by the entry key and the experiment type it belongs to. When loading sequences from the database, the list of *Entry keys* parameter determines which entries are used, and the *Experiment type* parameter decides for which experiment type sequences are loaded. See [18.7.5.1.6](#) for more information.



The content of the field *[Source]* is set to (*from database*).

18.5.1.1.2 Import references from FASTA file

This action imports sequences from a FASTA file, and flags them as reference sequences. More than one FASTA file can be imported at the same time, each file containing one or more sequences.

The name of the sequence is parsed from the header information in the sequence file. Each sequence begins with a line of the form

```
>[field1] [separator] [field2] [separator] ...
```

By specifying the separator between fields and the number of the field to use, the name of the sequence record is extracted from the sequence file.

Parameters

The files that will be imported, are listed in the *Sequence files* box. Pressing the **<Add>** button pops up a dialog box, where multiple files can be selected. Files in the list can be removed by highlighting them and pressing the **<Remove>** button. See [18.7.5.1.2](#) for more information. The name of the sequence record is determined by the following parameters:

- *Delimiter separating header info fields:* the delimiter that separates the fields in the header information of the file. See [18.7.5.1.2](#) for more information.
- *As name for the sequence record, use label:* the number of the field to be used as name for the sequence record. See [18.7.5.1.2](#) for more information.



The content of the field *[Source]* is set to the filename of the FASTA file the sequence was imported from.

18.5.1.2 Importing read sequences and sequence qualities

A variety of file formats are supported for importing read sequences. Apart from the FASTQ and FASTA formats, also read files from Roche/454[®] and Solexa/Illumina[®] can be imported directly into the Power Assembler.

Sequence qualities are imported together with the sequences, if requested. However, since the Power Assembler uses the Phred quality scores system [15], raw sequence qualities as provided in the sequence files are converted to Phred scores during import. See 18.7.2.8 for more information on the use of quality scores in the Power Assembler.

The Power Assembler can import paired-end reads, either from a pair of FASTQ or FASTA files or from a pair of Solexa/Illumina[®] files. Both files are then imported simultaneously, assuming that they contain the sequences in the same order. The distance between two ends is not set at this stage. For Roche/454[®] files containing paired-end reads, sequences are first imported as single-end reads and split into paired-end reads later based on the adapter sequence used.

As with reference sequence records, the name of the sequence record is filled in while importing the sequences.

For every read sequence file that is imported, the predefined import action creates a sample and links all read sequence records that were imported from the same file to the same sample, thus initiating the strategy that one file corresponds to one sample (or one experiment). However, when separate files need to be treated as corresponding to the same sample, samples need to be merged. This is done by the predefined preprocessing action *Merge samples* (see 18.5.2).



Sequence qualities are loaded in the field *[Sequence quality]*. All quality values in the Power Assembler are Phred scores between 0 and 63 (included).

The name of a sequence record is stored in the field *[Name]*.

18.5.1.2.1 Import reads from Roche/454 file

This action imports the sequences from a set of Roche/454[®] sequence files, and flags them as read sequences. Multiple files can be imported at the same time.

Optionally, sequence qualities can be imported along with the sequence itself. Files are coupled by their filename. A sequence file and its quality file should have the same name, except for the extension, which should be "fna" for the sequence file, and "qual" for the quality file,

`[filename].fna` and `[filename].qual`.

Both files are then imported simultaneously, assuming that they contain the sequences in the same order.

The name of the sequence is parsed from the header information in the sequence file. Each sequence begins with a line of the form

```
>[field1] [separator] [field2] [separator] ...
```

By specifying the separator between fields and the number of the field to use, the name of the sequence record is extracted from the sequence file.

Roche/454[®] paired-end sequences are imported as single-end sequences, but can be split up by the predefined action *Split Roche/454[®] paired-ends* (see 18.5.2).

Parameters

The files that will be imported, are listed in the *Sequence files* box. Pressing the **<Add>** button pops up a dialog box, where multiple files can be selected. Files in the list can be removed by highlighting them and pressing the **<Remove>** button. See 18.7.5.1.4 for more information. The name of the sequence record is

determined by the following parameters:

- *Delimiter separating header info fields:* the delimiter that separates the fields in the header information of the file. See [18.7.5.1.4](#) for more information.
- *As name for the sequence record, use label:* the number of the field to be used as name for the sequence record. See [18.7.5.1.4](#) for more information.

When the parameter *Import sequence quality* is checked, sequence quality scores are imported along with the sequences. The sequence quality file should have the same name as the sequence file it corresponds to, except for the extension, which should be ".qual". See [18.7.5.1.4](#) for more information.



A Roche/454[®] quality score is taken to be equal to a BioNumerics quality score.

18.5.1.2.2 Import Solexa/Illumina files

These actions import the sequences from a set of Solexa/Illumina[®] sequence files, and flag them as read sequences. Multiple files can be imported at the same time.

Optionally, sequence qualities, which are contained in the sequence file, can be imported along with the sequence itself. Since Solexa/Illumina[®] changed the way sequence qualities are stored in version 1.3 of their sequence processing software, the Power Assembler provides two predefined actions for importing Solexa/Illumina[®] files:

- *Import reads from Solexa/Illumina v1.3+ file:* This action imports the sequences in a set of Solexa/Illumina[®] sequencer files created by Solexa Pipeline[®] version 1.3 and later.
- *Import reads from Solexa/Illumina v1.3- file:* This action imports the sequences in a set of Solexa/Illumina[®] sequencer files created by Solexa Pipeline[®] prior to version 1.3.

The name of the sequence record is set to the identifier given in the sequence file.

Paired-end reads can be imported when the filenames have the appropriate form. Files are coupled by their filename: two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end,

[filename]_1.[extension] and [filename]_2.[extension].

Parameters

The files that will be imported, are listed in the *Sequence files* box. Pressing the <Add> button pops up a dialog box, where multiple files can be selected. Files in the list can be removed by highlighting them and pressing the <Remove> button. See [18.7.5.1.3](#) for more information.

When the parameter *Import sequence quality* is checked, sequence quality scores are imported along with the sequences. See [18.7.5.1.3](#) for more information.

When importing sequence files as paired-end reads by checking the parameter *Import as paired-end reads*, files are coupled based on the filename. Two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end. See [18.7.5.1.3](#) for more information.



For sequence files created by Solexa Pipeline[®] version 1.3 and later, a Solexa/Illumina[®] quality score *illQ* is converted to a BioNumerics quality score *bnQ* by the formula

$$bnQ = illQ - 64.$$

For sequence files created by Solexa Pipeline[®] prior to version 1.3, this is accomplished by the formula

$$bnQ = \frac{10}{\log_{10}} \log \left(1 + 10^{\frac{illQ}{10}} \right).$$

18.5.1.2.3 Import reads from FASTA file

This action imports the sequences from a set of FASTA files, and flags them as read sequences.

The name of the sequence is parsed from the header information in the sequence file. Each sequence begins with a line of the form

```
>[field1] [separator] [field2] [separator] ...
```

By specifying the separator between fields and the number of the field to use, the name of the sequence record is extracted from the sequence file.

Paired-end reads can be imported when the filenames have the appropriate form. Files are coupled by their filename: two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end,

```
[filename]_1.[extension] and [filename]_2.[extension].
```

Parameters

The files that will be imported, are listed in the *Sequence files* box. Pressing the **<Add>** button pops up a dialog box, where multiple files can be selected. Files in the list can be removed by highlighting them and pressing the **<Remove>** button.

The name of the sequence record is determined by the following parameters:

- *Delimiter separating header info fields:* the delimiter that separates the fields in the header information of the file. See [18.7.5.1.5](#) for more information.
- *As name for the sequence record, use label:* the number of the field to be used as name for the sequence record. See [18.7.5.1.5](#) for more information.

When importing sequence files as paired-end reads by checking the parameter **Import as paired-end reads**, files are coupled based on the filename. Two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end.

18.5.1.2.4 Import reads from FASTQ file

This action imports the sequences in a set of FASTQ files, and flags them as read sequences.

Optionally, sequence qualities, which are contained in the sequence file, can be imported along with the sequence itself.

The name of the sequence is parsed from the header information in the sequence file. Each sequence begins with a line of the form

@[field1] [separator] [field2] [separator] ...

By specifying the separator between fields and the number of the field to use, the name of the sequence record is extracted from the sequence file.

Paired-end reads can be imported when the filenames have the appropriate form. Files are coupled by their filename: two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end,

[filename]_1.[extension] and [filename]_2.[extension].

Parameters

The files that will be imported, are listed in the *Sequence files* box. Pressing the **<Add>** button pops up a dialog box, where multiple files can be selected. Files in the list can be removed by highlighting them and pressing the **<Remove>** button. The name of the sequence record is determined by the following parameters:

- *Delimiter separating header info fields:* the delimiter that separates the fields in the header information of the file. See [18.7.5.1.5](#) for more information.
- *As name for the sequence record, use label:* the number of the field to be used as name for the sequence record. See [18.7.5.1.5](#) for more information.

When the parameter *Import sequence quality* is checked, sequence quality scores are imported along with the sequences.

When importing sequence files as paired-end reads by checking the parameter *Import as paired-end reads*, files are coupled based on the filename. Two files containing each one end of a paired-end read should have the same name and extension, except for the last part of the name, which should be "_1" for the first end, and "_2" for the second end.



A FASTQ quality score is taken to be equal to a BioNumerics quality score.

18.5.1.3 Importing sample data from TAB-delimited file

The predefined action *Import sample data* imports sample data such as the entry key and the experiment name from a tab-delimited file into the sample data set. All information for one particular sample should be on one line in the file. The link between the lines in the file and the samples in the power assembly project is set by specifying a link column in the file and a link field in the sample record data set. Once this link is established, any number of columns from the file can be imported in the sample record data set of the power assembly project. When the tab-delimited file contains samples which are not yet in the sample record data set, one can choose to ignore them, or create new samples accordingly. However, in the latter case one should be aware of the fact that the predefined import actions for read sequences remove all the sample records that are not linked to any of the reads. Therefore, importing sample data is best done after the reads have been imported.

Parameters

The parameter *File to import* specifies the TAB-delimited file containing the sample data. See [18.7.5.12.6](#) for more information.

Linking the lines in the TAB-delimited file to the sample records in the sample data set of the power assembly project is done by specifying a *Link column index* for the column in the file, and a *Link field* for the field in the sample data set. When, for a particular sample record, the file does not contain a corresponding line, the sample record is left untouched. Conversely, when a line in the file does not correspond to an existing

sample record, a new sample record is created if the parameter *Create new samples if necessary* is checked, and, if unchecked, the line in the file is discarded. See [18.7.5.12.6](#) for more information.

The columns that need to be imported, can be specified in the list, together with the corresponding fields. See [18.7.5.12.6](#) for more information.

18.5.2 Preprocessing

The predefined preprocessing actions provide standard operations that prepare the read sequence records for assembly. When a variety of files has been imported but should be treated as one, the samples corresponding to the different files can be merged using the action *Merge samples*. The *De-multiplexing* action separates the multiplex identifiers from the sequences and links read sequence records with the same multiplex identifier to the same sample. For Roche/454[®] paired-end reads, the action *Split Roche/454[®] paired-ends* splits the sequences into two ends by aligning the adapter to the sequence and breaking up the initial sequence into two parts according to the adapter position that was found.



All preprocessing actions only consider read sequence records, that is, sequence records where the field *[Is reference]* is set to *False*.

18.5.2.1 Merge samples

This action merges all samples and removes the samples that are no longer in use. Merging samples is necessary when working with read sequences from different files that need to be treated as coming from the same sample, since most predefined actions treat samples separately. The content of the information fields that were identical for all the samples in use before this action was run, is retained in the corresponding information fields of the merged sample.

18.5.2.2 Demultiplexing

This action removes the multiplex identifiers (also called *bar codes*) and creates samples and sample links according to these identifiers. Multiplexing identifiers have a fixed sequence length, and are followed by a linker sequence of fixed length as well. If the multiplex identifier and the linker found at the beginning of the sequence, are also found reverse-complemented at the end of the sequence, both the multiplex identifier and the linker are removed from the read sequence, from the beginning and the end of the sequence. For every multiplex identifier, a sample is created and all sequences having a particular multiplex identifier are linked to the sample corresponding to this multiplex identifier. The multiplex identifiers are stored in the sample record data set in the field *[Barcode]*.

Parameters

Linker size: size of the linker between the bar code and the actual read sequence. See [18.7.5.3.1](#) for more information.

Barcode size: size of the multiplex identifier or bar code identifying the sample the read came from. See [18.7.5.3.1](#) for more information.

18.5.2.3 Demultiplexing with error correction

This action removes the multiplex identifiers (also called *bar codes*), performs an error correction on these identifiers and then creates samples and sample links according to these identifiers. Multiplexing identifiers have a fixed sequence length, and are followed by a linker sequence of fixed length as well. If the multiplex

identifier and the linker found at the beginning of the sequence, are also found reverse-complemented at the end of the sequence, both the multiplex identifier and the linker are removed from the read sequence, from the beginning and the end of the sequence. For every error-corrected multiplex identifier, a sample is created and all sequences having a particular multiplex identifier are linked to the sample corresponding to this multiplex identifier. The multiplex identifiers are stored in the sample record data set in the field *[Barcode]*.

Parameters

Linker size: size of the linker between the bar code and the actual read sequence. See [18.7.5.3.1](#) for more information.

Barcode size: size of the multiplex identifier or bar code identifying the sample the read came from. See [18.7.5.3.1](#) for more information.

To obtain the representatives: this parameter determines the correct multiplex identifiers, either by reading them from the sample records table or by taking the most frequent multiplex identifiers in the data set.

Maximum number of mismatches: maximum number of mismatches between two multiplex identifiers to be identified.

18.5.2.4 Split Roche/454 paired ends

This action takes read sequences from Roche/454[®] paired-end runs that were imported as single-end reads, and creates paired-end read sequences. The split position is determined by aligning the adapter sequence to the read. When the adapter does not match any part of the read sequence, the sequence is left as a whole. If the adapter matches but is too close to the beginning and/or the end of the read sequence, only the parts that are long enough are retained, thus yielding a single-end read sequence, or no sequence at all. Sequence qualities are split accordingly.

Parameters

Adaptors: the adaptor sequence(s) to search for when determining the split position between the two ends of the paired-end read. The available adaptor sequences are the Roche/454 Flx[®] palindromic adaptor

GTTGGAACCGAAAGGGTTTGAATTCAAACCCTTCGGTTCCAAC

and the Roche/454 Titanium[®] adaptor

TCGTATAACTTCGTATAATGTATGCTATACGAAGTTATTACG.

If the latter is selected, also the reverse complement is used for analysis. See [18.7.5.3.3](#) for more information.

The alignment of the adapter sequence to the read sequence is scored using the *Match score*, *Mismatch score*, the gap policy for the adapter sequence (with *Open gap score* and *Extend gap score*) and the gap policy for the read sequence (again, with *Open gap score* and *Extend gap score*). See [18.7.5.3.3](#) for more information.

An alignment is accepted when the sequence identity is at least the *Minimum sequence identity*, and the total penalty (that is, the sum of all negative contributions to the alignment score) does not exceed the *Maximum penalty*. See [18.7.5.3.3](#) for more information.

Once a sequence is split, a part of the paired-end sequence is retained only when the size of the part is at least the *Minimum sequence size*. If this is not the case, the part is discarded and the paired-end read is reduced to a single-end read, or no read at all. See [18.7.5.3.3](#) for more information.

18.5.2.5 Set sample entry key from barcode

This action parses the sample entry key (i.e. the entry key under which the sample data will be saved to the database) from the sample barcode field (i.e. the *Barcode* field as defined in the *Samples panel*).

18.5.2.6 Set entry key from source

18.5.2.7 Set sample entry key from source

This action parses the sample entry key (i.e. the entry key under which the sample data will be saved to the database) from the sample source field (i.e. the *Source* field as defined in the *Samples panel*).

18.5.2.8 Set experiment name from source

18.5.2.9 Set sample experiment name from source

This action parses the sample experiment name (i.e. the experiment under which the sample data will be saved to the database) from the sample source field (i.e. the *Source* field as defined in the *Samples panel*).

18.5.3 Trimming

The predefined trimming actions provide means to filter the set of read sequence record, and retain only those records that meet the quality standards or structural expectations. The *Quality trimming* action performs a sequence of trimming actions based on the length and the quality scores of the read sequences. The action *Remove polyA reads* removes those read sequences that contain too many A's. Analogously, the action *Remove polyGC reads* removes those read sequences whose percent GC is not within a certain window.

The predefined trimming actions also provide an overview of the properties used to perform their trimming in the form of summary plots. For instance, when excluding read sequences when their average read quality drops below a threshold, a histogram is created that shows the distribution of the average read quality over the complete set of read sequences, both before and after the trimming was executed. This gives a visual account of the effect of the trimming operation. Moreover, the minimum average quality threshold is visualized on the histogram, and can be changed graphically. This approach is used in all predefined trimming actions. See [18.3.6](#) for more details of graphical parameters.



All trimming actions use the fields *[Sequence]* and *[Sequence quality]* as source for the sequence and sequence quality data. Output is written in the same fields.

All trimming actions only consider read sequence records, that is, sequence records where the field *[Is reference]* is set to *False*.

18.5.3.1 Quality trimming (automatic)

18.5.3.2 Quality trimming

This action trims read sequences based on read sequence length and sequence quality. First, all reads with minimum quality below the minimum quality threshold are removed. The same is done for average quality instead of minimum quality. Next, bases at the beginning or the end of the sequence are removed as long as their quality is below the minimum tail quality threshold. Finally, reads that are too short are removed, and the low-quality bases in the remaining read sequences are replaced by N.

This action creates histograms for the following aspects of the read sequence records:

- minimum sequence quality,
- average sequence quality, and
- sequence length.

For each of the above three characteristics, a histogram is created before any trimming operation is executed (thus using the raw data) and after all trimming operations have finished (thus using the trimmed data). In this fashion the parameters for trimming can be set graphically, and the result of the trimming operations can be assessed visually.

Parameters

Exclude reads with minimum quality below: all reads with minimum quality below this threshold are excluded. See [18.7.5.2.1](#) for more information.

Exclude reads with average quality below: all reads with average quality below this threshold are excluded. See [18.7.5.2.2](#) for more information.

Remove bases at the end of a read while the quality is below: as long as the quality of the bases at the beginning or end of the sequence is lower than this threshold, the bases are removed from the sequence. See [18.7.5.2.5](#) for more information.

Exclude reads shorter than: all reads shorter than the length threshold are excluded. See [18.7.5.2.6](#) for more information.

Replace base by N when the quality is below: if the quality of a base is below the threshold, the base is replaced by N. See [18.7.5.2.8](#) for more information.

18.5.3.3 Overall quality trimming

This action trims overall read sequences based on sequence quality. First, all reads with minimum quality below the minimum quality threshold are excluded. Second, the same is done for average quality instead of minimum quality. Third, the low-quality bases in the read sequences are replaced by N.

This action creates histograms for the following aspects of the read sequence records:

- read length.
- minimum sequence quality,
- average sequence quality, and

For each of the above three characteristics, a histogram is created before any trimming operation is executed (thus using the raw data) and after all trimming operations have finished (thus using the trimmed data). In this fashion the parameters for trimming based on the minimum and average read quality can be set graphically, and the result of the trimming operations can be assessed visually.

Parameters

Exclude reads with minimum quality below: all reads with minimum quality below this threshold are excluded. See [18.7.5.2.1](#) for more information.

Exclude reads with average quality below: all reads with average quality below this threshold are excluded. See [18.7.5.2.2](#) for more information.

Replace base by N when the quality is below: if the quality of a base is below the threshold, the base is replaced by N. See [18.7.5.2.8](#) for more information.

18.5.3.4 Tail quality trimming

This action trims the tails of the reads based on read quality.

First, all reads with minimum quality below the minimum quality threshold are excluded. Second, the same is done for average quality instead of minimum quality. Third, the low-quality bases in the read sequences are replaced by N.

Parameters

Minimum tail quality: as long as the quality of the bases at the beginning or end of the sequence is lower than this threshold, the bases are removed from the sequence. See [18.7.5.2.5](#) for more information.

Minimum windowed average quality: as long as the average quality within a window in front of the specified base is lower than this threshold, the base is removed from the sequence. See [18.7.5.2.4](#) for more information.

Minimum rolling average quality: as long as the average quality within a window starting at the beginning of the read down to the specified base is higher than this threshold, the base is retained in the sequence. If not, the read is trimmed up to that specific base position. See [18.7.5.2.3](#) for more information.

18.5.3.5 Length trimming

This action trims read sequences based on their read sequence length. Reads that are too short or too long are removed from the data set.

A histogram is created for the read sequence length before any trimming operation is executed (thus using the raw data) and after the trimming operation has finished (thus using the trimmed data). In the first histogram, both the parameters for trimming can be set graphically, and the result of the trimming operation can be assessed visually in the latter histogram.

Parameters

Exclude reads shorter than: all reads shorter than the length threshold are excluded. See [18.7.5.2.6](#) for more information.

Restrict reads to at most: all reads longer than the length threshold are excluded. See [18.7.5.2.7](#) for more information.

18.5.3.6 Remove polyA reads

This action trims read sequences based on the frequency of the base A. When the number of appearances of the base A with respect to the total number of bases in the sequence exceeds a threshold, the sequence is removed.

A histogram is created for the percentage of the base A before any trimming operation is executed (thus using the raw data) and after all trimming operations have finished (thus using the trimmed data). In this fashion the parameter for trimming can be set graphically, and the result of the trimming operation can be assessed visually.

Parameters

Exclude reads with %A above: all reads with %A above this threshold are excluded. See [18.7.5.2.12](#) for more information.

18.5.3.7 Remove polyGC reads

This action trims read sequences based on the frequency of the bases G and C. When the number of appearances of the bases G or C with respect to the total number of bases in the sequence is not within a chosen range, the sequence is removed.

A histogram is created for the percentage of the bases G and C before any trimming operation is executed (thus using the raw data) and after all trimming operations have finished (thus using the trimmed data). In this fashion the range parameters for trimming can be set graphically, and the result of the trimming operation can be assessed visually.

Parameters

Exclude reads with %GC above: all reads with %GC above this threshold are excluded. See [18.7.5.2.11](#) for more information.

Exclude reads with %GC below: all reads with %GC below this threshold are excluded. See [18.7.5.2.11](#) for more information.

18.5.3.8 Remove reads with long homopolymers

This action removes read sequences with long homopolymers. Reads that have homopolymer regions larger than the *Homopolymer length threshold* are removed from the data set.

A histogram is created for the detected homopolymer lengths before any trimming operation is executed (thus using the raw data) and after the trimming operation has finished (thus using the trimmed data). In the first histogram, the maximum homopolymer length can be defined graphically, and the result of the trimming operation can be assessed visually in the latter histogram.

Parameters

Maximum homopolymer length: all reads with homopolymers longer than this threshold are excluded. See [18.7.5.2.10](#) for more information.

18.5.4 Statistics

The predefined statistics actions provide summary information for the data at hand. Apart from descriptive statistics such as the number of sequences in a sequence record dataset, the minimum, average and maximum length of the sequences and the frequency distribution of the bases A, C, G, T and N, they also provide summary graphs, thus giving visual feedback on the data. These actions provide both a preliminary look at the data and a way of assessing the result of a mapping action (see [18.5.5](#)).

18.5.4.1 Global reference statistics

This action calculates global statistics on the reference sequences in the sequence record data set. In particular, the following aspects are reported:

- the minimum, average and maximum length of the sequences, and
- the frequency distribution of the bases A, C, G, T and N, and the frequency distribution of the base pairs A/T versus G/C.



The *Global reference statistics* action only considers reference sequence records, that is, sequence records where the field *[Is reference]* is set to *True*.

18.5.4.2 Global read statistics

This action calculates global statistics on the read sequences and sequence qualities in the sequence record data set. In particular, the following aspects are reported:

- the minimum, average and maximum length of the sequences,
- the minimum, average and maximum quality of the sequence qualities, and
- the frequency distribution of the bases A, C, G, T and N, and the frequency distribution of the base pairs A/T versus G/C.

Moreover, histograms are created of the sequence length and the minimum, average and maximum sequence quality.



The *Global reference statistics* action only considers reference sequence records, that is, sequence records where the field *[Is reference]* is set to *True*.

18.5.4.3 Mapping statistics

This action creates histograms that are helpful in assessing the setting and the outcome of a mapping action (see 18.5.5) that positioned the read sequences with respect to the reference sequence(s), and created a target sequence record from this mapping information. As described in 18.7.2.6, the *mapping discrimination* and the *mapping degeneracy*, which were calculated during the mapping action, can be used for assessing both the parameters used in the mapping, and the quality of the mapping itself. The *coverage histogram*, *mapping degeneracy histogram*, and *mapping discrimination histogram* created by this action offer the graphical tools to analyze the quality of the mapping, and the suitability of the mapping parameters.

18.5.4.4 Contig statistics

This action calculates summary statistics on the length of the target sequences in a data set, such as average size, N50, and so on. Moreover, this action creates histograms that are helpful in assessing the setting and the outcome of a *de novo* assembly action (see 18.5.6), that created *de novo* target sequence records and that positioned the read sequences with respect to these target sequence(s). As described in 18.7.2.6, the *mapping discrimination* and the *mapping degeneracy*, which were calculated during the mapping action, can be used for assessing both the parameters used in the mapping, and the quality of the mapping itself. The *coverage histogram*, *mapping degeneracy histogram*, and *mapping discrimination histogram* created by this action offer the graphical tools to analyze the quality of the mapping, and the suitability of the mapping parameters.

18.5.5 Mapping

The predefined mapping actions try to position the reads against the reference sequences, and create target sequence records from this positioning information.

All mapping actions can handle more than one reference sequences at the same time.

First of all, every read is mapped on the best fitting reference sequence, and the optimal alignment is determined. Then, for every reference sequence, a target sequence is created using the mapping information of the reads. Together with the target sequence, also the target coverage matrix, the target quality matrix and the target sequence quality are calculated. Finally, for every target sequence record, an assembly view is created, allowing a visual inspection of the mapping results. Moreover, a sequence curve is created for the following aspects of the target sequence records:

- the sequence itself,
- the sequence quality,
- the coverage matrix, and
- the quality matrix.

The predefined mapping actions come in two flavors, depending on the application.

- In the predefined action **Create target (no inserts)**, it is assumed that the target sequence follows exactly the same frame as the reference sequence. In this case, the target sequence should be identical to the reference sequence it was created from, apart from individual base calls. The alignment of the reads with respect to the references can be gapped or ungapped, but all insertions and deletions are performed on the reads.
- In the predefined action **Create target (with inserts)**, the reference sequence is a more distant relative of the target sequence, and not only the reproduction of the reference sequence is the goal, but also the local structural variation between reference and target is of interest. The alignment procedure of the reads with respect to the references should allow gaps. Insertions are performed both on the reads and on the reference sequence, thus building the target sequence out of it.

Parameters

Alignment parameters (read-to-reference): the *match score*, *mismatch score*, whether or not to *allow gaps in the reads*, the *open gap cost* and *extend gap cost* for gaps in the reads, and whether or not to *allow gaps in the reference*, the *open gap cost* and *extend gap cost* for gaps in the references. See [18.7.5.4.1](#) for more information.

Alignment acceptance parameters (read-to-reference): the *minimum sequence identity*, the *maximum penalty score* and the *minimum overlap score*. See [18.7.5.4.1](#) for more information.

In case insertions are allowed on the reference sequences, there are also **alignment parameters (read-to-target)**: the *match score*, *mismatch score*, whether or not to *allow gaps in the reads*, the *open gap cost* and *extend gap cost* for gaps in the reads, and whether or not to *allow gaps in the target*, the *open gap cost* and *extend gap cost* for gaps in the target. See [18.7.5.4.2](#) for more information. In this case, additional **alignment acceptance parameters (read-to-target)** are defined: the *minimum sequence identity*, the *maximum penalty score* and the *minimum overlap score*. See [18.7.5.4.2](#) for more information.

The **target quality threshold** sets the importance balance between coverage and individual base qualities of the bases in the reads in the calculation of the quality matrix. See [18.7.5.4.3](#) for more information.

The base calling of the target sequence is determined by the *minimum coverage*, the *gap threshold* and the *single, double and triple base threshold*. See [18.7.5.4.6](#) for more information.



All mapping actions distinguish reads from reference sequences by the field **[Is reference]**.

Sequences are fetched from the field **[Sequence]**, and sequence qualities (if available) from **[Sequence quality]**. All mapping actions can simultaneously handle reads of different origin and lengths, single-end and paired-end reads or reads with and without qualities.

18.5.6 De novo assembly

The predefined *de novo* assembly actions build a set of contigs (called *de novo targets*) out of a collection of reads, and map the reads against them. The core of the *de novo* assembly actions has been implemented through the third-party tools **Velvet** and **Ray** (see [18.1](#) for more information).

First of all, the *de novo* targets are created. Then, every read is mapped on the best fitting target sequence, and the optimal alignment is determined. Then, every target sequence is updated using the mapping information of the reads. Together with the target sequence, also the target coverage matrix, the target quality matrix and the target sequence quality are calculated. Finally, for every target sequence record, an assembly view is created, allowing a visual inspection of the assembly and mapping results. Moreover, a sequence curve is created for the following aspects of the target sequence records:

- the sequence itself,
- the sequence quality,
- the coverage matrix, and
- the quality matrix.

The predefined *de novo* assembly actions come in two flavors, depending on assembly algorithm used.

- *Create de novo target (Velvet)*, and
- *Create de novo target (Ray)*.

Both *de novo* assembly actions can use single-end and paired-end reads.

There are also some predefined *de novo* finishing actions. In some cases, the *de novo* targets produced by the *de novo* assembly actions are not completely separated from each other. In that case, the predefined action *Merge overlapping de novo targets* can be used to merge the overlapping contigs. Second, when a set of reads has been considered for *de novo* assembly, but ultimately have not been used, the predefined action *Extend de novo targets* tries to extend the previously created *de novo* targets by the unmapped reads. This action will also try to merge the overlapping contigs.

Parameters

Read libraries: whether or not to use the single-end or the paired-end reads in the sequence record data set. When using the paired-end read library, the insert size can be determined automatically, or can be specified by setting an average insert size and standard error on the insert size. These parameters are queried for only in the actions *Create de novo target (Velvet)* and *Create de novo target (Ray)*.

Coverage parameters: these are available only when using the action *Create de novo target (Velvet)*. The expected coverage is used for repeat resolving, and can be determined automatically, or set to a fixed value. This procedure can also be switched off. Error correction is done by excluding low-coverage nodes from the contig construction procedure. A threshold for low-coverage nodes can be determined automatically, or can be set to a fixed value. This procedure can also be switched off.

Alignment parameters (assembly): the *match score*, *mismatch score*, whether or not to *allow gaps*, the *open gap cost* and *extend gap cost*. See [18.7.5.5.3](#) for more information.

Alignment acceptance parameters (assembly): the *minimum sequence identity*, the *maximum penalty score* and the *minimum overlap score*. See [18.7.5.5.3](#) for more information.

Alignment parameters (read-to-target): the *match score*, *mismatch score*, whether or not to *allow gaps in the reads*, the *open gap cost* and *extend gap cost* for gaps in the reads, and whether or not to *allow gaps in the target*, the *open gap cost* and *extend gap cost* for gaps in the references. See [18.7.5.4.1](#) for more information.

Alignment acceptance parameters (read-to-target): the *minimum sequence identity*, the *maximum penalty score* and the *minimum overlap score*. See [18.7.5.4.1](#) for more information.

The *target quality threshold* sets the importance balance between coverage and individual base qualities of the bases in the reads in the calculation of the quality matrix. See [18.7.5.4.3](#) for more information.

The base calling of the target sequence is determined by the *Minimum coverage*, the *gap threshold* and the *single*, *double* and *triple base threshold*. See 18.7.5.4.6 for more information.



All *de novo* assembly actions distinguish reads from other sequences by the field *[Is reference]*. Paired-end sequences are determined by the expression *IsPaired([Sequence])*.

Sequences are fetched from the field *[Sequence]*, and sequence qualities (if available) from *[Sequence quality]*. All *de novo* assembly actions can simultaneously handle reads of different origin and lengths, single-end and paired-end reads or reads with and without qualities.

18.5.7 Postprocessing

The predefined postprocessing actions provide standard operations that extract the relevant information after a mapping action has been performed.

18.5.7.1 Determine covered regions

This action determines all regions on the target sequence where the coverage is higher than the defined threshold value and a base call has been done. For every target sequence record, a sequence curve is created showing the regions that have been extracted, and a histogram of the region lengths is made. The statistics calculated on the regions (such as minimum and maximum length, total base coverage, N50, ...) are added to the report. Finally, these regions are extracted into new sequence records.

Parameters

Coverage must be at least: regions with a coverage below this threshold are excluded from the target sequence.

Minimum length: covered regions shorter than the length threshold are excluded from the target sequence.

Maximum gap size between regions: two regions with a gap size smaller than the threshold are exported as one region.



Only target sequence records are considered, that is, sequence records where the field *[Is target]* is set to *True*.

18.5.8 Export

The predefined export actions are designed to store the assembled sequences (called *targets*) of a power assembly project in the BioNumerics database. Next to the sequence itself, also the sequence quality and the coverage matrix can be stored for further in-depth analysis.

The predefined action *Export single sequence to database* concatenates the sequences from all target sequence records and saves this in the database as a single experiment. On the other hand, the action *Export multiple sequences to database* concatenates the sequences from all target sequence records that belong to the same sample, and uses the entry key and the experiment name from the sample record as entry key and experiment name in the database.



Saving a sequence, its sequence quality and, in particular, its coverage in the database can take a very long time, especially when using an Access database. It is advised to use a SQLite or SQL Server (Express) database when saving power assembly results.



All export actions only consider target sequence records, that is, sequence records where the field *[Is target]* is set to *True*.

18.5.8.1 Export multiple sequence read sets to FASTQ files

This action exports all read sequences and their sequence quality to multiple FASTQ files. For each sample record, a different file containing the sample read sequences will be created.

Parameters

Output directory: the full path to the directory where the sequence files will be exported to. See [18.7.5.1.16](#) for more information.

18.5.8.2 Export multiple sequence read sets to database

This action exports read sequences and their qualities for all reads associated to the same samples to one sequence read set in the database. This way, multiple sequence read set experiments are exported to the database. The entry key and the experiment name used to determine the location where the data is stored, is taken from the respective sample records.

18.5.8.3 Export multiple sequences and coverages to database

This action concatenates the sequence, sequence quality and coverage field of all target sequence records associated with the same sample, and writes them as separate experiments in the BioNumerics database. The entry key and the experiment name used to determine the location where the data is stored, is taken from the respective sample records.

18.5.8.4 Export multiple sequences to database

This action concatenates the sequence and sequence quality field of all target sequence records associated with the same sample, and writes them as separate experiments in the BioNumerics database. The entry key and the experiment name used to determine the location where the data is stored, is taken from the respective sample records.

18.5.8.5 Export sample barcode field to entry information field

This action exports the barcode for each sample to an entry information field. The entry key to determine the location where the data is stored, is taken from the respective sample records.

Parameters

Database field: the name of the database field where the sample barcode will be exported to. See [18.7.5.12.8](#) for more information.

18.5.8.6 Export sample data to tab-delimited file

This action exports the sample data set to a tab-delimited file.

Parameters

File: name and location of the file to write to. See [18.7.5.12.7](#) for more information.

18.5.8.7 Export single sequence and coverage to database

This action concatenates sequence, sequence quality and coverage fields of all target sequence records, and writes them in the BioNumerics database as a single experiment.

Parameters

Entry key: database key of the entry where the sequence will be stored. If left empty, an automatically generated key will be filled in. See [18.7.5.1.9](#) for more information.

Sequence experiment: name of the experiment type where the sequence will be stored. This should be an existing sequence experiment. See [18.7.5.1.9](#) for more information.

18.5.8.8 Export single sequence to database

This action concatenates the sequence and sequence quality field of all target sequence records, and writes these two fields in the BioNumerics database as a single experiment.

Parameters

Entry key: database key of the entry where the sequence will be stored. If left empty, an automatically generated key will be filled in. See [18.7.5.1.9](#) for more information.

Sequence experiment: name of the experiment type where the sequence will be stored. This should be an existing sequence experiment. See [18.7.5.1.9](#) for more information.

Chapter 18.6


User-specific actions

18.6.1 Introduction

The Power Assembler is shipped with a number of predefined actions, developed for high-throughput sequencing applications such as the assembly of a whole bacterial genome based on a reference sequence, partial targeted resequencing, high-throughput analysis of amplicon sequences etc. These actions are documented in [18.5](#).


If other functionality, which is not already present in the predefined actions, needs to be implemented, one can design a user-specific action. These actions are then build in the *Action design panel* from a combination of operators. The user-defined action can be combined with other predefined actions to construct a project pipeline.


An action is created from the *Action design panel*, which is updated when selecting an action from the project pipeline. If the current project is empty, a new action should first be created. The operators to be included in the action flowchart can be selected from the *Operators panel* tree structure.

To add an operator to the action flowchart, double-click the operator from the operators tree structure (see [18.3.9](#)), or select **Action > Action design > Add operator...** (.

In the *Settings dialog box* of the operator, the settings of the operator are defined. Once all parameter values are defined, pressing <Next> on the last parameter page of the operator settings will automatically add the operator to the operator pipeline.

Adding an operator to the action has two possible outcomes. If one adds an operator that needs an input data set, the operator will be appended to the previous operator or to the current data set (represented by the green block). If an operator is added which does not need any input data set, then a separate operator pipeline is started next to the existing one. Later on, these two operator lineages can be merged into one data set by the operator **Merge**.

In the *Action flowchart panel*, the purple box represents the operator, and the green box represents the resulting data set from this operator. To reopen the operator settings, double-click the operator box, or select the operator box and press **Action > Action design > Operator properties...** (). Detailed information on the specific operator settings are discussed in the operator descriptions of [18.7](#).

At any time the selected action can be executed by selecting **Action > Execute** (). The action results can be checked in the *Report panel*.

18.6.2 The use of action templates

Once an action has been created, the action flowchart is automatically saved when saving the corresponding power assembly project. To make a selected action accessible for usage in any project created within a BioNumerics database, the action can be stored as a XML template. Exchanging actions between different databases or multiple users is possible through the use of these action templates. Action templates include the action name, the action description, and all detailed information of the operators present in the action. The power of the action template is that one action outline can be used for different problems, e.g. running the same operators (including the creation of sequence curves, histogram, assemblies...) on different input data sets, or using different operator parameter settings, without the need to completely rebuild the action.

18.6.3 Creating action templates

A user-defined action template is created from the selected action by selecting **Action > Store action as template...** from the *Power assembly* window. The action template is stored within the current database, and listed as user-defined action in the *Add action dialog box* (see Figure 18.6.1).

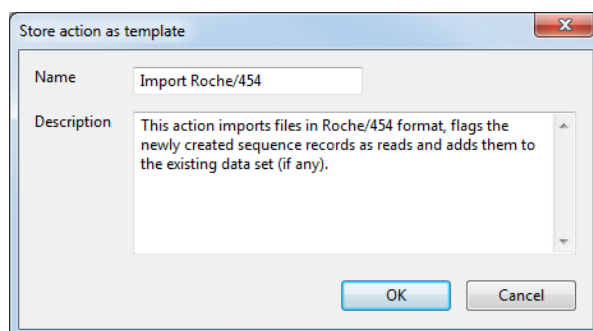


Figure 18.6.1: The *Store action template dialog box*.

In this dialog box, the name and the description for the action can be entered. By default, the name and description as defined in the *Action properties* are filled in.

18.6.4 Removing action templates

To remove user-defined action templates, select **Action > Remove action templates...** from the *Power assembly* window. This pops up the *Remove action templates dialog box* as shown in Figure 18.6.2. If no user-defined actions exist, an error is displayed.

In this dialog box, the action(s) to be removed from the list of user-defined actions can be selected. Hold down the **Ctrl**-key to select multiple actions.

18.6.5 Exporting action templates

To export an action as a template, select the action in the *Project pipeline panel*, and select **Action > Export action template....** Now the *Export action as XML dialog box* pops up (see Figure 18.6.3).

In this dialog box, the detailed information that is exported together with the action template should be provided. The name, the category (e.g. import, export, trimming) and the description for the action can be specified.

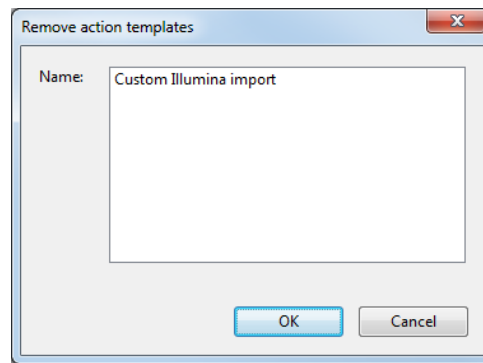


Figure 18.6.2: The *Remove action templates* dialog box.

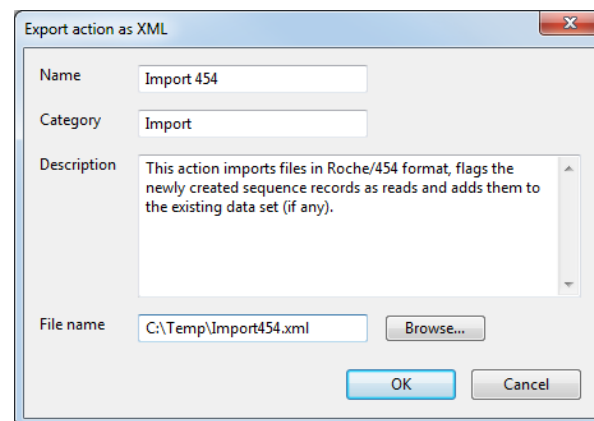


Figure 18.6.3: The *Export action as XML* dialog box.

18.6.6 Importing action templates

To import an action template, first create a new action in the project pipeline, or select an existing action where the import has to take place. Once the action template is injected, all information present in the selected action is permanently lost. Select **Action > Import action template...** from the *Power assembly* window to import an action template. The *Import action from XML* dialog box pops up (see Figure 18.6.4).

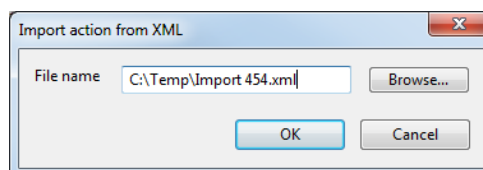


Figure 18.6.4: The *Import action from XML* dialog box.

In this dialog box, the XML file which contains the action template needs to be specified.

After import, the action name, the description and the action-defined operators are updated. The action is now ready to be executed.

Chapter 18.7

Overview of the operators

18.7.1 Introduction

Operators are the basic building blocks used to construct an action. Generally, each operator is characterized by its general parameters, the input and output parameters, and the specific operator parameters. The parameters require the input of data set fields of different specific field types. For example, when prompted for a *Sequence*, the data set fields listed in the drop down menu are all of the data type *Sequence*; when prompted for a sequence quality field, the data set fields from the drop down menu are only those data set fields of the data type *SequenceQuality*. Detailed information on the data set structure: the data set field types, the default data set fields, and the default sample set fields can be found in [18.7.2.5](#), [18.7.2.6](#) and [18.7.2.7](#), respectively.

18.7.2 The operator parameters

18.7.2.1 Background

In this section, the general operator parameters are discussed to prevent recurrence in the operator descriptions ([18.7.5.1](#) to [18.7.5.13](#)).

The *General parameters* include the user-defined *Operator Name*, a *Restriction test* on the data set to control the data records used when executing the operator, and a *Group-by modus* to run actions separately for different groups of sequence records.

18.7.2.2 General parameters

Operator name The name of the operator is required to refer to this operator at other locations in the Power Assembler.

Applications which require a reference to the operator, include:

- referring to the operator in subsequent expressions,
- defining runtime parameter settings for an operator parameter, and
- defining graphic parameter settings for an operator parameter.

If no name is specified, the operator cannot be referred to, and none of the above applications can be implemented.

Restriction test Logical test determining to which sequence records the operator applies.

Sequencing techniques typically result in very large data sets. As it is not always desirable to perform operators on the entire data set, a restriction test is used to restrict the application range of the operator. The restriction test is a logical expression. For more information on expressions, see [18.7.4.4](#). Sequence records that pass the restriction test are subjected to the defined operator, whereas sequence records that do not comply with the restriction are omitted from the operator.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately.

The **Group By** option combines data records into groups, based on the value in the specified field. In this way, sequence records with identical values in the specified field are grouped into a single group, and for each group, the operator is executed. If a record does not contain data for the specified field, this record is omitted from the operator execution.

This option is particularly useful when working e.g. with multiplex identifiers (MIDs). MIDs are part of the sequence reads and allow to link the reads to the different samples. For more information on MIDs, see [18.3.8](#). After the import of the reads, each read is assigned to the correct sample, based on the identifier tag. The different samples are further characterized by a specific sample key. Next, one can execute the same project workflow for each sample separately, by using the **Group by** modus on the field **Sample key**.

18.7.2.3 Background

Further, the operator parameters consist of:

- The *Input fields* which define the data set fields, needed for the execution of the operator.
- The *Output fields* which define the fields of the data set where the data, manipulated or created by the operator, is stored.
- The specific *Operator parameters*, which can take many forms e.g. data sets to be loaded, trimming thresholds, alignment algorithm parameters etc.

These parameters are all operator-specific, and therefore they are documented in the detailed overview of the operators, starting from [18.7.5.1](#) to [18.7.5.13](#).

18.7.2.4 The data set structure

In this subsection, the data set structure is elaborated. In [18.7.2.5](#), the field types defined in the Power Assembler are listed. In [18.7.2.6](#), an overview of the default data set fields is given, and finally, in [18.7.2.7](#), the default sample set fields are listed.

18.7.2.5 Data set field types

In this section, more information on the data set field types is reported.

Int A variable of this type contains an integer number.

Float A variable of this type contains a floating-point number.

Bool A variable of this type contains a boolean value.

String A variable of this type contains a string.

SeqKey A variable of this type contains a sequence record key.

SampleKey A variable of this type contains a sample record key.

FloatArray A variable of this type contains a list of floating-point numbers.

ManagerId A variable of this type contains the identifier of a user management operator.

Sequence A variable of this type contains a (possibly paired-end) sequence.

SequenceQuality A variable of this type contains quality scores for a (possibly paired-end) sequence.

CoverageMatrix A variable of this type contains coverage counts, both per-base and in total.

QualityMatrix A variable of this type contains per-base quality scores.

RegionList A variable of this type contains a list of regions.

PositionList A variable of this type contains a list of positions.

SeqKeyArray A variable of this type contains a list of sequence record keys.

OperatorId A variable of this type contains an operator identifier.

SeqDir A variable of this type contains the forward/reverse orientation in which a sequence was aligned.

18.7.2.6 Data set fields

In this section, the default data set fields, i.e. data set fields already defined in the dataset, are listed alphabetically. Additional fields can be added to the data set by the *Add data set field operator*. For each field, the *Field name*, the description and the field type are provided.

Associated alignment This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Associated aligned quality This field is designed to contain the aligned sequence quality with respect to the associated sequence record. This field is of type *SequenceQuality*.

Reverse complement This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Associated position This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated sequence This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated sequences list This field is designed to contain a list of keys from sequence records associated to this sequence record. This field is of type *SeqKeyArray*.

Barcode This field is designed to contain the multiplex identifier of the sample record.

Barcode sequence quality This field is designed to contain the qualities associated to a multiplex identifier sequence.

Coverage matrix This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Created by This field contains the operator identifier of the operator that created the sequence record. This field is of type *OperatorId*.

Group number This field is designed to contain the number of the group the sequence record was assigned to. This field is of type *Int*.

Is accepted This field is designed to discern between read sequence records whose alignment with respect to a reference sequence has been accepted. This field is of type *Bool*.

Is in region list This field is designed to keep track of those sequence records that have a substantial overlap with a list of regions. This field is of type *Bool*.

Is read This field is designed to discern read sequence records from all other sequence records. This field is of type *Bool*.

Is reference This field is designed to discern between reference sequence records and read sequence records. This field is of type *Bool*.

Is target This field is designed to discern between target sequence records and input and intermediate sequence records. This field is of type *Bool*.

Key This field contains the sequence record key, which can be used to identify the sequence record throughout the database. This field is of type *SeqKey*.

Last modified by This field contains the operator identifier of the last operator that modified the sequence record. This field is of type *OperatorId*.

Managed This field contains the operator identifier of the user management operator handling the sequence record. This field is of type *ManagerId*.

Mapping degeneracy This field is designed to contain the mapping degeneracy of a sequence record when positioned with respect to a reference sequence. This field is of type *Int*.

Mapping discrimination This field is designed to contain the mapping discrimination of a sequence record when positioned with respect to a reference sequence. This field is of type *Float*.

Matching position This field is designed to contain the best position of a sequence record without gapped alignment when positioned with respect to a reference sequence. This field is of type *Int*.

Name This field contains the user-friendly name of the sequence record. This field is of type *String*.

Position list This field is designed to contain a list of positions. This field is of type *PositionList*.

Quality matrix This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Read type This field is designed to contain the type of the read sequence. This field is of type *String*.

Region list This field is designed to contain a list of regions. This field is of type *RegionList*.

Sample key This field contains the key of the sample record associated to the sequence record. This field is of type *SampleKey*.

Sequence This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Sequence identity This field is designed to contain the sequence identity score of a sequence record when positioned with respect to a reference sequence. This field is of type *Float*.

Source This field contains the original source of the sequence record. This field is of type *String*.

Trimmed sequence This field is designed to contain a trimmed sequence. This field is of type *Sequence*.

Trimmed sequence quality This field is designed to contain the qualities associated to a trimmed sequence. This field is of type *SequenceQuality*.

Value profile This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

18.7.2.7 Sample set fields

In this section, the sample set fields and their field types are listed. For more information on the *Entry key*, the *Experiment*, and the *Source* field, see [18.3.8](#), the *Samples panel*.

Barcode This field is designed to contain the multiplex identifier of the sample record. This field is of type *Sequence*.

Entry key This field contains the BioNumerics entry key corresponding to the sample record. This field is of type *String*.

Experiment This field contains the BioNumerics experiment name corresponding to the sample record. This field is of type *String*.

Sample key This field contains the key of the sample record associated to the sequence record. This field is of type *SampleKey*.

Source This field contains the original source of the sequence records associated to this sample record. This field is of type *String*.

18.7.2.8 Power assembler qualities

The Power Assembler uses a Phred-like system for quality scores. A quality score q can be interpreted as the log-scaled probability p that the corresponding base call is wrong,

$$p = 10^{-\frac{q}{10}}$$

The quality scores in the Power Assembler can range from 0 to 64, or, in terms of probability, from 1 to about 0.000000389.

18.7.2.8.1 Importing quality scores

Quality scores can be imported together with the corresponding sequences. Since every type of data has its own quality score scaling, the raw sequence quality can be converted to a sequence quality that makes sense in the context of the Power Assembler. This conversion is done automatically in the predefined actions that import read data (see [18.5.1.2](#)).

18.7.2.8.2 Quality-based trimming

Quality scores can be used for selecting high-quality reads and for trimming. The trimming operators provide the possibility to exclude read sequences when the minimum or average quality is too low, and provide means to remove individual low-quality reads or low-quality regions at the beginning or the end of a read. The predefined action *Quality trimming* (see [18.5.3.2](#)) uses these operators to provide an easy-to-use trimming action based on quality scores.

18.7.2.8.3 How quality scores are used in mapping

The use of quality scores in the mapping operators is twofold. On the one hand, quality scores are used in the alignment procedure to determine a weighted alignment score. On the other hand, they have an influence on the quality score of the assembled sequence.

18.7.2.8.4 Exporting quality scores

Quality scores can be exported to the BioNumerics database along with the sequence. This is done automatically by all of the predefined export actions (see [18.5.8](#)).

18.7.3 The operator categories

The existing operators are functionally divided into a number of categories:

- **Import & Export**

The import operators create the data set used throughout the project pipeline. The import of data includes opening a data set generated by a previous operator, the import of sequences from the BioNumerics database or the import of raw sequence data (reads) obtained from a sequencing project. The export functionalities can be used to export different types of information e.g. to export sequences to the BioNumerics database, to export sequences to a FASTA/FASTQ file and/or to export specific information fields from the data set into the BioNumerics entry information fields or into a txt file.

See [18.7.5.1](#) for detailed information.

- **Trimming**

Trimming operators are used to trim and filter the sequence reads in the data set. Different trimming operators are available, based on e.g. the quality of the reads, the length of the reads, a specific signature present in the reads, the GC-content of the reads or a combination of these criteria.

See [18.7.5.2](#) for detailed information.

- **Preprocessing**

The **Preprocessing** operators include e.g. operators for processing the multiplex identifiers for read sequences from multiple samples, splitting the 454[®] paired-end read (which exists of a first part of the paired-end read, the adapter and the second part of the paired-end read) into the adapter and the two paired-end reads, searching for sequence signatures or extracting subsequences.

See [18.7.5.3](#) for detailed information.

- **Mapping**

Operators related to the assembly of the read sequences, e.g. positioning of the reads on the reference sequence(s), creating a gapped target sequence for the assembly, calculating the target coverage and performing the consensus base calling or creating the assembly map, are brought together in the **Mapping** operators.

See [18.7.5.4](#) for detailed information.

- **De novo assembly**

Within the category of **De novo assembly**, the operators which are based on the tools Velvet and Ray, are available. In addition, a greedy assembler for de novo finishing is provided.

See [18.7.5.5](#) for detailed information.

- **Sequence clustering**

Currently, the **Sequence clustering** category contains only one operator that assigns read sequences to groups based on the barcode information present in the sample records or based on representative sequences from the data set.

See [18.7.5.6](#) for detailed information.

- **Region tools**

The **Region tools** contain operators to identify regions. These regions can then be manipulated or extracted as separate sequence records.

See [18.7.5.6](#) for detailed information.

- **Sequence profiles & curves**

The operators grouped within the **Sequence profiles & curves** are used to create a profile over a sequence (e.g. sequence quality, coverage) (see [18.3.5](#)), and subsequently create a curve of this sequence profile which can then be displayed in the *Sequence curves panel*.

See [18.7.5.8](#) for detailed information.

- **Summary graphs**

The operators that provide **Summary graphs** are used to summarize global properties (entered as numerical expressions) for the data set e.g. the GC-content, or the length of the sequences. Also histograms on the coverage, sequence quality, as well as any other windowed average, minimum and maximum curve analysis on a sequence profile can be calculated (see also [18.3.6](#)).

See [18.7.5.9](#) for detailed information.

- **Statistics**

The **Statistics** operator provides a global summary statistic calculated on (a part of) the data set.

See [18.7.5.10](#) for detailed information.

- **Data set tools**

A variety on operators acting on the sequence records that form the data set can be found in the **Data set tools**. They are used for manipulations of the entire data set and/or for the manipulation of information fields within these data sets. These operators include e.g. the addition of a field into a data set, performing an arithmetic calculation on a field, setting a specific value for a particular field, merging different data sets into one data set, removing sequence records, and removing fields from the data set.

See [18.7.5.11](#) for detailed information.

- **Sample tools**

In the case of simultaneous sequencing of multiple samples, a strategy to assign each sequence record to the correct sample is achieved by the use of a unique identity tag for each sample. The operators that perform these assignments are collected in the **Sample tools**. Using these operators, one can add sample links, parse sample information from a specific field, shortly, manipulate the sample record data set to achieve a unique sample identity (e.g. through the use of MIDs). See also [18.3.8](#) for more information.

See [18.7.5.12](#) for detailed information.

- **Project tools**

The **Project tools** allow one to define multiple project properties. This option is very useful for managing project-specific settings.


See [18.7.5.13](#) for detailed information.

The following sections provide a detailed description of each operator present in the *Operators tree* from the *Action design panel* (see 18.3.9).

Please note that the description of the general operator parameters can be found in 18.7.2, and is not repeated for each operator. For each operator the general parameters, and the input and output parameters are just listed, while the operator parameters are discussed in detail.

18.7.4 The operator parameter properties

18.7.4.1 Displaying parameter properties of an operator

To display the parameter properties of a selected operator from the *Action flowchart panel*, select **Action > Action design > Operator properties...** (⚙️), and subsequently press the  button next to one of the operator parameters. This will launch the *Parameter properties dialog box*.

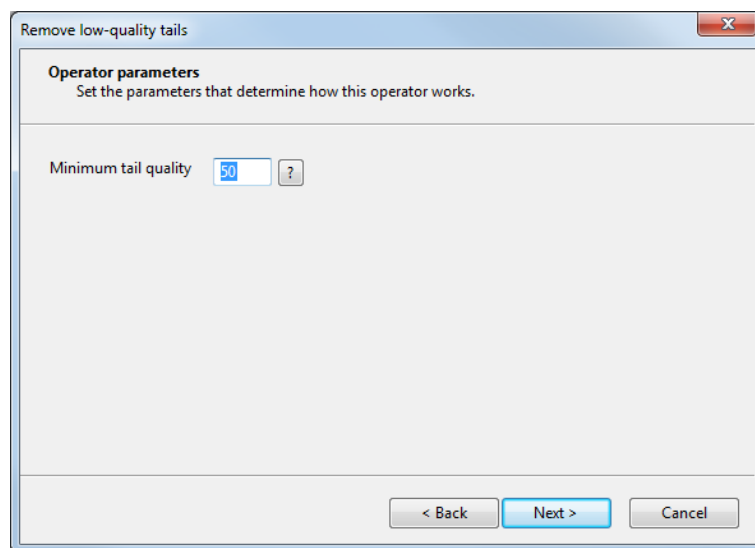


Figure 18.7.1: Example of runtime questionable operator parameter.

The *Parameter properties dialog box* summarizes the **Expression**, the **Description**, the **Graphical link** and the **Runtime properties** from the operator parameter.

- In the **Expression** dialog, a logical expression can be entered which defines the value of the operator parameter.
- The **Description** contains a brief description of the operator parameter.
- The **Graphical link** displays the linkage information of the parameter with a summary graph. More information on parameters linked to summary graphs can be found in 18.7.4.3.
 - If the parameter cannot be linked, the message *This parameter cannot be linked with a summary graph* is displayed (Figure 18.7.3).
 - If a link to a summary graph can be defined, the message *Link parameter with summary graph* is displayed. Defined graphic links are documented by the *Linked summary graph*, the *Link type*, and the *Link name* (Figure 18.7.4).
- The **Runtime properties** indicate whether the parameter will be queried at runtime. More information on runtime parameter questioning can be found in 18.7.4.2.

- If the parameter is not prompted for at runtime, the message *Query parameter at runtime: No* is displayed (Figure 18.7.2).
- If the parameter is prompted for at runtime, the *Label*, the *Group*, and the *Page* where the parameter is queried, are displayed (Figure 18.7.3).

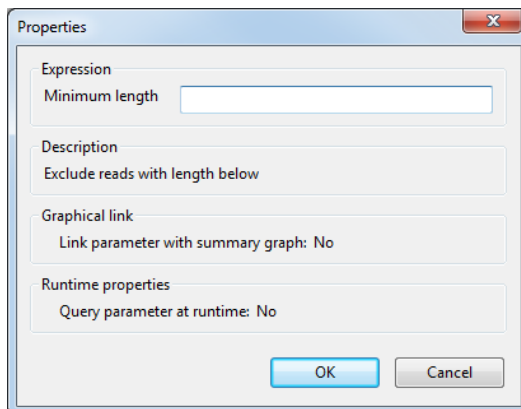


Figure 18.7.2: The *Parameter properties dialog box* for a parameter that is not prompted for at runtime. This parameter can be linked with a summary graph, but no links have been defined.

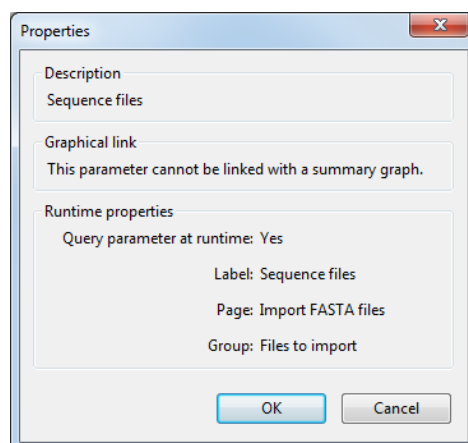



Figure 18.7.3: The *Parameter properties dialog box* for a parameter that is prompted for at runtime. This parameter cannot be linked with a summary graph.

18.7.4.2 Runtime parameter questioning

As operators contain a lot of parameters, many default or predefined parameter values are chosen design time. These parameter values are defined within the operator properties, and can be accessed or altered from the *Action design panel*. Besides the predefined parameters, some parameters are very specific, and therefore need to be questioned to the user at runtime. When executing an action, predefined parameters are not prompted for but the default parameter values as defined in the operator properties are used, whereas the parameters queried at runtime need to be filled in by the user before the calculation of the action starts.

Each operator does have one or multiple parameters that can be prompted for at runtime. These parameters can be set from the *Action flowchart panel*. When looking into the dialog of the operator, some parameters have a  button displayed, which implies that these parameter values can be made questionable at runtime.

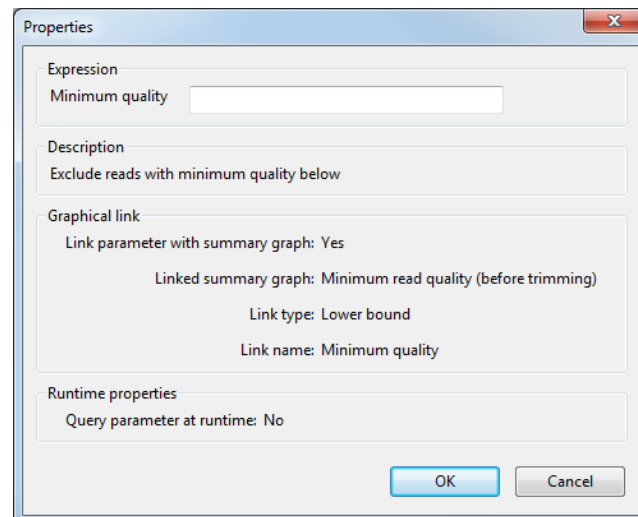


Figure 18.7.4: The *Parameter properties dialog box* for a parameter that is not prompted for at runtime. This parameter value **Minimum read quality** is a lower bound threshold linked with a summary graph.

For each action, the runtime parameters need to be defined. To modify the runtime parameters of a selected action, select **Action** > **Action design** > **Runtime parameters...** (▶) from the *Action flowchart panel*. This launches the *Runtime parameters dialog box* (see Figure 18.7.5).

The *Runtime parameters dialog box* is used to manage the runtime dialog box of the selected action, and gives an overview of the different groups and pages that host the runtime parameters. A page is displayed as a separate wizard page in the runtime dialog box, whereas a group contains a variety of related parameters prompted for at runtime.

A new *page* can be added by pressing the <Add page...> button. In the *Add page dialog box*, the page name and the description for this page can be entered. Once a page is created, an empty group list is automatically added, displayed as a sub-node <no groups present>. On each page, different groups can be defined.

A new *group* can be added by first selecting the page that should contain the new group, and then pressing the <Add group...> button.

The group name and a description for this group can be entered in the *Add group dialog box* (Figure 18.7.6). When no group name is defined, the parameters will be listed on the page without group structure. Parameters belonging to a specified group will be structured in the runtime dialog box within a separate box, defined by the group name. When a group is created, automatically an empty parameter list is added, displayed as a sub-node <no parameters present>.

Once pages and groups are defined, the *parameters* can be added to the runtime dialog structure. Therefore, first select the group to which the parameter will be added, and then press the <Add parameter...> button. This will launch the *Add parameter dialog box* (Figure 18.7.7). The *Add parameter dialog box* displays the available parameters in a tree structure. The operators present in the action are shown as separate folders. When clicking on one of these operators, all parameters from the selected operator that can be queried at runtime, are displayed. To add a parameter, click the parameter name. Automatically the parameter label is filled with the parameter name. Take into account that parameters must be structured within a specific *group* of a page, and cannot be added just to the page itself.

In general, if no page name, group name or parameter name is defined, the identifier is displayed as a number, e.g. <5> (see Figure 18.7.5), and no name will be displayed in the runtime dialog box. Operator parameters can only be managed if a unique operator name was defined for the operator. If not, the operator and the related parameters cannot be referred to (see 18.7.2.2). The **Name** and **Description** of a selected node (whether it is a page, group or parameter) are displayed in the *Node details box* at the bottom of the

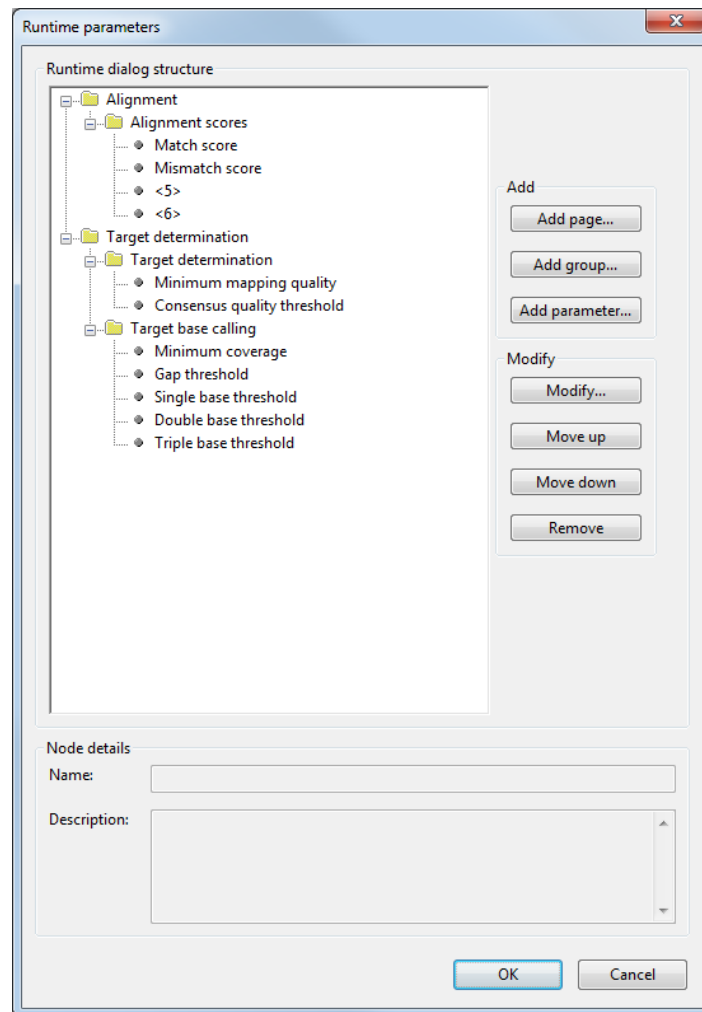


Figure 18.7.5: The *Runtime parameters* dialog box.

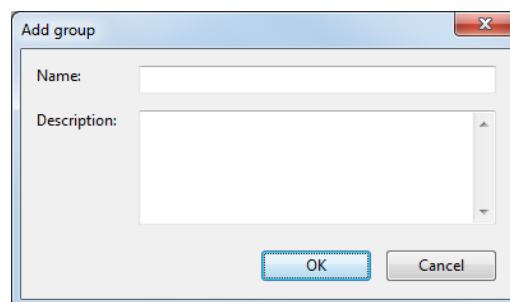


Figure 18.7.6: The *Add group* dialog box, identical to the *Add page* dialog box.

Runtime parameters dialog box.

The structure of the wizard pages can be modified in various ways.

To modify the name and description of a selected page, group or parameter, press **<Modify...>**. This will launch the *Modify page* dialog box, the *Modify group* dialog box, or the *Modify parameter* dialog box, respectively (Figure 18.7.8).

In these dialog boxes, the name and description of an existing page, group or parameter can be altered. The default parameter name is automatically filled as parameter label.

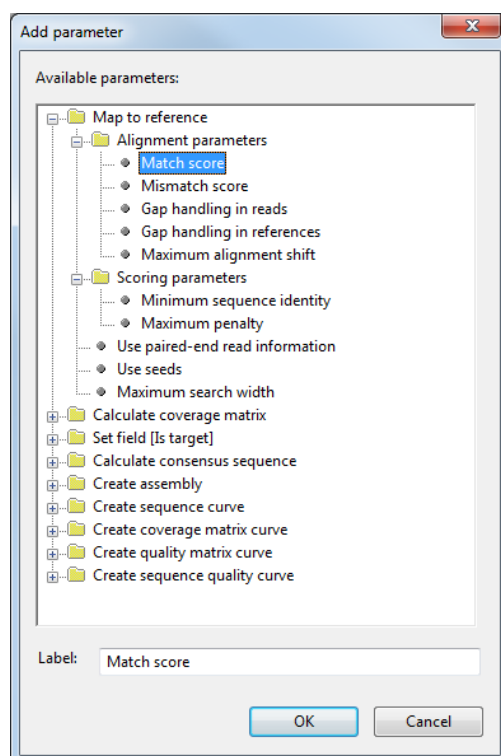


Figure 18.7.7: The *Add parameter dialog box*.

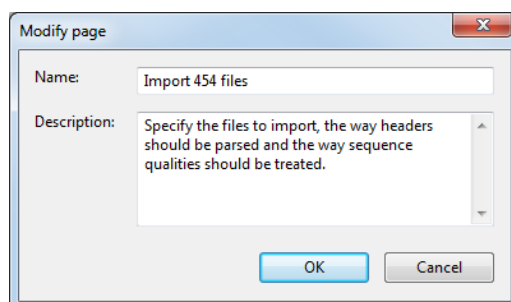


Figure 18.7.8: The *Modify page dialog box*, identical to the *Modify group dialog box* and the *Modify parameter dialog box*.

The order in which the pages, groups and parameters are displayed in the runtime dialog box can be changed by selecting the item in the tree structure, and pressing the **<Move up>** or **<Move down>** buttons. This will move the selected item one position in the tree. A selected item can be removed by pressing the **<Remove>** button.

If an action is executed, the wizard, organized as in the *Runtime parameters dialog box*, pops up, prompting the user for the runtime parameters for that action. If a series of actions is launched and multiple actions are executed, the wizard is composed by a sequence of the action-specific dialog boxes.



The Power assembler project needs to be closed and re-opened before the changes in the *Runtime parameters dialog box* are actualized.

18.7.4.3 Linking parameters to summary graphs

Summary graphs provide an overview of the characteristics of a set of sequences (see 18.3.6). Parameter values queried for at runtime can also be defined by linking the parameter values to thresholds, set in these

summary graphs. To set the operator parameters for e.g. trimming, one can manually enter the threshold value in the dialog box, or link this trimming threshold to a summary graph (Figure 18.7.9). The advantage of this approach is that thresholds are defined from the data summaries, displayed as summary graphs. By using these graphical links, thresholds can be easily assessed, and manually adjusted by the user.

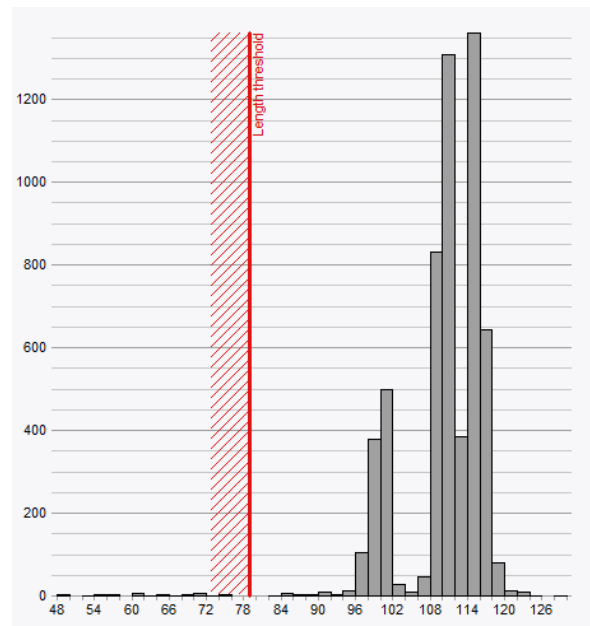


Figure 18.7.9: Summary graph with linked operator parameter

The graphical link settings are defined in the *Graphic parameters dialog box* (Figure 18.7.10). This dialog box is launched from the *Action flowchart panel* by selecting **Action** > **Action design** > **Summary graph boundaries...** (📊).

The *Graphic parameters dialog box* displays the existing summary graphs present in the project. If no parameters are linked to the graphic, <no parameters> is displayed. If a parameter is linked to the graph, the display name of the parameter is shown.

To link a parameter to a summary graph, first select the summary graph that will deliver the parameter value. Next, press <Add parameter...>. This will launch the *Add graphic parameter dialog box* (Figure 18.7.11).

The *Add graphic parameter dialog box* consists of different folders in a tree-like structure. The folders represent the actions of the project. By selecting a folder, the corresponding operators defined in that action are displayed. Parameters can only be managed if a unique name was given to the operator. If not, the operator and the related parameters cannot be referred to. When selecting an operator from the action, all the possible graphic operator parameters are displayed. Select the graphic parameter that should be linked to the summary graph and specify the **Parameter type**:

- *No bound* means that no graphical display is defined,
- *Upper bound* means that the parameter value will be displayed as a maximum value, and
- *Lower bound* means that the parameter value will be displayed as a minimum value.

The **Display name** is the name displayed next to the red line that appears on the summary graph once the parameter is graphically linked (Figure 18.7.9).

To change the graphic parameter properties, select the graphic parameter and press <Modify...>. This will launch the *Modify graphic parameter dialog box* (Figure 18.7.12).

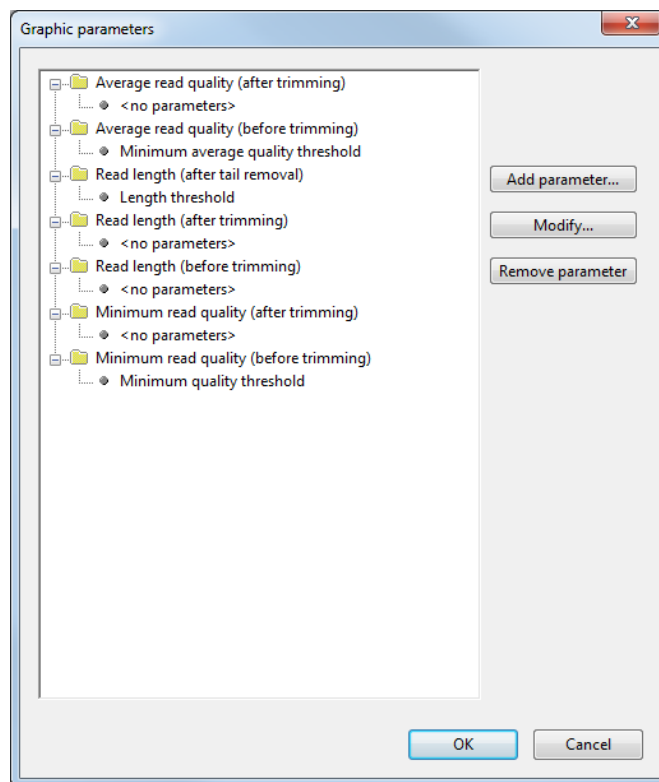


Figure 18.7.10: The *Graphic parameters* dialog box.

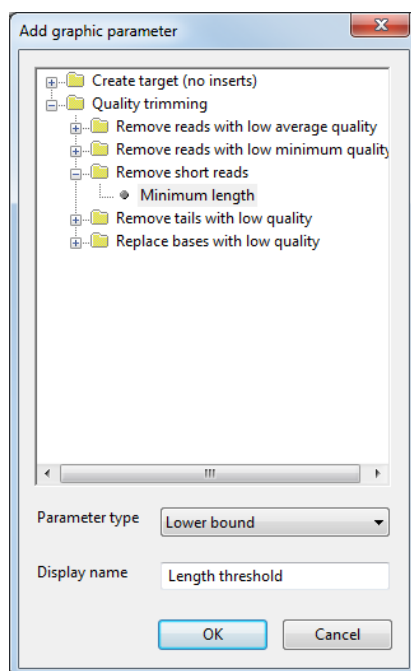


Figure 18.7.11: The *Add graphic parameter* dialog box.

In this dialog, the *Parameter type* and *Display name* of the graphic parameter can be set.

A selected graphic parameter can be removed by the *<Remove parameter>* button.

From the moment the graphic parameter is added, the current parameter value is displayed as a red line on the related summary graph. The next time the action is executed, the parameter value will be updated from

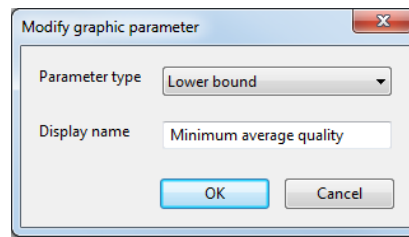


Figure 18.7.12: The *Modify graphic parameter* dialog box.

the summary graph.

18.7.4.4 Expressions

High-throughput sequencing techniques typically result in very large data sets. As it is not always desirable to perform operators on the entire data set, restriction tests are used to restrict the application range of an operator. Sequence records that pass the restriction test are subjected to the defined operator, whereas sequence records that do not comply with the restriction are omitted from the operator.

When prompted for a restriction test, one can manually insert the logical expression in the *Operators restriction test dialog box* (see Figure 18.7.13), or one can build the expression by using the expression builder (see Figure 18.7.14). The expression builder can be launched by selecting **<Build...>** next to the *Operators restriction test dialog box*.

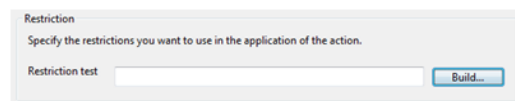


Figure 18.7.13: The *Operators restriction test* dialog box.

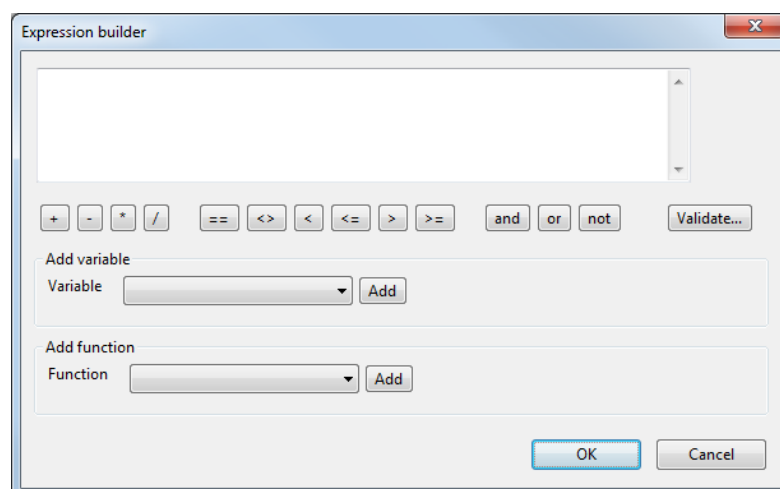


Figure 18.7.14: The *Expression builder* dialog box.

The expression builder offers easy access to the names of variables and functions present in the data set. Therefore, it makes the construction of expressions more accessible and facilitates the creation of an expression from scratch. The dialog box on top contains the build expression. One can directly type the expression into the dialog box or use the drop-down list under **Add variable** or **Add function** combined with the **<Add>** button to insert a variable or function into the expression. Variables will be entered as [Variable], whereas

the functions will be inserted as `Function(...)`. Strings can be entered between quotes ("example string"). Below the box, some commonly used operators are displayed. To add an operator to the expression, simply click the appropriate button.

To check the validity of an expression, press **<Validate>**. The software then checks the syntax of the expression and reports an error if it detects an inconsistency.

Some examples of simple valid expressions are:

- `[IsReference]==True`: to select all reference sequence(s) from the data set.
- `[Last modified by]==[Actions.TrimSeqQual]`: to select all sequence records that were last modified by the trimming operator (Operator Name: TrimSeqQual) which filtered for the sequence quality.
- `"Target" + ToString([Index])`: can be used as a name expression when creating new sequence records. The name of the sequence record will be constructed by combining the string "Target" and the string created from the index e.g. *Target 1*.

If an expression is validated as being correct, exit the *Expression builder dialog box* and the expression is automatically filled in the restriction test from the *General settings dialog box* of the operator. Besides restriction tests, expressions can also be set in the parameter properties and e.g. to define export labels and custom penalty functions.

Following, an overview of the available expression functions:

* Takes the product of two objects.

```
Int *(Int 'arg1', Int 'arg2')
Float *(Float 'arg1', Float 'arg2')
FloatArray *(Float 'arg1', CoverageMatrix 'arg2')
FloatArray *(Float 'arg1', SequenceQuality 'arg2')
FloatArray *(Float 'arg1', Sequence 'arg2')
FloatArray *(Float 'arg1', FloatArray 'arg2')
FloatArray *(CoverageMatrix 'arg1', Float 'arg2')
FloatArray *(SequenceQuality 'arg1', Float 'arg2')
FloatArray *(Sequence 'arg1', Float 'arg2')
FloatArray *(FloatArray 'arg1', Float 'arg2')
```

+ Takes the sum of two objects.

```
Int +(Int 'arg1', Int 'arg2')
Int +(Int 'arg')
Float +(Float 'arg1', Float 'arg2')
Float +(Float 'arg')
String +(String 'arg1', String 'arg2')
FloatArray +(Float 'arg1', CoverageMatrix 'arg2')
FloatArray +(Float 'arg1', SequenceQuality 'arg2')
FloatArray +(Float 'arg1', Sequence 'arg2')
FloatArray +(Float 'arg1', FloatArray 'arg2')
FloatArray +(CoverageMatrix 'arg1', Float 'arg2')
FloatArray +(SequenceQuality 'arg1', Float 'arg2')
FloatArray +(Sequence 'arg1', Float 'arg2')
FloatArray +(FloatArray 'arg1', Float 'arg2')
FloatArray +(CoverageMatrix 'arg1', CoverageMatrix 'arg2')
FloatArray +(CoverageMatrix 'arg1', Sequence 'arg2')
```

```

FloatArray +(CoverageMatrix 'arg1', SequenceQuality 'arg2')
FloatArray +(CoverageMatrix 'arg1', FloatArray 'arg2')
FloatArray +(Sequence 'arg1', CoverageMatrix 'arg2')
FloatArray +(Sequence 'arg1', Sequence 'arg2')
FloatArray +(Sequence 'arg1', SequenceQuality 'arg2')
FloatArray +(Sequence 'arg1', FloatArray 'arg2')
FloatArray +(SequenceQuality 'arg1', CoverageMatrix 'arg2')
FloatArray +(SequenceQuality 'arg1', Sequence 'arg2')
FloatArray +(SequenceQuality 'arg1', SequenceQuality 'arg2')
FloatArray +(SequenceQuality 'arg1', FloatArray 'arg2')
FloatArray +(FloatArray 'arg1', CoverageMatrix 'arg2')
FloatArray +(FloatArray 'arg1', Sequence 'arg2')
FloatArray +(FloatArray 'arg1', SequenceQuality 'arg2')
FloatArray +(FloatArray 'arg1', FloatArray 'arg2')

```

- Takes the difference of two objects.

```

Int -(Int 'arg1', Int 'arg2')
Int -(Int 'arg')
Float -(Float 'arg1', Float 'arg2')
Float -(Float 'arg')
FloatArray -(Float 'arg1', CoverageMatrix 'arg2')
FloatArray -(Float 'arg1', SequenceQuality 'arg2')
FloatArray -(Float 'arg1', Sequence 'arg2')
FloatArray -(Float 'arg1', FloatArray 'arg2')
FloatArray -(CoverageMatrix 'arg1', Float 'arg2')
FloatArray -(SequenceQuality 'arg1', Float 'arg2')
FloatArray -(Sequence 'arg1', Float 'arg2')
FloatArray -(FloatArray 'arg1', Float 'arg2')
FloatArray -(CoverageMatrix 'arg1', CoverageMatrix 'arg2')
FloatArray -(CoverageMatrix 'arg1', Sequence 'arg2')
FloatArray -(CoverageMatrix 'arg1', SequenceQuality 'arg2')
FloatArray -(CoverageMatrix 'arg1', FloatArray 'arg2')
FloatArray -(Sequence 'arg1', CoverageMatrix 'arg2')
FloatArray -(Sequence 'arg1', Sequence 'arg2')
FloatArray -(Sequence 'arg1', SequenceQuality 'arg2')
FloatArray -(Sequence 'arg1', FloatArray 'arg2')
FloatArray -(SequenceQuality 'arg1', CoverageMatrix 'arg2')
FloatArray -(SequenceQuality 'arg1', Sequence 'arg2')
FloatArray -(SequenceQuality 'arg1', SequenceQuality 'arg2')
FloatArray -(SequenceQuality 'arg1', FloatArray 'arg2')
FloatArray -(FloatArray 'arg1', CoverageMatrix 'arg2')
FloatArray -(FloatArray 'arg1', Sequence 'arg2')
FloatArray -(FloatArray 'arg1', SequenceQuality 'arg2')
FloatArray -(FloatArray 'arg1', FloatArray 'arg2')

```

- / Takes the quotient of two objects.

```

Int /(Int 'arg1', Int 'arg2')
Float /(Float 'arg1', Float 'arg2')
FloatArray /(Float 'arg1', CoverageMatrix 'arg2')
FloatArray /(Float 'arg1', SequenceQuality 'arg2')
FloatArray /(Float 'arg1', Sequence 'arg2')

```

```
FloatArray /(Float 'arg1', FloatArray 'arg2')
FloatArray /(CoverageMatrix 'arg1', Float 'arg2')
FloatArray /(SequenceQuality 'arg1', Float 'arg2')
FloatArray /(Sequence 'arg1', Float 'arg2')
FloatArray /(FloatArray 'arg1', Float 'arg2')
```

< Tests whether the first object is smaller than the second.

```
Bool <(Int 'arg1', Int 'arg2')
Bool <(Float 'arg1', Float 'arg2')
```

<= Tests whether the first object is smaller than or equal to the second.

```
Bool <=(Int 'arg1', Int 'arg2')
Bool <=(Float 'arg1', Float 'arg2')
```

<> Tests whether two objects are different.

```
Bool <>(Int 'arg1', Int 'arg2')
Bool <>(Float 'arg1', Float 'arg2')
Bool <>(String 'arg1', String 'arg2')
```

== Tests whether two objects are equal.

```
Bool ==(Bool 'arg1', Bool 'arg2')
Bool ==(Int 'arg1', Int 'arg2')
Bool ==(Float 'arg1', Float 'arg2')
Bool ==(String 'arg1', String 'arg2')
Bool ==(OperatorId 'arg1', OperatorId 'arg2')
```

> Tests whether the first object is bigger than the second.

```
Bool >(Int 'arg1', Int 'arg2')
Bool >(Float 'arg1', Float 'arg2')
```

>= Tests whether the first object is bigger than or equal to the second.

```
Bool >=(Int 'arg1', Int 'arg2')
Bool >=(Float 'arg1', Float 'arg2')
```

ACount Counts the number of A's in a sequence.

```
Int ACount(Sequence 'arg1')
```

Avg Computes the average of a profile.

```
Float Avg(SequenceQuality 'arg1')
Float Avg(CoverageMatrix 'arg1')
Float Avg(FloatArray 'arg1')
```

CCount Counts the number of C's in a sequence.

```
Int CCount(Sequence 'arg1')
```


FALSE false

```
Bool false()
```

GCCCount Counts the number of G/C's in a sequence.

```
Int GCCCount(Sequence 'arg1')
```

GCount Counts the number of G's in a sequence.

```
Int GCount(Sequence 'arg1')
```

HasField Checks whether a field is present in the data set.

HasProperty Checks whether a property is present.

HasSampleField Checks whether a field is present in the samples data set.

If If

```
Bool If(Bool 'condition', Bool 'then-statement', Bool 'else-statement')
Int If(Bool 'condition', Int 'then-statement', Int 'else-statement')
Float If(Bool 'condition', Float 'then-statement', Float 'else-statement')
String If(Bool 'condition', String 'then-statement', String 'else-statement')
OperatorId If(Bool 'condition', OperatorId 'then-statement', OperatorId 'else-statement')
```

IsPaired Checks whether a sequence is paired.

```
Bool IsPaired(Sequence 'arg1')
Bool IsPaired(SequenceQuality 'arg1')
```

IsPresent Checks whether a value is present.

Length Determines the size of a sequence.

```
Int Length(Sequence 'arg1')
Int Length(SequenceQuality 'arg1')
Int Length(Sequence 'arg1')
Int Length(CoverageMatrix 'arg1')
Int Length(QualityMatrix 'arg1')
Int Length(Sequence 'seq', Int 'partnr')
Int Length(SequenceQuality 'seq', Int 'partnr')
```

Max Computes the maximum of a profile.

```
Int Max(SequenceQuality 'arg1')
Int Max(CoverageMatrix 'arg1')
Float Max(FloatArray 'arg1')
```

MaxHomopolymerLength Determines the length of the longest homopolymer in a sequence.

Min Computes the minimum of a profile.

```
Int Min(SequenceQuality 'arg1')
Int Min(CoverageMatrix 'arg1')
Float Min(FloatArray 'arg1')
```

NCount Counts the number of N's in a sequence.

```
Int NCount(Sequence 'arg1')
```

RollingAvg Calculates the rolling average of a profile.

SampleKey Creates a sample record key.

```
SampleKey SampleKey(Int 'projectid', Int 'sampleid')
```

SequenceKey Creates a sequence record key.

```
SeqKey SequenceKey(Int 'projectid', Int 'seqid')
```

SpaceBetweenParts Determines the space between two parts of a sequence.

```
Int SpaceBetweenParts(Sequence 'arg1')
```

TCount Counts the number of T's in a sequence.

```
Int TCount(Sequence 'arg1')
```

TRUE true

```
Bool true()
```

ToFloat Converts a value to a float.

```
Float ToFloat(Int 'arg1')
```

ToString Converts a value to a string.

```
String ToString(Int 'arg1')
String ToString(Float 'arg1')
```

WindowAvg

^ Returns the 'arg2' power of 'arg1'.

```
Float ^(Float 'arg1', Float 'arg2')
Float ^(Float 'arg1', Int 'arg2')
```

abs Returns the absolute value of 'arg'.

```
Int abs(Int 'arg')
Float abs(Float 'arg')
FloatArray abs(CoverageMatrix 'arg1')
FloatArray abs(SequenceQuality 'arg1')
FloatArray abs(Sequence 'arg1')
FloatArray abs(FloatArray 'arg1')
```

acos Returns the arc cosine of 'arg' (in radians).

```
Float acos(Float 'arg')
```

and AND operator

```
Bool and(Bool 'arg1', Bool 'arg2')
```

asin Returns the arc sine of 'arg' (in radians).

```
Float asin(Float 'arg')
```

atan Returns the arc tangent of 'arg' (in radians).

```
Float atan(Float 'arg')
```

cos Returns the cosine of 'arg' (in radians).

```
Float cos(Float 'arg')
```

exp Returns the exponential of 'arg'.

```
Float exp(Float 'arg')  
Float exp(Float 'arg1')  
Float exp(Int 'arg1')
```

floor Returns the largest integral value smaller than or equal to 'arg'.

```
Float floor(Float 'arg')  
FloatArray floor(FloatArray 'arg1')
```

ln Natural logarithm.

```
Float ln(Float 'arg1')  
Float ln(Int 'arg1')
```

not NOT operator

```
Bool not(Bool 'arg1')
```

or OR operator

```
Bool or(Bool 'arg1', Bool 'arg2')
```

sign Returns the sign of 'arg'.

```
Int sign(Int 'arg')  
Int sign(Float 'arg')
```

sin Returns the sine of 'arg' (in radians).

```
Float sin(Float 'arg')
```

ToInt

tan Returns the tangent of 'arg' (in radians).

```
Float tan(Float 'arg')
```

18.7.5 Power assembler operators ---

18.7.5.1 Import & Export ---

The import and export operators make both data from external files and data from the database available to the Power Assembler, and offer the possibilities to write data to both external files and the database.

18.7.5.1.1 Open data set ---

This operator reads a sequence record data set from an action in a power assembly project. This sequence record data set can be the data set from the previous action in the same project, or a data set with a non-empty name from any project in the same BioNumerics database.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Dataset Name of the data set.

The name of a data set from an action in a power assembly, or the placeholder *Use last data set from current project* can be selected. The latter option is used to load the data set from the previous component in the project. If, in this case, there is no previous component, no data set is opened. If the previous component has more than one result data set, only the first result data set is opened.

18.7.5.1.2 Import FASTA files ---

This operator imports FASTA sequence files into a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Sequence files Location of the sequence files.

This parameter contains the full path to the sequence files to be imported. More than one file can be specified. All sequences are imported in one sequence record data set.

Delimiter separating header info fields Delimiter for parsing the header information.

The delimiter separates the information fields in the header of the sequence. The label number (see below) determines which information field is used to set the *Name* field of the sequence record that is being imported.

As name for the sequence record, use label Number of the information field to be used as sequence record name.

The content of the information field with this number is used to set the *Name* field of the sequence record being imported.

Source label Label assigned to the source field of a sequence record.

While importing sequences, the *Source* field of a sequence record in the data set can be filled in with information about the source of this sequence record. This information is formed by the string expression provided in this parameter. The variable *[Filename]* can be used to refer to the name of the file that is being imported.

Import as paired-end reads Whether or not to import sequences as paired-end.

When importing sequences as paired-end reads, files are coupled based on the filename. Two files with the same name except for the last two characters (which should be "_1" and "_2") are supposed to contain each one end of a paired-end read.

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.3 Import Solexa/Illumina files

This operator imports Solexa/Illumina[®] sequence files into a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Sequence files Location of the sequence files.

This parameter contains the full path to the sequence files to be imported. More than one file can be specified. All sequences are imported in one sequence record data set.

Source label Label assigned to the source field of a sequence record.

While importing sequences, the *Source* field of a sequence record in the data set can be filled in with information about the source of this sequence record. This information is formed by the string expression provided in this parameter. The variable *[Filename]* can be used to refer to the name of the file that is being imported.

Import sequence quality Whether or not to import sequence qualities.

If checked, the raw sequence qualities are imported and converted to Phred qualities using the conversion formula (see below). If unchecked, no sequence qualities are imported.

Conversion formula Formula for converting raw sequence qualities to Phred qualities.

Quality scores in the Power assembler are Phred scores on a scale from 0 to 63 included. For Solexa/Illumina[®] reads processed with Solexa Pipeline v1.3 or above, the conversion formula is

$$bnQ = illQ - 64,$$

whereas for Solexa/Illumina[®] reads processed with Solexa Pipeline earlier than v1.3, the conversion formula is

$$bnQ = \frac{10}{\log_{10}} \log \left(1 + 10^{\frac{illQ}{10}} \right).$$

Import as paired-end reads Whether or not to import sequences as paired-end.

When importing sequences as paired-end reads, files are coupled based on the filename. Two files with the same name except for the last two characters (which should be "_1" and "_2") are supposed to contain each one end of a paired-end read.

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.4 Import Roche/454 files

This operator imports Roche/454[®] sequence files into a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Sequence files Location of the sequence files.

This parameter contains the full path to the sequence files to be imported. More than one file can be specified. All sequences are imported in one sequence record data set.

Delimiter separating header info fields Delimiter for parsing the header information.

The delimiter separates the information fields in the header of the sequence. The label number (see below) determines which information field is used to set the *Name* field of the sequence record that is being imported.

As name for the sequence record, use label Number of the information field to be used as sequence record name.

The content of the information field with this number is used to set the *Name* field of the sequence record being imported.

Source label Label assigned to the source field of a sequence record.

While importing sequences, the *Source* field of a sequence record in the data set can be filled in with information about the source of this sequence record. This information is formed by the string expression provided in this parameter. The variable *[Filename]* can be used to refer to the name of the file that is being imported.

Import sequence quality Whether or not to import sequence qualities.

If checked, the raw sequence qualities are imported and converted to Phred qualities using the conversion formula (see below). If unchecked, no sequence qualities are imported.

When importing sequence quality, quality files are linked to sequence files based on the filename. A quality file should have the same filename as its sequence file, except for the extension. Files with extension '.fna' are considered to be sequence files, and files with extension '.qual' quality files.

Conversion formula Formula for converting raw sequence qualities to Phred qualities.

Quality scores in the Power assembler are Phred scores on a scale from 0 to 63 included. For Roche/454[®] reads, the conversion formula is

$$bnQ = rQ.$$

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.5 Import FASTQ files

This operator imports FASTQ sequence files into a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Sequence files Location of the sequence files.

This parameter contains the full path to the sequence files to be imported. More than one file can be specified. All sequences are imported in one sequence record data set.

Delimiter separating header info fields Delimiter for parsing the header information.

The delimiter separates the information fields in the header of the sequence. The label number (see below) determines which information field is used to set the *Name* field of the sequence record that is being imported.

As name for the sequence record, use label Number of the information field to be used as sequence record name.

The content of the information field with this number is used to set the *Name* field of the sequence record being imported.

Source label Label assigned to the source field of a sequence record.

While importing sequences, the *Source* field of a sequence record in the data set can be filled in with information about the source of this sequence record. This information is formed by the string expression provided in this parameter. The variable *[Filename]* can be used to refer to the name of the file that is being imported.

Import sequence quality Whether or not to import sequence qualities.

If checked, the raw sequence qualities are imported and converted to Phred qualities using the conversion formula (see below). If unchecked, no sequence qualities are imported.

Conversion formula Formula for converting raw sequence qualities to Phred qualities.

Quality scores in the Power assembler are Phred scores on a scale from 0 to 63 included. For FASTQ reads, the conversion formula is

$$bnQ = fQ.$$

Import as paired-end reads Whether or not to import sequences as paired-end.

When importing sequences as paired-end reads, files are coupled based on the filename. Two files with the same name except for the last two characters (which should be "_1" and "_2") are supposed to contain each one end of a paired-end read.

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.6 Load from database

Loads a sequence or a set of sequences from a sequence experiment in the database into a sequence record set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Entry list List of entry keys to fetch sequences from.

This operator reads the sequences of the entries with the specified keys. Keys can be added to the list by manually typing the entry key and pushing the **<Add>** button, or by adjusting the current selection in the BioNumerics main window to the desired entries and pushing the **<Add current selection>** button. Entries selected in the list can be removed by pushing the **<Remove selected>** button.

Experiment Name of a sequence experiment type.

For the keys specified, the sequence from the experiment type chosen here will be loaded by the operator.

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.7 Load from sequence read set

Loads a sequence or a set of sequences from one or multiple sequence read sets from the database into a sequence record set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Entry list List of entry keys to fetch sequences from.

This operator reads the sequences of the entries with the specified keys. Keys can be added to the list by manually typing the entry key and pushing the **<Add>** button, or by adjusting the current selection in the BioNumerics main window to the desired entries and pushing the **<Add current selection>** button. Entries selected in the list can be removed by pushing the **<Remove selected>** button.

Experiment Name of a sequence experiment type.

For the keys specified, the sequence from the experiment type chosen here will be loaded by the operator.

Import sequences in hard-disk based data set Whether or not to use a hard-disk based dataset.

18.7.5.1.8 Export sequence to single experiment

Exports a sequence to an experiment in the database for a single sequence record.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Operator parameters

Entry key Key of the entry where the information should be written.

The entry key is the string used as key when creating a new entry in the BioNumerics database for the export of information from the Power assembler. If this parameter is left empty, an automatically generated key will be filled in.

Sequence experiment Name of the sequence experiment type where the information should be written.

The information exported from the Power Assembler will be loaded in the specified sequence experiment. If no sequence experiment is present, first create this experiment in the *Experiment panel* of the BioNumerics main window.

18.7.5.1.9 Export sequences to multiple experiments

Exports a sequence to an experiment in the database for a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Operator parameters

Entry key expression Expression for the key of the entry where the information should be written.

For every sequence record that is treated by this operator, an entry key is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

Experiment name expression Expression for the name of the experiment type where the information should be written.

For every sequence record that is treated by this operator, the name of an experiment type is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

18.7.5.1.10 Export sequences to multiple read set experiments

Exports a set of sequences to a sequence read set experiment in the database for a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Name field This field is designed to contain the name information for the reads. This can be Name, Source or Read type information. This field is of type *String*.

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Entry key expression Expression for the key of the entry where the information should be written.

For every sequence record that is treated by this operator, an entry key is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

Experiment name expression Expression for the name of the experiment type where the information should be written.

For every sequence record that is treated by this operator, the name of an experiment type is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

18.7.5.1.11 Export field to single entry

Exports a field to an information field of an entry for a single sequence record.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Value expression Expression that determines the value to be written.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Entry key Key of the entry where the information should be written.

The entry key is the string used as key when creating a new entry in the BioNumerics database for the export of information from the Power assembler. If this parameter is left empty, an automatically generated key will be filled in.

Entry information field Name of the information field where the information should be written.

18.7.5.1.12 Export field to multiple entries

Exports a field to an information field of an entry for a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Value expression Expression that determines the value to be written.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Entry key expression Expression for the key of the entry where the information should be written.

For every sequence record that is treated by this operator, an entry key is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

Entry information field Name of the information field where the information should be written.

18.7.5.1.13 Export field to single experiment field

Exports a field to a custom information field of an experiment for a single sequence record.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Value expression Expression that determines the value to be written.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Entry key Key of the entry where the information should be written.

The entry key is the string used as key when creating a new entry in the BioNumerics database for the export of information from the Power assembler. If this parameter is left empty, an automatically generated key will be filled in.

Sequence experiment Name of the sequence experiment type where the information should be written.

The information exported from the Power Assembler will be loaded in the specified sequence experiment. If no sequence experiment is present, first create this experiment in the *Experiment panel* of the BioNumerics main window.

Sequence experiment field Name of the experiment information field where the information should be written.

18.7.5.1.14 Export field to multiple experiment fields

Exports a field to an information field of an experiment for a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Value expression Expression that determines the value to be written.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Entry key expression Expression for the key of the entry where the information should be written.

For every sequence record that is treated by this operator, an entry key is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

Experiment name expression Expression for the name of the experiment type where the information should be written.

For every sequence record that is treated by this operator, the name of an experiment type is calculated from this expression. More information on expressions can be found in [18.7.4.4](#).

Sequence experiment field Name of the experiment information field where the information should be written.

18.7.5.1.15 Export sequences to FASTA/FASTQ file

Exports sequences for a set of sequence records to a FASTA or a FASTQ file.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Output directory Location of the exported sequence files.

This parameter contains the full path to the directory where the sequence files will be exported to.

File Name and location of the file to write to.

Select the file from the *Browse for file dialog box*. To create a new file for this export action, just manually type in the name of the export file and press **<Open>**. The new file will automatically be created when running the operator.

Label expression Expression for the label of a sequence in the file.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Quality score range start Offset for the quality scores.

Export paired-end sequences Whether or not to export the sequences as paired-end sequences.

If checked, paired-end sequences will be exported to two files which are supposed to contain each one end of a paired-end read. Both files have the same name except for the last two characters (which will be "_1" and "_2").

18.7.5.1.16 Export sequences to multiple FASTA/FASTQ files

Exports sequences for a set of sequence records to multiple FASTA or FASTQ file.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Output directory Location of the exported sequence files.

This parameter contains the full path to the directory where the sequence files will be exported to.

Filename expression Expression for the file name of a sequence in the file.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Label expression Expression for the label of a sequence in the file.

This expression can be of any type. More information on expressions can be found in [18.7.4.4](#).

Quality score range start Offset for the quality scores.

Export paired-end sequences Whether or not to export the sequences as paired-end sequences.

If checked, paired-end sequences will be exported to two files which are supposed to contain each one end of a paired-end read. Both files have the same name except for the last two characters (which will be "_1" and "_2").

18.7.5.1.17 Export fields to file

Exports fields to a tab-delimited file for a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Output directory Location of the exported sequence files.

This parameter contains the full path to the directory where the sequence files will be exported to.

File Name and location of the file to write to.

Select the file from the *Browse for file dialog box*. To create a new file for this export action, just manually type in the name of the export file and press <**Open**>. The new file will automatically be created when running the operator.

Fields to export The list of fields to export.

Select the fields to be exported. Multiple fields can be selected by holding the Ctrl-key.

18.7.5.1.18 Concatenate sequences

This operator concatenates sequences, sequence qualities, target coverages and target qualities.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Associated sequences field This field is designed to contain a list of keys from sequence records associated to this sequence record. This field is of type *SeqKeyArray*.

Operator parameters

Separator Separator character to put between two sequences.

The default separator is a | (pipe character). When concatenating sequences, keep in mind that only a | is recognized by BioNumerics as a concatenated sequence. A space is interpreted as a 1bp gap between both sequences.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.1.19 Unconcatenate sequences

This operator un-concatenates sequences, sequence qualities, target coverages and target qualities.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Operator parameters

Separator Separator character that has been put between two sequences.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.2 Trimming

The trimming operators select and process the raw sequences and sequence qualities to obtain a set of sequence records that can be used for assembly.

18.7.5.2.1 Exclude low minimum-quality reads

This operator removes all sequences with a low minimum quality.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum quality In order to keep the sequence, the minimum value of the read sequence quality should not be lower than this threshold.

18.7.5.2.2 Exclude low average-quality reads

This operator removes all sequences with a low average quality.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum average quality In order to keep the sequence, the average value of the read sequence quality should not be lower than this threshold.

18.7.5.2.3 Remove low rolling-average-quality tail

This operator removes bases at the end of the sequences as soon as the rolling average quality is too low.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum rolling average quality From the moment the average base quality, calculated from the beginning of a sequence to the position of the base, is below this value, the read sequence is trimmed to this base position.

18.7.5.2.4 Remove low average-quality tail

This operator removes bases at the end of the sequences as soon as the windowed average quality is too low.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum windowed average quality In order to keep the sequence, the windowed average value of the read sequence quality should not be lower than this threshold.

Window size Size of the window to calculate the average in.

18.7.5.2.5 Remove low-quality tails

This operator removes bases from both ends of a sequence until the minimum tail quality is reached.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum tail quality As long as the base quality at the beginning or the end of a sequence is below this value, the base is removed from the read sequence.

18.7.5.2.6 Exclude short reads

This operator removes all sequences that are too short.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum length In order to keep the sequence, the length of the read sequence should not be lower than this threshold.

18.7.5.2.7 Restrict reads to maximum length

This operator restricts reads to a maximum number of bases.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Maximum length The length of the read sequence is limited to this threshold.

18.7.5.2.8 Replace low-quality bases

This operator replaces low-quality bases by N.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum quality If the quality of a base drops below this value, the base is replaced by N.

18.7.5.2.9 Remove bases N at the end

18.7.5.2.10 Exclude reads based on homopolymer length

This operator removes all sequences containing homopolymers that are too long.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Maximum homopolymer length In order to keep the sequence, the homopolymer length of the read sequence should not be higher than this threshold.

18.7.5.2.11 Exclude reads based on %GC

This operator removes all sequences with a %GC that is not in the expected range.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum %GC In order to keep the sequence, the %GC value of the read sequence should not be lower than this threshold.

Maximum %GC In order to keep the sequence, the %GC value of the read sequence should not be higher than this threshold.

18.7.5.2.12 Exclude reads based on %A

This operator removes all sequences with a %A above a threshold.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Maximum %A In order to keep the sequence, the %A value of the read sequence should not be higher than this threshold.

18.7.5.2.13 Remove tails by signature

This operator searches for the first occurrence of a start signature string and the first occurrence of a stop signature string after the occurrence of the start signature, and restricts the sequence and the sequence quality to the part between the locations of the start and the stop signatures (signatures not included). If one of the two signatures could not be found, the sequence is kept as a whole.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Start signature string String determining the start of the sequence part to keep.

Stop signature string String determining the end of the sequence part to keep.

Maximum number of mismatches Number of mismatches allowed when comparing a signature to the sequence.

18.7.5.2.14 Remove fixed-size tails

This operator removes a fixed number of bases from the beginning of a sequence, and a fixed number of bases from the end of the sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Trimmed sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Trimmed sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Offset from the beginning Number of bases to remove at the beginning of the sequence.

Offset from the end Number of bases to remove at the end of the sequence.

18.7.5.3 Preprocessing ---

The preprocessing operators provide tools to prepare read and reference sequences for assembly.

18.7.5.3.1 Process barcodes ---

This operator removes the multiplex identifier barcodes and links sequence records with the same barcode to the same sample record.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Sequence field This field is designed to contain a trimmed sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a trimmed sequence. This field is of type *SequenceQuality*.

Operator parameters

Linker size Size of the linker.

Barcode size Size of the multiplex identifier barcode.

Barcode is located Location of the bar code. This can be at the beginning of the sequence, at the end of the sequence or at both ends of the sequence.

Remove barcode and linker from the sequence Whether or not to remove the bar code and the linker from the sequence.

18.7.5.3.2 Separate barcode from sequence

This operator splits a sequence into the actual sequence part and the bar code part, thus removing the linker.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Barcode field This field is designed to contain the multiplex identifier of the sample record.

Barcode quality field This field is designed to contain the qualities associated to a multiplex identifier sequence.

Operator parameters

Linker size Size of the linker.

Barcode size Size of the multiplex identifier barcode.

Barcode is located Location of the bar code. This can be at the beginning of the sequence, at the end of the sequence or at both ends of the sequence.

Remove barcode and linker from the sequence Whether or not to remove the bar code and the linker from the sequence.

18.7.5.3.3 Split pairs

This operator splits a sequence, consisting of two sequence parts linked by an adaptor sequence, in the two sequence parts and removes the adaptor sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Splitted sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Splitted sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Alignment parameters

Match score Score for two identical bases.

Mismatch score Score for two non-identical bases.

Allow gaps in the adapter Whether or not to allow gaps in a read sequence.

Open gap score read Penalty score for introducing a gap in a read sequence.

Extend gap score read Penalty score for extending an existing gap in a read sequence.

Allow gaps in the read Whether or not to allow gaps in an adapter sequence.

Open gap score read Penalty score for introducing a gap in an adapter sequence.

Extend gap score read Penalty score for extending an existing gap in an adapter sequence.

Scoring parameters

Minimum sequence identity The minimum sequence identity for an alignment to be acceptable.

Maximum penalty The maximum penalty for an alignment to be acceptable.

Penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Custom penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Remove read when no adapter could be found

Minimum sequence size Minimum sequence length for one part of the paired-end sequence to be retained.

Adaptors Adaptor sequence.

The adaptor sequences already present in the software are the 454flx[®] palindromic adaptor

GTTGGAACCGAAAGGGTTTGAATTCAAACCCTTTCGGTTCCAAC

and the 454titanium[®] adaptor

TCGTATAACTTCGTATAATGTATGCTATACGAAGTTATTACG.

If the latter is selected, of course, also the reverse complement is used for analysis.

18.7.5.3.4 Split pairs separated by barcode

This operator splits a raw sequence into a paired-end sequence and the multiplexing barcode, and removes both adaptor and multiplexing barcode from the sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Splitted sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Splitted sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Size of the first read Sequence length of the first part of the paired-end sequence.

Size of the linker between the first read and the barcode Sequence length of the linker between the first read and the barcode.

Barcode size Sequence length of the barcode.

Size of the linker between the barcode and the second read Sequence length of the linker between the barcode and the second read.

18.7.5.3.5 Find signature

This operator searches for a specific signature on a sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Position field This field is designed to contain a list of positions. This field is of type *PositionList*.

Operator parameters

Signature string String to search for.

Maximum number of mismatches Number of mismatches allowed when comparing a signature to the sequence.

18.7.5.3.6 Extract subsequence

This operator cuts a part out of a sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Operator parameters

Start position First position to include in the subsequence.

End position One position past the last position to include in the subsequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.3.7 Wrap-around

This operator wraps a sequence around, simulating a circular sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Operator parameters

Wrap-around size Length of the sequence to add at the beginning and at the end of the original sequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.3.8 Undo wrap-around

This operator restores a sequence that has been wrapped around.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Operator parameters

Wrap-around size Length of the sequence to add at the beginning and at the end of the original sequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.3.9 Replace in sequence

This operator replaces all occurrences of a character in a sequences by another character.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Operator parameters

Character to replace The character that needs to be replaced.

Character to replace by The character used as a replacement.

18.7.5.4 Mapping

The mapping operators position the read sequences with respect to one or more reference sequences, and use this information to create a target sequence or an assembly.

18.7.5.4.1 Map to reference

This operator aligns read sequences against the reference sequence(s).

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Reference sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Reference sequence filter Logical test determining which sequence records are considered reference sequences.

The restriction test is an expression of boolean type that determines whether or not a sequence record is considered a reference sequence record. See [18.7.2.2](#) for detailed information.

Output fields

Aligned sequence field This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Aligned sequence quality field This field is designed to contain the aligned sequence quality with respect to the associated sequence record. This field is of type *SequenceQuality*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated orientation field This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Mapping discrimination field This field is designed to contain the mapping discrimination of a sequence record when positioned with respect to a reference sequence. This field is of type *Float*.

Mapping degeneracy field This field is designed to contain the mapping degeneracy of a sequence record when positioned with respect to a reference sequence. This field is of type *Int*.

Sequence identity field This field is designed to contain the sequence identity score of a sequence record when positioned with respect to a reference sequence. This field is of type *Float*.

Matching position field This field is designed to contain the best position of a sequence record without gapped alignment when positioned with respect to a reference sequence. This field is of type *Int*.

Operator parameters

Alignment parameters

Match score Score for two identical bases.

Mismatch score Score for two non-identical bases.

Allow gaps in the read Whether or not to allow gaps in a read sequence.

Open gap score read Penalty score for introducing a gap in a read sequence.

Extend gap score read Penalty score for extending an existing gap in a read sequence.

Allow gaps in the reference Whether or not to allow gaps in a reference sequence.

Open gap score reference Penalty score for introducing a gap in a reference sequence.

Extend gap score reference Penalty score for extending an existing gap in a reference sequence.

Maximum alignment shift Maximum shift in the alignment of a read sequence with respect to the reference sequence.

Scoring parameters

Minimum sequence identity The minimum sequence identity for an alignment to be acceptable.

Maximum penalty The maximum penalty for an alignment to be acceptable.

Minimum overlap The minimum overlap between the two sequences for an alignment to be acceptable.

Penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Custom penalty function A custom penalty function.

A custom penalty function is an expression composed of a set of variables. The available variables are: length of the alignment, number of matches, number of mismatches, number of open-gap operations on the read sequence, number of extend-gap operations on the read sequence, number of open-gap operations on the reference sequence and number of extend-gap operations on the reference sequence.

Enforce paired-end read constraints Whether or not to use a preset distance between two ends of a paired-end read.

Expected inter-read distance Average distance between two ends of a paired-end read. This parameter is used only when the parameter 'Perform paired-end read alignment' is checked.

Maximum distortion of inter-read distance Maximum deviation from the average distance between two ends of a paired-end read which produces an acceptable alignment of a paired-end read. This parameter is used only when the parameter 'Perform paired-end read alignment' is checked.

Use seeds Whether or not to use seeds. If unchecked, the seed length is set to the full length of the read sequence.

Seed start position Start position of the seed.

Maximum seed size Maximum size of the seed.

The actual size of the seed also depends on the seed start position and the length of the read sequence.

Maximum number of mismatches Maximum number of mismatches in the seed.

Maximum number of gaps in the reference Maximum number of gaps that can be inserted in a reference sequence while aligning the seed.

Maximum number of gaps in the read Maximum number of gaps that can be inserted in a read sequence while aligning the seed.

Maximum search width Maximum deviation from the optimal seed position at any stage in the seed determination process.

Minimum absolute score (seed) Minimum score of a seed to be acceptable.

18.7.5.4.2 Create inserts on target

This operator aligns read sequences against the reference sequence and constructs a target sequence by introducing the required insertions on the reference.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Aligned sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Matching position field This field is designed to contain the best position of a sequence record without gapped alignment when positioned with respect to a reference sequence. This field is of type *Int*.

Associated orientation field This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Reference sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Reference sequence filter Logical test determining which sequence records are considered reference sequences.

The restriction test is an expression of boolean type that determines whether or not a sequence record is considered a reference sequence record. See [18.7.2.2](#) for detailed information.

Output fields

Aligned sequence field This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Aligned sequence quality field This field is designed to contain the aligned sequence quality with respect to the associated sequence record. This field is of type *SequenceQuality*.

Target sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated orientation field This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Operator parameters

Minimum homology The minimum sequence identity for an alignment to be acceptable.

Alignment parameters

Match score Score for two identical bases.

Mismatch score Score for two non-identical bases.

Allow gaps in the read Whether or not to allow gaps in a read sequence.

Open gap score read Penalty score for introducing a gap in a read sequence.

Extend gap score read Penalty score for extending an existing gap in a read sequence.

Allow gaps in the reference Whether or not to allow gaps in a reference sequence.

Open gap score reference Penalty score for introducing a gap in a reference sequence.

Extend gap score reference Penalty score for extending an existing gap in a reference sequence.

Maximum alignment shift Maximum shift in the alignment of a read sequence with respect to the reference sequence.

Scoring parameters

Minimum sequence identity The minimum sequence identity for an alignment to be acceptable.

Maximum penalty The maximum penalty for an alignment to be acceptable.

Minimum overlap The minimum overlap between the two sequences for an alignment to be acceptable.

Penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Custom penalty function A custom penalty function.

A custom penalty function is an expression composed of a set of variables. The available variables are: length of the alignment, number of matches, number of mismatches, number of open-gap operations on the read sequence, number of extend-gap operations on the read sequence, number of open-gap operations on the reference sequence and number of extend-gap operations on the reference sequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.4.3 Calculate target coverage

This operator aligns read sequences against the reference sequence and constructs a target sequence by introducing the required insertions and deletions on the reads.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Aligned sequence field This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Aligned sequence quality field This field is designed to contain the aligned sequence quality with respect to the associated sequence record. This field is of type *SequenceQuality*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated orientation field This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Reference sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Reference sequence filter Logical test determining which sequence records are considered reference sequences.

The restriction test is an expression of boolean type that determines whether or not a sequence record is considered a reference sequence record. See [18.7.2.2](#) for detailed information.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Coverage matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Quality matrix field

Operator parameters

Target quality threshold Threshold for determining the individual base qualities of the target sequence.

Allow extension of the reference sequence (begin and end) Whether or not to create a target sequence that follows the frame of the reference sequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.4.4 Assemble map

This operator creates an assembly for the aligned read sequences against the defined reference sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Aligned sequence field This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated orientation field This field is designed to contain the forward/reverse orientation of the sequence with respect to the associated sequence record. This field is of type *SeqDir*.

Target sequence

Target sequence expression

Reference sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Reference sequence filter Logical test determining which sequence records are considered reference sequences.

The restriction test is an expression of boolean type that determines whether or not a sequence record is considered a reference sequence record. See [18.7.2.2](#) for detailed information.

Operator parameters

18.7.5.4.5 All-in-one map to reference

18.7.5.4.6 Consensus base calling

This operator calculates a consensus base calling.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Operator parameters

Minimum coverage Minimum coverage of a base to be considered for consensus base calling. If the coverage is too low, the base is replaced by a gap.

Gap threshold Minimum frequency of a gap before that position is considered as a gap in the consensus sequence.

Single base threshold Minimum frequency of the most frequent base before this base is considered the unique base at a certain position.

Double base threshold Minimum frequency of the two most frequent bases before these bases are considered the two possible bases at a certain position.

Triple base threshold Minimum frequency of the three most frequent bases before these bases are considered the three possible bases at a certain position.

18.7.5.5 De novo assembly

The de novo assembly operators create a set of contigs from one or more libraries of read sequences, and provide means to assess the results of this assembly.

18.7.5.5.1 De novo assembly (Velvet)

This operator performs a de novo assembly using Velvet.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Read sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Operator parameters

Use this type of input Whether or not to use the first short single-end read library in the data set.

Restriction test Logical test determining which sequence records are short single-end reads of the first library.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a single-end read of the first short read library. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the second short single-end read library in the data set.

Restriction test Logical test determining which sequence records are short single-end reads of the second library.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a single-end read of the second short read library. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the long single-end read library in the data set.

Restriction test Logical test determining which sequence records are long single-end reads.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a long single-end read of the long read library. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the first short paired-end read library in the data set.

Restriction test Logical test determining which sequence records are short paired-end reads of the first library.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a paired-end read of the first short read library. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the second short paired-end read library in the data set.

Restriction test Logical test determining which sequence records are short paired-end reads of the second library.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a paired-end read of the second short read library. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the long paired-end read library in the data set.

Restriction test Logical test determining which sequence records are long paired-end reads.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a paired-end read of the long read library. See [18.7.2.2](#) for detailed information.

Expected coverage Type of expected coverage to use. This parameter can be set to *Don't use*, *Determine automatically* or *Custom value*. The expected coverage is used for repeat resolving.

Custom expected coverage Expected value of the coverage.

Coverage cutoff Type of coverage cutoff to use. This parameter can be set to *Don't use*, *Determine automatically* or *Custom value*. The coverage cutoff is used to exclude low-coverage nodes, and as such has an error correcting effect.

Custom coverage cutoff Coverage cutoff value.

Maximum coverage Type of maximum coverage to use. This parameter can be set to *Don't use* or *Custom value*. The maximum coverage is used to exclude nodes with extremely high coverage.

Custom maximum coverage Maximum coverage value.

Insert length determination This parameters allows to determine the insert size in the first short paired-end read library automatically, or to specify it by setting an average insert size and standard error on the insert size.

Insert size Average insert size in the paired-end read library.

Insert size standard deviation Standard deviation on the insert size in the paired-end read library.

Mate-pair library with read-pair contamination.

Insert length determination This parameters allows to determine the insert size in the second short paired-end read library automatically, or to specify it by setting an average insert size and standard error on the insert size.

Insert size Average insert size in the paired-end read library.

Insert size standard deviation Standard deviation on the insert size in the paired-end read library.

Mate-pair library with read-pair contamination.

Insert length determination This parameters allows to determine the insert size in the long paired-end read library automatically, or to specify it by setting an average insert size and standard error on the insert size.

Insert size Average insert size in the paired-end read library.

Insert size standard deviation Standard deviation on the insert size in the paired-end read library.

Mate-pair library with read-pair contamination.

k-mer length Length of the k-mers used in hashing the reads.

Minimum contig length When a contig is shorter than the length threshold, it is discarded.

Maximum number of gaps used when aligning two branches of a bubble

Perform scaffolding. Whether or not to perform scaffolding, that is, linking non-overlapping contigs based on paired-end read information.

Perform strand-specific assembly. Whether or not to perform an assembly that is strand-aware.

Calculation priority

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.5.2 De novo assembly (Ray)

This operator performs a de novo assembly using Ray.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Read sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Operator parameters

Use this type of input Whether or not to use the single-end reads in the data set.

Restriction test Logical test determining which sequence records are single-end reads.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a single-end read. See [18.7.2.2](#) for detailed information.

Use this type of input Whether or not to use the paired-end reads in the data set.

Restriction test Logical test determining which sequence records are paired-end reads.

The restriction test is an expression of boolean type that determines whether or not a sequence record is a paired-end read. See [18.7.2.2](#) for detailed information.

Insert length determination This parameters allows to determine the insert size in the paired-end read library automatically, or to specify it by setting an average insert size and standard error on the insert size.

Insert size Average insert size in the paired-end read library.

Insert size standard deviation Standard deviation on the insert size in the paired-end read library.

k-mer length Length of the k-mers used in hashing the reads.

Perform scaffolding.

Calculation priority

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.5.3 De novo finishing

This operator performs a de novo assembly finishing.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Operator parameters

Restriction test Logical test determining which sequence records are to be extended.

The restriction test is an expression of boolean type that determines whether or not a sequence record is eligible for extension. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining which sequence records can be used for extension.

The restriction test is an expression of boolean type that determines whether or not a sequence record can be used for extension. See [18.7.2.2](#) for detailed information.

K-mer size Size of the k-mer used for the initial screening of association between sequences.

Overlap size Minimum overlap between two sequences in order to merge them.

Maximum number of mismatches Maximum number of mismatches between two sequences in the minimum overlap window in order to consider these two sequences associated.

Maximum number of gaps per sequence Maximum number of gaps allowed in the alignment of two sequences in the minimum overlap window in order to consider these two sequences associated.

Maximum search width Maximum deviation from the optimal overlap position at any stage in the association determination process.

Alignment parameters

Match score Score for two identical bases.

Mismatch score Penalty score for extending an existing gap in a read sequence.

Allow gaps Whether or not to allow gaps.

Open gap score Penalty score for introducing a gap.

Extend gap score Penalty score for extending an existing gap.

Maximum alignment shift Maximum shift in the alignment of two sequences.

Scoring parameters

Minimum sequence identity The minimum sequence identity for an alignment to be acceptable.

Maximum penalty The maximum penalty for an alignment to be acceptable.

Minimum overlap The minimum overlap between the two sequences for an alignment to be acceptable.

Penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Custom penalty function A custom penalty function.

A custom penalty function is an expression composed of a set of variables. The available variables are: length of the alignment, number of matches, number of mismatches, number of open-gap operations on the read sequence, number of extend-gap operations on the read sequence, number of open-gap operations on the reference sequence and number of extend-gap operations on the reference sequence.

Name expression Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.6 Sequence clustering

The sequence clustering operators create groups of similar sequences.

18.7.5.6.1 Clustering with predefined representatives

Performs a sequence clustering by identifying the representative sequences and assigning the sequences to the nearest representative one.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Operator parameters

Representative sequences Sets the source of the representatives. This can be read from the field *[Barcode]* in the sample records data set, or can be determined from the data at hand. In the latter case, all sequences that are at least as frequent as the frequency threshold set in the **Representative sequence frequency threshold** are considered as representatives.

Representative sequence frequency threshold Minimum frequency of a sequence to be considered a representative.

Compare representative sequences Whether or not to compare the representative sequences, and merge them if necessary. This can be done both before assigning sequences to the representatives, or after.

Alignment parameters

Match score Score for two identical bases.

Mismatch score Score for two nonidentical bases.

Allow gaps in the representative sequence Whether or not to allow gaps in a representative sequence.

Open gap score representative sequence Penalty score for introducing a gap in a representative sequence.

Extend gap score representative sequence Penalty score for extending an existing gap in a representative sequence.

Allow gaps in the sequence to be assigned Whether or not to allow gaps in a sequence that is to be assigned.

Open gap score sequence to be assigned Penalty score for introducing a gap in a sequence that is to be assigned.

Extend gap score sequence to be assigned Penalty score for extending an existing gap in a sequence that is to be assigned.

Maximum alignment shift Maximum shift in the alignment of a sequence with respect to the representative sequence.

Scoring parameters

Minimum sequence identity The minimum sequence identity for an alignment to be acceptable.

Maximum penalty The maximum penalty for an alignment to be acceptable.

Minimum overlap The minimum overlap between the two sequences for an alignment to be acceptable.

Penalty function The type of penalty function.

The standard penalty function is the sum of all open-gap and extend-gap operations weighted with their respective penalty score. Alternatively, a custom penalty function can be set.

Custom penalty function A custom penalty function.

A custom penalty function is an expression composed of a set of variables. The available variables are: length of the alignment, number of matches, number of mismatches, number of open-gap operations on the read sequence, number of extend-gap operations on the read sequence, number of open-gap operations on the reference sequence and number of extend-gap operations on the reference sequence.

Tie handling Determines the strategy when ties occur. When a sequence is equally similar to more than one representative, one can choose to either assign this sequence randomly to one of the representatives, or to leave the sequence unassigned.

Unassigned sequence handling Determines the strategy for unassigned sequences. When a sequence has not been assigned to a representative, one can choose to either leave the sequence as it is, or assign it to the empty representative.

18.7.5.7 Region tools

The region tools operators provide region-specific manipulations of a sequence record data set.

18.7.5.7.1 Find regions

This operator determines regions by start and stop signatures.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Start signature string Start signature string of a region.

Stop signature string Stop signature string of a region.

Maximum number of mismatches Number of mismatches allowed when comparing a signature to the sequence.

Minimum length Minimal sequence length of the region between consecutive start and stop signatures to be retained.

18.7.5.7.2 Find regions on sequence

This operator creates a list of regions that do not contain a specific character.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Signature Character to avoid.

18.7.5.7.3 Find covered regions

This operator creates a list of regions with coverage above a threshold.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Coverage must be at least Minimum coverage.

The minimum coverage threshold for a sequence position to be included in the region list.

Minimum length Minimal sequence length of a sufficiently covered region to be retained.

18.7.5.7.4 Find regions on profile

This operator determines those regions where a sequence curve lies between an upper and a lower bound.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence curve field

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Minimum value Minimum value of the profile.

Maximum value Maximum value of the profile.

Minimum length Minimal length of an admissible region to be retained.

18.7.5.7.5 Set fixed-position regions

This operator determines a region using a fixed start and stop position.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Start position Start position of the region.

Stop position Stop position of the region.

18.7.5.7.6 Detect sequences in region list

This operator detects which sequences have a substantial overlap within a region list.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain the aligned sequence with respect to the associated sequence record. This field is of type *Sequence*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Output fields

Region list membership field This field is designed to keep track of those sequence records that have a substantial overlap with a list of regions. This field is of type *Bool*.

Operator parameters

Minimum overlap size Minimum size of the overlap between a sequence and the regions in the region list.

18.7.5.7.7 Intersect regions

This operator takes the intersection of two sets of regions.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Region list field 1 This field is designed to contain a list of regions. This field is of type *RegionList*.

Region list field 2 This field is designed to contain a list of regions. This field is of type *RegionList*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

18.7.5.7.8 Unite regions

This operator takes the union of two sets of regions.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Region list field 1 This field is designed to contain a list of regions. This field is of type *RegionList*.

Region list field 2 This field is designed to contain a list of regions. This field is of type *RegionList*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

18.7.5.7.9 Invert regions

This operator inverts a set of regions.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Region list field 1 This field is designed to contain a list of regions. This field is of type *RegionList*.

Scale sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

18.7.5.7.10 Glue regions

This operator creates a new list of regions by joining consecutive regions in an existing region list as long as they are close enough to one another.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Output fields

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Operator parameters

Maximum gap size between regions The maximum distance between two regions if they are to be joined.

18.7.5.7.11 Extract regions

This operator extracts regions from a sequence record, and creates a new sequence record for each region.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Region list field This field is designed to contain a list of regions. This field is of type *RegionList*.

Output fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Operator parameters **Name expression** Expression for the name of the sequence records created by this operator.

Since sequence records are empty upon creation, no sequence record variables are available in the name expression. However, there is a variable *[Index]* that indicates the number of the sequence record currently being added. The name expression can be of any type, but is converted to a string. More information on expressions can be found in [18.7.4.4](#).

18.7.5.8 Sequence profiles & curves

The sequence profiles operators create and manipulate profiles over a sequence (%GC profile, sequence quality, coverage, ...). The sequence curve operators create a displayable curve over a sequence. The following built-in types of profiles can be used:

- sequence
- sequence quality
- target coverage counts (both global and per base)
- target quality (per base)
- regions on a curve

In addition to these standard profiles, integer and float profiles can be created.

18.7.5.8.1 Create simple curve

This operator creates a sequence curve based on a profile.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Operator parameters

Only create a curve when the sequence is long enough.

Name of the curve Name for the curve used in the action data panel.

18.7.5.8.2 Create compound curve

This operator creates a sequence curve consisting of multiple profiles associated to one sequence.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Scale sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Associated position field This field is designed to contain the position of the sequence with respect to the associated sequence record. This field is of type *Int*.

Associated sequence field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Operator parameters

Name of the curve Name for the curve used in the action data panel.

18.7.5.8.3 Calculate signature profile

This operator calculates a new profile by presence/absence of a signature.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

Operator parameters

Pattern to search for The signature string to search for.

18.7.5.8.4 Calculate GC profile

This operator calculates a new profile by computing the GC value of every base in the sequence. If a base at a certain position equals G or C, the profile contains a value 1 at that position. If not, the profile value is set to 0.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

18.7.5.8.5 Calculate windowed average profile

This operator calculates a new profile by replacing every value of the input profile by the average value in a window.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

Operator parameters

Half-window size Half-size of the window to calculate the average in.

18.7.5.8.6 Calculate windowed minimum profile

This operator calculates a new profile by replacing every value of the input profile by the minimum value in a window.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record.
This field is of type *FloatArray*.

Operator parameters

Half-window size Half-size of the window to calculate the minimum in.

18.7.5.8.7 Calculate windowed maximum profile

This operator calculates a new profile by replacing every value of the input profile by the maximum value in a window.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record.
This field is of type *FloatArray*.

Operator parameters

Half-window size Half-size of the window to calculate the maximum in.

18.7.5.8.8 Calculate coverage profile

This operator calculates a new profile from an expression based on coverage values.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Coverage matrix field This field is designed to contain the coverage matrix of a target sequence records. This field is of type *CoverageMatrix*.

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

Operator parameters

Value expression Numerical expression determining the value of the profile.

Available variables are:

- ACovFwd: the number of A's counted on forward mapped reads,
- CCovFwd: the number of C's counted on forward mapped reads,
- GCovFwd: the number of G's counted on forward mapped reads,
- TCovFwd: the number of T's counted on forward mapped reads,
- ACovRev: the number of A's counted on reverse mapped reads,
- CCovRev: the number of C's counted on reverse mapped reads,
- GCovRev: the number of G's counted on reverse mapped reads, and
- TCovRev: the number of T's counted on reverse mapped reads.

18.7.5.8.9 Calculate quality profile

This operator calculates a new profile from an expression based on quality values.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Quality matrix field This field is designed to contain the quality matrix of a target sequence records. This field is of type *QualityMatrix*.

Output fields

Profile field This field is designed to contain a value profile over a sequence in this sequence record. This field is of type *FloatArray*.

Operator parameters

Value expression Numerical expression determining the value of the profile.

Available variables are:

- AQual: the quality of the base calling if the call would be A,
- CQual: the quality of the base calling if the call would be C,
- GQual: the quality of the base calling if the call would be G, and
- TQual: the quality of the base calling if the call would be T.

18.7.5.9 Summary graphs

The summary plots operators create histograms that provide a summary of the data set. Histograms can be calculated for every numerical expression evaluated on all sequence records, or from any sequence profile: minimum quality, average quality, maximum quality, coverage, %A, %GC, ...

18.7.5.9.1 Create histogram

This operator creates a sequence curve based on a profile.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Value expression Numerical expression, evaluated for every sequence record, that results in a value and is used as input data for the histogram.

Set the bin size to If checked, a fixed bin size is used.

Use a fixed range If checked, a fixed plot range is used.

18.7.5.9.2 Create profile histogram

This operator creates a histogram of a profile.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Profile field A field of one of the types mentioned in [18.7.5.8](#).

Operator parameters

Set the bin size to If checked, a fixed bin size is used.

Use a fixed range If checked, a fixed plot range is used.

18.7.5.10 Statistics

The statistics operator calculates textual statistics on the data set.

18.7.5.10.1 Global statistics

This operator calculates global statistics for a set of sequence records: number of sequences present, minimum, maximum and average sequence length, base frequencies, number of unknown bases, ...

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

Sequence quality field This field is designed to contain the qualities associated to a raw sequence. This field is of type *SequenceQuality*.

18.7.5.10.2 Contig statistics

This operator calculates contig statistics for a set of sequence records: number of sequences present, minimum, maximum and average sequence length, N50, ...

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Sequence field This field is designed to contain a raw sequence. This field is of type *Sequence*.

18.7.5.11 Data set tools

The data set tools operators provide generic manipulations of a sequence record data set.

18.7.5.11.1 Add data set field

This operator adds a field to a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Field name Name of the field to add.

Field type Type of the field to add.

18.7.5.11.2 Remove data set field

This operator removes a field from a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Field to remove Name of the field to remove.

18.7.5.11.3 Set data set field

This operator sets the content of a field in a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Output fields

Output field Field in the sequence record data set.

This field can be an existing field, or one of the default fields.

Operator parameters

Expression Expression determining, for each sequence record individually, the value of a field.

18.7.5.11.4 Set data set field using associated record

This operator sets the content of a data set field using a field from an associated record.

General parameters

Input fields

Associated key field This field is designed to contain the key of the sequence record this sequence record was associated with. This field is of type *SeqKey*.

Source field Existing field in the sequence record data set.

Output fields

Destination field Field in the sequence record data set.
This field can be an existing field, or one of the default fields.

Operator parameters

Expression Expression determining, for each sequence record individually, the value of a field.
The value of the field from the associated sequence record is contained in the variable *[Result]*.

18.7.5.11.5 Transfer field (majority rule)

This operator copies the content of a field from a first set of sequence records to the same field for another set of sequence records. The content of the field is determined by majority consensus from the first set of sequence records.

General parameters

Operator name Name of the operator.
The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.
The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Field to summarize Existing field in the sequence record dataset.

Output fields

Field to write summary to Field in the sequence record dataset.
This field can be an existing field, or one of the default fields.

Operator parameters

Source sequences test Logical test for determining the source sequence records.

Destination sequences test Logical test for determining the destination sequence records.

Result expression Expression determining, for all destination sequence records at once, the value of a field. The value of the majority vote is put in the variable *[Result]*.

18.7.5.11.6 Transfer field (strict rule)

This operator copies the content of a field from a first set of sequence records to the same field for another set of sequence records. The content of the field is determined by strict consensus from the first set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Input fields

Field to summarize Existing field in the sequence record dataset.

Output fields

Field to write summary to Field in the sequence record dataset.

This field can be an existing field, or one of the default fields.

Operator parameters

Source sequences test Logical test for determining the source sequence records.

Destination sequences test Logical test for determining the destination sequence records.

Result expression Expression determining, for all destination sequence records at once, the value of a field. The value of the strict vote is put in the variable *[Result]*.

18.7.5.11.7 Clone data set

This operator clones the sequence records from a sequence record data set and creates a second data set ('Copy' data set) from them.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

18.7.5.11.8 Merge data sets

This operator merges two sequence record data sets.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Use second data set as main data set Whether or not to merge the first dataset into the second, or vice versa.

18.7.5.11.9 Split data set

This operator splits a sequence record data set into two data sets according to a logical expression.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Split test Logical test for dividing the sequence records into two groups.

18.7.5.11.10 Remove sequence records

This operator removes sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

18.7.5.11.11 Filter randomly

This operator reduces the size of a sequence record data set to a desired fraction by randomly selecting sequence records for removal.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Retain probability Tentative percentage of records to retain in the sequence record data set.

18.7.5.11.12 Perform acceptance/rejection test

This operator checks which sequence records have a numeric value within a range.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Field for boundary test A numerical field of *Integer* or *Float* type.

Output fields

Accept/reject field This field is designed to discern between read sequence records whose alignment with respect to a reference sequence has been accepted. This field is of type *Bool*.

Operator parameters

Minimum value Minimum for the value of the numerical field.

Maximum value Maximum for the value of the numerical field.

18.7.5.11.13 User interaction logging

This operator keeps track of the manual removal of reads from an assembly.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

User actions List of actions performed by the user.

18.7.5.11.14 Set data set property

This operator sets the name and the description of a sequence record data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Data set name Name for the sequence record data set.

Data set description Description for the sequence record data set.

18.7.5.12 Sample tools

The sample set tools operators provide generic manipulations of a sample record data set.

18.7.5.12.1 Add sample field

This operator adds a field to the samples data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Field name Name of the field to add.

Field type Type of the field to add.

18.7.5.12.2 Remove sample field

This operator removes a field from the samples data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Field to remove Name of the field to remove.

18.7.5.12.3 Set sample field

This operator sets the content of a field in the samples data set.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Field Name of the field to set.

Value Value to put into the field.

18.7.5.12.4 Set sample field (advanced)

This operator sets the content of a field in the samples data set according the defined expression.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Field Name of the field to set.

Expression Expression determining, for each sample individually, the value of a field.

18.7.5.12.5 Parse sample field

This operator parses a string in a field from the samples data set and writes the value to another field.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Input fields

Input field This field contains the original source of the sequence record. This field is of type *String*.

Output fields

Output field This field contains the BioNumerics entry key corresponding to the sample record. This field is of type *String*.

Operator parameters

Pattern Pattern to search for.

Token IDs Comma-separated list of token IDs.

Token to save Part of the pattern (ID) to use as key.

18.7.5.12.6 Import sample data

This operator imports sample data from a tab-delimited file.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

File Location of the file to import.

Field Name of the link field.

Column index Column index of the link field.

Field Name of the entry key field.

Column index Column index of the entry key field.

Create new samples if necessary. Whether or not to create new samples when the external file contains lines that cannot be linked to an existing sample.

Field Name of the experiment name field.

Column index Column index of the experiment name field.

18.7.5.12.7 Export sample data

This operator exports all sample data to a tab-delimited file.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

File Location of the file to export to.

18.7.5.12.8 Export sample field

This operator exports the contents of a sample field to an entry information field.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Sample field Name of the sample field.

Database field Name of the BioNumerics database field.

18.7.5.12.9 Remove sample record

This operator removes sample records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

18.7.5.12.10 Add sample links

This operator adds a link between a sequence record and a sample.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

Operator parameters

If available, use existing samples

18.7.5.12.11 Merge samples

This operator merges the samples associated to a set of sequence records.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Group by When set, the operator determines groups based on the field specified, and runs once for each group separately. See [18.7.2.2](#) for detailed information.

18.7.5.12.12 Remove unlinked samples

This operator removes all samples that are not linked to any sequence record.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

18.7.5.13 Project tools

The project tools operators provide generic manipulations of a project.

18.7.5.13.1 Set project property

This operator sets a global property of the power assembly project.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Property name Name of the property to add.

Property type Type of the property to add.

Property value Value of the property to add.

Don't change the property if it already exists.

18.7.5.13.2 Set project property by expression

This operator sets a global property of the power assembly project by evaluating an expression.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Restriction test Logical test determining to which sequence records the operator applies.

The restriction test is an expression of boolean type that determines whether or not a sequence record is used by the operator. See [18.7.2.2](#) for detailed information.

Operator parameters

Property name Name of the property to add.

Take the Defines that part of the expression that is taken as the value of the property.

from the expression Expression which contains the value of the project property.

18.7.5.13.3 Change project property

This operator changes a global property of the power assembly project.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Property to change Name of the property to change.

Property value Value of the property to change.

18.7.5.13.4 Remove project property

This operator removes a global property of the power assembly project.

General parameters

Operator name Name of the operator.

The name of the operator is used to refer to this operator in subsequent expressions. If no name is given, the operator cannot be referred to. See [18.7.2.2](#) for detailed information.

Operator parameters

Property to remove Name of the property to remove.

Part 19

Metagenomics analysis

NOTES

Parts of the documentation on the Metagenomics analysis is based on the mothur wiki (http://www.mothur.org/wiki/Main_Page) and is available under the GNU Free Documentation License 1.2.

Copyright ©2018, Applied Maths NV.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The copy of the "GNU Free Documentation License" is available online: <http://www.gnu.org/copyleft/fdl.html>.

Chapter 19.1

An introduction to metagenomics

BioNumerics is a single solution for metagenomics analyses typically performed by microbial ecologists. Although existing metagenomics analysis workflows are already available elsewhere as a combination of separate programs, scripts and procedures, the availability of one single program that merges all this functionality, the presence of an intuitive graphical user interface and the integration of the sequence read data in the existing BioNumerics platform used by the microbiologist is certainly of added value. BioNumerics provides one environment to start from the raw read sequences; to proceed by trimming, removing chimeras and clustering of the sequences to the final visualization of the operational taxonomic unit (further called OTU) abundances or to the evaluation of the alpha diversity using a plethora of indices. The BioNumerics program makes use of the *mothur* [35] project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). The *mothur* project filled the needs of the microbial ecology community by incorporating the functionality of numerous applications e.g. *dotur*, *treeclimber*, *s-libshuff*, *unifrac* . . . BioNumerics uses the flexibility of the algorithms incorporated in *mothur* and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results. An important advantage is that the metagenomics functionality is developed within the logic of the existing BioNumerics functionality, allowing the integration of metagenomics analysis with existing methods already analyzed in this environment. In this way scientists are able to combine the formerly commonly used environmental analyses such as DGGE, TGGE, or ARISA with the newly obtained metagenomics results and to compare these different methods. Additionally, the existing methodology present in BioNumerics can be used for the analysis of metagenomics data sets e.g. one can perform diversity analysis for one metagenomics entry and use the *Comparison* window with all its elaborated functionality to perform a more in depth comparative analysis of different metagenomics samples.

19.1.1 Alignment of metagenomics sequences using a reference alignment

The alignment algorithm implemented in the metagenomics analysis in BioNumerics requires the presence of a template alignment, also called reference alignment, which is used to align the sample sequences to.

In brief, the general approach to align the sample sequences is first to find the closest template for each read using *kmer* search, *blastn*, or *suffix tree* search; then, to make a pairwise alignment between the read and the degapped template sequences using the *Needleman-Wunsch* or *Gotoh* algorithms; and third, to re-insert gaps to the read and template pairwise alignments using the *NAST* algorithm so that the read sequence alignment is compatible with the original template alignment.

The reference alignment file is a *fasta*-formatted template alignment that should be imported into the BioNumerics database by the user. Once present in the database, it is available for all the metagenomics projects in that specific database. A reference alignment file is expected to have the extension ".align".

Different reference alignments for 16S and 18S rRNA gene sequences are available online and can be downloaded from the Greengenes, RDP or SILVA repositories. However, also custom alignments for any other DNA sequences can be used as a reference template. Example reference alignments, i.e. the Greengenes (<http://greengenes.lbl.gov>) and the SILVA (<http://arb-silva.de>) reference alignment can be downloaded from http://www.mothur.org/wiki/Alignment_database.

A reference alignment can be added to the BioNumerics database by selecting **Database > Sequence databases > Reference alignments...** from the *Main* window. This opens the *Manage reference alignments* dialog box (see Figure 19.1.1).

The *Manage reference alignments* dialog box is used to manage the existing reference alignments in the database and to create new ones.

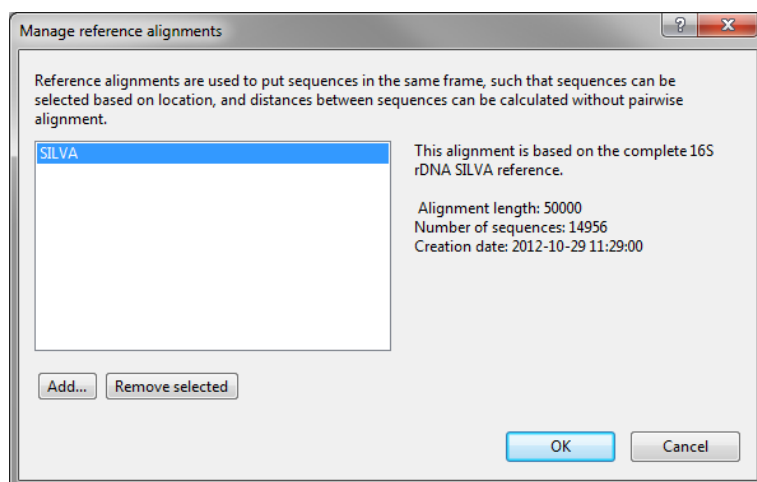


Figure 19.1.1: The *Manage reference alignments* dialog box.

19.1.1.1 Add reference alignment

To add a new reference alignment to the BioNumerics database, select **<Add...>** from the *Manage reference alignments* dialog box. This opens the *Add reference alignment* dialog box (see Figure 19.1.2).

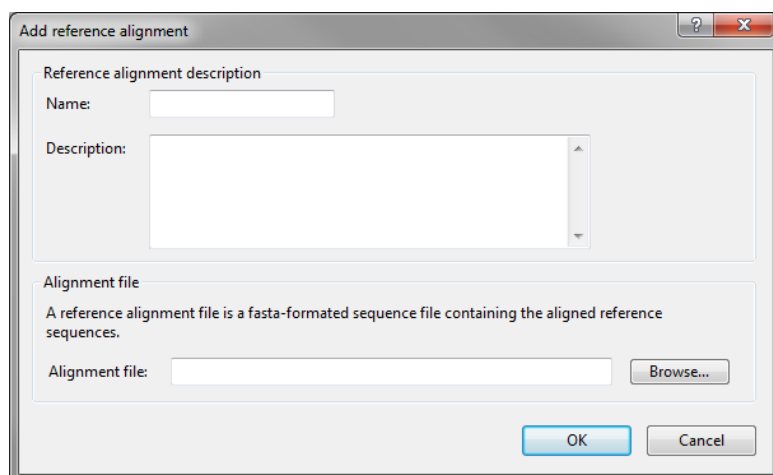


Figure 19.1.2: The *Add reference alignment* dialog box.

In the *Add reference alignment* dialog box, the **Name** and **Description** for the reference alignment can be defined. The fasta-formatted file containing the reference alignment can be selected from the dialog that

appears when hitting the **<Browse...>** button.

Once the correct reference alignment file is loaded in the dialog, click **<OK>** to save the reference alignment to the database. Once created, one can select the reference alignment from the list and have a look at the updated description information that is displayed at the right of the dialog (see Figure 19.1.1). Next to the description, some additional basic information is displayed e.g.: the length of the alignment, the number of sequences present in the alignment and the date the reference alignment was created.

19.1.1.2 Edit reference alignment

The name and description of the imported reference alignments can be edited at any time. Thereto, select a reference alignment and press **<Edit...>**. This will open the *Edit reference alignment* dialog box.

The *Edit reference alignment* dialog box allows to change the name and description of the imported reference alignments(see Figure 19.1.3). Press **<OK>** to save the changes to the database.

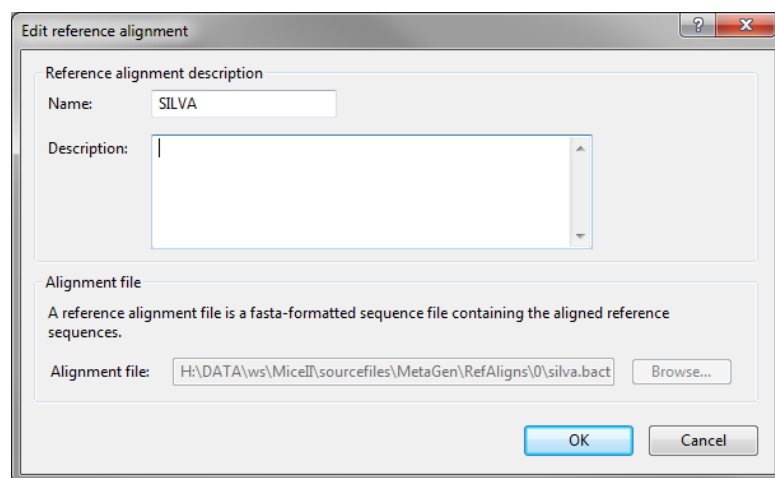


Figure 19.1.3: The *Edit reference alignment* dialog box.

Note that the alignment file cannot be altered from this dialog. Deleting and adding reference alignment source files should be done through the *Manage reference alignments* dialog box.

19.1.1.3 Remove reference alignment

From the *Manage reference alignments* dialog box, a selected reference alignment can be deleted from the database by selecting **<Remove selected>**. This action will remove the reference alignment without warning and cannot be undone.

19.1.2 Identification of metagenomics sequences using a taxonomy database

The methods implemented in the software to obtain classification results on the read sequences rely on the presence of a reference taxonomy. The classification methods currently implemented include the k-nearest neighbor method and the Bayesian approach, and will assign the read sequences onto the taxonomy outline as defined in the metagenomics project. The taxonomy outline contains the taxonomic information of a set of reference sequences. As such, each taxonomy database consists of two files: the reference sequence file and the taxonomy file for these reference sequences. The reference sequence file is a fasta-formatted

(non-)aligned sequence file, and the taxonomy file is a two column text file where the first column is the name of the sequence and the second column is a string of taxonomic information separated by semicolons. This information should not include spaces and the last character must be a semi-colon. For example, one line of the taxonomy file could be as follows:

```
AY553109.1 Bacteria;Firmicutes;Bacillales;Mollicutes;Bacillus_subtilis_et_rel.;Bacillus_carboniphilus_et.
```

One or multiple taxonomy databases need to be imported to the database before being able to start a metagenomics project. The reference sequence and the taxonomy file are expected to have the extension ".fasta" and ".tax", respectively. Once imported, these taxonomy outlines can be used for any metagenomics project created in the database.

Different taxonomies based on 16S rRNA gene sequences are available online and can be downloaded. However, also custom taxonomies for any other DNA sequences can be used as a reference. Example reference taxonomies, i.e. the Greengenes (<http://greengenes.lbl.gov>), the RDP (<http://rdp.cme.msu.edu>) and the SILVA (<http://arb-silva.de>) reference taxonomy can be downloaded from http://www.mothur.org/wiki/Taxonomy_outline.

A taxonomy outline can be added to the BioNumerics database by selecting **Database > Sequence databases > Taxonomic databases...** from the *Main* window. This opens the *Manage taxonomic databases* dialog box (see Figure 19.1.4).

The *Manage taxonomic databases* dialog box is used to upload the new and to manage the existing taxonomic databases.

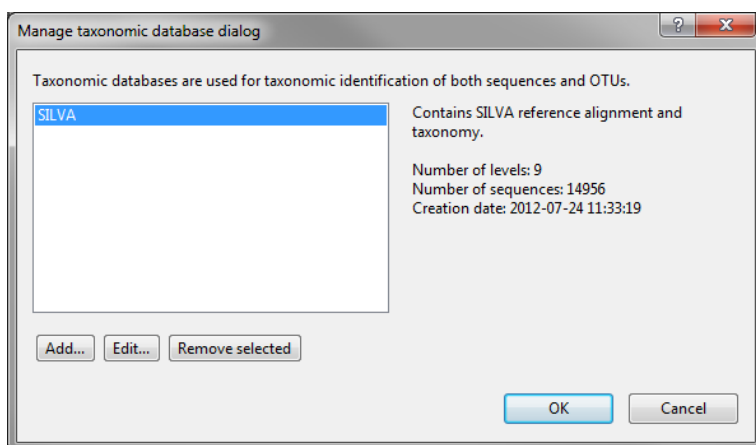


Figure 19.1.4: The *Manage taxonomic databases* dialog box.

19.1.2.1 Add taxonomic database

To add a new reference taxonomy to the BioNumerics database, select **<Add...>** from the *Manage taxonomic databases* dialog box. This opens the *Add taxonomic database* dialog box.

In the *Add taxonomic database* dialog box, basic information on the new taxonomic database such as the **Name** and **Description** can be defined. Further, the two source files need to be loaded:

- First, the taxonomy file needs to be uploaded after selecting the **<Browse...>** button. This taxonomy file is a two column text file where the first column is the name of the sequence, as it appears in the corresponding reference sequence file, and the second column is a string of taxonomic information separated by semicolons. See 19.1.2 for more information on this file format.
- Second, the reference sequence file can be selected from the dialog that appears after hitting the

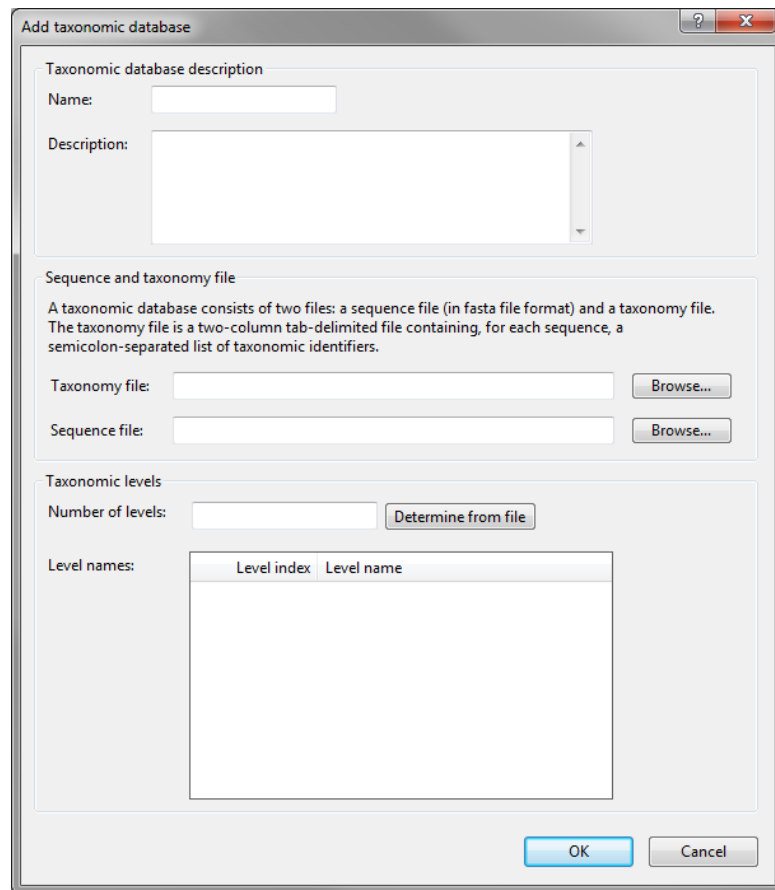


Figure 19.1.5: The *Add taxonomic database* dialog box.

<**Browse...**> button. As indicated earlier, this reference file is a fasta-formatted (non-)aligned sequence file.

After selection of the correct input files, the different number of taxonomic levels linked to this reference taxonomy can be defined. There are two ways to proceed:

- The first option is to manually enter any number between 0 and the maximum number of taxonomic levels present in the uploaded taxonomy file. When editing the number of levels, the overview of the level names is automatically updated.
- The second option is to let BioNumerics search for the number of levels present in the uploaded taxonomy file. Thereto, select **<Determine from file>**. This will automatically update the number of taxonomic levels, and provide the detailed level names in the table beneath. Default level names are assigned to the different level indices. One can manually correct the existing level names, or add a level name where not available by default.

When the correct files and taxonomic information are entered in the dialog, the taxonomic database is saved to the BioNumerics database by selecting **<OK>**.

Once created in the database, the taxonomy can be selected from the list of taxonomic databases, and detailed information is displayed at the right of the dialog (see Figure 19.1.4). Next to the description, some additional basic information is displayed e.g. the number of taxonomic levels defined for this reference taxonomy, the number of sequences present in the taxonomy and the date the taxonomy was created.

19.1.2.2 Edit taxonomic database

The information for a taxonomic database can be edited at any time. Thereto, select a taxonomy and press <**Edit...**>. This will open the *Edit taxonomic database* dialog box.

The *Edit taxonomic database* dialog box allows to change the database name and description information and the level names for each of the level indices of the imported taxonomy (see Figure 19.1.5). Press <**OK**> to save the changes on the reference taxonomy to the database.

Note that the files that define the reference taxonomy cannot be altered afterwards. The only possibility to change any of the input files is to add a new taxonomic database from these files and, if necessary, delete a previously imported database. Next to the reference files, the number of taxonomic levels is also fixed upon import. Although all available taxonomic levels are imported from the taxonomic reference (e.g. domain on to species level), one can always decide to use only a subset of these taxonomic levels in the actual metagenomics analysis (e.g. to use only class to family levels). For each analysis, the taxonomic levels to be used can be changed separately from within the projects.

19.1.2.3 Remove taxonomic database

From the *Manage taxonomic databases* dialog box, a selected taxonomic database can be deleted from the database by selecting <**Remove selected**>. This action will remove the reference taxonomy without warning and cannot be undone.

Chapter 19.2

Creating a new metagenomics project

Metagenomics projects can be initiated from the following windows: the *Main* window, the *Sequence read set experiment* window, as well as from the *Metagenomics* window.

These three different ways to start a metagenomics project all have in common that a sequence read set must be present before the analysis can be started. This implies that the high-throughput sequencing data already needs to be imported via the import functionality BioNumerics. See [9.1.4](#) for more information on the import of sequence read sets.



Before the metagenomics analyses can be started, a reference alignment and reference taxonomy might be needed. Select **File > Reference alignments...** and **File > Taxonomic databases...** to upload these to the database. More information on the reference alignment and reference taxonomy can be found under [19.1.1](#) and [19.1.2](#), respectively.

19.2.1 From the main window

To create identical metagenomic projects for multiple entries at once, first make the entry selection in the *Main* window. Next, select the required analysis by selecting **Analysis > Sequence read set types > Identify against taxonomic database** or **Analysis > Sequence read set types > Single-sample diversity analysis**. For the selected entries, a metagenomics project is now created, and the dedicated analysis dialog *Identify against taxonomic database* wizard or *Single sample diversity analysis* wizard appears. After completion of the information in the analysis dialog, calculation of the different metagenomics projects is automatically started. If only one entry was selected to perform the analysis on, the software will ask whether or not to run the analysis in the *Metagenomics* window or to run the analysis in the background, without opening the window.

19.2.2 From the Sequence read set experiment window

Upon import of a sequence read set in the database, some basic read and quality statistics are calculated and displayed in the *Sequence read set experiment* window. A typical workflow will include the import of the data and subsequently, quality checking and specific trimming of the data where needed. After inspection of the resulting data set (or the original data set if no trimming was performed), it is possible to directly create a metagenomics analysis from this window. Thereto, select **Analysis > Single-sample diversity analysis** or **Analysis > Identify against taxonomic database**. This will open the *Single sample diversity analysis* wizard or *Identify against taxonomic database* wizard, respectively. After completion of the information in the dialog, the analysis is automatically created and added to the *Analysis* panel of the *Sequence read set experiment* window. Upon calculation of only one entry, the software asks whether or not the analysis should be displayed in the *Metagenomics* window or should be executed in background.

19.2.3 From the Metagenomics window

A third option to create a metagenomics project is to start from a new metagenomics analysis. To create a metagenomics analysis, highlight the *Metagenomics projects* tab and select **Edit > Create new object...** (➕) from the *Main* window.

Enter a Project name in the *Create new metagenomics project* dialog box and confirm by <OK>. This will open an empty *Metagenomics* window, together with the *Create metagenomics project* dialog box.

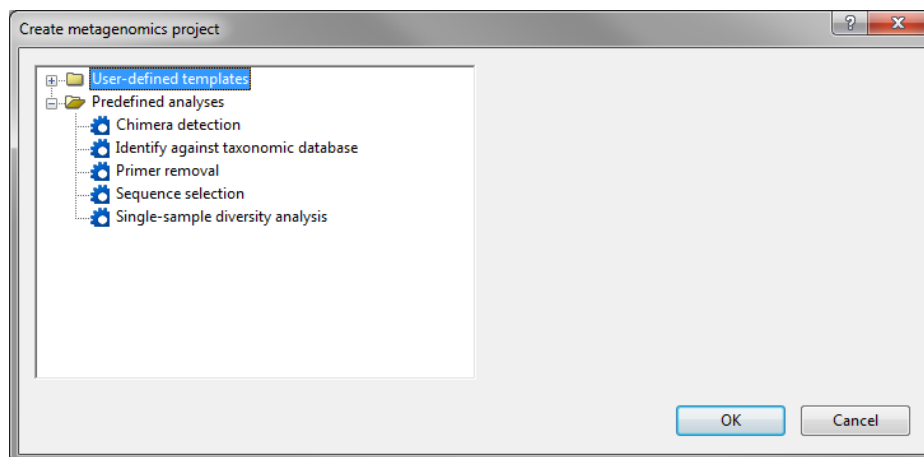


Figure 19.2.1: The *Create metagenomics project* dialog box.

In the *Create metagenomics project* dialog box, a template can be selected to create the new metagenomics project. Two different types of analyses can be selected. The first option is to select a user-defined project template that was previously saved to the database. Secondly, one of the predefined analyses templates can be selected. Note that both preprocessing (Chimera detection, Primer removal and Sequence selection) and processing templates (Identification against a taxonomic database and Single-sample diversity analysis) are present in this list. To load the selected project as a new analysis, select the template from the list and press <OK> to confirm. The selected project pipeline is then updated in the *Project* panel of the *Metagenomics* window.

Depending on the chosen metagenomics analysis template, the *Single sample diversity analysis* wizard or the *Identify against taxonomic database* wizard will open. After completion of the information in the dialog, calculation of the analysis is automatically started.

Chapter 19.3

Preprocessing analyses for metagenomics data

19.3.1 Chimera detection

A common source of 16S sequence artifacts is the formation of chimeric sequences during PCR amplification of the 16S genes. Chimeras are formed when an aborted extension product from an earlier cycle of PCR can function as a primer in a subsequent PCR cycle. Studies have indicated that $\pm 5\%$ of the sequences within curated collections are anomalous or suspect, with chimeras accounting for the majority of problematic sequences [8]. Experimental measurements of chimera formation during PCR co-amplification of 16S rRNA sequences from cloned 16S genes or from mixed bacterial genomic DNA have indicated chimera formation rates of over 30%. Multiple factors including pairwise sequence identity between 16S rRNA genes, number of PCR cycles, and relative abundance of gene-specific PCR templates have been shown to influence chimera formation [39] [38] [1].

Although chimera formation rates can be lowered experimentally, no method has been shown to eliminate these artifacts entirely. Hence, the ability to recognize chimeric sequences is critical in using 16S sequences to profile microbial communities. Several computational methods have been used to identify chimeric sequences. We integrated the chimera-detection algorithm, Chimera Slayer [17] from the mothur tool [35], which can be applied to large datasets, performs well on short sequences, and is sensitive to chimeras between closely related 16S genes.

Chimera detection on a sequence read set can be initiated in different ways:

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis** > **Sequence read set types** > **Chimera detection**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing** > **Chimera detection**; or
- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File** > **New project...** : **Chimera detection** in the *Create metagenomics project* dialog box.

When launching a chimera detection analysis, the *Chimera detection* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <Next> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

The chimera detection reads the sequence read set to be preprocessed and a reference alignment from the

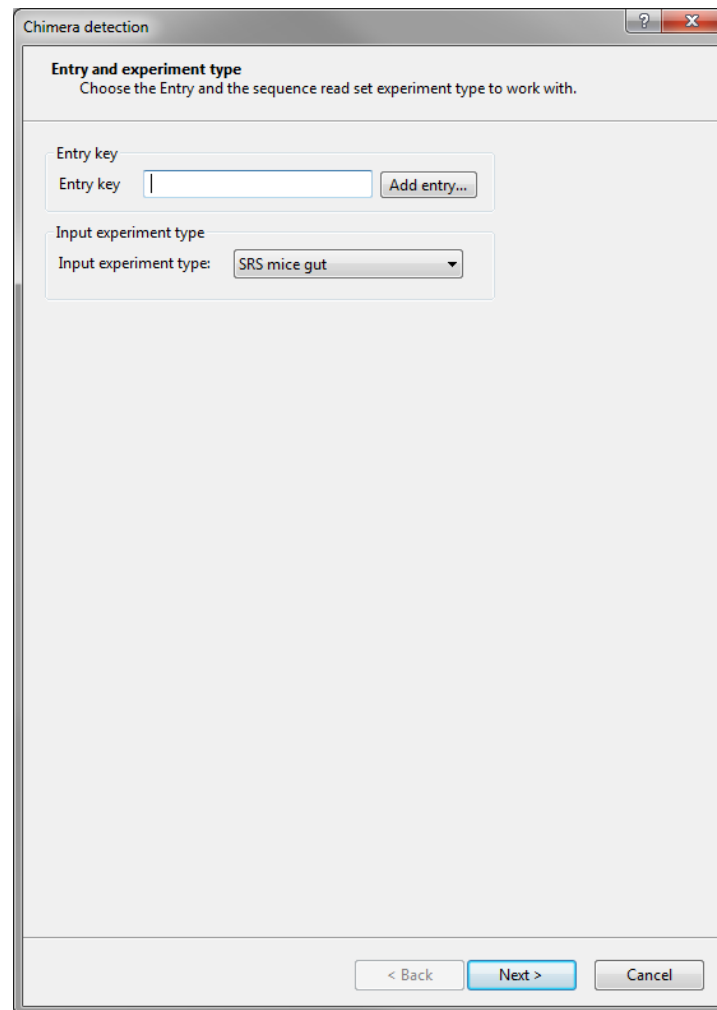


Figure 19.3.1: The *Chimera detection* dialog box: Entry and experiment type settings.

database e.g. the Silva-based reference alignment, and outputs potentially chimeric sequences. Finally, the non-chimeric sequences are saved to the output sequence read set experiment in the database.

In the *Chimera screening and comparison settings*, the following parameters can be defined:

- For the *Chimera detection procedure* one has the option to do a self-comparison, or to compare the read sequences to a standard.
 - The *Self-comparison* does not need a reference alignment but will use the more abundant reads from the sample to check all the reads in the sequence read set.
 - The *Comparison to a standard* will compare the reads from the sequence read set to the sequences from the reference set to decide whether or not the read is identified as a chimera. When this option is used, select the *Reference alignment* to be used from the *Reference alignment* drop down box at the bottom of the dialog. When the reference alignment is defined, the alignment settings can be defined from the *Alignment details* dialog after selecting **<Alignment details ... >**. In the *Alignment details* dialog box, the dedicated settings to align the reads from the sequence read set to the reference alignment can be defined. These settings include the search method to find the closest template for each read, the method to create a pairwise alignment between the read and the de-gapped template sequences, and the settings for the alignment assessment.
 - * There are three methods implemented to find the template sequence for each of the reads:

The screenshot shows a software dialog box titled "Chimera detection". Inside, there is a section "Chimera screening and comparison" with the instruction "Specify the screening and comparison settings for chimera detection." The settings are organized into several groups:

- Chimera detection procedure:** A dropdown menu for "Detection type" is set to "Use self-comparison".
- Screening window:** Two input fields: "Window size" is 50 bases, and "Step size" is 5 bases.
- Sequence comparison:** A group of six input fields: "Number of potential parents (initial screening):" is 15, "Number of potential parents (in-depth screening):" is 3, "Match score:" is 5, "Mismatch score:" is -4, "Minimum similarity between query and parent fragments:" is 90 %, and "Minimum overlap between query and parent fragments:" is 70 %.
- Final sequence selection:** An input field for "Minimum length for the larger part of the sequence to be retained:" is 200 bases.
- Reference alignment:** A dropdown menu for "Reference alignment:" is set to "SILVA", and there is a button labeled "Alignment details...".

At the bottom of the dialog are three buttons: "< Back", "Next >", and "Cancel".

Figure 19.3.2: The *Chimera detection* dialog box: Chimera screening and comparison settings.

- *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- * After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
- *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
 - *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh

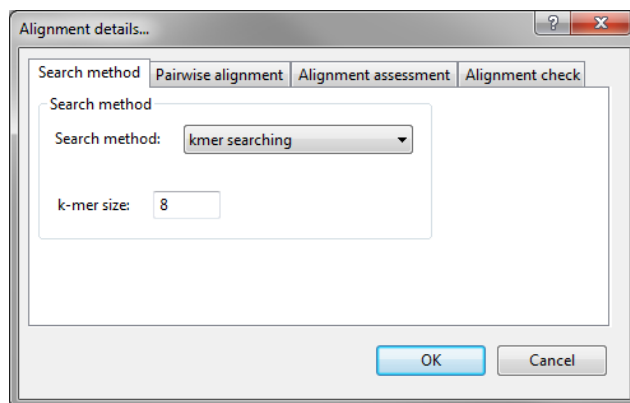


Figure 19.3.3: The *Alignment details* dialog box: Search method settings.

algorithm increases the calculation time but does not result in an improved pairwise alignment.

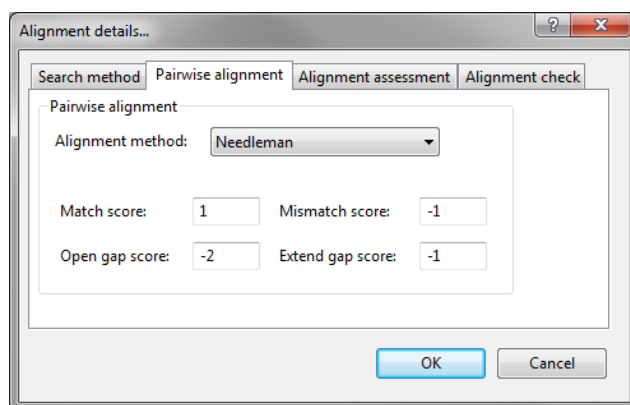


Figure 19.3.4: The *Alignment details* dialog box: Pairwise alignment settings.

- * Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

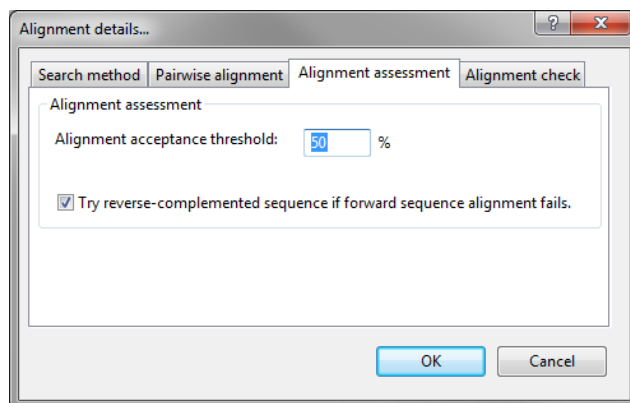


Figure 19.3.5: The *Alignment details* dialog box: Alignment assessment settings.

Press <OK> to save the modified settings to the project. Press <Cancel> to return to the *Chimera detection* dialog box without altering the alignment settings.


- In the settings for the *Screening window*, the *Window size*, i.e. the window size for searching for chimeras, and the *Step size*, i.e. how many base pairs a window is moved each time while screening for chimeric sequences, can be defined.
- The parameters under *Sequence comparison* are used while screening the read sequences for a number of potential parents. The different parameters include:
 - *Number of potential parents (initial screening)*: how many potential parent sequences each read sequence should be compared with.
 - *Number of potential parents (in-depth screening)*: how many potential parent sequences to investigate from the best rated matches.
 - *Match score* and *Mismatch score*: score and penalty values to reward a matched base and penalize a mismatch base, respectively, while screening potential parent sequences.
 - *Minimum similarity between query and parent fragments (%)*: sequence similarity threshold between the read and the parent fragments.
 - *Minimum overlap between query and parent fragments (%)*: coverage threshold of the closest matches found in the read and the parent fragments.
- The only parameter available under *Final sequence selection* is the *Minimum fragment length for a sequence to be retained*. Chimeric sequences are trimmed to include only the longest piece. To be retained in the data set, this piece should be longer than the minimum fragment length.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press **<Finish>** to start the actual preprocessing.

19.3.1.1 Chimera detection : Project element settings

The chimera detection consists of three different project elements:

- *Input sequences*: imports the selected sequence read set from the database and calculates a basic sequence summary.
- *Chimera detection*: performs the chimera trimming on the imported reads.
- *Save to database*: saves the modified sequence read set to the database.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File > Element settings...** .

19.3.1.1.1 Input sequences

In the *Input sequences* dialog box, only settings for the project element *Load sequence read sets from the database* are available.

These settings include:

- *Entry key* information, indicating for which entries sequence read sets should be imported. The buttons **<Add...>**, **<Add current selection>** and **<Remove selected>** can be used to select entries from the database and add them to the list, to add the currently selected database entries to the list, and to remove the currently selected entries from the dialog, respectively.

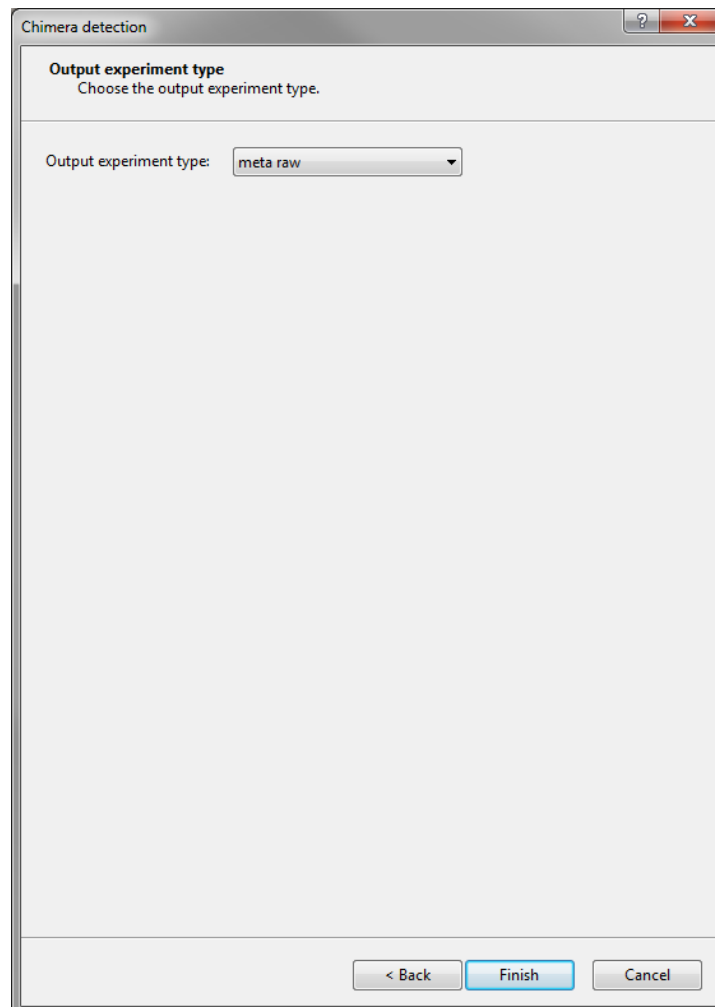


Figure 19.3.6: The *Chimera detection* dialog box: Output experiment type settings.

- *Experiment type* information, indicating from which sequence read set experiment type the data should be loaded.

Press **<OK>** to close the dialog and update the settings in the project. Press **<Cancel>** to close the dialog without altering any of the project settings.

19.3.1.1.2 Chimera detection

The first step in the chimera analysis is the alignment of the reads to the uploaded reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,
2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps into the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *Chimera detection* dialog box, the **Alignment settings** can be modified.

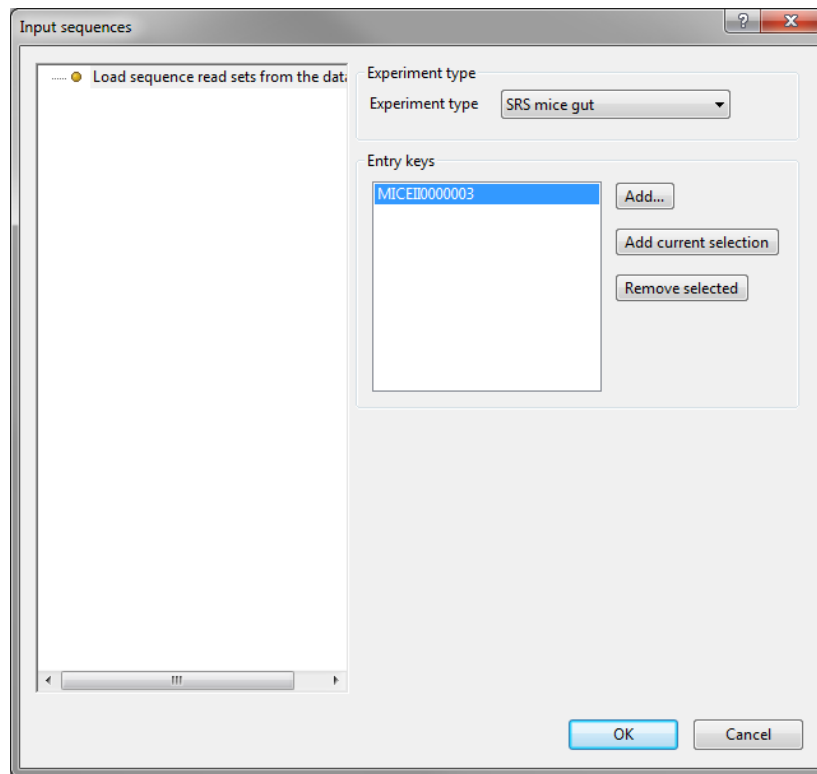


Figure 19.3.7: The *Input sequences* dialog box.

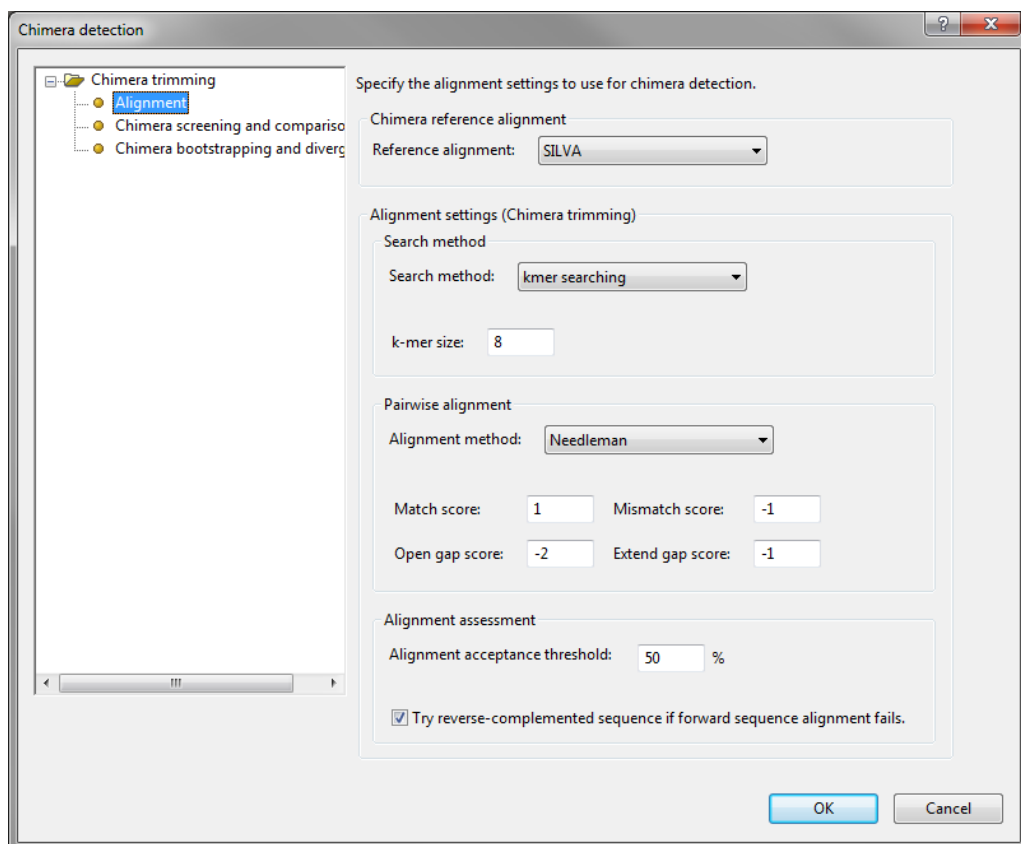


Figure 19.3.8: The *Chimera detection* dialog box: Alignment settings.

- The ***Chimera reference alignment*** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See 19.1.1 for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
 - *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

In the ***Chimera screening and comparison settings***, the following parameters can be defined:

- For the ***Chimera detection procedure*** one has the option to do a self-comparison, or to compare the read sequences to a standard.
 - The ***Self-comparison*** does not need a reference alignment but will use the more abundant reads from the sample to check all the reads in the sequence read set.
 - The ***Comparison to a standard*** will compare the reads from the sequence read set to the sequences from the reference set to decide whether or not the read is identified as a chimera.
- In the settings for the ***Screening window***, the ***Window size***, i.e. the window size for searching for chimeras, and the ***Step size***, i.e. how many base pairs a window is moved each time while screening for chimeric sequences, can be defined.
- The parameters under ***Sequence comparison*** are used while screening the read sequences for a number of potential parents. The different parameters include:
 - ***Number of potential parents (initial screening)***: how many potential parent sequences each read sequence should be compared with.

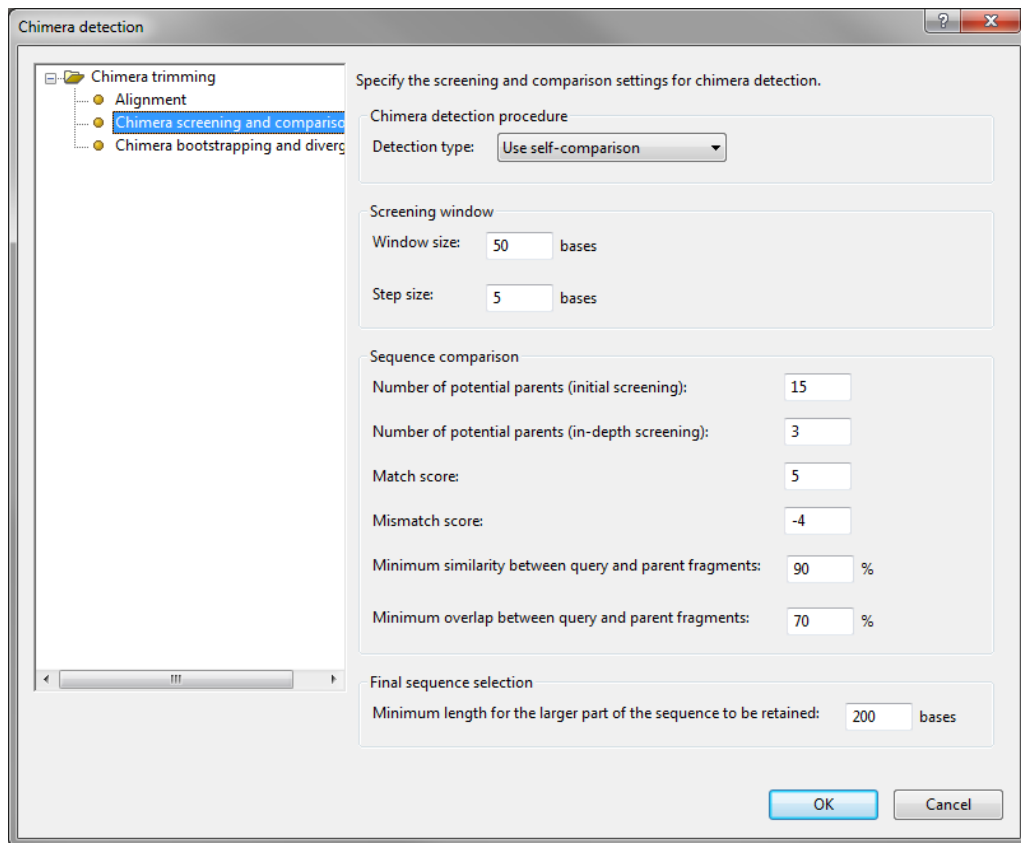


Figure 19.3.9: The *Chimera detection* dialog box: Chimera screening and comparison settings.

- **Number of potential parents (in-depth screening):** how many potential parent sequences to investigate from the best rated matches.
- **Match score** and **Mismatch score:** score and penalty values to reward a matched base and penalize a mismatch base, respectively, while screening potential parent sequences.
- **Minimum similarity between query and parent fragments (%):** sequence similarity threshold between the read and the parent fragments.
- **Minimum overlap between query and parent fragments (%):** overlap threshold of the closest matches found in the read and the parent fragments.
- The only parameter available under **Final sequence selection** is the **Minimum fragment length for a sequence to be retained**. Chimeric sequences are trimmed to include only the longest piece. To be retained in the data set, this piece should be longer than the minimum fragment length.

Additionally, the *Chimera bootstrapping and divergence settings* can be specified.

- The **Bootstrapping** settings include:
 - the **Number of bootstrap iterations** to perform,
 - the **Minimum bootstrap support** for calling a sequence chimeric, and
 - the **Percent of SNPs to sample** allows to specify the percent of SNPs that is analyzed on each side of the breakpoint for computing bootstrap support.

More detailed information on these parameter settings can be found in [17].

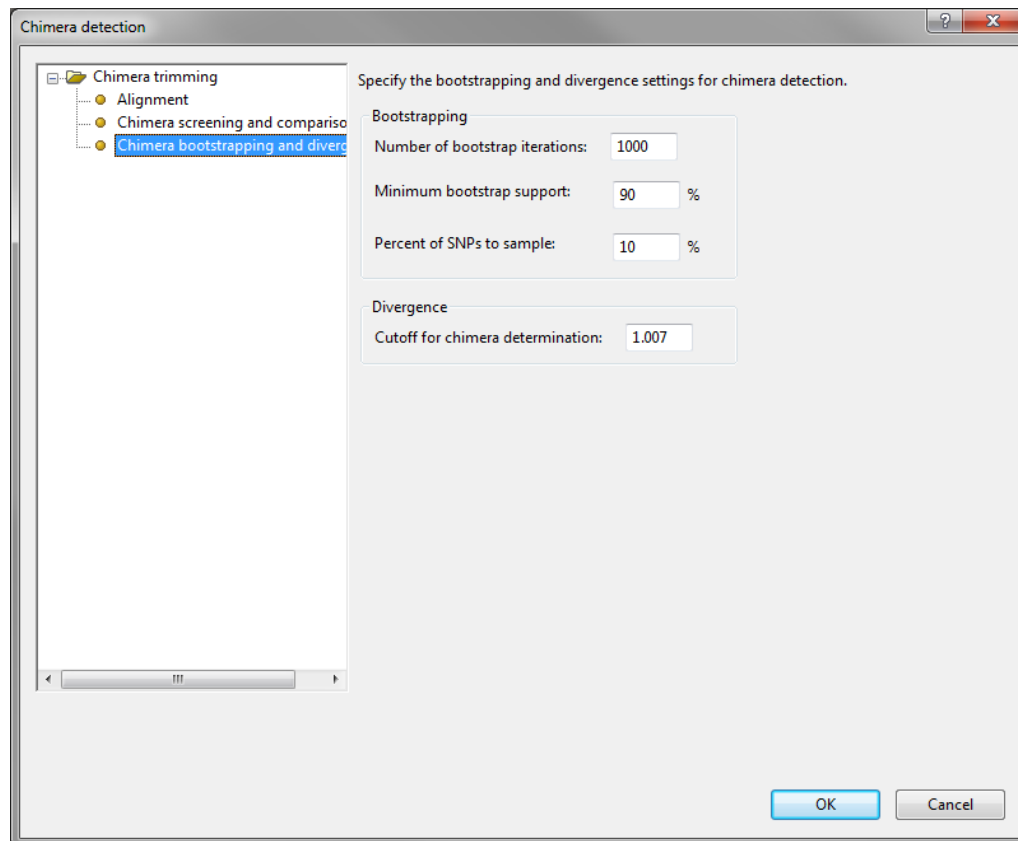


Figure 19.3.10: The *Chimera detection* dialog box: Chimera bootstrapping and divergence settings.

- The only **Divergence** parameter is the **Cutoff for chimera determination**, representing a cutoff value for chimera determination. The default value is 1.007, indicating that for a perfect alignment between the query and the chimera sequence, a maximum sequence identity of 99.3% for the query sequence against one of the parent sequences should be obtained.

19.3.1.1.3 Save to database

In the *Save to sequence read set* dialog box, the settings for the project element **Store sequences** can be defined.

This element is needed to store the resulting data set to the database. To store the data to the entry of choice, the following settings are queried for:

- **Entry key** information, indicating to which database entries the resulting data sets should be saved to. The different options are
 - Create new: use this option to save the resulting sequence read set to a new entry in the database.
 - Use entry keys used in previous run: will save the data to the same entry key as used when running the project before.
 - Use entry key from input: will save the data to the same entry key as used during the import of the data.
- **Experiment type** information, indicating to which sequence read set experiment type the data should be saved to. Any of the sequence read sets available in the database can be selected from the drop down list.

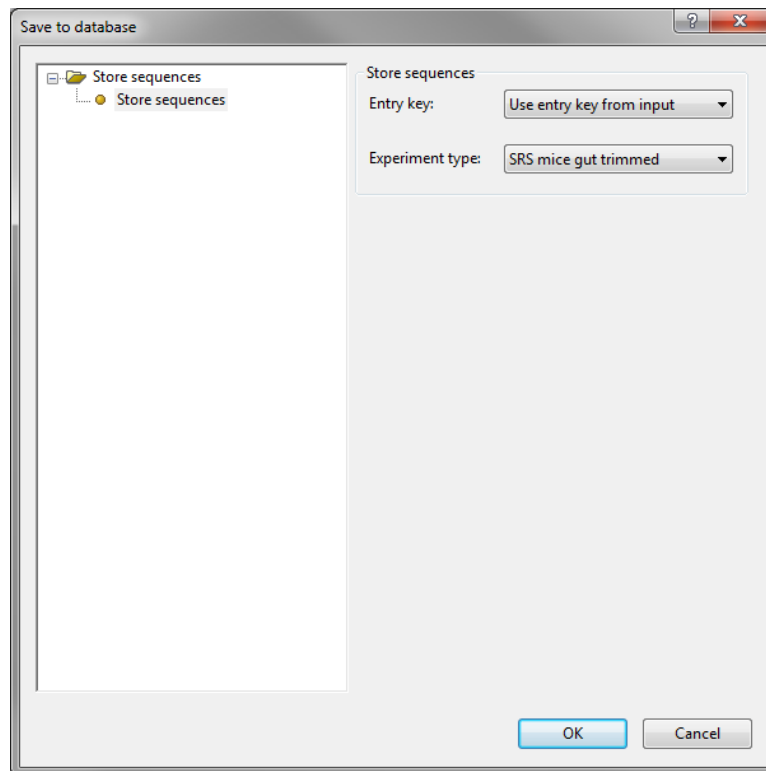


Figure 19.3.11: The *Save to sequence read set* dialog box.

Press **<OK>** to close the dialog and update the settings in the project. Press **<Cancel>** to close the dialog without altering any of the project settings.

19.3.2 Primer removal

This preprocessing step will enable you to trim off primer sequences. The forward primer is defined as the forward sequencing primer. So if you are using the 16S rRNA primers 27F and 338R to generate sequences, but you are sequencing off from the 338R end of the fragment, you would list 338R as the forward primer and 27F as the reverse. Note that forward and reverse primers can be degenerate using standard IUPAC nomenclature. Primer oligos can be entered both as upper or lowercase letters. It has been shown that sequencing errors in the PCR primer region of a sequence correlate highly with poor sequence quality. Therefore, default settings only allow exact matches. However, one can define a number of mismatches against the primer sequences to avoid this strict screening and allow for inexact matches.

Primer removal on a sequence read set can be initiated in different ways:

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis > Sequence read set types > Primer removal**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing > Primer removal**; or
- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File > New project... : Primer removal** in the *Create metagenomics project* dialog box.

When launching a primer removal analysis, the *Primer removal* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <Next> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

The primer removal imports the sequence read set to be preprocessed and only saves the trimmed reads to the output sequence read set in the database.

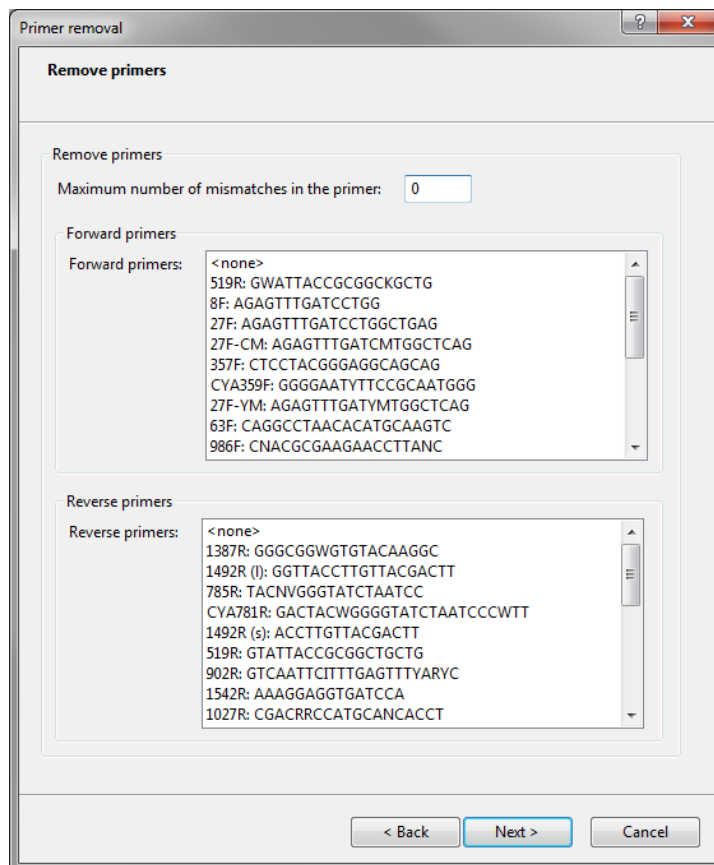


Figure 19.3.12: The *Primer removal* dialog box: Remove primers settings.


Primers to be trimmed off can be selected from the lists in the *Forward primers* and *Reverse primers*. Multiple primers can be selected by holding the **Ctrl** button. On top of the dialog, the *Maximum number of mismatches in the primer* can be defined, allowing for inexact primer matches in the reads. Press <Next> to proceed.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press <Finish> to start the actual preprocessing.

19.3.2.1 Primer removal : Project element settings

The primer removal consists of three different project elements:

- *Input sequences*: imports the selected sequence read set from the database and calculates a basic sequence summary.
- *Primer removal*: removes the forward and reverse primers from the imported reads.
- *Save to database*: saves the modified sequence read set to the database.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File** > **Element settings...** ().

19.3.2.1.1 Input sequences

The *Input sequences* dialog box is discussed in 19.3.1.1.1.

19.3.2.1.2 Primer removal

In the *Primer removal* dialog box, the settings for the project element **Remove primers** are available.

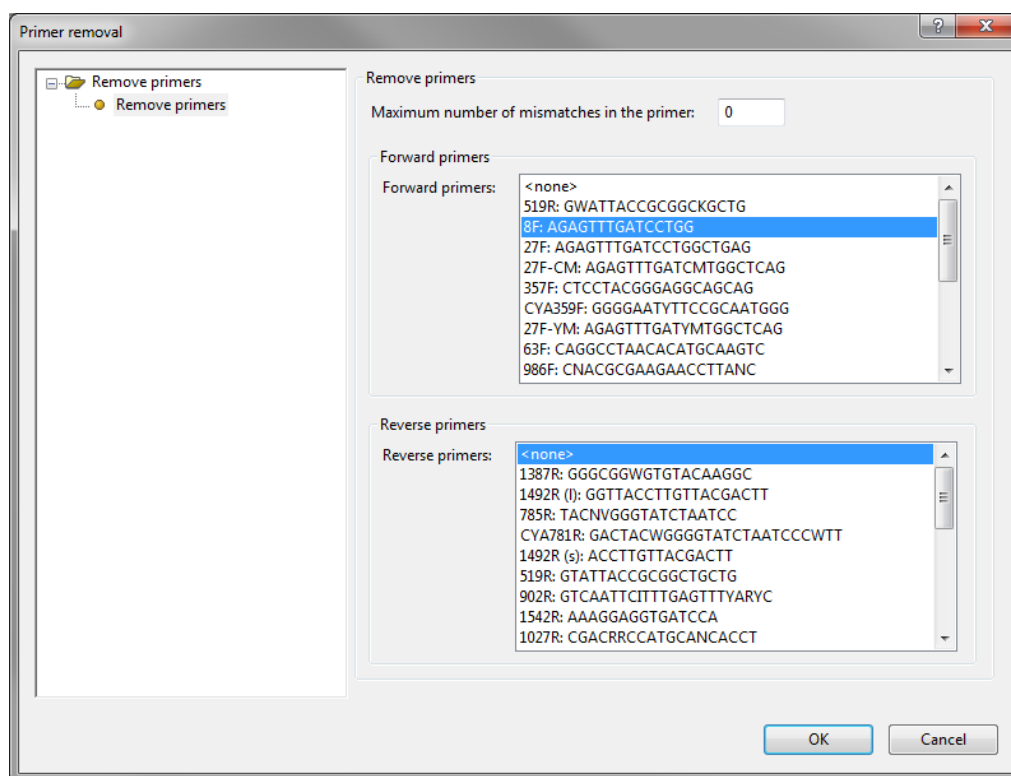


Figure 19.3.13: The *Primer removal* dialog box.

These settings include:

- **Maximum number of mismatches in the primer**, allowing for inexact primer matches against the reads.
- A list of **Forward primers** and **Reverse primers** to select the primers that will be used for the primer removal operation. Multiple primers can be selected by holding the **Ctrl** button.

Press <**OK**> to close the dialog and update the settings in the project. Press <**Cancel**> to close the dialog without altering any of the project settings.

19.3.2.1.3 Save to database

In the *Save to sequence read set* dialog box, the settings for the project element **Store sequences** can be defined.

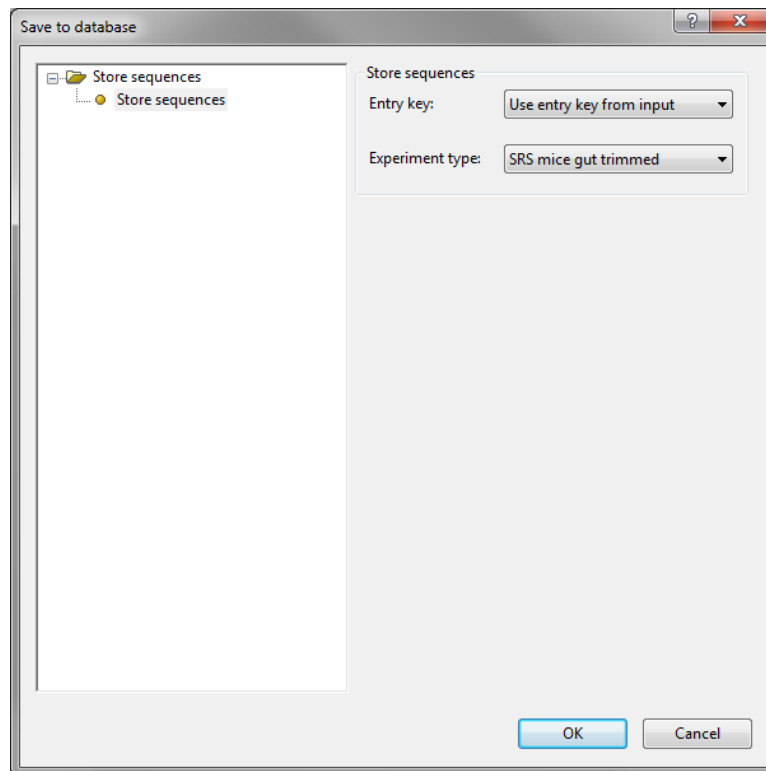


Figure 19.3.14: The *Save to sequence read set* dialog box.

This element is needed to store the resulting data set to the database. To store the data to the entry of choice, the following settings are queried for:

- **Entry key** information, indicating to which database entries the resulting data sets should be saved to. The different options are
 - Create new: use this option to save the resulting sequence read set to a new entry in the database.
 - Use entry keys used in previous run: will save the data to the same entry key as used when running the project before.
 - Use entry key from input: will save the data to the same entry key as used during the import of the data.
- **Experiment type** information, indicating to which sequence read set experiment type the data should be saved to. Any of the sequence read sets available in the database can be selected from the drop down list.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.3.3 Sequence selection

This preprocessing option performs sequence selection based on certain user defined criteria such as start and end position, and based on minimum and maximum length. The sequence selection functionality is based on the *mothur* command *screen.seqs* [35].

Sequence selection on a sequence read set can be initiated in different ways:

- from the *Main* window: make a selection of the sequence read sets that need to be preprocessed and select **Analysis** > **Sequence read set types** > **Sequence selection**;
- from the *Sequence read set experiment* window: open the *Sequence read set experiment* window of the sequence read set and select **Preprocessing** > **Sequence selection**; or
- from the *Metagenomics* window, by creating an empty *Metagenomics* window first, and selecting **File** > **New project...** : **Sequence selection** in the *Create metagenomics project* dialog box.

When executing a screening of the read sequences, the *Sequence selection* dialog box will appear.

When starting the preprocessing analysis from the *Main* window or the *Metagenomics* window, the first page of the dialog will ask for the **Entry key**, the **Input experiment type** or both. Press <**Next**> to proceed. When starting from the *Sequence read set experiment* window, this information is already present and this first page will be skipped in the dialog.

Different *Selection methods* can be checked and combined in the *Sequence selection* dialog box to create a custom sequence selection based on the alignment outcome.

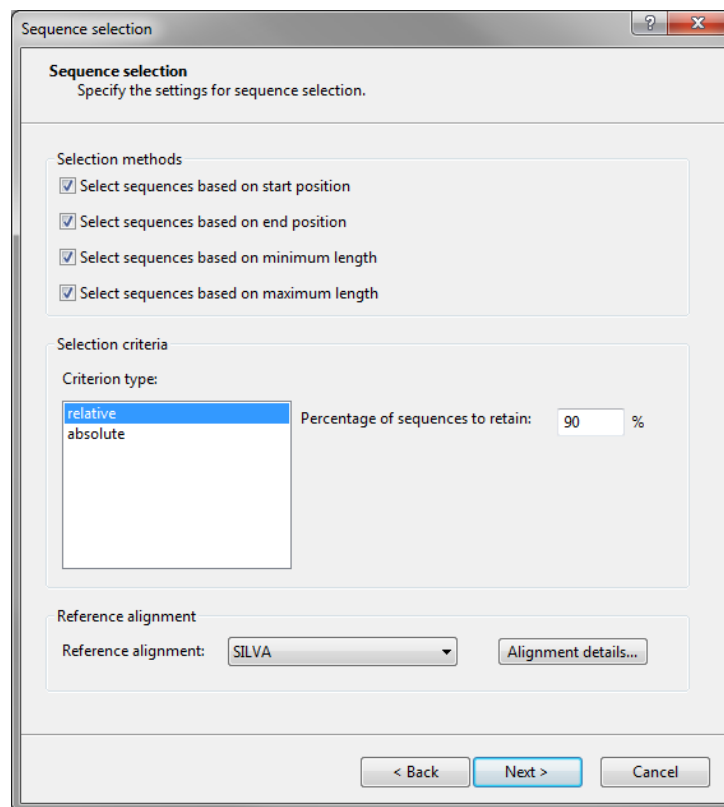


Figure 19.3.15: The *Sequence selection* dialog box: Sequence selection settings.

Some reads from the data set may not align in the same region as most of the reads that are analyzed. In a situation where read alignments started at deviating alignment positions, the following options can be used:

- *Select sequences based on start position*: reads starting the alignment after the start position entered in the dialog box, will be omitted from the output read set.
- *Select sequences based on end position*: reads ending the alignment before the end position entered in the dialog box, will be omitted from the output read set.

In some pyrosequencing studies, the reads will differ in length. Trimming these reads within the expected window of minimum and maximum size can be done by selecting the options:

- *Select sequences based on minimum length*: reads spanning an alignment region shorter than the minimum length entered in the dialog box, will be omitted from the output read set.
- *Select sequences based on maximum length*: reads spanning an alignment region larger than the maximum length entered in the dialog box, will be omitted from the output read set.

The *Selection criteria* can be defined in an *absolute* manner, as threshold values entered in the dialog box, or in a *relative* one. When using the relative criteria, the *percentage of sequences to retain* needs to be entered in the dialog box. The percentage of sequences to retain is evaluated for the combination of selection methods checked in the dialog, meaning that e.g. when the percentage to retain is set at 90%, the sequence read set will be trimmed off by 10% of the reads, using all the selected criteria.

All positions and lengths refer to positions and lengths on the reference alignment selected from the *Reference alignment* drop down box at the bottom of the dialog. When the reference alignment is defined, the alignment settings can be entered from the *Alignment details* dialog after selecting *<Alignment details ...>*. In the *Alignment details* dialog box, the dedicated settings to align the reads from the sequence read set to the reference alignment can be defined. These settings include the search method to find the closest template for each read, the method to create a pairwise alignment between the read and the de-gapped template sequences, and the settings for the alignment assessment.

- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.

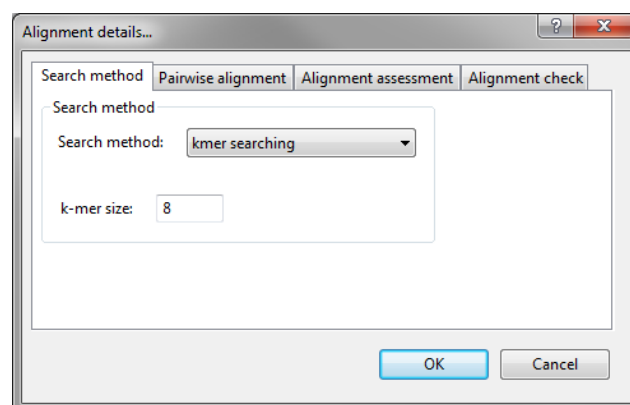


Figure 19.3.16: The *Alignment details* dialog box: Search method settings.

- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.

- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.

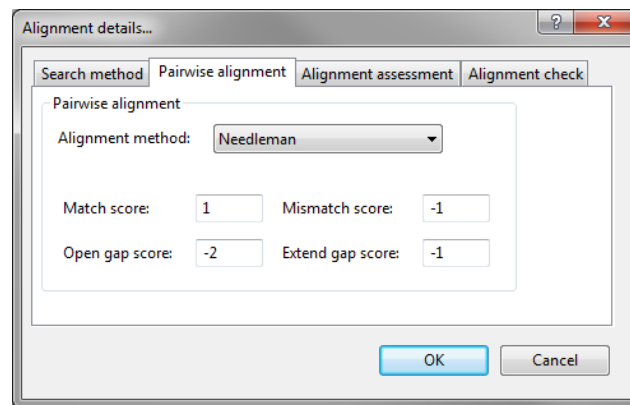


Figure 19.3.17: The *Alignment details* dialog box: Pairwise alignment settings.

- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

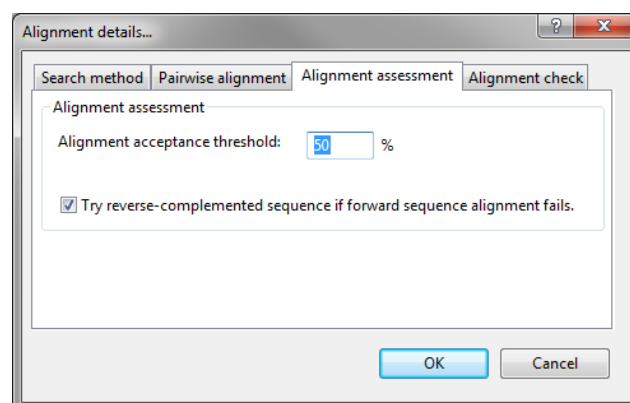


Figure 19.3.18: The *Alignment details* dialog box: Alignment assessment settings.

Press **<OK>** to save the modified settings to the project. Press **<Cancel>** to return to the *Sequence selection* dialog box without altering the alignment settings.

Press **<Next>** to proceed.

At the last page, select an output experiment type from the drop down list. All sequence read sets present in the database are listed here. Select the experiment type to export the preprocessed reads to, and press **<Finish>** to start the actual preprocessing.

19.3.3.1 Sequence selection : Project element settings

The sequence selection consists of three different project elements:

- *Input sequences*: imports the selected sequence read set from the database and calculates a basic sequence summary.
- *Sequence selection*: removes reads based on their start and end position in the alignment, and based on minimum and maximum alignment length of the read.
- *Save to database*: saves the modified sequence read set to the database.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File** > **Element settings...** (⚙️).

19.3.3.1.1 Input sequences

The *Input sequences* dialog box is discussed in 19.3.1.1.1.

19.3.3.1.2 Sequence selection

In the *Sequence selection* dialog box, the settings for the project element *Sequence selection* are available. See 19.3.3 for more information on the details of the *Selection methods*, the *Selection criteria* and the *Reference alignment*.

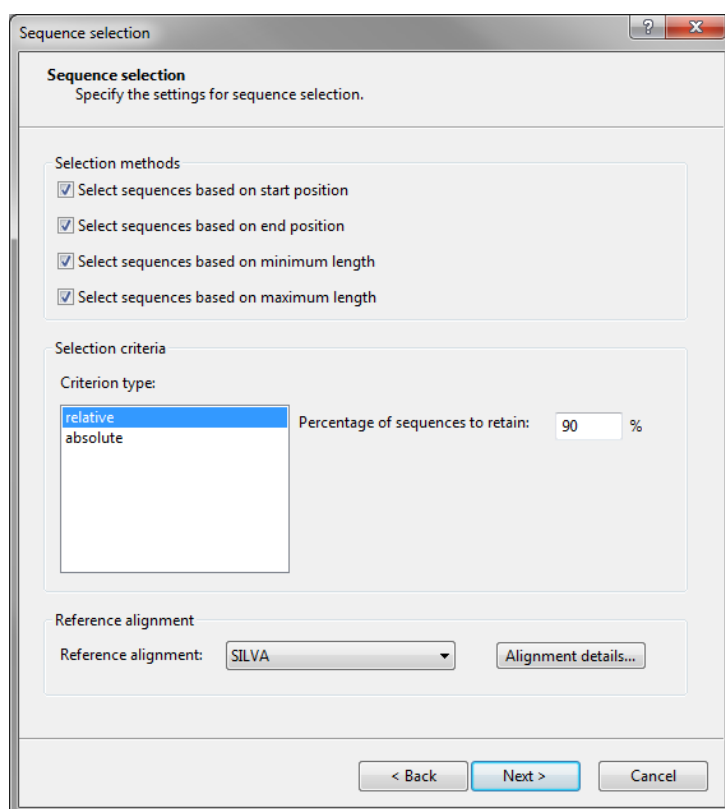


Figure 19.3.19: The *Sequence selection* dialog box.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.3.3.1.3 Save to database

In the *Save to sequence read set* dialog box, the settings for the project element *Store sequences* can be defined.

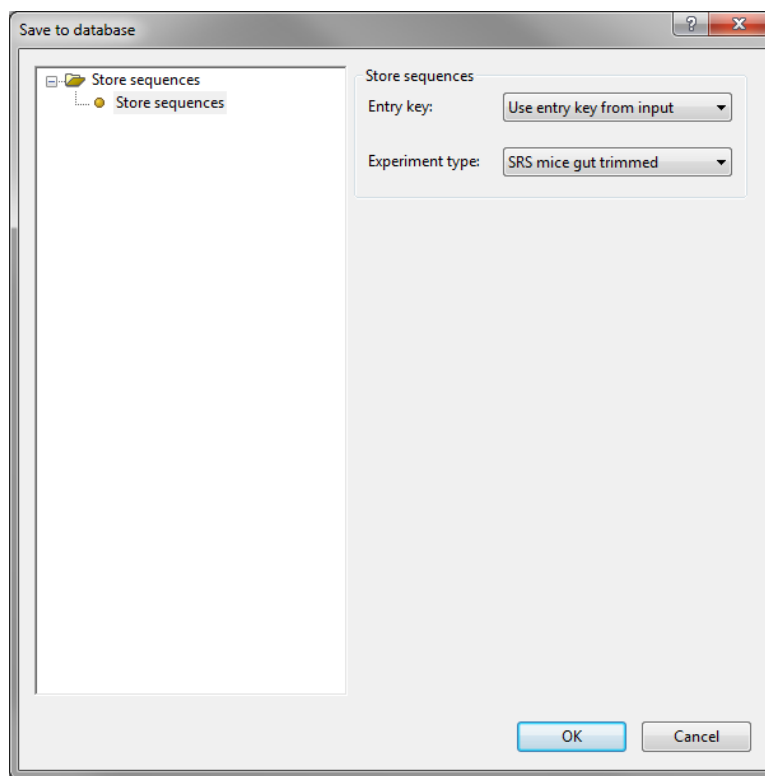


Figure 19.3.20: The *Save to sequence read set* dialog box.

This element is needed to store the resulting data set to the database. To store the data to the entry of choice, the following settings are queried for:

- **Entry key** information, indicating to which database entries the resulting data sets should be saved to. The different options are
 - Create new: use this option to save the resulting sequence read set to a new entry in the database.
 - Use entry keys used in previous run: will save the data to the same entry key as used when running the project before.
 - Use entry key from input: will save the data to the same entry key as used during the import of the data.
- **Experiment type** information, indicating to which sequence read set experiment type the data should be saved to. Any of the sequence read sets available in the database can be selected from the drop down list.

Press <**OK**> to close the dialog and update the settings in the project. Press <**Cancel**> to close the dialog without altering any of the project settings.

Chapter 19.4

Predefined metagenomics workflows

Fully elaborated analysis templates are available in the software for the following predefined metagenomics analyses.

- *Identification against a taxonomic database:* In this analysis, all sample reads are identified against a taxonomic database. The operational taxonomic units (further called OTUs) can be defined in two ways: either directly from the phylotypic levels defined in the taxonomic database used for the automated identification of the reads, or from the sequence data itself by performing a sequence clustering first, and then determining the consensus taxonomy per cluster. Finally, for each sample the OTU abundances are stored as characters in the database (see [19.4.1](#)).

For this analysis, BioNumerics makes use of the [mothur \[35\]](#) project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). BioNumerics uses the flexibility of the algorithms incorporated in [mothur](#) and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results.

- *Single-sample diversity analysis:* In this analysis, the alpha-diversity of a single sample is assessed. For the operational taxonomic units (further called OTUs) obtained, the within-sample diversity, the community evenness, the community richness and the community diversity indices are calculated (see [19.4.2](#)).

For this analysis, BioNumerics makes use of the [mothur \[35\]](#) project, initiated by Dr. Patrick Schloss and colleagues (Department of Microbiology & Immunology, University of Michigan). BioNumerics uses the flexibility of the algorithms incorporated in [mothur](#) and further elaborates on these results by creating a fully interactive reporting service for the interpretation and manipulation of the results.

19.4.1 Identification against a taxonomic database

When starting the analysis from the *Main* window, one additional dialog page is displayed where you can select the sequence read set experiment type that should be used for the analysis. All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select **<Next>** to proceed.

Similarly, when starting the analysis from within the *Metagenomics* window, another additional dialog page is displayed (see below). In this dialog, the entry information and the input sequence read set experiment type need to be specified. Press **<Add entry>** to select the entry to analyze. This opens the *Select entry* dialog box where the entry can be highlighted in the database list, press **<OK>** to return to the *Single sample diversity analysis* wizard where the sequence read set experiment type to be used for the analysis can be selected from the drop down list with available sequence read set experiment types present in the database. Select **<Next>** to proceed.

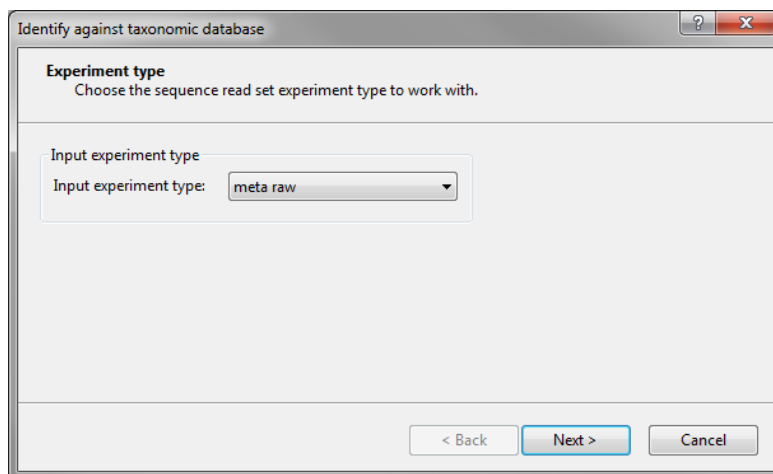


Figure 19.4.1: The *Identify against taxonomic database* wizard: Experiment type settings.

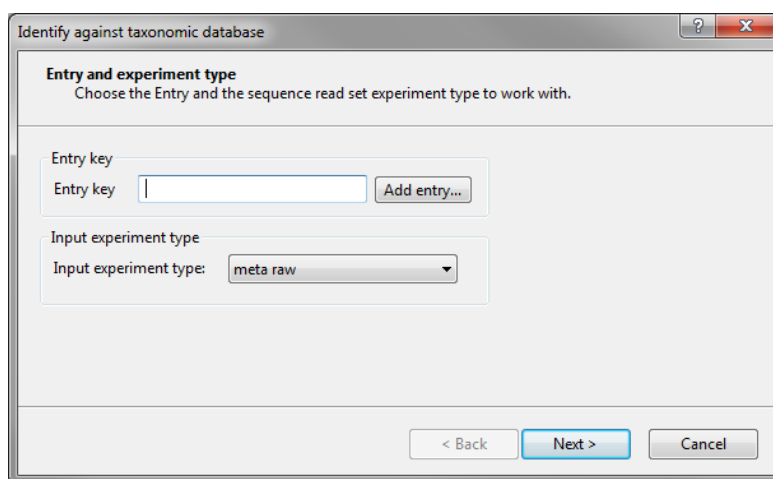


Figure 19.4.2: The *Identify against taxonomic database* wizard: Entry and Experiment type settings.

When starting the metagenomics analysis from the *Sequence read set experiment* window, the first page of the *Identify against taxonomic database* wizard asks for the OTU determination settings (see below). When starting the analysis from the *Main* window or the *Metagenomics* window, this will be the second page of the wizard. The OTUs can be determined two ways.

- The option ***Determine OTUs by taxonomic identification*** will determine the unique sequences in the sequence read set, identify the sequences against the reference taxonomy and create OTUs from the taxonomic identification results.
- The option ***Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs*** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. For each of the reads in the cluster the taxonomic identification is calculated and based on these results, the cluster consensus taxonomy is defined.

Select one of the two options and press <Next> to proceed.

The taxonomic identification settings only require the taxonomic reference database to be selected from the drop down list and to specify the start and end level of the taxonomic identification for the current analysis. Select <Next> to proceed.

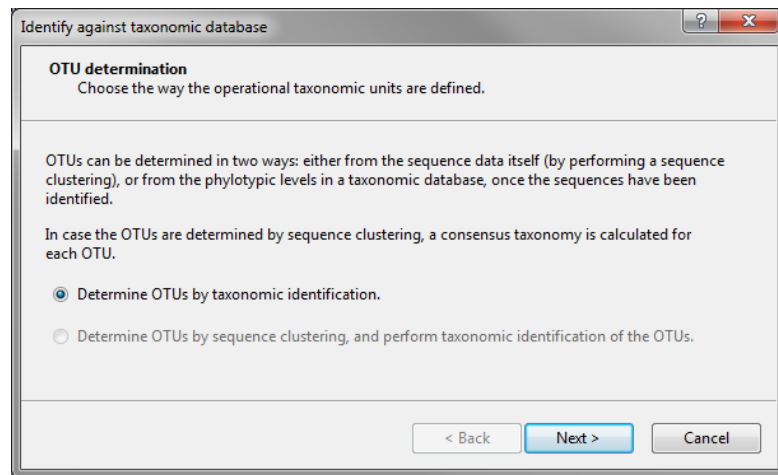


Figure 19.4.3: The *Identify against taxonomic database* wizard: OTU determination settings.



See [19.1.2](#) for more information on the taxonomic reference database.

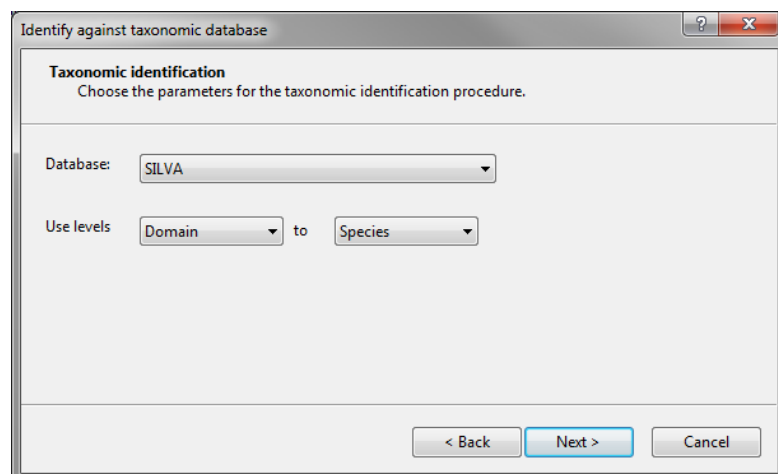


Figure 19.4.4: The *Identify against taxonomic database* wizard: Taxonomic identification settings.

The output experiment type settings define to which character experiment type the OTU abundance results will be saved to the database. You can leave the default settings “Choose automatically”. Doing so, a character set is created for each unique combination of a taxonomic database and a taxonomic level. If desired, a custom prefix can be added to be used in front of the unique character experiment type names that will be created. Leaving all settings default enables you to save the analysis results for multiple entries based on the same reference taxonomy to the same character experiment types, allowing a follow-up analysis on these character values in e.g. the *Comparison* window. Select **<Finish>** to complete the wizard, and to start the analysis in the *Metagenomics* window.

The taxonomic identification project consists of three different project elements:

- *Input sequences*: imports the selected sequence read set from the database and calculates a basic sequence summary.
- *Phylotype determination*: filters the unique sequences from the sequence read set and calculates a cluster analysis on these reads. Based on a cluster cutoff value, OTUs are created and later identified against the taxonomic database.

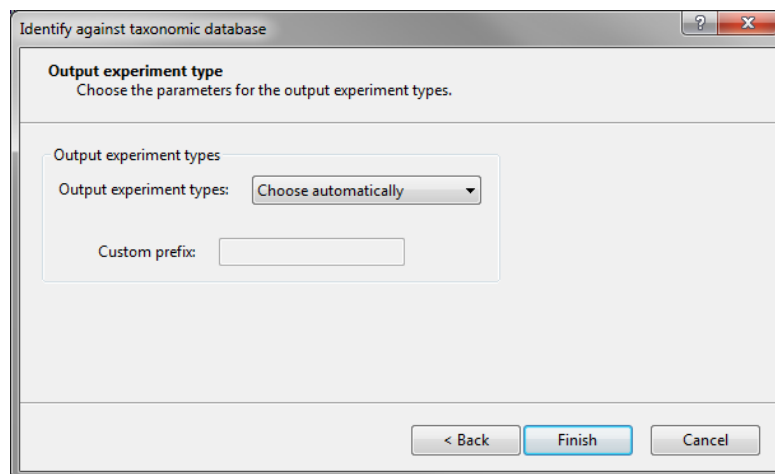


Figure 19.4.5: The *Identify against taxonomic database* wizard: Output experiment type settings.

- *Save to character set*: saves the OTU abundances to character sets in the database.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File > Element settings...** (🔧).

19.4.1.1 Input sequences

The *Input sequences* dialog box is discussed in 19.3.1.1.1.

19.4.1.2 Phylotype determination: OTU determination by taxonomic identification

The phylotype determination assigns the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

In the *Phylotype determination* dialog box, the settings for the read identification can be defined.

- First, the ***Taxonomic database*** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the ***Identification method*** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:
 - ***Number of bootstrap iterations***: identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
 - ***Bootstrap score threshold***: the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic

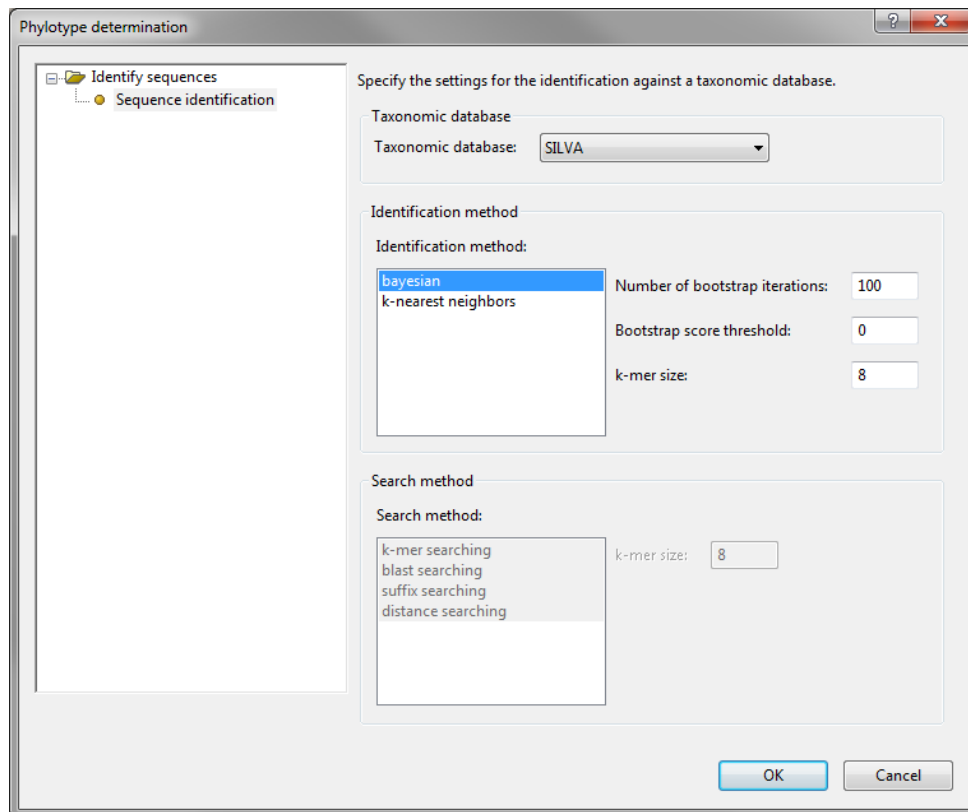


Figure 19.4.6: The *Phylotype determination* dialog box.

assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).

- ***k-mer size***: k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- ***Number of sequences to retain***: identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the ***Search method*** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - ***k-mer searching*** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - ***blast searching***: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.

- **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.1.3 Phylotype determination: OTU determination by sequence clustering and taxonomic identification of the OTUs

The phylotype determination aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, the identification of the OTUs can be obtained using an identification method of choice against the taxonomic reference database.

The first step in the analysis is to align the reads from the sequence read set to the uploaded reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,
2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *Phylotypic OTU preparation* dialog box, the **Alignment settings** can be modified.

- The **Reference alignment** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See 19.1.1 for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.

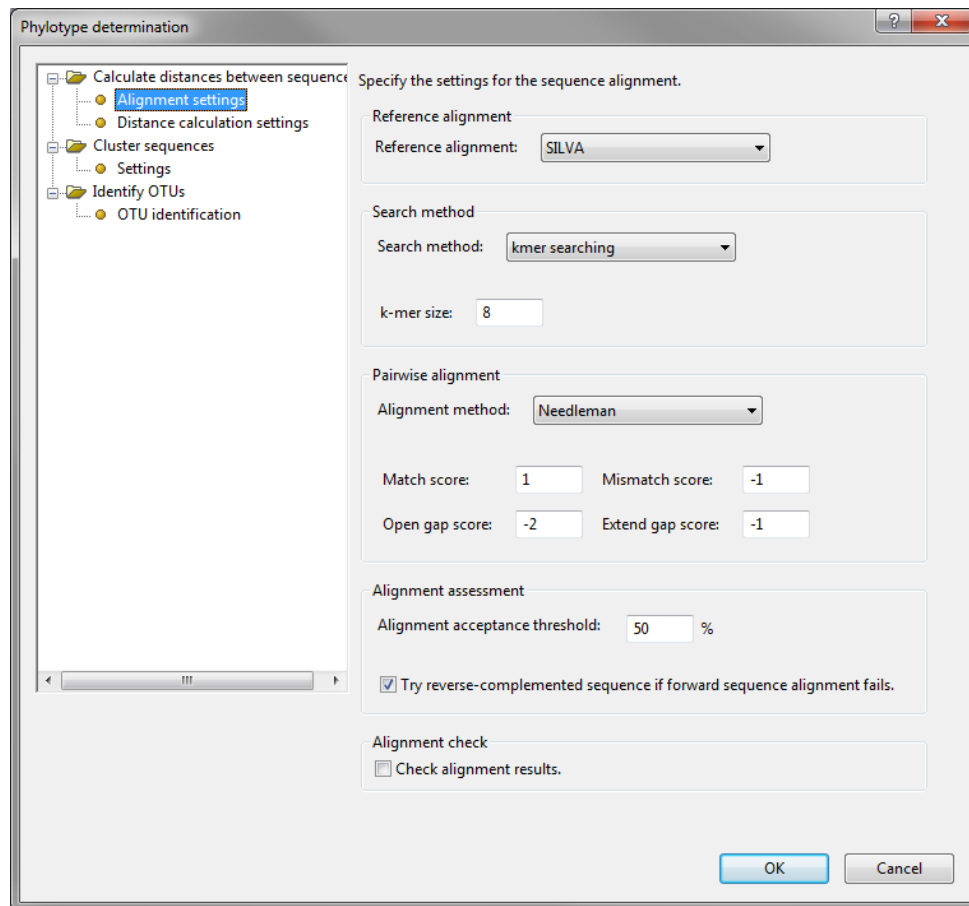


Figure 19.4.7: The *Phylotypic OTU preparation* dialog box: Alignment settings.

- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Once the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

- The *Gap treatment* lists three different options on how to handle gap comparisons and terminal gaps.
 - *Count consecutive gaps as a single gap*: counts a string of gaps as a single gap i.e. the gap is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.

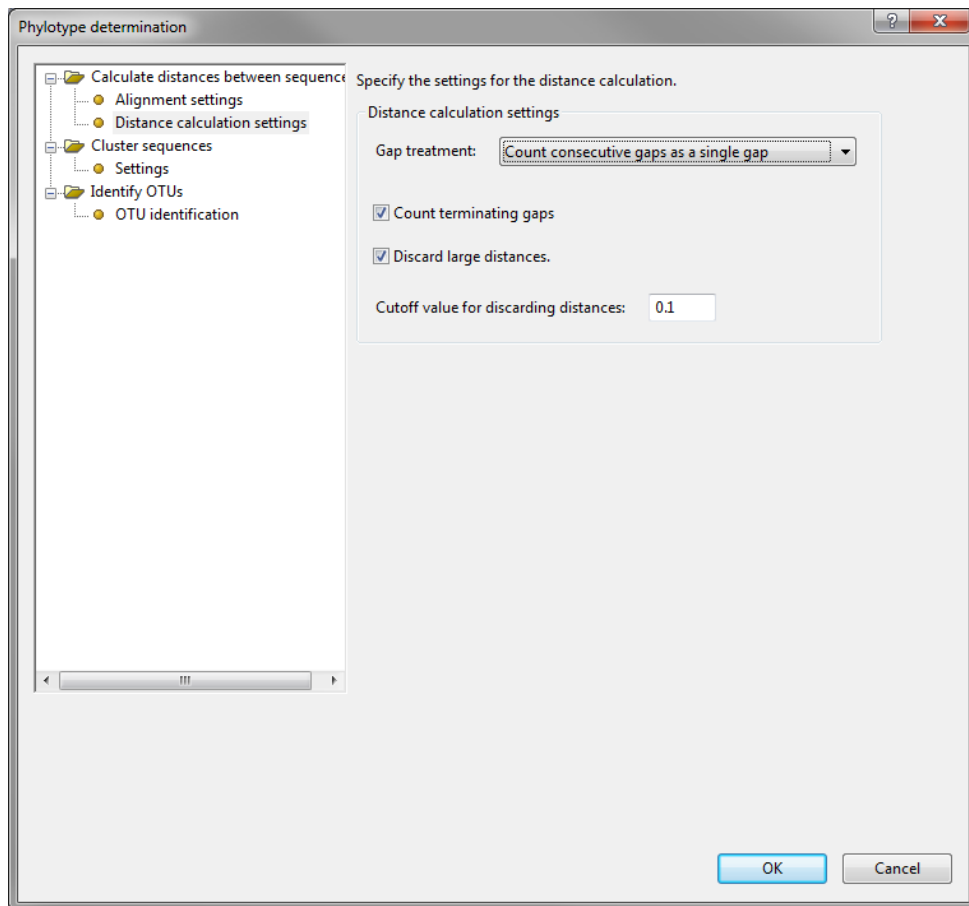


Figure 19.4.8: The *Phylotypic OTU preparation* dialog box: Distance calculation settings.

- **Ignore gaps completely:** this distance calculation does not take into account any gaps or insertions.
- **Count each gap individually:** will penalize each position of the gap or insert as a single mismatch.
- The option **Count terminating gaps** determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
- The option **Discard large distances** can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the **Cutoff value for discarding distances**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

Once a distance matrix is calculated for the reads in the project, the sequence clustering can be started to assign sequences to OTUs. The settings for the **Sequence clustering** include:

- The choice of **clustering method**. Four clustering methods are implemented:
 - **Average neighbor clustering:** this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.

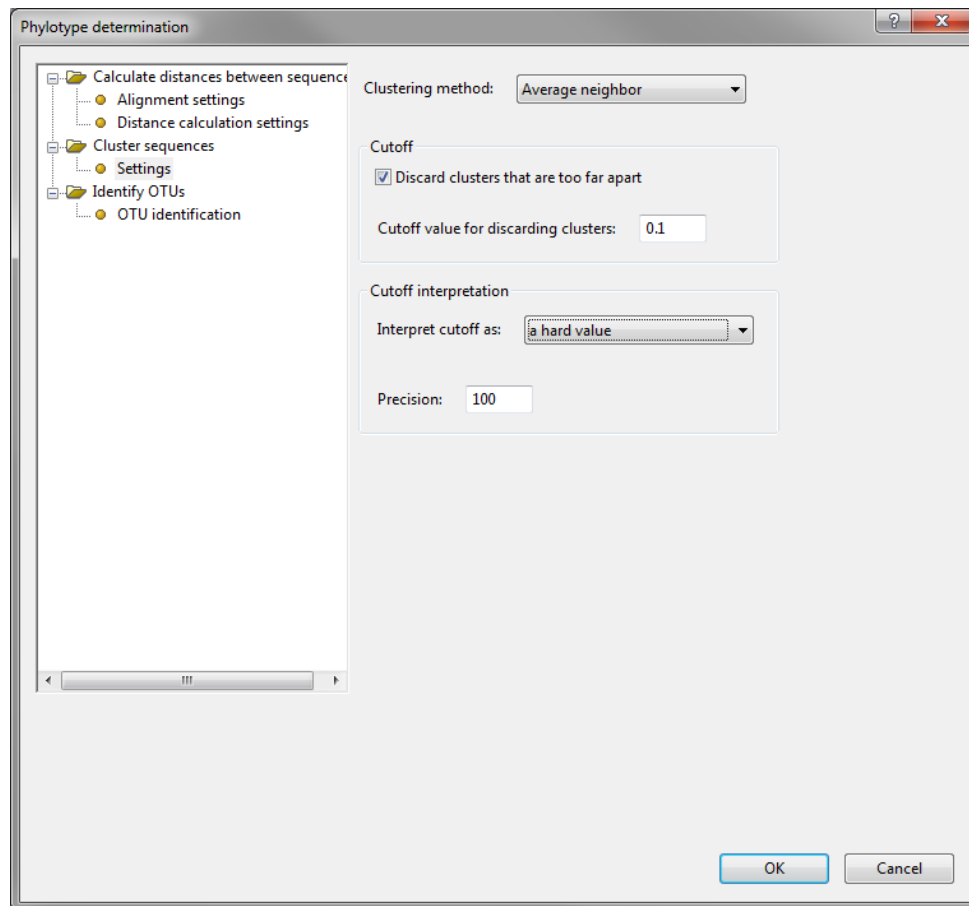


Figure 19.4.9: The *Phylotypic OTU preparation* dialog box: Cluster settings.

- **Nearest neighbor clustering:** each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
- **Furthest neighbor clustering:** all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
- **Weighted neighbor clustering:** this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.
 - The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

The identification of OTUs will assign the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

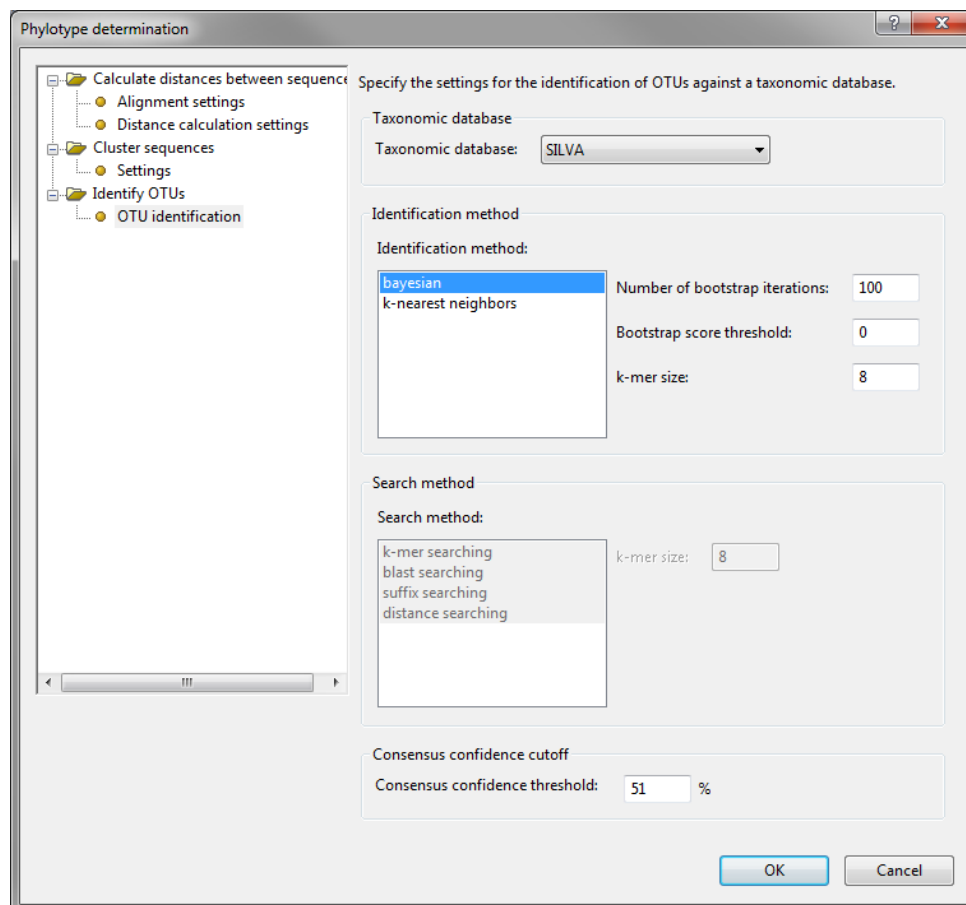


Figure 19.4.10: The *Phylotypic OTU preparation* dialog box: OTU identification settings.

- First, the ***Taxonomic database*** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the ***Identification method*** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:
 - ***Number of bootstrap iterations***: identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
 - ***Bootstrap score threshold***: the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
 - ***k-mer size***: k-mer size to be used in screening the taxonomy and read sequences [5-12,default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - **k-mer searching** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - **blast searching**: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.
- The **Consensus confidence cutoff** is the minimum value to specify a consensus taxonomy on the OTUs. The default is 51%, which is the minimum cutoff value that can be set for this parameter.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.1.4 Save to character set

Once the OTUs have been taxonomically identified, the taxonomic assignments and abundances are saved to the database. The settings to define this export to the database character sets are given in the *Save to character set* dialog box.

This element is needed to store a separate character set for each taxonomic level to the database. To store the data to the entries of choice, the following settings are queried for:

- **Level:** a specific sequence identity threshold will be used as cluster cutoff value to determine the OTUs. When the final sequence identity level is set to unique, the cutoff value is indicated as -1.
- **Taxonomic level:** the level assignment (e.g. Class) as indicated on top of the dialog refers to a taxonomic level defined in the taxonomic reference database (e.g. taxonomic level 2 containing Class information). By default, the taxonomic levels are displayed as imported from the taxonomic reference file and should not be altered upon export.
- **Entry key** information, indicating to which database entries the resulting data sets should be saved to. The different options are
 - Create new: use this option to create a new entry for each character experiment (i.e. taxonomic level abundances) saved to the database.

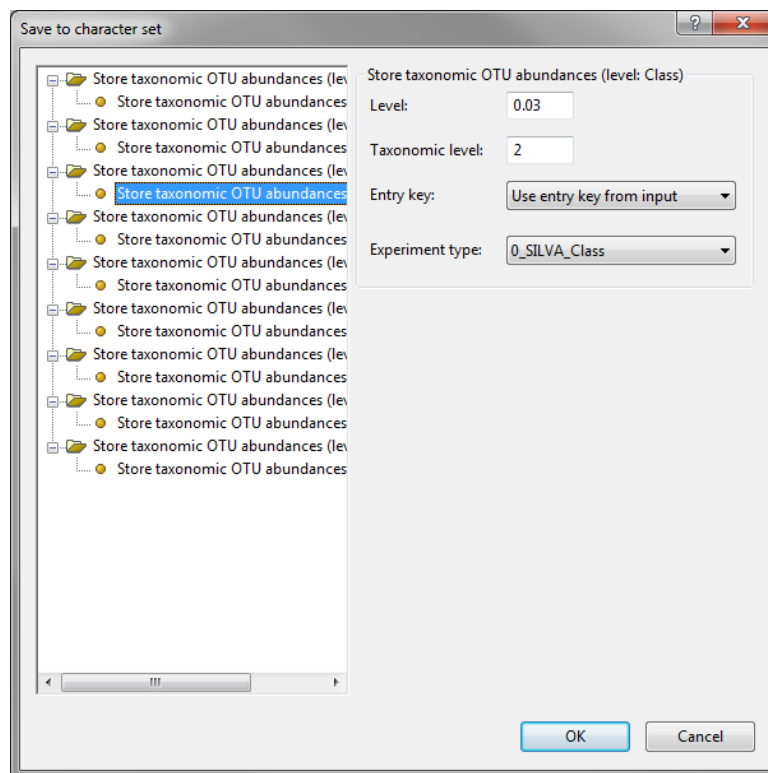


Figure 19.4.11: The *Save to character set* dialog box.

- Use entry keys used in previous run: will save the data to the same entry key as used when running the project before.
- Use entry key from input: will save the data to the same entry key as used during the import of the data. Use entry key from input is the default setting and should not be changed, as it allows to save the OTU abundances to the same entry as the original sequence read set used in the analysis.
- **Experiment type** information, indicating to which sequence character experiment type the data should be saved to. Default the experiment types names are composed of [unique number]_[name of reference alignment]_[taxonomic level name].

Press **<OK>** to close the dialog and update the settings in the project. Press **<Cancel>** to close the dialog without altering any of the project settings.

19.4.1.5 Identify against taxonomic database : Project element settings

The taxonomic identification project consists of three different project elements:

- **Input sequences:** imports the selected sequence read set from the database and calculates a basic sequence summary.
- **Phylotype determination:** filters the unique sequences from the sequence read set and calculates a cluster analysis on these reads. Based on a cluster cutoff value, OTUs are created and later identified against the taxonomic database.
- **Save to character set:** saves the OTU abundances to character sets in the database.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File** > **Element settings...** (🔧).

19.4.1.5.1 Input sequences

The *Input sequences* dialog box is discussed in 19.3.1.1.1.

19.4.1.5.2 Phylotype determination: OTU determination by taxonomic identification

The phylotype determination assigns the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

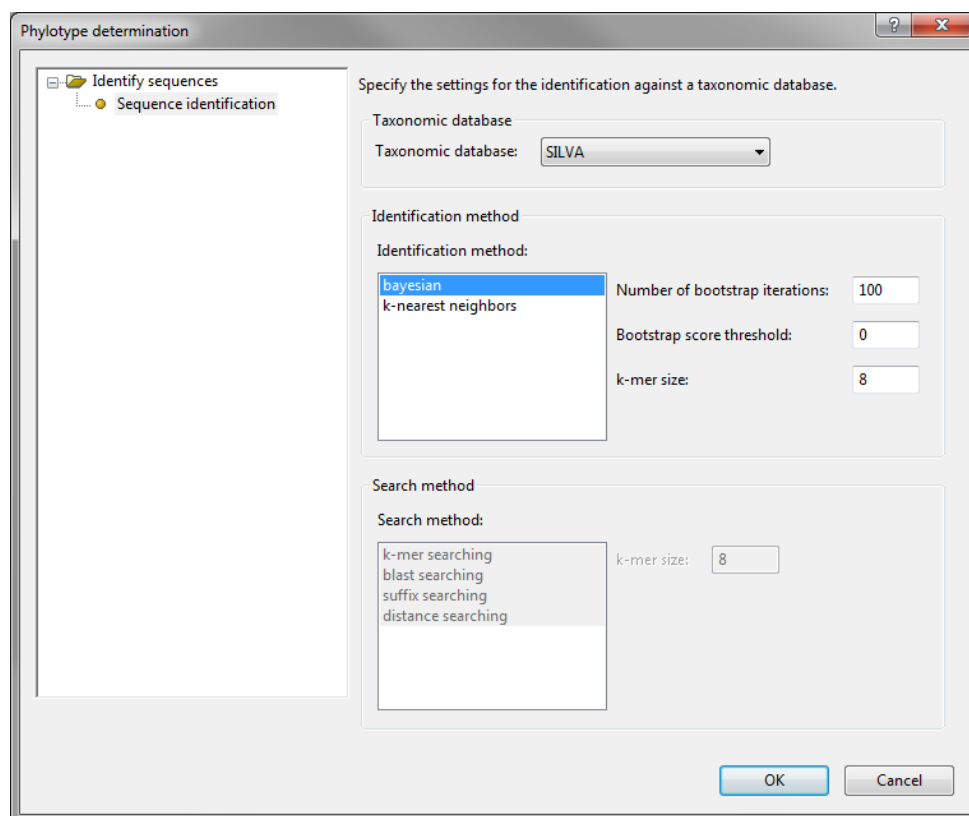


Figure 19.4.12: The *Phylotype determination* dialog box.

In the *Phylotype determination* dialog box, the settings for the read identification can be defined.

- First, the **Taxonomic database** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the **Identification method** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:

- **Number of bootstrap iterations:** identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
- **Bootstrap score threshold:** the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
- **k-mer size:** k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - **k-mer searching** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - **blast searching**: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.1.5.3 Phylotype determination: OTU determination by sequence clustering and taxonomic identification of the OTUs

The phylotype determination aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, the identification of the OTUs can be obtained using an identification method of choice against the taxonomic reference database.

The first step in the analysis is to align the reads from the sequence read set to the uploaded reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,

2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *Phylotypic OTU preparation* dialog box, the **Alignment settings** can be modified.

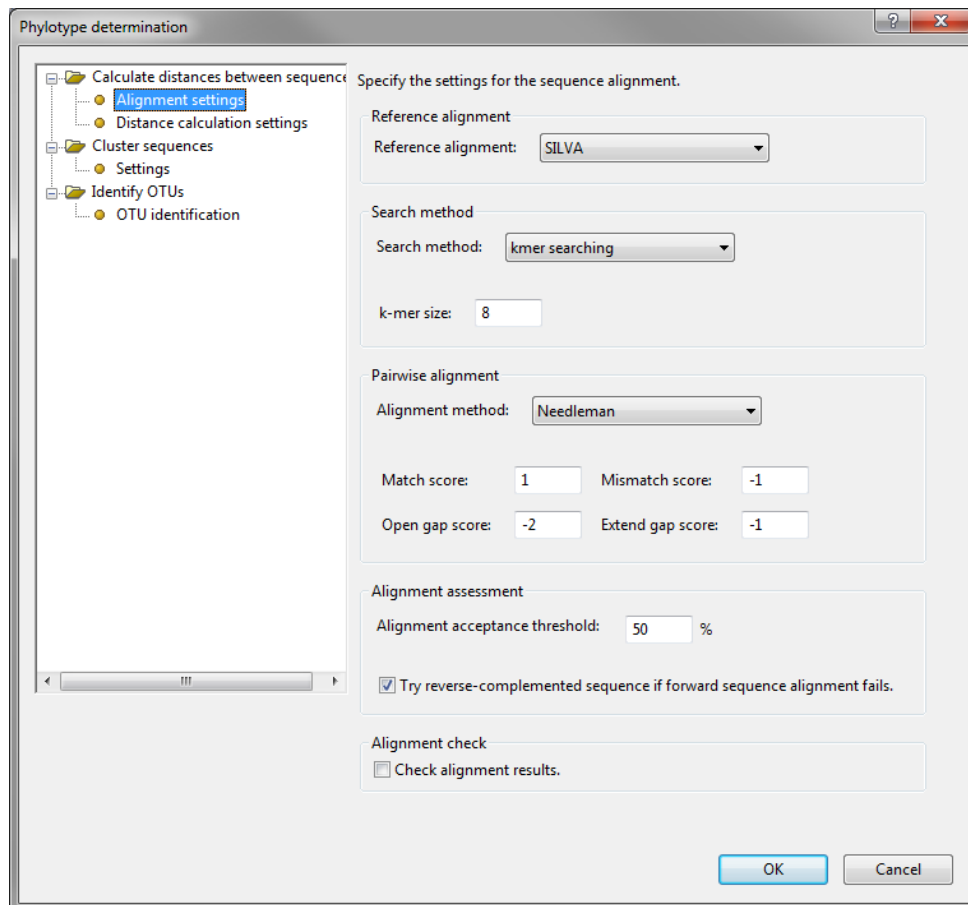


Figure 19.4.13: The *Phylotypic OTU preparation* dialog box: Alignment settings.

- The **Reference alignment** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See 19.1.1 for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:

- *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Once the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

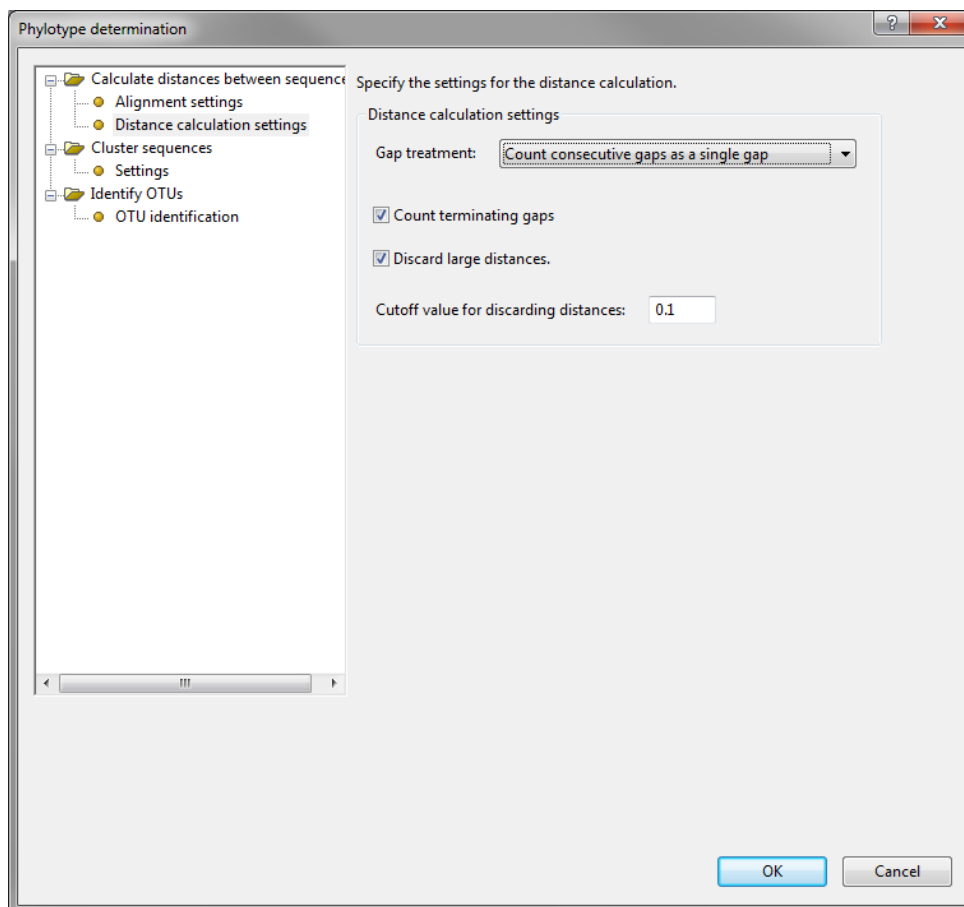


Figure 19.4.14: The *Phylotypic OTU preparation* dialog box: Distance calculation settings.

- The *Gap treatment* lists three different options on how to handle gap comparisons and terminal gaps.

- **Count consecutive gaps as a single gap**: counts a string of gaps as a single gap i.e. the gap is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.
 - **Ignore gaps completely**: this distance calculation does not take into account any gaps or insertions.
 - **Count each gap individually**: will penalize each position of the gap or insert as a single mismatch.
- The option **Count terminating gaps** determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
 - The option **Discard large distances** can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the **Cutoff value for discarding distances**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

Once a distance matrix is calculated for the reads in the project, the sequence clustering can be started to assign sequences to OTUs. The settings for the **Sequence clustering** include:

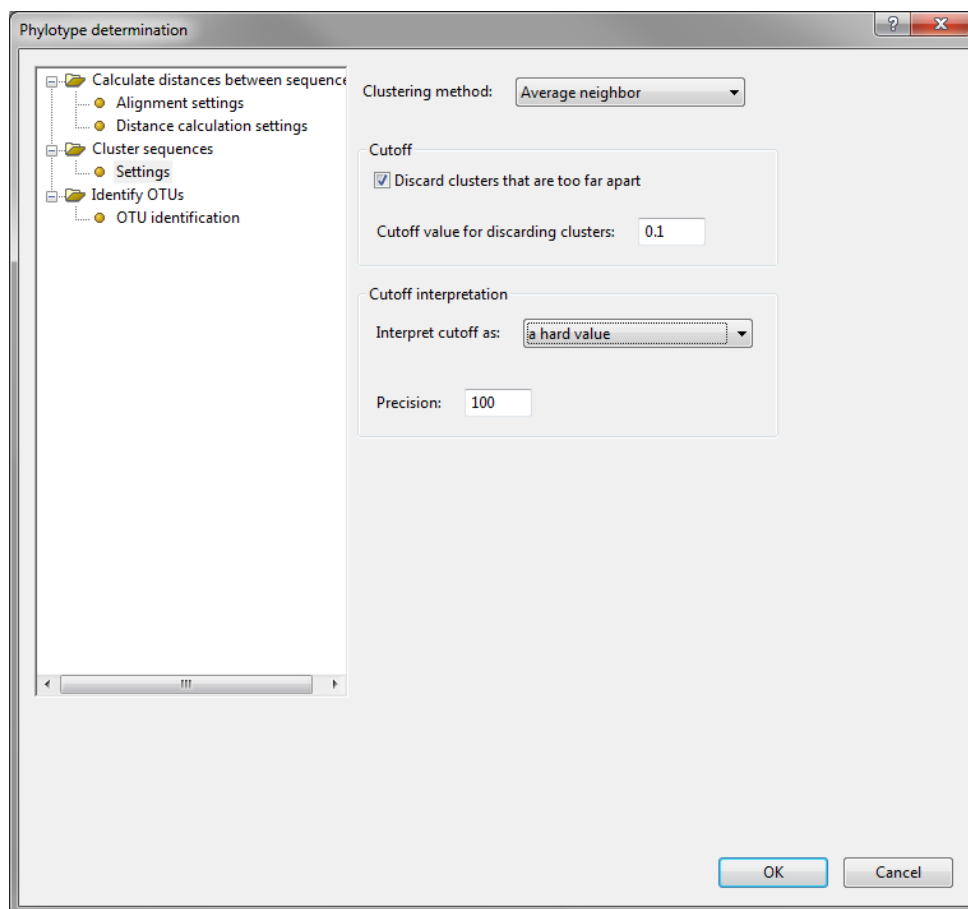


Figure 19.4.15: The *Phylotypic OTU preparation* dialog box: Cluster settings.

- The choice of **clustering method**. Four clustering methods are implemented:

- **Average neighbor clustering:** this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.
 - **Nearest neighbor clustering:** each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
 - **Furthest neighbor clustering:** all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
 - **Weighted neighbor clustering:** this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.
 - The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

The identification of OTUs will assign the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

- First, the **Taxonomic database** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the **Identification method** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:
 - **Number of bootstrap iterations:** identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
 - **Bootstrap score threshold:** the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
 - **k-mer size:** k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

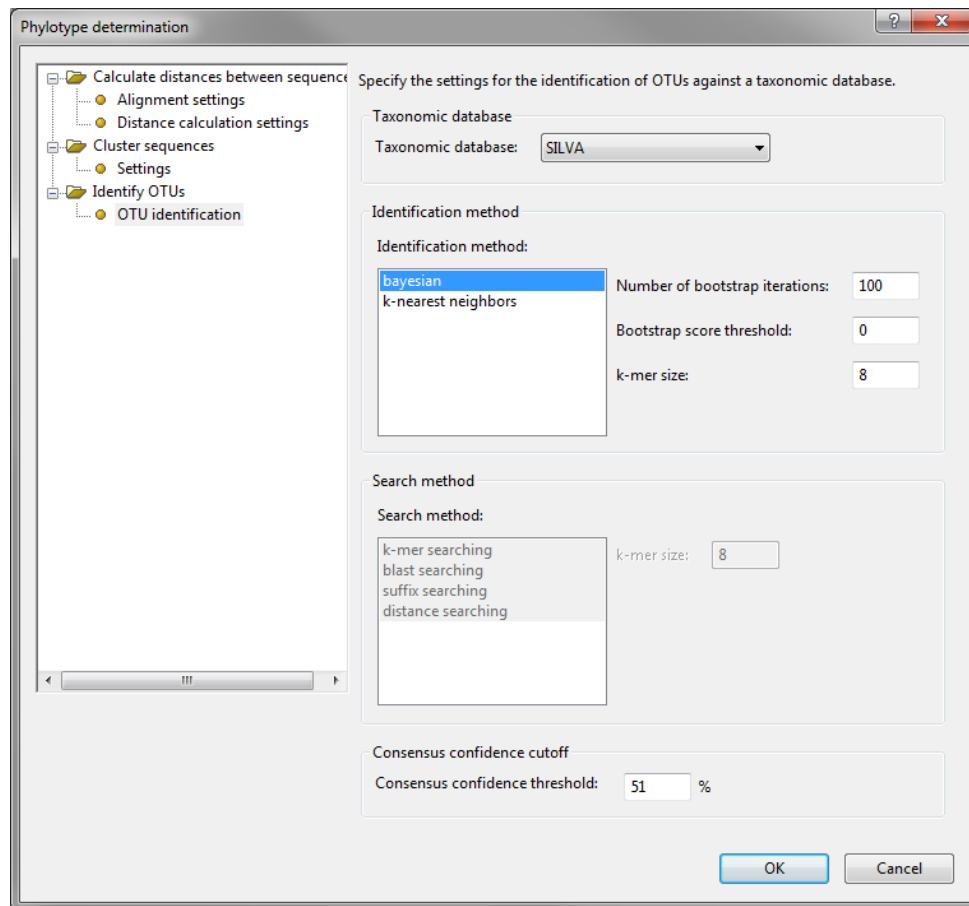


Figure 19.4.16: The *Phylotypic OTU preparation* dialog box: OTU identification settings.

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - **k-mer searching** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - **blast searching**: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

- The **Consensus confidence cutoff** is the minimum value to specify a consensus taxonomy on the OTUs. The default is 51%, which is the minimum cutoff value that can be set for this parameter.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.1.5.4 Save to character set

Once the OTUs have been taxonomically identified, the taxonomic assignments and abundances are saved to the database. The settings to define this export to the database character sets are given in the *Save to character set* dialog box.

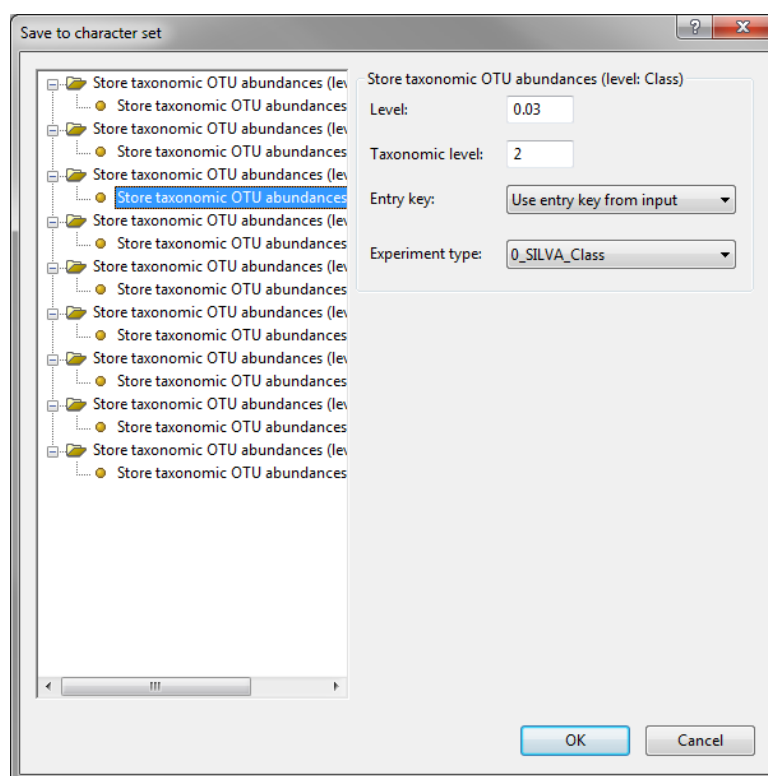


Figure 19.4.17: The *Save to character set* dialog box.

This element is needed to store a separate character set for each taxonomic level to the database. To store the data to the entries of choice, the following settings are queried for:

- **Level:** a specific sequence identity threshold will be used as cluster cutoff value to determine the OTUs. When the final sequence identity level is set to unique, the cutoff value is indicated as -1.
- **Taxonomic level:** the level assignment (e.g. Class) as indicated on top of the dialog refers to a taxonomic level defined in the taxonomic reference database (e.g. taxonomic level 2 containing Class information). By default, the taxonomic levels are displayed as imported from the taxonomic reference file and should not be altered upon export.
- **Entry key** information, indicating to which database entries the resulting data sets should be saved to. The different options are
 - Create new: use this option to create a new entry for each character experiment (i.e. taxonomic level abundances) saved to the database.

- Use entry keys used in previous run: will save the data to the same entry key as used when running the project before.
- Use entry key from input: will save the data to the same entry key as used during the import of the data. Use entry key from input is the default setting and should not be changed, as it allows to save the OTU abundances to the same entry as the original sequence read set used in the analysis.
- **Experiment type** information, indicating to which sequence character experiment type the data should be saved to. Default the experiment types names are composed of [unique number]_[name of reference alignment]_[taxonomic level name].

Press **<OK>** to close the dialog and update the settings in the project. Press **<Cancel>** to close the dialog without altering any of the project settings.

19.4.2 Single-sample diversity analysis

When starting the analysis from the *Main* window, one additional dialog page, compared to the dialog when starting from the sequence read set, is displayed where you can select the sequence read set experiment type that should be used for the analysis. All available sequence read set experiment types are listed in the drop down. Select the required sequence read set experiment type and select **<Next>** to proceed.

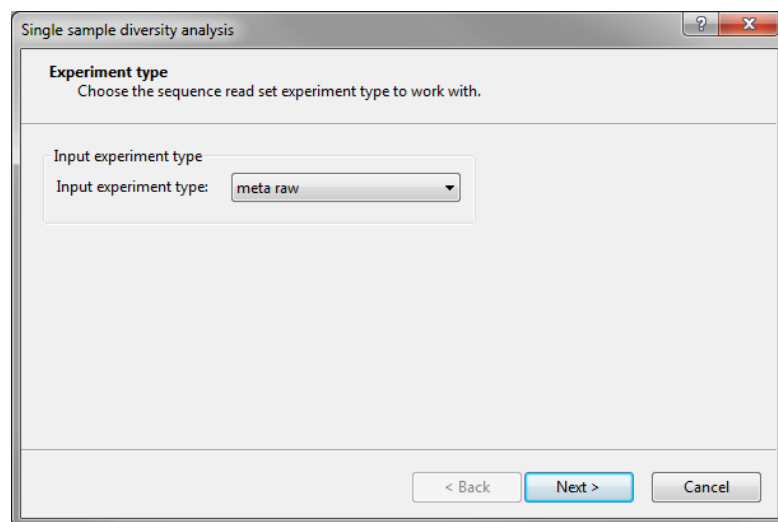


Figure 19.4.18: The *Single-sample diversity analysis* dialog box: Experiment type settings.

Similarly, when starting the analysis from within the *Metagenomics* window, another additional dialog page, compared to the dialog when starting from the sequence read set, is displayed. In this dialog, the entry information and the input sequence read set experiment type need to be specified. Press **<Add entry>** to select the entry to analyze. This opens the *Select entry* dialog box where the entry can be highlighted in the database list and press **<OK>** to return to the *Single sample diversity analysis* wizard where the sequence read set experiment type to be used for the analysis can be selected from the drop down list with available sequence read set experiment types present in the database. Select **<Next>** to proceed.

When starting the metagenomics analysis from the *Sequence read set experiment* window, the first page of the *Single sample diversity analysis* wizard asks for the OTU determination settings. When starting the analysis from the *Main* window or the *Metagenomics* window, this will be the second page of the wizard. The OTUs can be determined three ways. The first two options are similar to the ones described under *Identification against a taxonomic database* (see 9.5.6). Additionally, the third option will define OTUs based on sequence clustering and a similarity cutoff value on these sequence clustering results.

The screenshot shows a window titled "Single sample diversity analysis" with a subtitle "Entry and experiment type" and the instruction "Choose the Entry and the sequence read set experiment type to work with." Below this, there is a section for "Entry key" with a text input field and an "Add entry..." button. Another section for "Input experiment type" shows a dropdown menu currently set to "meta raw". At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Figure 19.4.19: The *Single sample diversity analysis* wizard: Entry and Experiment type settings.

The screenshot shows a window titled "Single sample diversity analysis" with a subtitle "OTU determination" and the instruction "Choose the way the operational taxonomic units are defined." The main text explains that OTUs can be determined in two ways: either from the sequence data itself (by performing a sequence clustering), or from the phylotypic levels in a taxonomic database, once the sequences have been identified. It also notes that in case the OTUs are determined by sequence clustering, a consensus taxonomy can be calculated for each OTU. Below this, there are three radio button options: "Determine OTUs by sequence clustering." (which is selected), "Determine OTUs by taxonomic identification.", and "Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs." At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Figure 19.4.20: The *Single sample diversity analysis* wizard: OTU determination settings.

- The option ***Determine OTUs by sequence clustering*** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. Based upon the OTUs defined from this cluster analysis, the diversity analyses are calculated.
- The option ***Determine OTUs by taxonomic identification*** will determine the unique sequences in the sequence read set, identify the sequences against the reference taxonomy and create OTUs from the taxonomic identification results. Based upon the taxonomically defined OTUs, the diversity analyses are calculated.
- The option ***Determine OTUs by sequence clustering, and perform taxonomic identification of the OTUs*** will determine the unique sequences in the sequence read set, calculate the distances between the reads and create a cluster analysis. For each of the reads in the cluster the taxonomic identification is calculated and based on these results, the cluster consensus taxonomy is defined. The OTUs derived from this consensus taxonomy are then used as input to calculate the diversity analyses.

Select one of these options and press <Next> to proceed.

When taxonomic identification is involved, the taxonomic identification settings require the taxonomic reference database to be selected from the drop down list, and the start and end levels of the taxonomic identification for the current analysis need to be specified. Select *<Next>* to proceed.



See [19.1.2](#) for more information on the taxonomic reference database.

The screenshot shows a dialog box titled "Single sample diversity analysis" with a subtitle "Taxonomic identification" and the instruction "Choose the parameters for the taxonomic identification procedure." The "Database:" dropdown menu is set to "SILVA". The "Use levels:" section shows "Domain" selected for the start and "Species" selected for the end, separated by the word "to". At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 19.4.21: The *Single sample diversity analysis* wizard: Taxonomic identification settings.

When OTU determination is based on sequence clustering, these settings need to be defined in the *Single sample diversity analysis* wizard: Sequence clustering options. From this dialog, the sequence identity threshold needs to be set. The sequence identity threshold is a cutoff value applied to the clustering of the reads. Using this threshold, only sequences with a small distance will be clustered together, and not the complete matrix. This may result in a significant speedup of the cluster analysis involved in the OTU preparation. Next to the threshold, the reference alignment needs to be selected from the drop down list. Select *<Next>* to proceed.



See [19.1.1](#) for more information on the reference alignment.

The screenshot shows a dialog box titled "Single sample diversity analysis" with a subtitle "Sequence clustering options" and the instruction "Choose the parameters for the sequence clustering procedure." The "Sequence identity threshold:" is set to "90" followed by a "%" symbol. The "Reference alignment:" dropdown menu is set to "SILVA". At the bottom, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

Figure 19.4.22: The *Single sample diversity analysis* wizard: Sequence clustering options.

When launching the single sample diversity analysis, one can specify the set of calculators that should be used in the analysis from the *Single sample diversity analysis* wizard: Diversity calculators settings. The options *Use default calculators* uses a set of calculators defined upon installation. The different calculators activated as default can be visualized by selecting the option *Use default calculators* and pressing the **<Custom calculators...>** button. This opens the *Custom Calculators* dialog box where the different summary diversity calculators, the collector curve calculators and the rarefaction curve calculators can be viewed.

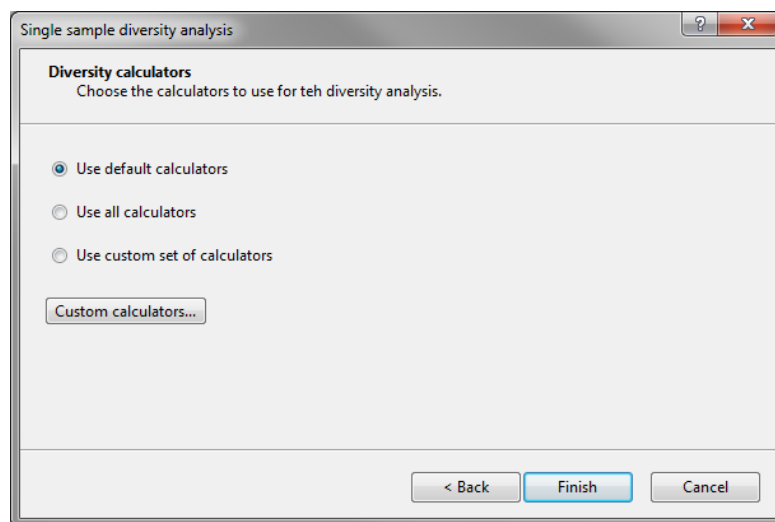


Figure 19.4.23: The *Single sample diversity analysis* wizard: Diversity calculators settings.

When the default selection of calculators is not satisfactory, one should choose the option *Use custom set of calculators*. In this case, opening the *Custom Calculators* dialog box by pressing **<Custom calculators...>** allows to select/unselect a custom set of summary diversity, collector curve and the rarefaction curve calculators that will be used in the current analysis.

A third option is to *Use all calculators*. When checking this option, all calculators displayed in the *Custom Calculators* dialog box are calculated in the analysis at hand. Press **<Finish>** to complete the dialog and create the analysis in the *Metagenomics* window.

When calculating a single sample diversity analysis, the different summary diversity calculators, the collector curve calculators and the rarefaction curve calculators can be viewed from the *Custom Calculators* dialog box. Initially, the selection shows the default activated calculators. When using a custom set of calculators, these calculators need to be defined from the same dialog by (un-)selecting the check boxes in front of the different calculators. Press **<OK>** to update the calculator selection in the analysis or press **<Cancel>** to leave the dialog without making any modifications on the predefined calculators.

The single-sample diversity analysis consists of three different project elements:

- *Input sequences*: imports the selected sequence read set from the database and calculates a basic sequence summary.
- *OTU preparation*: filters the unique sequences from the sequence read set and calculates a cluster analysis on these reads. Based on a cluster cutoff value, OTUs are created and later identified against the taxonomic database.
- *Single-sample diversity analysis*: calculates the within-sample diversity indices, the within-sample collector curves and the rarefaction curves.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File > Element settings...** (⚙️).

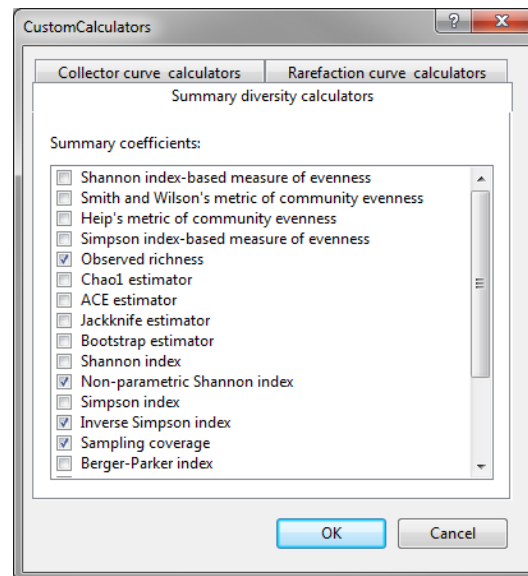


Figure 19.4.24: The *Custom Calculators* dialog box.

19.4.2.1 Input sequences

The *Input sequences* dialog box is discussed in [19.3.1.1.1](#).

19.4.2.2 OTU determination by sequence clustering

The OTU preparation aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, the single-sample diversity can be analyzed.

The first step in the analysis is to align the reads from the sequence read set to the reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,
2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *OTU preparation* dialog box, the *Alignment settings* can be modified.

- The **Reference alignment** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See [19.1.1](#) for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.

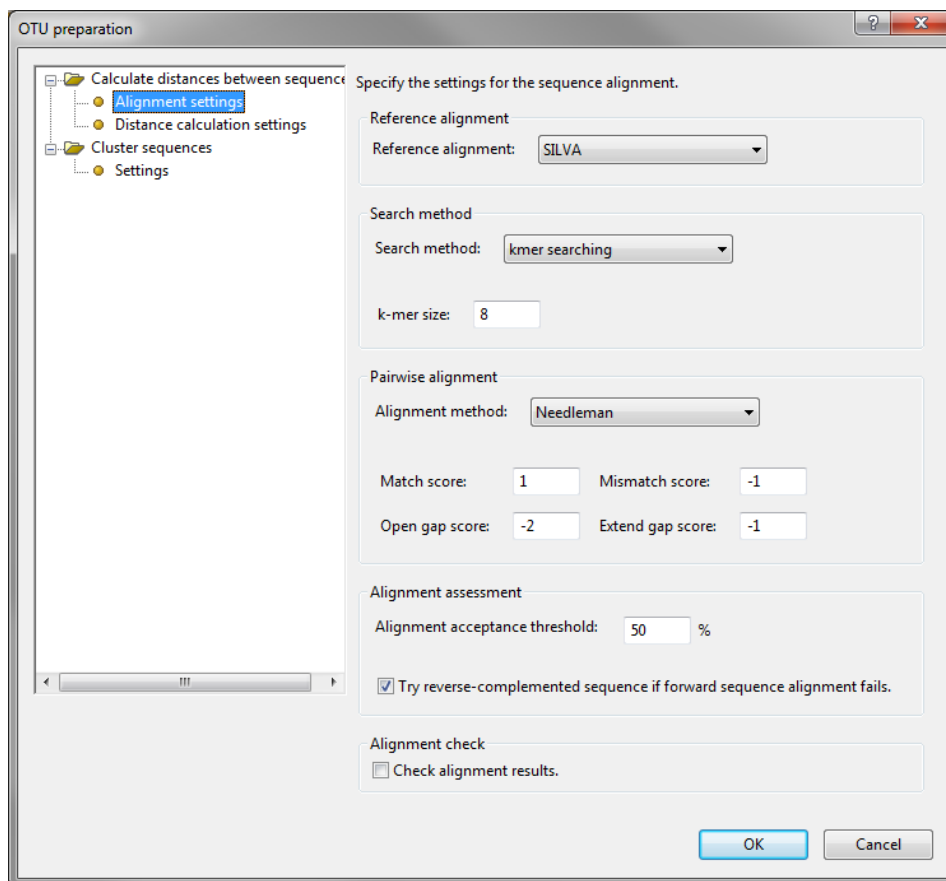


Figure 19.4.25: The *OTU preparation* dialog box: Alignment settings.

- *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
 - *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Now that the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might

not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

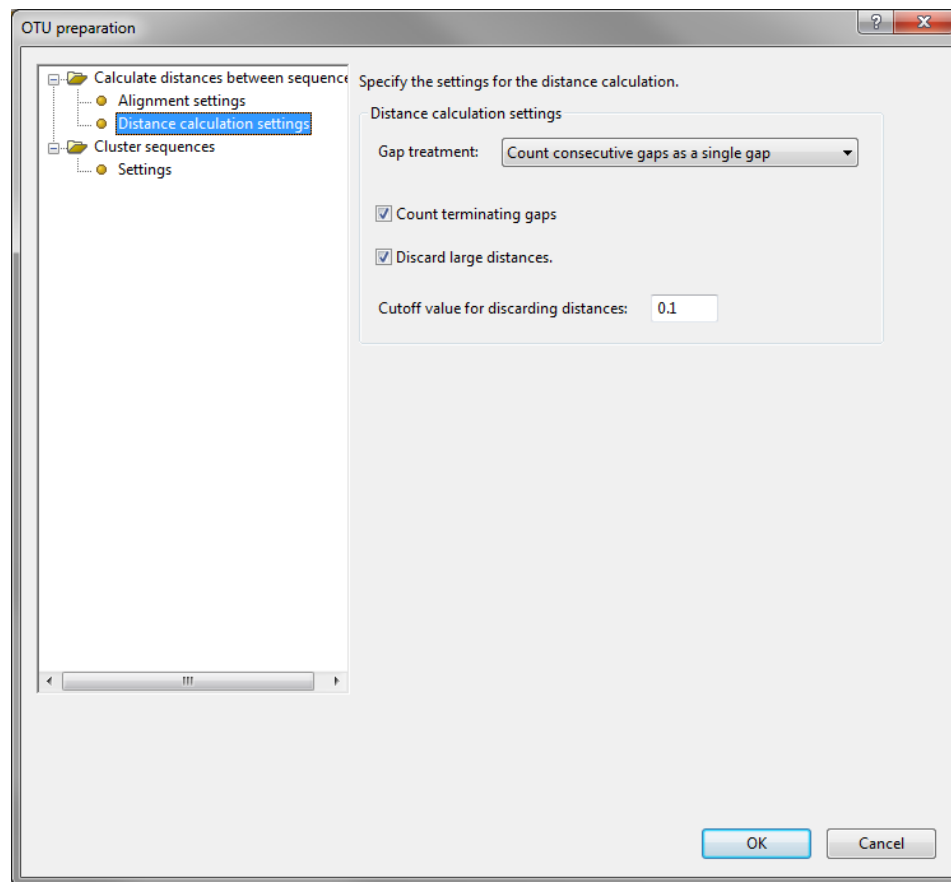


Figure 19.4.26: The *OTU preparation* dialog box: Distance calculation settings.

- The **Gap treatment** lists three different options on how to handle gap comparisons and terminal gaps.
 - **Count consecutive gaps as a single gap**: counts a string of gaps as a single gap i.e. the gap is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.
 - **Ignore gaps completely**: this distance calculation does not take into account any gaps or insertions.
 - **Count each gap individually**: will penalize each position of the gap or insert as a single mismatch.
- The option **Count terminating gaps** determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
- The option **Discard large distances** can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the **Cutoff value for discarding distances**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

Once a distance matrix is calculated, the sequence clustering can be started. The settings for the *Sequence clustering* include:

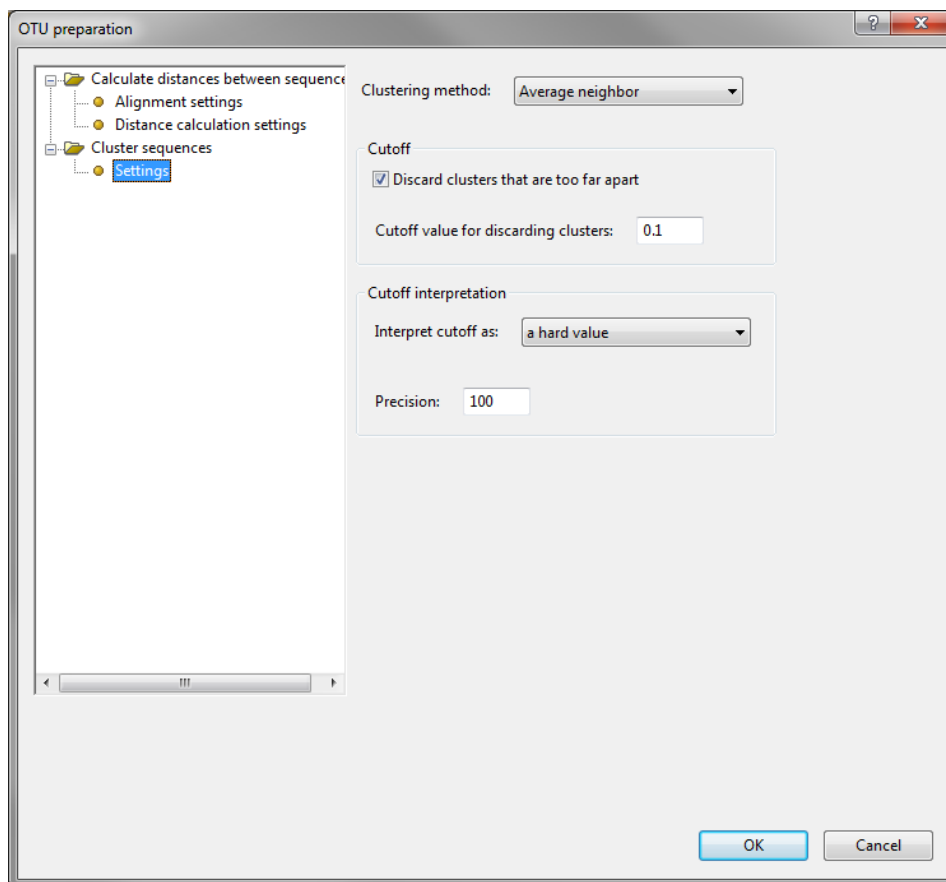


Figure 19.4.27: The *OTU preparation* dialog box: Cluster settings.

- The choice of **clustering method**. Four clustering methods are implemented:
 - **Average neighbor clustering:** this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.
 - **Nearest neighbor clustering:** each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
 - **Furthest neighbor clustering:** all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
 - **Weighted neighbor clustering:** this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.

- The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

19.4.2.3 OTU determination by taxonomic identification

The OTU preparation assigns the unique sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method. After identification of the unique reads, consensus OTUs are created from the taxonomic results.

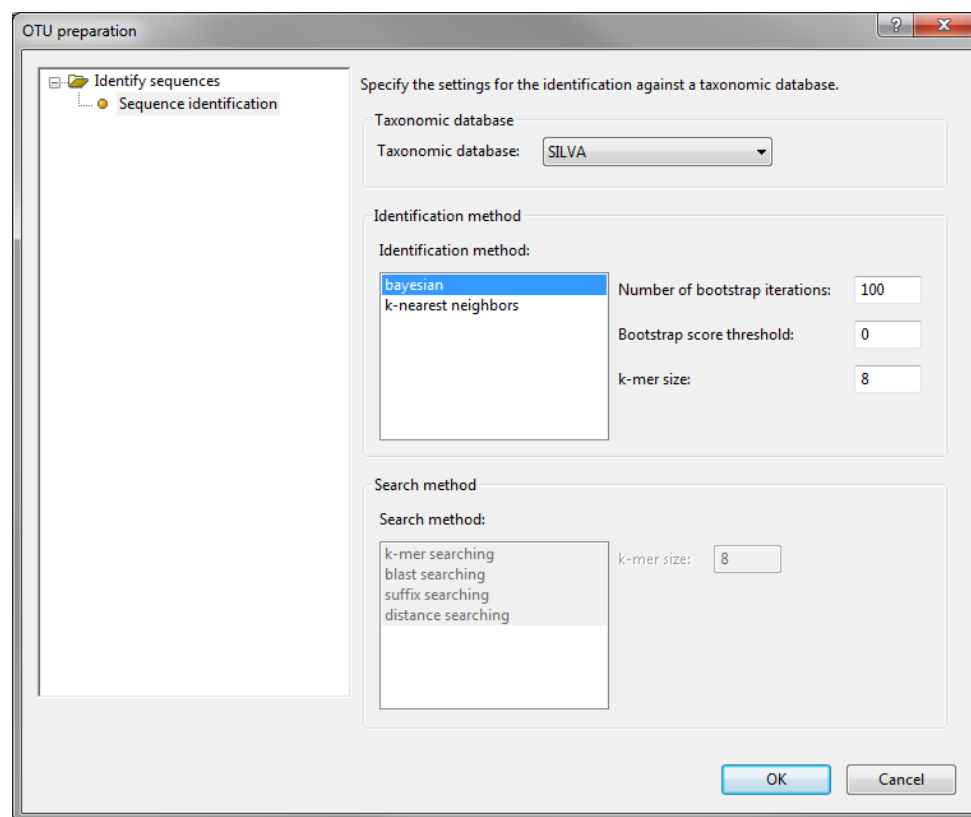


Figure 19.4.28: The *OTU preparation* dialog box.

In the *OTU preparation* dialog box, the settings for the read identification can be defined.

- First, the **Taxonomic database** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the **Identification method** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:

- **Number of bootstrap iterations:** identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
- **Bootstrap score threshold:** the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
- **k-mer size:** k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - **k-mer searching** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - **blast searching**: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.2.4 OTU determination by sequence clustering and taxonomic identification of the OTUs

The OTU preparation aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, they are taxonomically identified against a reference taxonomy of choice and subsequently, the single-sample diversity can be calculated.

The first step in the analysis is to align the unique reads from the sequence read set to the uploaded reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,

2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *OTU preparation* dialog box, the *Alignment settings* can be modified.

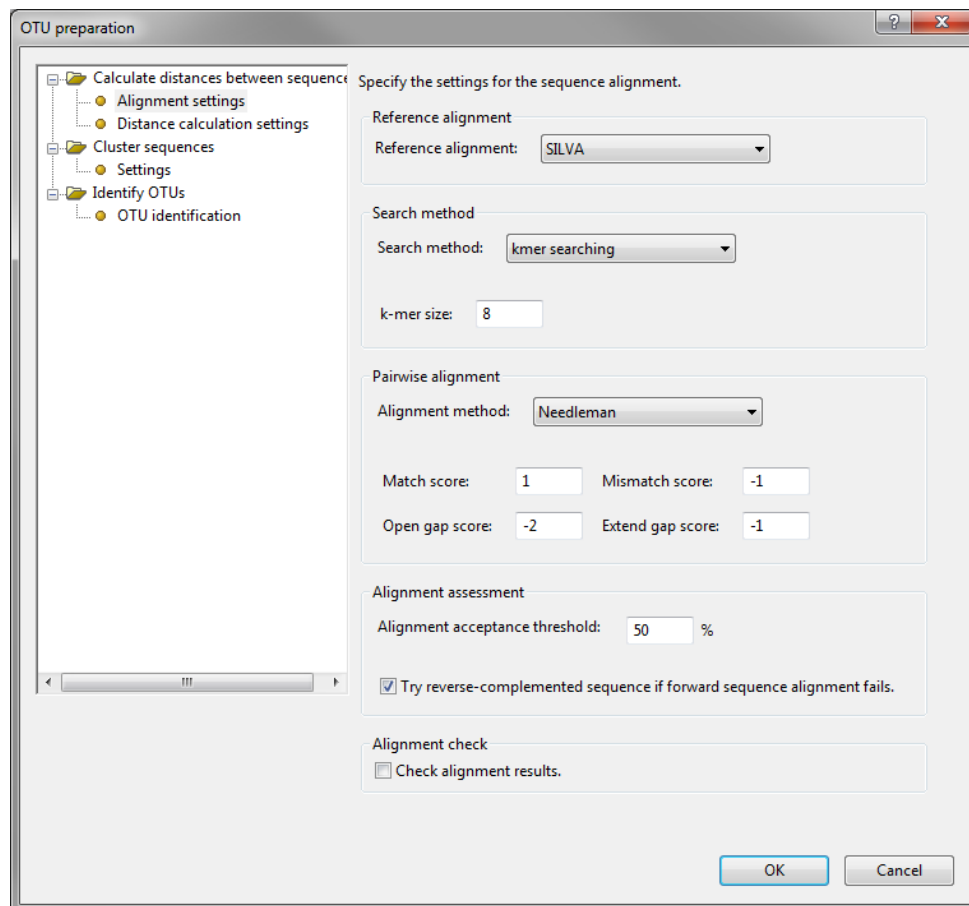


Figure 19.4.29: The *OTU preparation* dialog box: Alignment settings.

- The **Reference alignment** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See 19.1.1 for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:

- *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Once the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

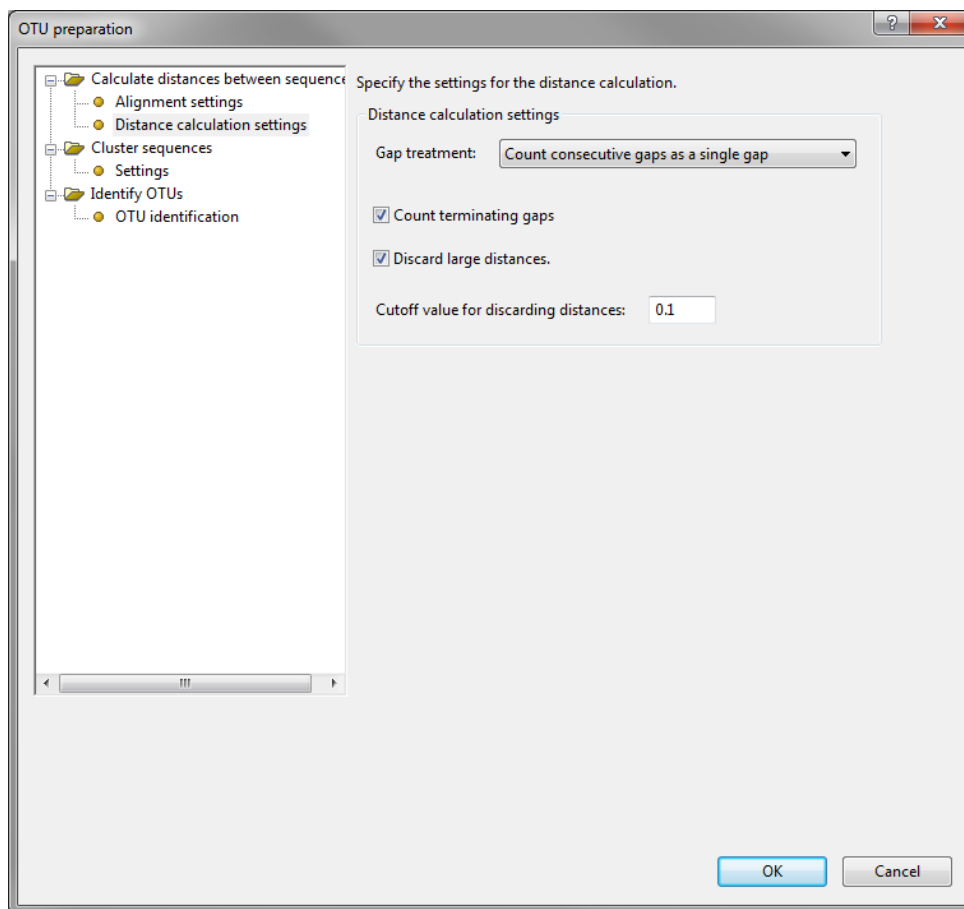


Figure 19.4.30: The *OTU preparation* dialog box: Distance calculation settings.

- The *Gap treatment* lists three different options on how to handle gap comparisons and terminal gaps.

- **Count consecutive gaps as a single gap**: counts a string of gaps as a single gap i.e. the gap is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.
 - **Ignore gaps completely**: this distance calculation does not take into account any gaps or insertions.
 - **Count each gap individually**: will penalize each position of the gap or insert as a single mismatch.
- The option **Count terminating gaps** determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
 - The option **Discard large distances** can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the **Cutoff value for discarding distances**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

Once a distance matrix is calculated, the sequence clustering can be started. The settings for the *Sequence clustering* include:

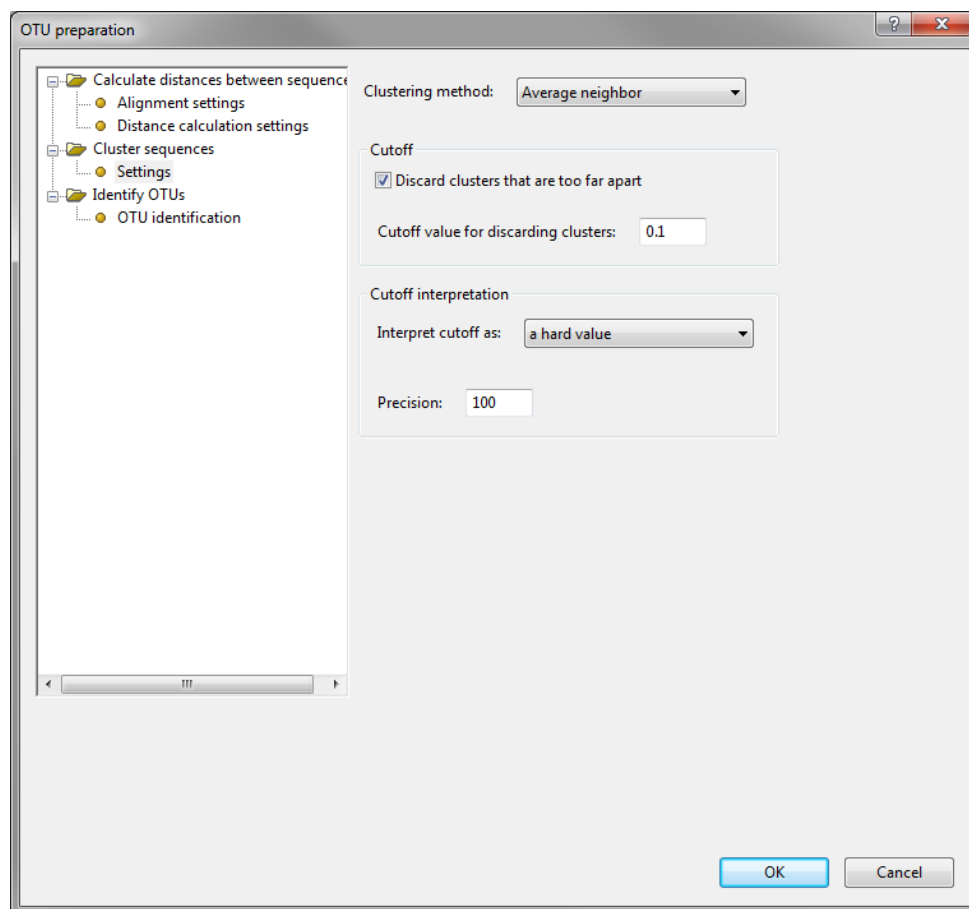


Figure 19.4.31: The *OTU preparation* dialog box: Cluster settings.

- The choice of **clustering method**. Four clustering methods are implemented:

- **Average neighbor clustering:** this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.
 - **Nearest neighbor clustering:** each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
 - **Furthest neighbor clustering:** all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
 - **Weighted neighbor clustering:** this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.
 - The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

The identification of OTUs will assign the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

- First, the **Taxonomic database** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the **Identification method** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:
 - **Number of bootstrap iterations:** identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
 - **Bootstrap score threshold:** the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
 - **k-mer size:** k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

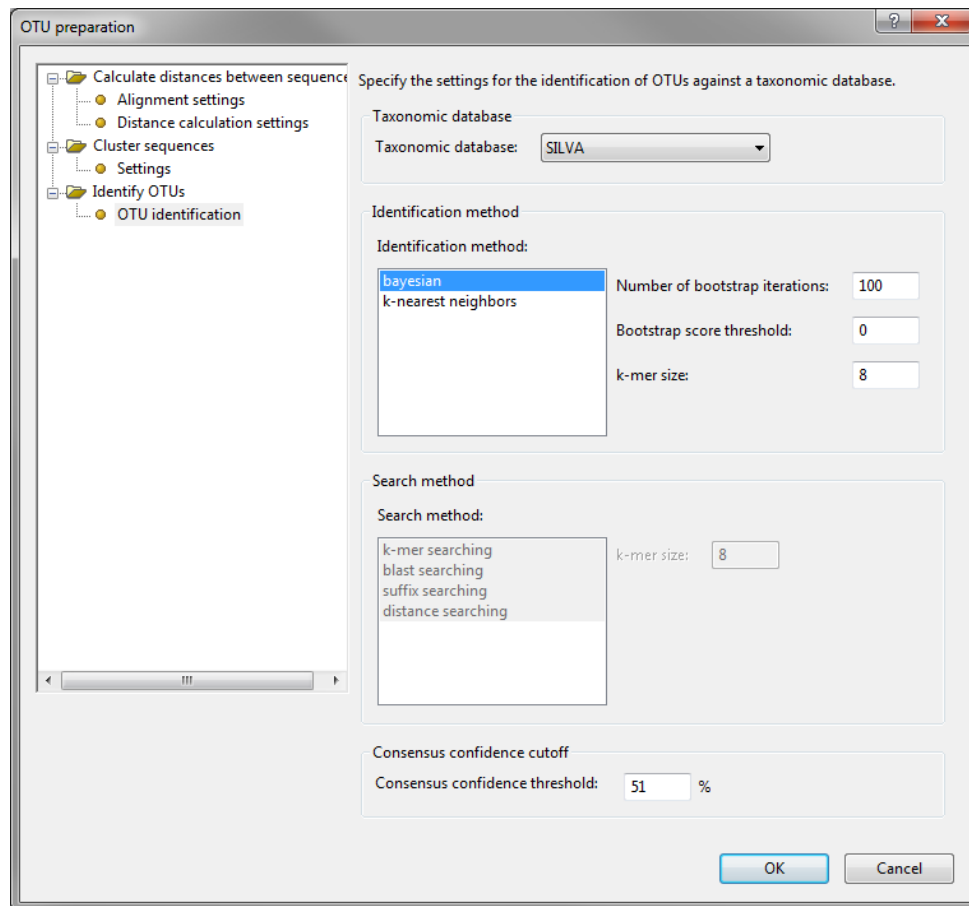


Figure 19.4.32: The *OTU preparation* dialog box: OTU identification settings.

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - **k-mer searching** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - **blast searching**: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - **suffix searching**: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - **distance searching**: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

- The **Consensus confidence cutoff** is the minimum value to specify a consensus taxonomy on the OTUs. The default is 51%, which is the minimum cutoff value that can be set for this parameter.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.2.5 Single-sample diversity analysis

The *Single-sample diversity analysis* dialog box gives an overview of the different diversity indices, and the collector and rarefaction curves that will be calculated on the sample data. For each of these categories, the settings are defined and the different options for the calculators are listed. Changing the default calculators as defined upon the first calculation of the project can be done by selecting the type of diversity analysis, and simply (un-)checking the box in front of the calculator. Recalculating the project will then update the diversity analysis results. An overview of the available calculators and their settings is listed.

Calculation of the within-sample diversity indices includes calculation of the actual estimators, and if available, the 95% confidence intervals i.e. the value for the lower and upper bound on the interval.



Note that these diversity indices only make sense when the sampling depth was sufficient. If not, the diversity indices will be sensitive to sampling and cannot be trusted. The influence of sampling depth on diversity indices can be checked by looking at the collector's curves (see further).

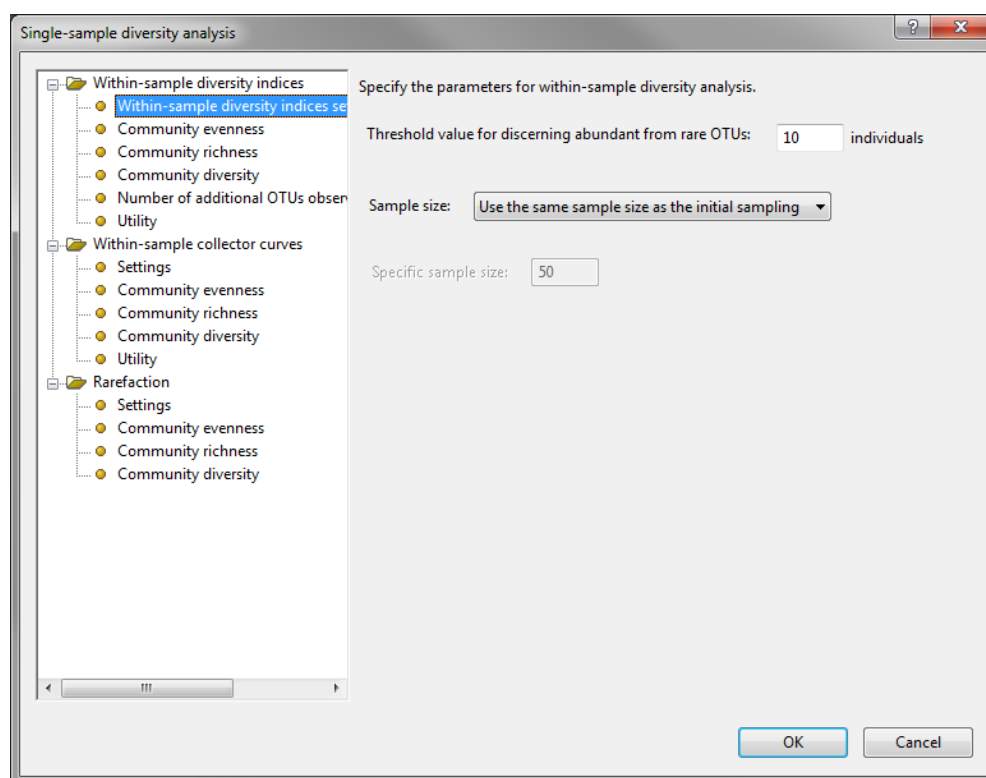


Figure 19.4.33: The *Single-sample diversity analysis* dialog box: The settings for the within-sample diversity indices.

- The **Settings** for the *within-sample diversity indices* include:
 - The **threshold value for discerning abundant from rare OTUs** is default set at 10 individuals, meaning that an OTU is considered abundant when having more than 10 individual reads assigned to it.

- For calculating the number of additional OTUs observed with extra sampling, the sample size used in this calculation needs to be set. The sample size can be defined as *the same size as the initial sampling* or as a *custom sample size*. The value of this *specific sample size* can be fixed in the dialog box.

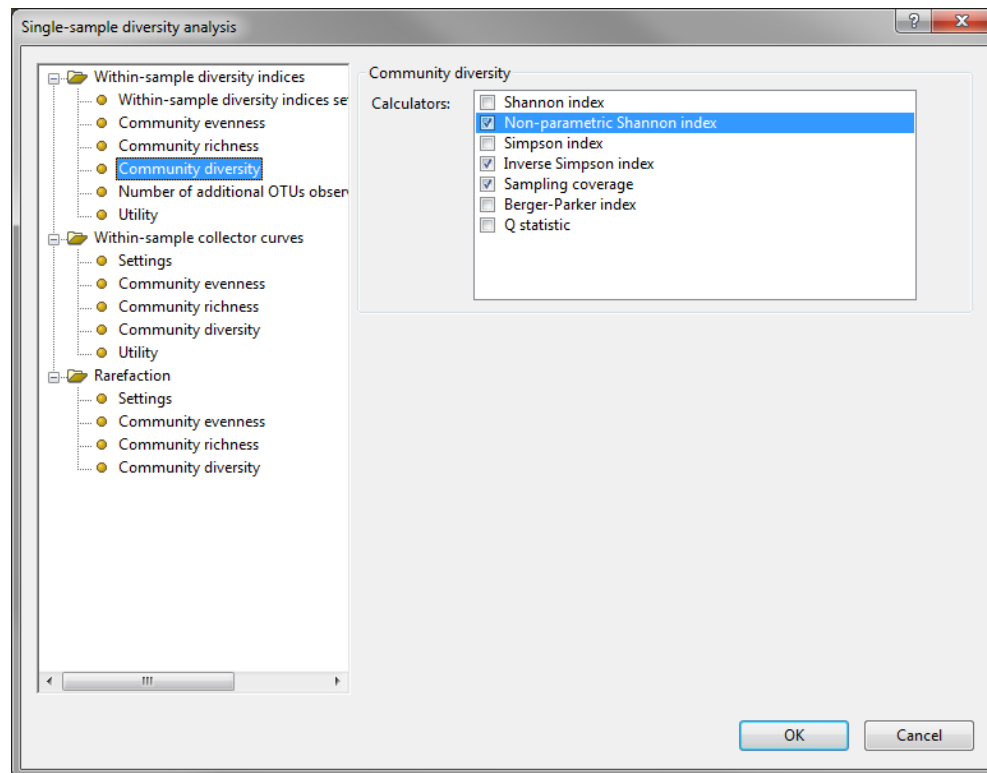


Figure 19.4.34: The *Single-sample diversity analysis* dialog box: The community diversity calculators.

- The *Community evenness* calculators include the *Shannon index-based measure of evenness*, the *Smith and Wilson's metric of community evenness*, the *Heip's metric of community evenness* and the *Simpson index-based measure of evenness*.
- The *Community richness* calculators include the *Observed richness*, the *Chao1 estimator*, the *ACE estimator*, the *Jackknife estimator* and the *Bootstrap estimator*.
- The *Community diversity* calculators include the *Shannon index*, the *Non-parametric Shannon index*, the *Simpson index*, the *Inverse Simpson index*, the *Sampling coverage*, the *Berger-Parker index* and the *Q statistic*.
- The *Number of additional OTUs observed with extra sampling* include the *Shen's estimator*, the *Boneh's estimator*, the *Efron's estimator* and the *Solow's estimator*.
- In the *Utility*, the *number of sequences in a sample* is calculated, resulting in the number of sequences that were sampled for each OTU definition.

Calculating the within-sample collector curves generates collector curves using calculators that describe the evenness, richness, diversity, and other features of individual samples. Collector curves describe how richness or diversity change as you sample additional individuals. If a collector curve becomes parallel to the x-axis, sampling depth has been deep enough and the final values of the evenness, richness and diversity calculators can be trusted. Otherwise, these values cannot be relied upon. When recalculating the

collector curves, the results might slightly differ. This is normal, as the calculation is each time performed on individuals being sampled in a randomized order.

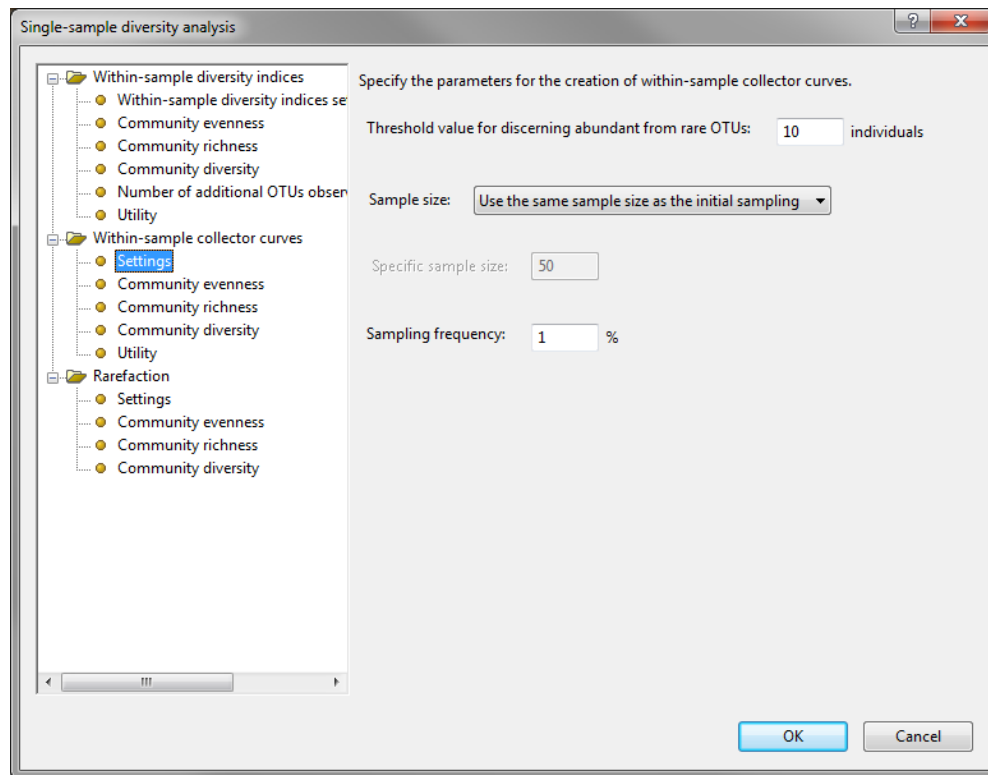


Figure 19.4.35: The *Single-sample diversity analysis* dialog box: The settings for the within-sample collector curves.

For the ***within-sample collector curves***, the same settings as for the within-sample diversity indices apply. One additional setting is the ***Sampling frequency*** for the collector curves. For large data sets i.e. 100,000 reads, it would be very time consuming and of limited information to have data calculated each 150 sequences. For that, the sampling frequency for the collector curves can be expressed as a percentage of the number of reads in the analysis. By default, 1% is entered, meaning 100 data points are calculated for each collector curve.

From the options under ***within-sample collector curves*** in the *Single-sample diversity analysis* dialog box, one can modify the diversity indices for which the collector curves will be calculated. An overview of the available indices on evenness, richness and diversity is listed under the settings for the within-sample diversity indices.

Rarefaction curves allow to compare the richness observed in different samples by displaying the number of OTUs, on average, that would have been observed when not having sampled as many individuals as present in the sample. The intra-sample rarefaction curve is calculated by a random sampling without replacement procedure. The rarefaction curve assesses the sampling intensity i.e. when the rarefaction curve becomes parallel to the x-axis, sampling depth is sufficient and the observed level of diversity and richness can be trusted.

The settings for the ***rarefaction curves*** include:

- The ***threshold value for discerning abundant from rare OTUs***, default set at 10 individuals, meaning that an OTU is considered abundant when having more than 10 individual reads assigned to it.
- The ***number of iterations*** used in the randomization process of the resampling. To improve the accuracy of the calculations the number of iterations can be increased, but this has major implications on the speed of the calculations.

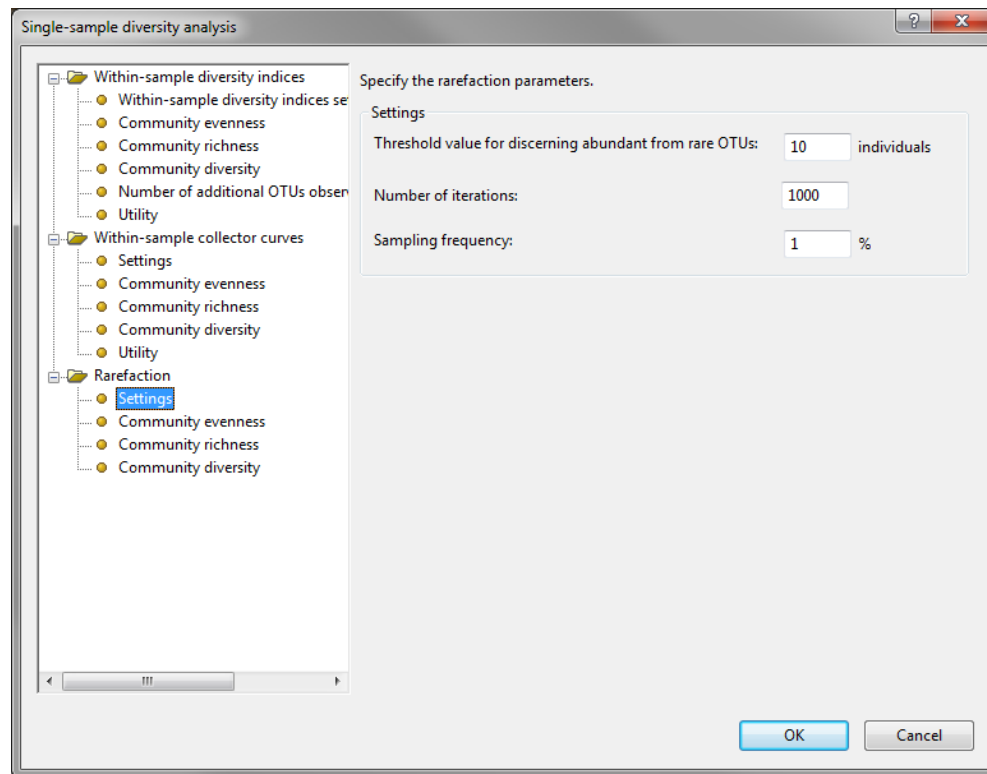


Figure 19.4.36: The *Single-sample diversity analysis* dialog box: The settings for the rarefaction analysis.

- The **sampling frequency** for the rarefaction curves, expressed as a percentage of the number of reads in the analysis.

From the options under **Rarefaction** in the *Single-sample diversity analysis* dialog box, one can modify the diversity indices for which the rarefaction curves will be calculated. An overview of the available indices on evenness, richness and diversity is listed under the settings for the within-sample diversity indices.

19.4.2.6 Single-sample diversity analysis : Project element settings

The single-sample diversity analysis consists of three different project elements:

- **Input sequences:** imports the selected sequence read set from the database and calculates a basic sequence summary.
- **OTU preparation:** filters the unique sequences from the sequence read set and calculates a cluster analysis on these reads. Based on a cluster cutoff value, OTUs are created and later identified against the taxonomic database.
- **Single-sample diversity analysis:** calculates the within-sample diversity indices, the within-sample collector curves and the rarefaction curves.

These project elements are further elaborated in this section. To visualize the element settings, select the element box from the *Project* panel and select **File > Element settings...** (🔧).

19.4.2.6.1 Input sequences

The *Input sequences* dialog box is discussed in [19.3.1.1.1](#).

19.4.2.6.2 OTU determination by sequence clustering

The OTU preparation aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, the single-sample diversity can be analyzed.

The first step in the analysis is to align the reads from the sequence read set to the reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,
2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *OTU preparation* dialog box, the *Alignment settings* can be modified.

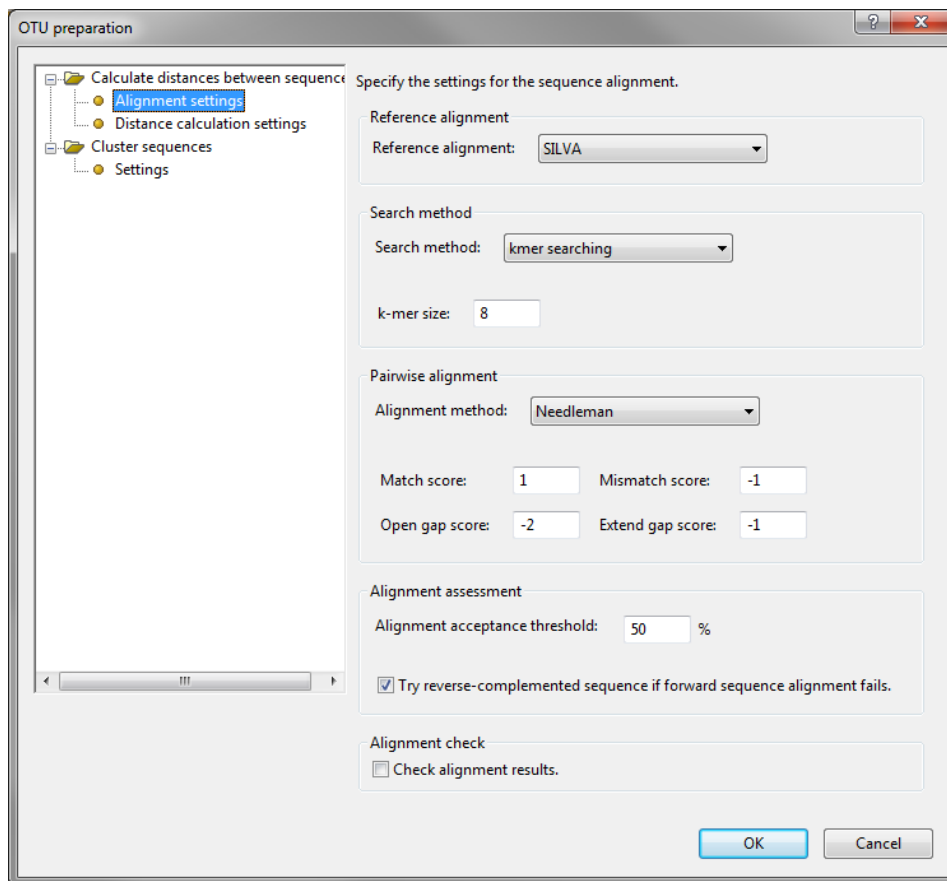


Figure 19.4.37: The *OTU preparation* dialog box: Alignment settings.

- The **Reference alignment** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See 19.1.1 for more information.
- There are three methods implemented to find the template sequence for each of the reads:

- *k-mer searching*: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
- *blast searching*: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
- *suffix searching*: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:
 - *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
 - *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Now that the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

- The *Gap treatment* lists three different options on how to handle gap comparisons and terminal gaps.
 - *Count consecutive gaps as a single gap*: counts a string of gaps as a single gap i.e. the gap is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.
 - *Ignore gaps completely*: this distance calculation does not take into account any gaps or insertions.
 - *Count each gap individually*: will penalize each position of the gap or insert as a single mismatch.
- The option *Count terminating gaps* determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
- The option *Discard large distances* can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the *Cutoff value for discarding distances*. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

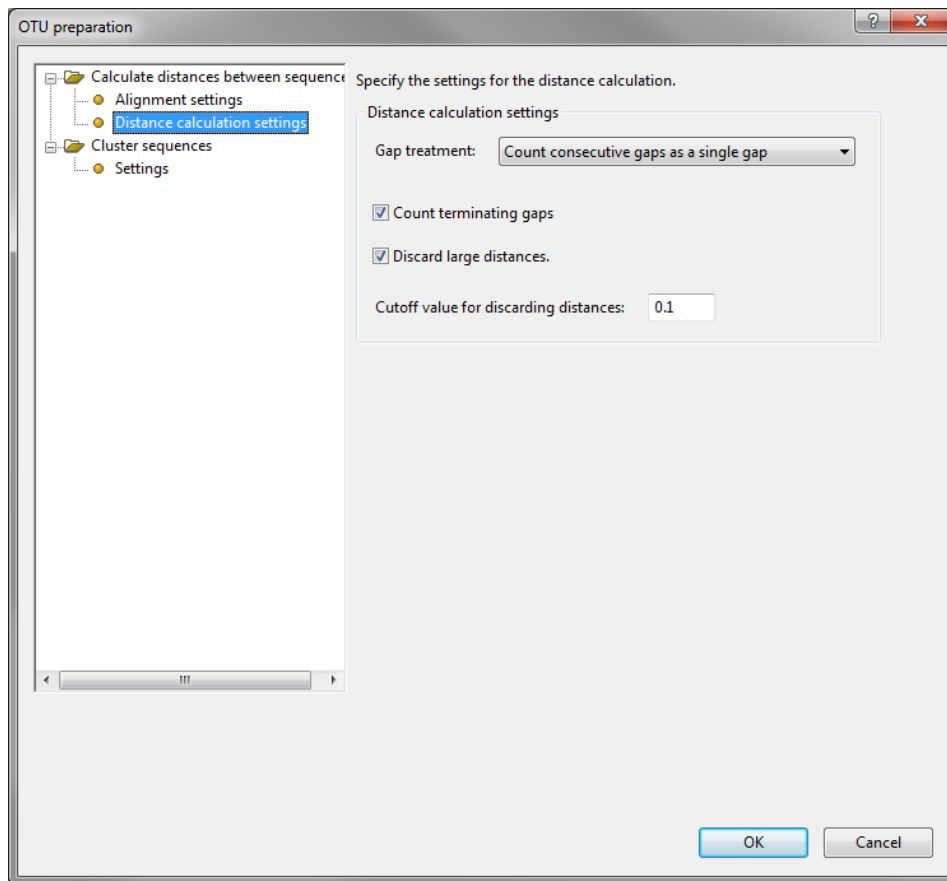


Figure 19.4.38: The *OTU preparation* dialog box: Distance calculation settings.

Once a distance matrix is calculated, the sequence clustering can be started. The settings for the *Sequence clustering* include:

- The choice of **clustering method**. Four clustering methods are implemented:
 - **Average neighbor clustering**: this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.
 - **Nearest neighbor clustering**: each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
 - **Furthest neighbor clustering**: all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
 - **Weighted neighbor clustering**: this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.

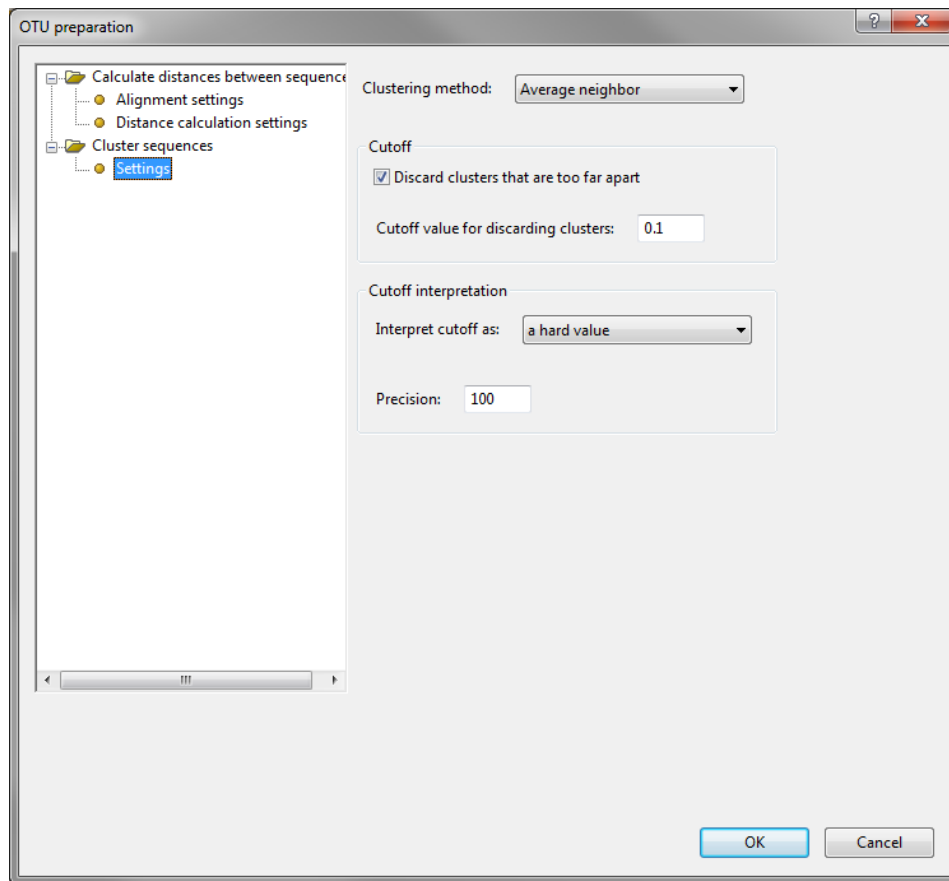


Figure 19.4.39: The *OTU preparation* dialog box: Cluster settings.

- The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

19.4.2.6.3 OTU determination by taxonomic identification

The OTU preparation assigns the unique sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method. After identification of the unique reads, consensus OTUs are created from the taxonomic results.

In the *OTU preparation* dialog box, the settings for the read identification can be defined.

- First, the **Taxonomic database** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the **Identification method** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to

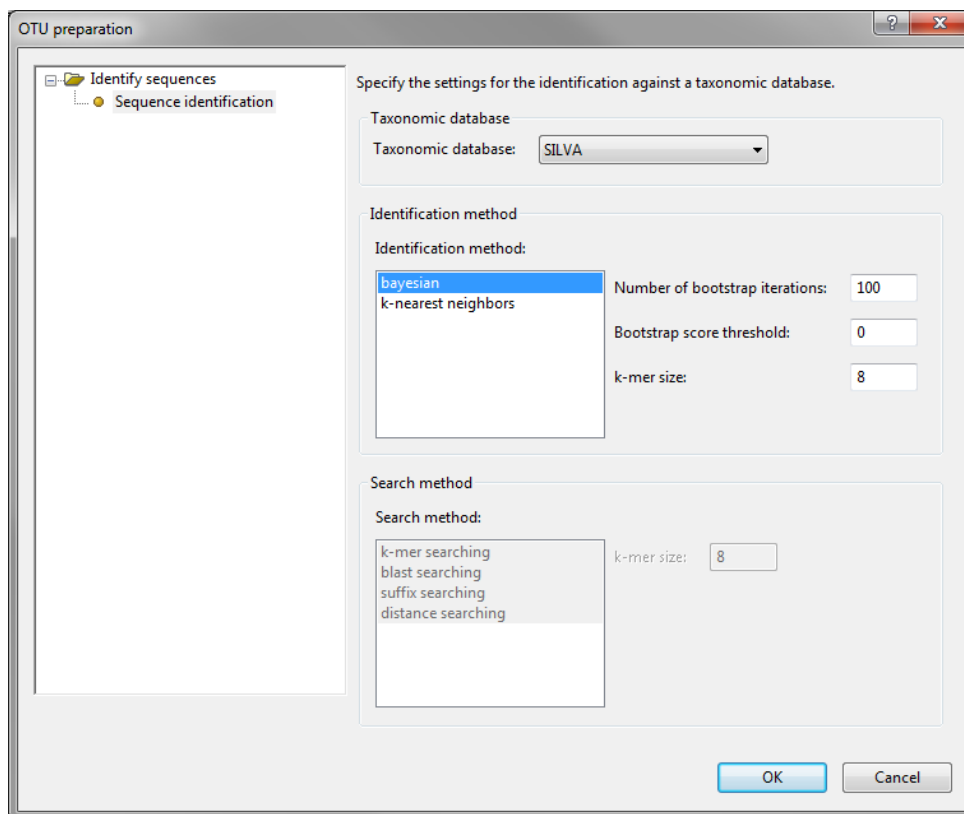


Figure 19.4.40: The *OTU preparation* dialog box.

find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:

- **Number of bootstrap iterations:** identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
- **Bootstrap score threshold:** the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).
- **k-mer size:** k-mer size to be used in screening the taxonomy and read sequences [5-12, default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- **Number of sequences to retain:** identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the **Search method** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.

- ***k-mer searching*** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
- ***blast searching***: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
- ***suffix searching***: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- ***distance searching***: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.

Press <**OK**> to close the dialog and update the settings in the project. Press <**Cancel**> to close the dialog without altering any of the project settings.

19.4.2.6.4 OTU determination by sequence clustering and taxonomic identification of the OTUs

The OTU preparation aligns the sequence reads into the reference alignment and, based on this alignment, sequence distances are calculated. From the distance matrix, different clustering methods can be applied to obtain a sequence clustering of the complete sequence read set. Once the clusters (OTUs) are defined, they are taxonomically identified against a reference taxonomy of choice and subsequently, the single-sample diversity can be calculated.

The first step in the analysis is to align the unique reads from the sequence read set to the uploaded reference alignment. The general approach is to

1. find the closest template for each read using k-mer searching, blastn, or suffix tree searching,
2. make a pairwise alignment between the read and the de-gapped template sequences using the Needleman-Wunsch or Gotoh algorithm, and
3. re-insert gaps to the read and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment.

From the *OTU preparation* dialog box, the *Alignment settings* can be modified.

- The ***Reference alignment*** can be selected from the drop down box. This reference alignment is a template alignment used to put the read sequences in the same alignment frame. See [19.1.1](#) for more information.
- There are three methods implemented to find the template sequence for each of the reads:
 - ***k-mer searching***: appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - ***blast searching***: uses traditional blast analysis to search for the template sequences. No further parameters are queried for this option.
 - ***suffix searching***: uses a suffix tree to search for the template sequences. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
- After the closest template sequence is found for each of the unique reads, a *Pairwise alignment* is calculated between each of the reads and its de-gapped template sequence. Different alignment methods are available:

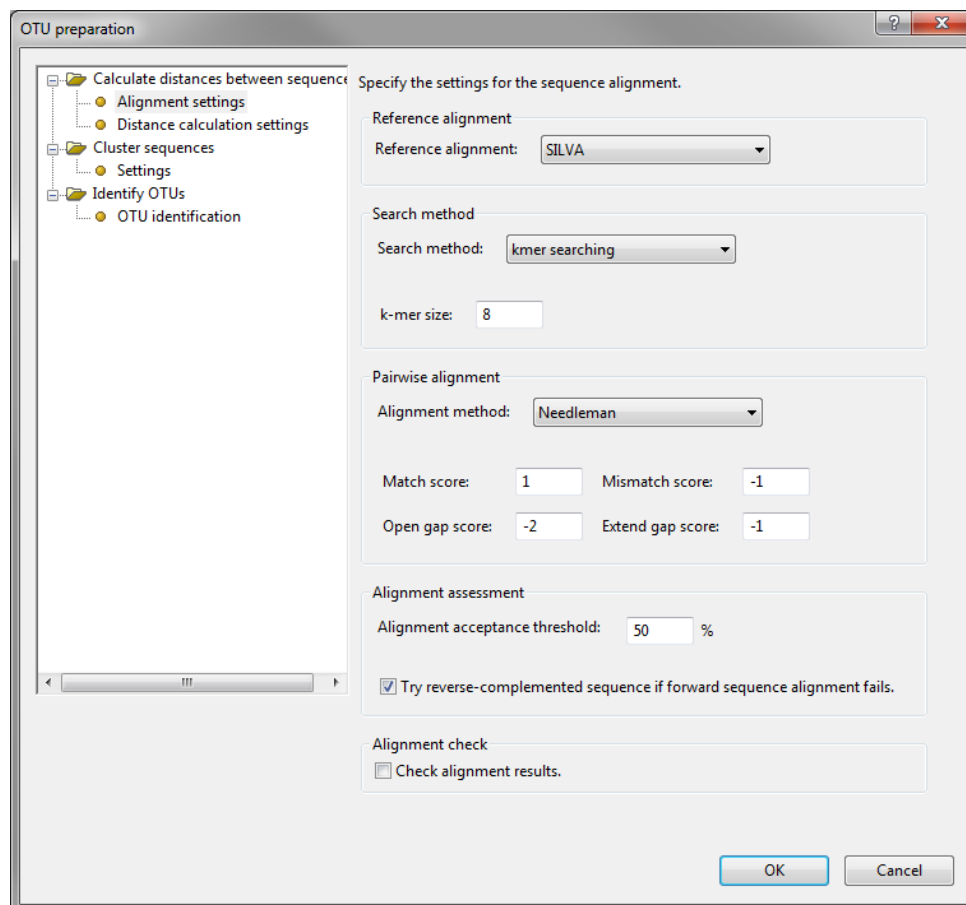


Figure 19.4.41: The *OTU preparation* dialog box: Alignment settings.

- *Needleman-Wunsch*: global alignment algorithm for which match, mismatch, open gap and extend gap scores can be modified. Default values are 1, -1, -2 and -1, respectively. The Needleman-Wunsch method penalizes the same amount for opening and extending a gap.
- *Gotoh*: this alignment algorithm is comparable to Needleman-Wunsch, as it is a global alignment algorithm for which match, mismatch and open gap scores can be modified. Default values are 1, -1 and -2, respectively. In contrast to the Needleman-Wunsch method, the Gotoh method penalizes a different amount for opening and extending a gap, -2 and -1, respectively. Compared to the Needleman-Wunsch algorithm, the Gotoh algorithm increases the calculation time but does not result in an improved pairwise alignment.
- Once the alignment is calculated, it will be assessed by using the *Alignment assessment* parameters. The *Alignment acceptance threshold* indicates when the reverse complement of the read may be tried to be aligned in case of a low quality alignment of the original read. The default threshold is set at 50%, meaning the reverse complement will be tried if similarity of candidate to template is less than 50

Once the sequence alignment is available, the uncorrected pairwise distances between the aligned DNA sequences can be calculated. During this calculation, it is possible to ignore "large" distances that one might not be interested in and this way, save calculation time and disk space. Specific settings can be defined in the *Distance calculation settings*.

- The *Gap treatment* lists three different options on how to handle gap comparisons and terminal gaps.
 - *Count consecutive gaps as a single gap*: counts a string of gaps as a single gap i.e. the gap

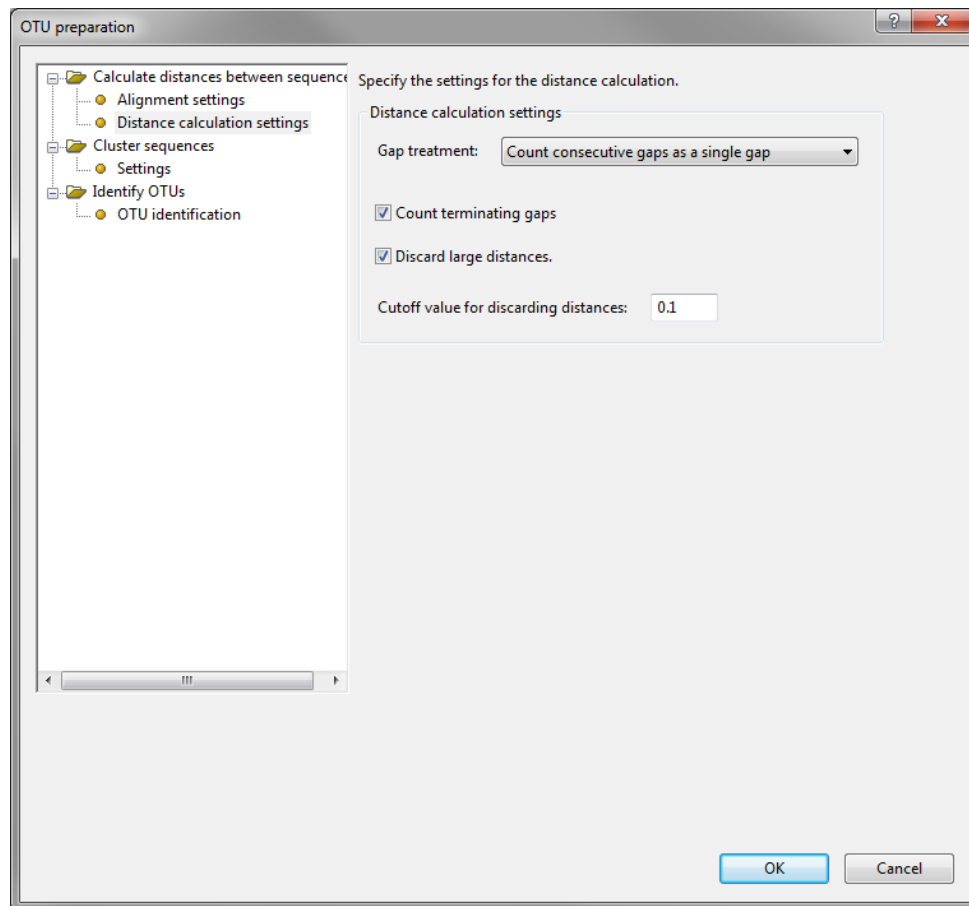


Figure 19.4.42: The *OTU preparation* dialog box: Distance calculation settings.

is considered as one position. The logic behind this type of penalty is that a gap represents an insertion and it is likely that a gap of any length represents only a single insertion.

- ***Ignore gaps completely***: this distance calculation does not take into account any gaps or insertions.
- ***Count each gap individually***: will penalize each position of the gap or insert as a single mismatch.
- The option ***Count terminating gaps*** determines whether gaps that occur at the end of sequences are penalized. This option is default checked, meaning that if all reads were aligned over the same region, the gaps at the end will be penalized the same way as gaps within the reads. Uncheck this option to ignore the penalization of end gaps.
- The option ***Discard large distances*** can be used when knowing in advance that the OTUs with distances larger than the threshold value will not be taken into account for the identification analysis. This option allows to significantly cut down the amount of hard drive space required to store the distance matrix. The threshold can be defined in the ***Cutoff value for discarding distances***. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 (i.e. 90% sequence similarity) will be saved to the distance matrix.

Once a distance matrix is calculated, the sequence clustering can be started. The settings for the *Sequence clustering* include:

- The choice of ***clustering method***. Four clustering methods are implemented:

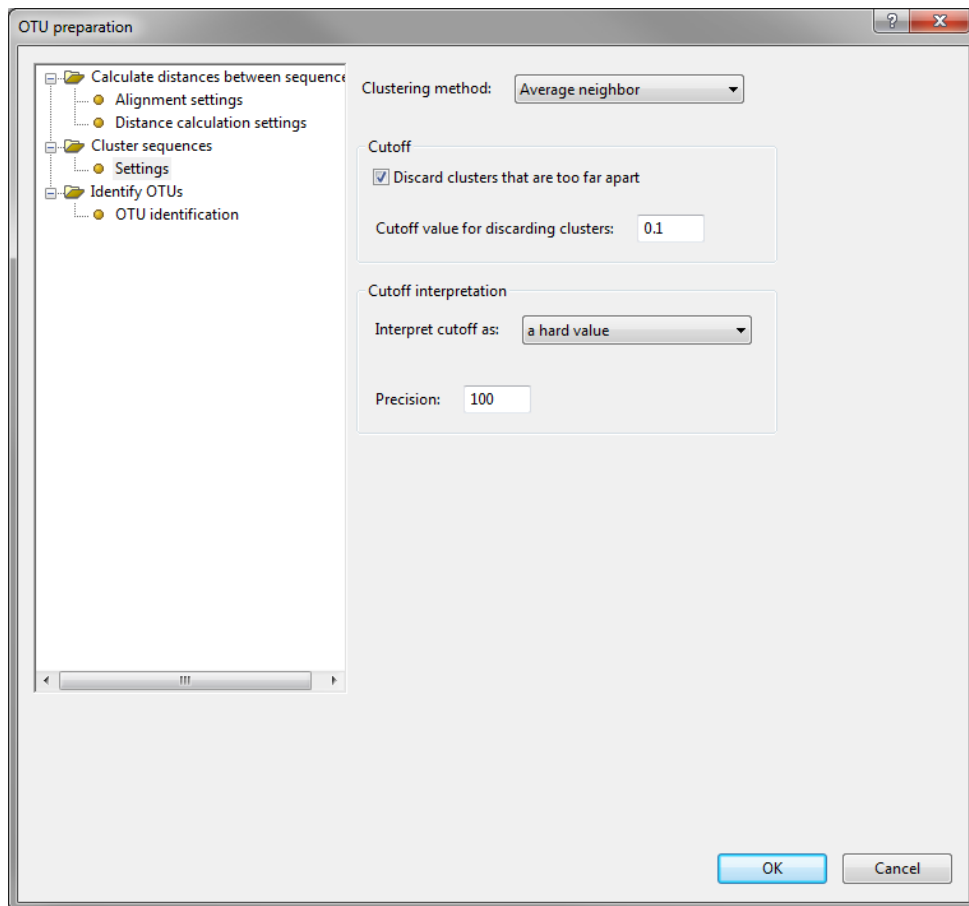


Figure 19.4.43: The *OTU preparation* dialog box: Cluster settings.

- **Average neighbor clustering:** this method is an intermediate between the nearest and furthest neighbor clustering and takes into account the weighted average distances between clusters.
 - **Nearest neighbor clustering:** each of the sequences within an OTU are at most x% distant from the most similar sequence in the OTU. This method is also known as single linkage clustering.
 - **Furthest neighbor clustering:** all of the sequences within an OTU are at most x% distant from all of the other sequences within the OTU. This method is also known as complete linkage clustering.
 - **Weighted neighbor clustering:** this method takes the distance between the centroids of pairs of clusters into account.
- To perform the clustering calculation, a **Cutoff** can be set to reduce the calculation time. Check the option **Discard clusters that are too far apart** when the cutoff should be applied and enter the cutoff value in the **Cutoff value for discarding clusters**. Default, a cutoff value of 0.1 is used, meaning that no distances larger than 0.1 will be taken into account when calculating the cluster analysis. When a cutoff value is used in the clustering, the **Cutoff interpretation** can also be defined. Two options are available:
 - The cutoff is interpreted as a **hard value**: uses the cutoff value as entered in the **Cutoff** settings.
 - The cutoff is interpreted as a **soft value**: the actual cutoff value is calculated by the formula:

$$\text{defined cutoff value} + \frac{5}{(10 \times \text{precision})}$$

with the precision being the value entered in the **precision** of the **Cutoff interpretation**.

The identification of OTUs will assigns the sequence reads to the taxonomy outline of choice. Current methods include a Bayesian classifier as described by Wang et al. [40] and a k-nearest neighbor consensus method.

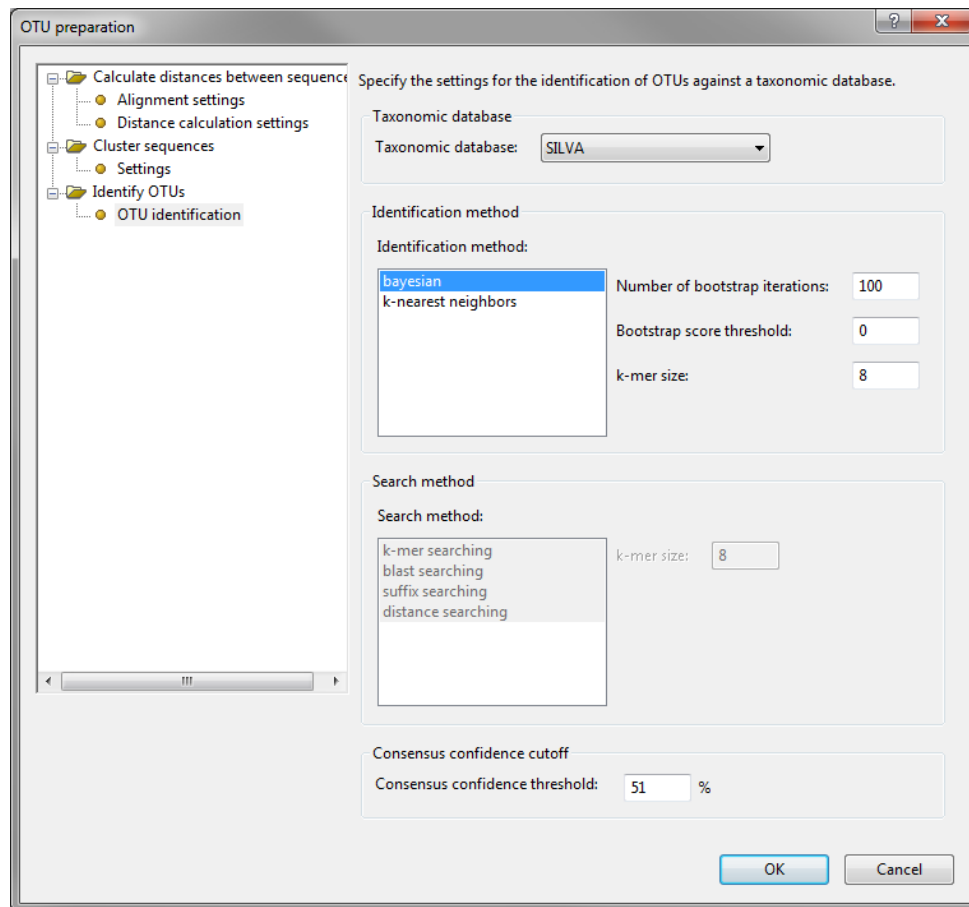


Figure 19.4.44: The *OTU preparation* dialog box: OTU identification settings.

- First, the ***Taxonomic database*** needs to be selected from the drop down box. This taxonomy database consists of a set of reference sequences with known taxonomic information. Reads are assigned to this taxonomic outline and, based on these results, are given a taxonomy label. See 19.1.2 for more information.
- Next, the ***Identification method*** can be selected. The Bayesian classifier[40] is selected by default. This method looks at all taxonomies represented in the taxonomic database, and calculates the probability a sequence from a given taxonomy would contain a specific kmer. Then it calculates the probability that a read would be in a given taxonomy based on the kmers it contains, and assign the read to the taxonomy with the highest probability. This method also runs a bootstrapping algorithm to find the confidence limit of the assignment by randomly choosing with replacement 1/8 of the kmers in the read and then finding the taxonomy. This is also the default method that is implemented in the RDP classifier pipeline. When selecting the Bayesian approach, the following settings are available:
 - ***Number of bootstrap iterations***: identifies how many iterations are performed when calculating the bootstrap confidence score on the obtained taxonomy [0-100, default 100].
 - ***Bootstrap score threshold***: the minimum bootstrap value to define a confident, classified taxonomy assignment [0-100, default 0]. Default, this threshold is set to 0, meaning that all taxonomic assignments are taken into account. When e.g. setting this threshold to 60, a taxonomic assignment with bootstrap score of 56 will remain 'unclassified' on the current level, but will get the

taxonomic identification of the higher taxonomic level (if bootstrap score on this level is above 60).

- ***k-mer size***: k-mer size to be used in screening the taxonomy and read sequences [5-12,default 8].

The second identification method that can be used, is the k-nearest neighbor algorithm. This approach will identify the k most similar sequences in the taxonomic database to the read sequence. For the k-nearest neighbor method, the following setting is available:

- ***Number of sequences to retain***: identifies how many most similar sequences are searched for in the taxonomic database [default 10].

Once the k most similar sequence are identified, the taxonomic information for each of the k sequences will be used to determine the consensus taxonomy of the OTUs.

- When selecting this k-nearest neighbor method, the method that is used to find the closest matches can also be determined from the ***Search method*** options. By default, the k-nearest neighbor approach searches for nearest neighbors by using the k-mer searching.
 - ***k-mer searching*** : appears to be superior in accuracy and speed when compared to the blast and suffix searching methods. The default k-mer size is 8.
 - ***blast searching***: uses traditional blast analysis to search for the nearest neighbors. When using blast, the Match score, Open gap score, Mismatch score and Extend gap score can be specified. The default scores are defined as 1 -2 -1 -1, respectively.
 - ***suffix searching***: uses a suffix tree to search for the nearest neighbors. Using the suffix tree search is comparable to blast in terms of quality and speed. No further parameters are queried for this option.
 - ***distance searching***: uses the distance from the reads to each of the taxonomic template sequences to determine the nearest neighbors. No further parameters are queried for this option.
- The ***Consensus confidence cutoff*** is the minimum value to specify a consensus taxonomy on the OTUs. The default is 51%, which is the minimum cutoff value that can be set for this parameter.

Press <OK> to close the dialog and update the settings in the project. Press <Cancel> to close the dialog without altering any of the project settings.

19.4.2.6.5 Single-sample diversity analysis

The *Single-sample diversity analysis* dialog box gives an overview of the different diversity indices, and the collector and rarefaction curves that will be calculated on the sample data. For each of these categories, the settings are defined and the different options for the calculators are listed. Changing the default calculators as defined upon the first calculation of the project can be done by selecting the type of diversity analysis, and simply (un-)checking the box in front of the calculator. Recalculating the project will then update the diversity analysis results. An overview of the available calculators and their settings is listed.

Calculation of the within-sample diversity indices includes calculation of the actual estimators, and if available, the 95% confidence intervals i.e. the value for the lower and upper bound on the interval.



Note that these diversity indices only make sense when the sampling depth was sufficient. If not, the diversity indices will be sensitive to sampling and cannot be trusted. The influence of sampling depth on diversity indices can be checked by looking at the collector's curves (see further).

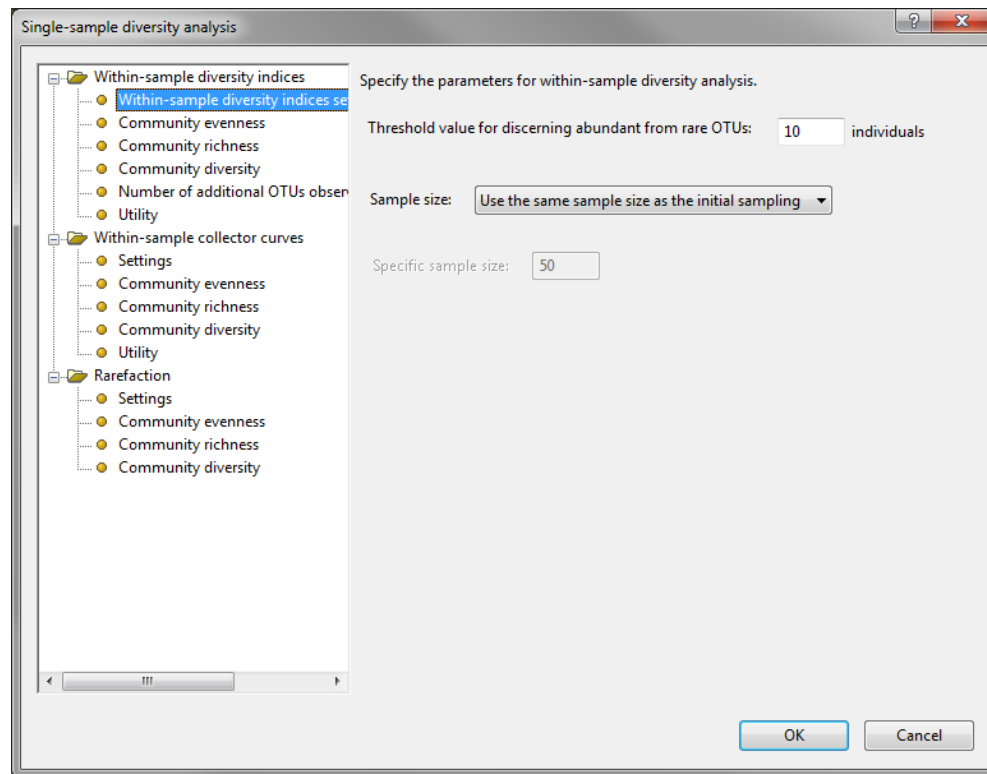


Figure 19.4.45: The *Single-sample diversity analysis* dialog box: The settings for the within-sample diversity indices.

- The *Settings* for the *within-sample diversity indices* include:
 - The *threshold value for discerning abundant from rare OTUs* is default set at 10 individuals, meaning that an OTU is considered abundant when having more than 10 individual reads assigned to it.
 - For calculating the number of additional OTUs observed with extra sampling, the sample size used in this calculation needs to be set. The sample size can be defined as *the same size as the initial sampling* or as a *custom sample size*. The value of this *specific sample size* can be fixed in the dialog box.
- The *Community evenness* calculators include the *Shannon index-based measure of evenness*, the *Smith and Wilson's metric of community evenness*, the *Heip's metric of community evenness* and the *Simpson index-based measure of evenness*.
- The *Community richness* calculators include the *Observed richness*, the *Chao1 estimator*, the *ACE estimator*, the *Jackknife estimator* and the *Bootstrap estimator*.
- The *Community diversity* calculators include the *Shannon index*, the *Non-parametric Shannon index*, the *Simpson index*, the *Inverse Simpson index*, the *Sampling coverage*, the *Berger-Parker index* and the *Q statistic*.
- The *Number of additional OTUs observed with extra sampling* include the *Shen's estimator*, the *Boneh's estimator*, the *Efron's estimator* and the *Solow's estimator*.
- In the *Utility*, the *number of sequences in a sample* is calculated, resulting in the number of sequences that were sampled for each OTU definition.

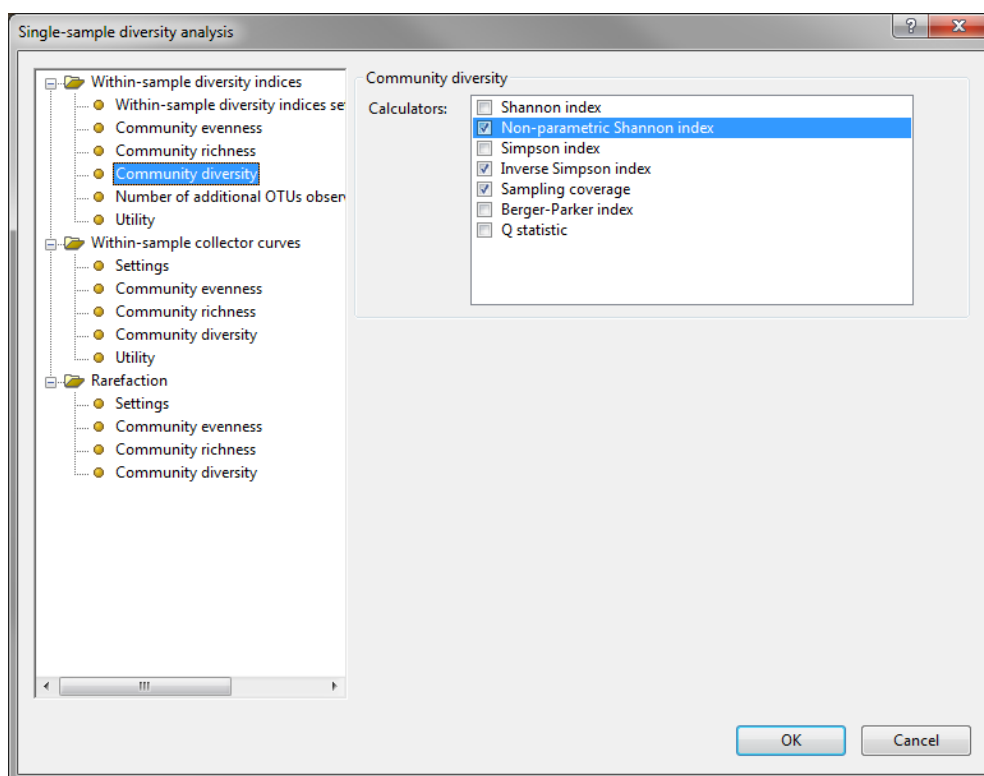


Figure 19.4.46: The *Single-sample diversity analysis* dialog box: The community diversity calculators.

Calculating the within-sample collector curves generates collector curves using calculators that describe the evenness, richness, diversity, and other features of individual samples. Collector curves describe how richness or diversity change as you sample additional individuals. If a collector curve becomes parallel to the x-axis, sampling depth has been deep enough and the final values of the evenness, richness and diversity calculators can be trusted. Otherwise, these values cannot be relied upon. When recalculating the collector curves, the results might slightly differ. This is normal, as the calculation is each time performed on individuals being sampled in a randomized order.

For the *within-sample collector curves*, the same settings as for the within-sample diversity indices apply. One additional setting is the *Sampling frequency* for the collector curves. For large data sets i.e. 100,000 reads, it would be very time consuming and of limited information to have data calculated each 150 sequences. For that, the sampling frequency for the collector curves can be expressed as a percentage of the number of reads in the analysis. By default, 1% is entered, meaning 100 data points are calculated for each collector curve.

From the options under *within-sample collector curves* in the *Single-sample diversity analysis* dialog box, one can modify the diversity indices for which the collector curves will be calculated. An overview of the available indices on evenness, richness and diversity is listed under the settings for the within-sample diversity indices.

Rarefaction curves allow to compare the richness observed in different samples by displaying the number of OTUs, on average, that would have been observed when not having sampled as many individuals as present in the sample. The intra-sample rarefaction curve is calculated by a random sampling without replacement procedure. The rarefaction curve assesses the sampling intensity i.e. when the rarefaction curve becomes parallel to the x-axis, sampling depth is sufficient and the observed level of diversity and richness can be trusted.

The settings for the *rarefaction curves* include:

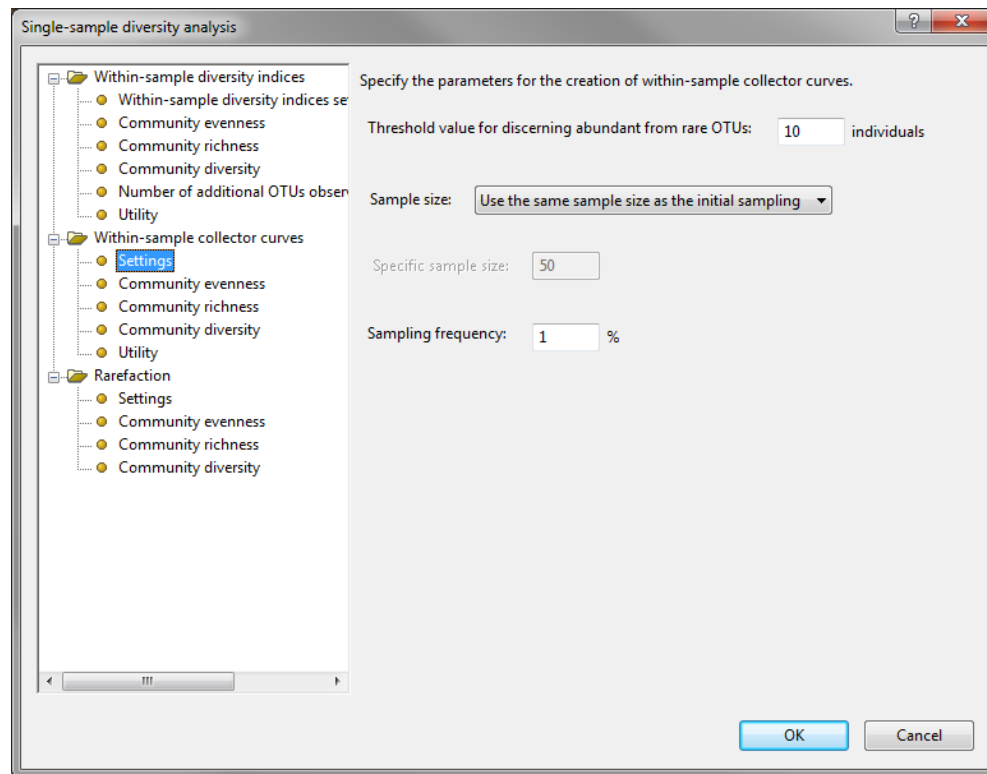


Figure 19.4.47: The *Single-sample diversity analysis* dialog box: The settings for the within-sample collector curves.

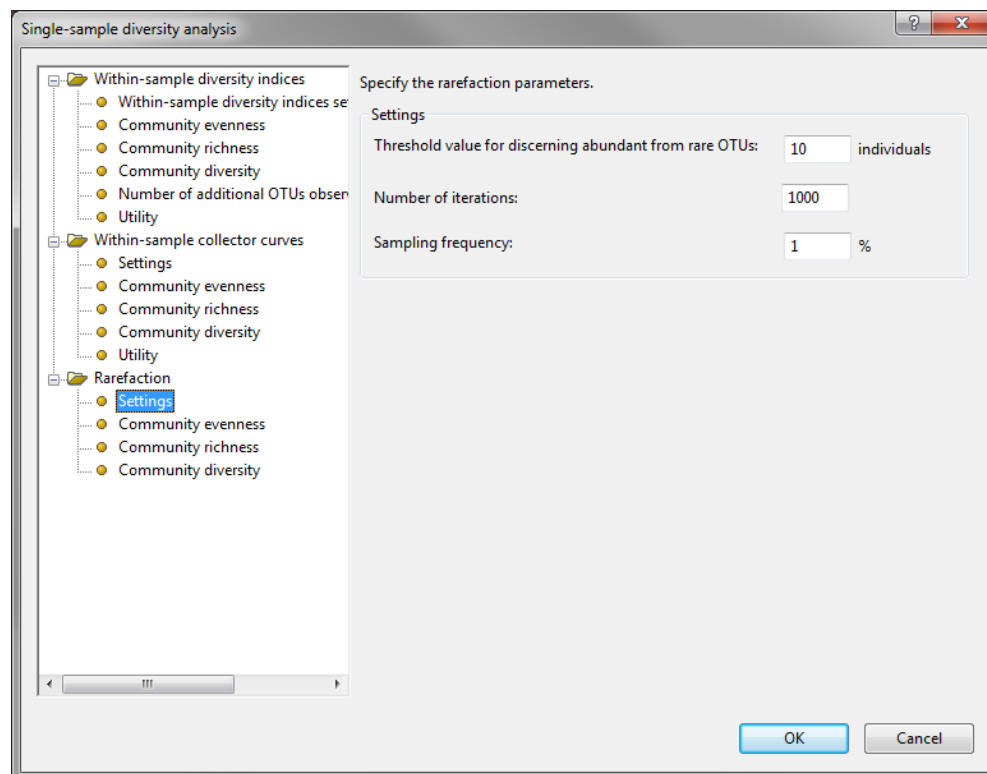


Figure 19.4.48: The *Single-sample diversity analysis* dialog box: The settings for the rarefaction analysis.

- The ***threshold value for discerning abundant from rare OTUs***, default set at 10 individuals, meaning that an OTU is considered abundant when having more than 10 individual reads assigned to it.
- The ***number of iterations*** used in the randomization process of the resampling. To improve the accuracy of the calculations the number of iterations can be increased, but this has major implications on the speed of the calculations.
- The ***sampling frequency*** for the rarefaction curves, expressed as a percentage of the number of reads in the analysis.



From the options under ***Rarefaction*** in the *Single-sample diversity analysis* dialog box, one can modify the diversity indices for which the rarefaction curves will be calculated. An overview of the available indices on evenness, richness and diversity is listed under the settings for the within-sample diversity indices.

Chapter 19.5

The Metagenomics window

19.5.1 Panel structure

There are a number of possibilities to open an existing metagenomics project:

- From the *Metagenomics projects* panel: double-click the metagenomics project, or alternatively, highlight the project in the *Metagenomics projects* panel and select **Edit > Open highlighted object...** (, **Enter**). Multiple metagenomics analyses may be open at the same time.
- From the *Sequence read set experiment* window: double-click the metagenomics project in the *Analyses* panel, or alternatively, highlight the project in the *Metagenomics projects* panel and select **File > Open selected analyses** (.
- From the *Main* window: for the entry selection in the database, multiple projects can be created at once by selecting **Analysis > Sequence read set types > Identify against taxonomic database** or **Analysis > Sequence read set types > Single-sample diversity analysis**. When multiple entries were selected, the *Metagenomics* window will not open and the projects are executed in the background. However, if only one was selected, the newly created project opens directly in the *Metagenomics* window upon the start of the calculation if desired.

The *Metagenomics* window (see Figure [19.5.1](#)) contains five dockable panels:

- The *Project* panel contains the different elements of the metagenomics project (see also [19.5.2](#)).
- The *Report list* panel gives an overview of the data reports which contain specific combinations of information on the input data and result data. Both reports generated from report templates, and custom created reports are listed here (see also [19.5.4](#)).
- The *Data source overview* panel displays the data sources e.g. visualizations, data sets and text tables made available by the calculated project. These data sources can be used for evaluation of the results and can be combined in different analysis reports (see also [19.5.5](#)).
- The *Report* panel displays the different analysis reports, including a selection of data sources i.e. text information and specific visualizations on results calculated by the project (see also [19.5.6](#)).
- The *Log* panel displays the run information about the calculated parts of the metagenomics project (see also [19.5.3](#)).

The default configuration of the *Metagenomics window* is presented in Figure [19.5.1](#). One can alter the configuration of the different panels by re-docking. In this way, a personalized configuration can be obtained and stored. The default configuration can be restored at any time (see [2.3.4](#)).

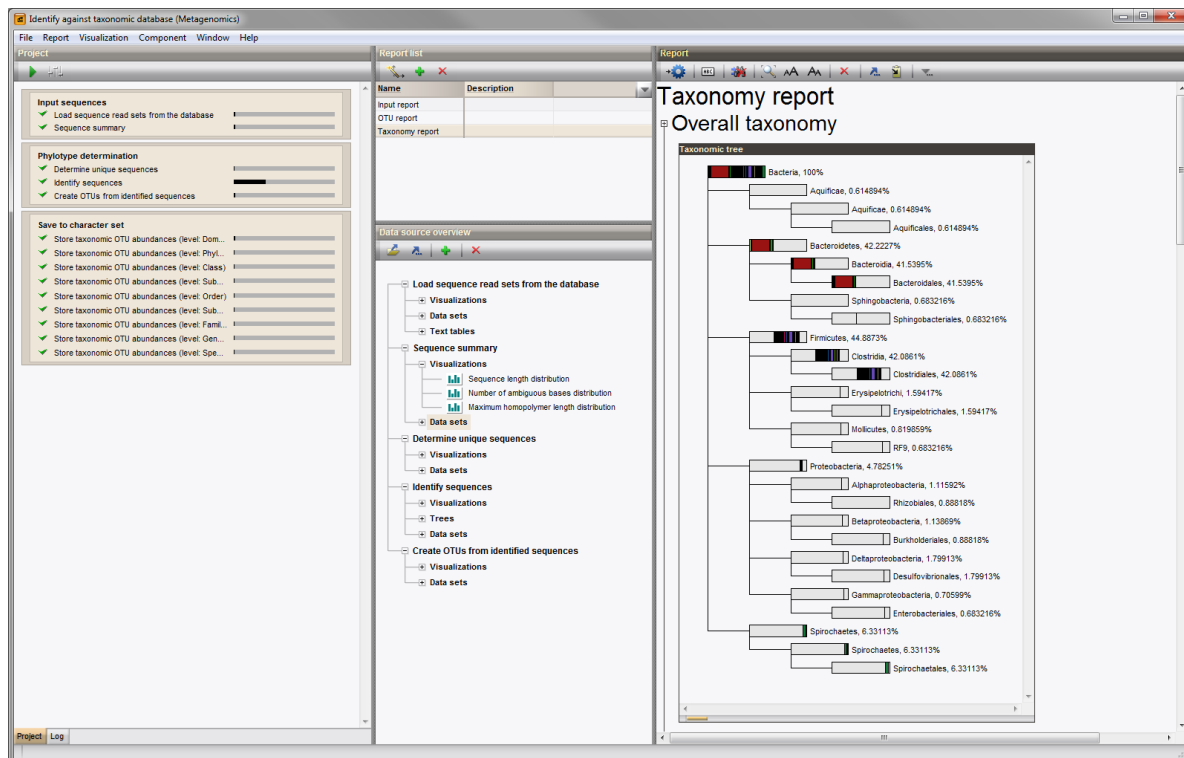


Figure 19.5.1: The *Metagenomics* window.

In the following sections, the different panels will be discussed.

19.5.2 The Project panel

19.5.2.1 Introduction

The *Project* panel provides a way to get immediate feedback on the status of the different elements in a metagenomics project. Depending on the project of choice, the analysis pipeline is updated and calculation progress is displayed.

From this *Project* panel, the project can be (re)calculated and, for each of the different elements in the project, specific settings can be queried for and modified. Note that each element block is divided into separate steps, each having its own status and progress information.

A project can be constructed in three different ways, by

- starting a new project, based on one of the predefined analysis templates present in the database,
- loading a customized project template that was already saved to the database from earlier analyses, or
- loading a project pipeline from an external xml template file.

Note that, when starting the analysis from the *Sequence read set experiment* window, the metagenomics project is automatically created and calculation is started after completion of the metagenomics analysis wizard. At the start of the calculation, the user has the option to open the dedicated *Metagenomics* window or to run the analysis without opening the window.

19.5.2.2 Working with project templates

Project templates contain a specific analysis workflow. A metagenomics project template includes all the elements of the project, including all the parameters settings. They are particularly useful when repeatedly performing the same analysis on different data sets. Moreover, they can be exported and shared between different BioNumerics users.

An existing project pipeline can be stored as a template in the database by selecting **File > Save project as template....**

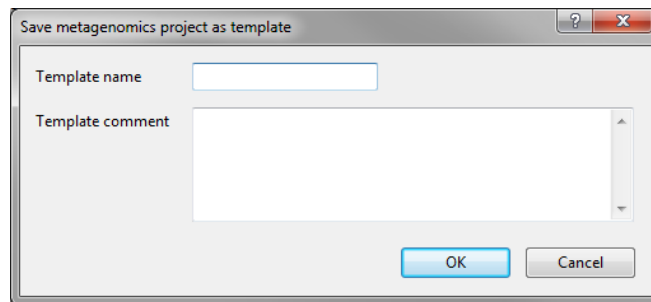


Figure 19.5.2: The *Save metagenomics project as template* dialog box.

In the *Save metagenomics project as template* dialog box, a **Template name** and **Template comment** can be entered.

Any project pipeline that is stored as a customized template in the database, can be loaded into a metagenomics project by selecting **File > New project....**

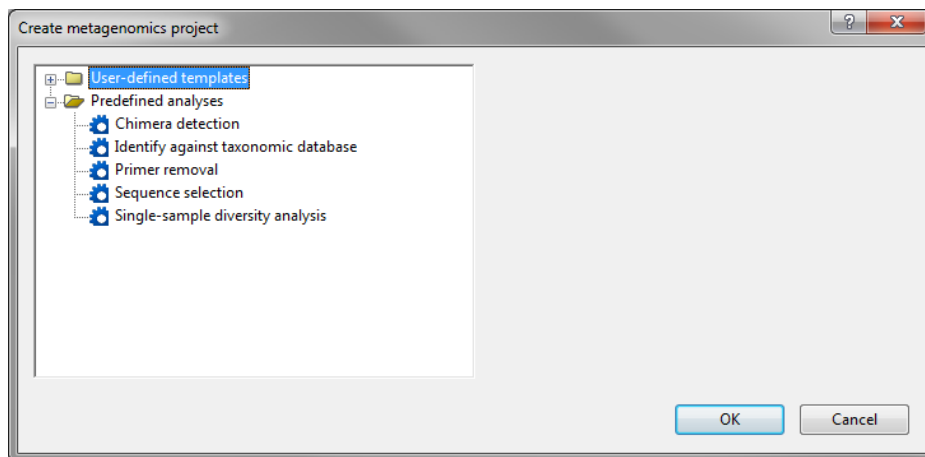


Figure 19.5.3: The *Create metagenomics project* dialog box.

In the *Create metagenomics project* dialog box, a template can be selected to create the new metagenomics project. Two different types of analyses can be selected. The first option is to select a user-defined project template that was previously saved to the database. Secondly, one of the predefined analyses templates can be selected. Note that both preprocessing (Chimera detection, Primer removal and Sequence selection) and processing templates (Identification against a taxonomic database and Single-sample diversity analysis) are present in this list. To load the selected project as a new analysis, select the template from the list and press **<OK>** to confirm. The selected project pipeline is then updated in the *Project* panel of the *Metagenomics* window.

The user-defined project templates can be deleted from the database by selecting **File > Remove project templates....** This will open the *Remove metagenomics project templates* dialog box.

To delete a user-defined metagenomics project template from the database, select the template from the *Remove metagenomics project templates* dialog box, and press <OK>. Deleting user-defined templates is an irreversible operation.

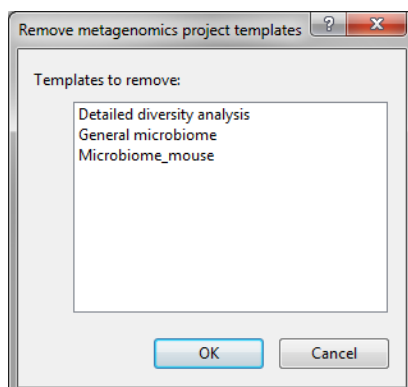


Figure 19.5.4: The *Remove metagenomics project templates* dialog box.

The functionality of the project templates listed above is only valid within the same database. To exchange projects templates between different databases or even different users, the same project templates can be used once they are exported from the database as external files.

To export a metagenomics project, select **File > Export project...**. The *Export project* dialog box is now displayed.

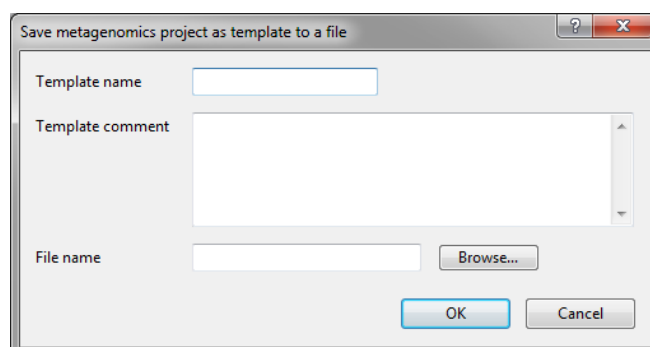


Figure 19.5.5: The *Save metagenomics project as template to a file* dialog box.

In this dialog box, enter the template name and optionally, a template comment (Figure 19.5.5).

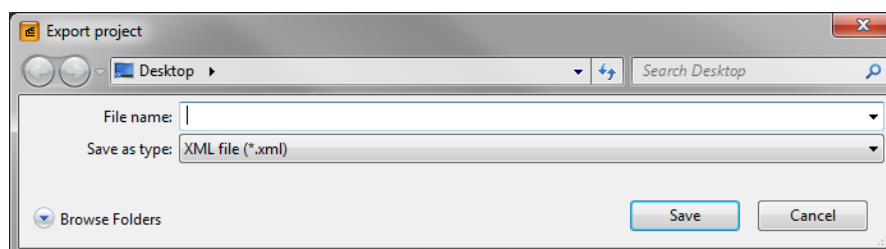


Figure 19.5.6: The *Export project* dialog.

Next, use the <**Browse**> option to navigate to the path where the XML file should be stored (Figure 19.5.6). Enter the **File name** and press <**Save**> to export the project as an XML file.

To import a project template from an XML file, select **File > Import project...**. This opens the *Import project* dialog box.

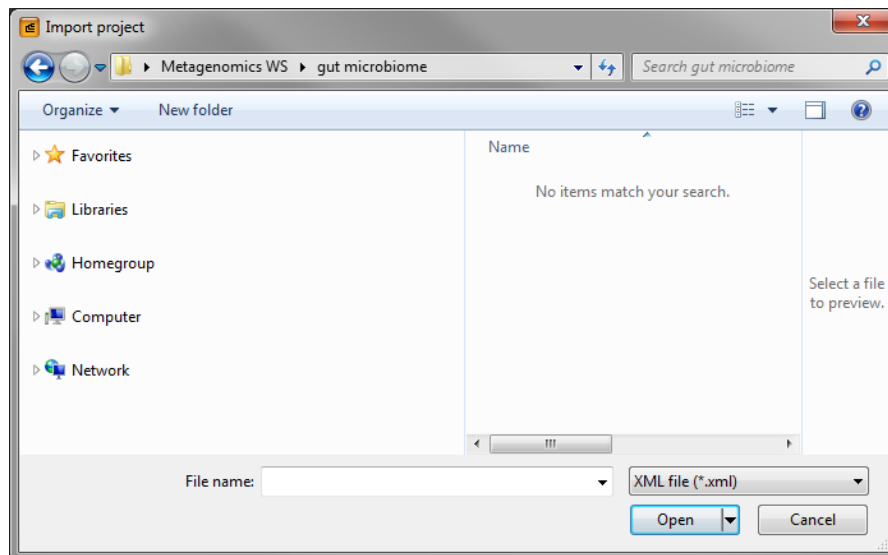


Figure 19.5.7: The *Import project* dialog box.

In this dialog, browse to the correct XML file that contains the metagenomics analysis template, and confirm the import by selecting **<Open>**. This will load the project pipeline from the template in the *Project* panel of the *Metagenomics* window.

19.5.2.3 Executing the metagenomics analysis projects

When starting the analysis from the *Main* window or from the *Sequence read set experiment* window, a metagenomics project is automatically created and calculated upon completion of the analysis wizard.

However, when starting the analysis from the *Metagenomics* window and selecting any of the elements from the project, the command **File > Execute project** (▶) will start querying the parameters in the dialog wizard and once the wizard is completed, the calculation of the project will be launched.

When calculating the project for the first time, the calculation starts from the first element. For each element step, a green V-sign is displayed in front of the element name when the calculation is finished. Right after the element name, the grey bar indicates the total project time whereas the black bar indicates the relative amount of time needed to complete the respective element step.

In case one or multiple elements are already calculated, calculation starts from the first unfinished element and proceeds for the complete project. From the moment element settings are updated, the element status changes to *to be calculated*, and the green check mark indicating that an element step has been completely calculated, disappears.

At any moment during the project calculations, the calculations can be canceled by selecting **<Stop>** on the *Calculation progress* dialog box, or by selecting **File > Cancel execution**. This will stop the calculation process.

19.5.2.4 The project settings

The metagenomics projects consist of several elements, combined according to the selected type of analysis. The settings for each of these elements can be called by **File > Element settings...** (⚙️). This opens the specific *Element settings* dialog box for the selected element in the project (see Figure 19.5.8 for an example of one of the specific *Element settings* dialog boxes).

Each of the elements has its specific *Element settings* dialog box. In general, an overview of the different

element steps is included, and for each selected element step, the specific parameter settings can be defined at the right side of the dialog. Selecting **<OK>** will remove the green check marks and the time bars from the selected element, and the element status becomes 'to be calculated'. Selecting **<Cancel>** closes the dialog box without altering the project parameters and without any adjustments to the project.

In the example shown in Figure 19.5.8, the settings for the *Phylotype determination* step in the *Identification against a taxonomic database* analysis are displayed. In this case, the taxonomic database can be changed and the identification and search method used in the identification process can be altered.

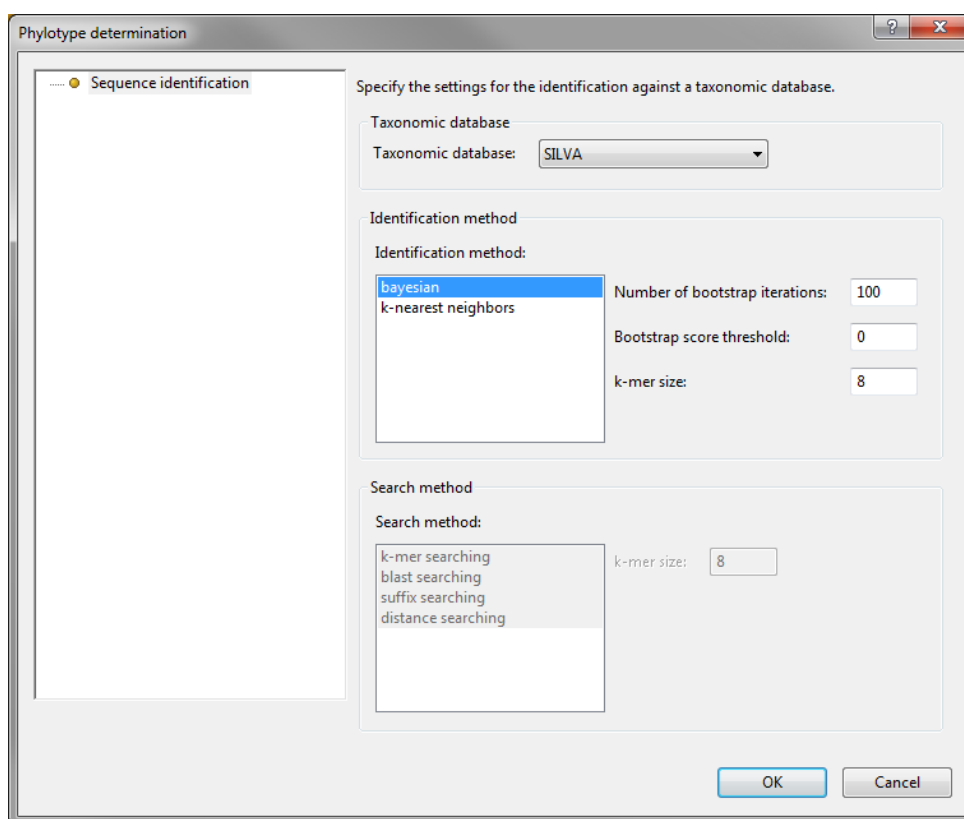


Figure 19.5.8: The *Element settings* dialog box for the Phylotype determination step in the Identification against a taxonomic database.

When calculating a metagenomics project, all analysis levels are calculated, but the analysis levels to be used in the reports are queried for. If the analysis is based on sequence clustering, the *Sequence identity threshold* is queried for, if the analysis is based on both sequence clustering and taxonomic identification, then the *Sequence identity threshold* as well as the *Taxonomic identification levels* to be used are queried for.

When these parameters need to be altered after a first iteration, these settings can be called by selecting **File > Select analysis levels....** This opens the *Select analysis levels* dialog box (see Figure 19.5.9).

The analysis levels used for reporting can be altered from the *Select analysis levels* dialog box. In this dialog, three tabs are displayed. When sequence clustering is part of the project, the sequence identity thresholds are displayed in the *Clustering levels* tab. The taxonomic identification levels can be altered from the *Taxonomic levels* tab, and for projects using a sequence clustering and taxonomic identification, a combination of both clustering and taxonomic levels is displayed in the *Mixed levels* tab. Each of the levels displayed in these tabs can be activated or deactivated within the current project by selecting or unselecting the check box in front of the level. Press **<OK>** to save the changes to the project, or **<Cancel>** to close the dialog without altering the project settings.

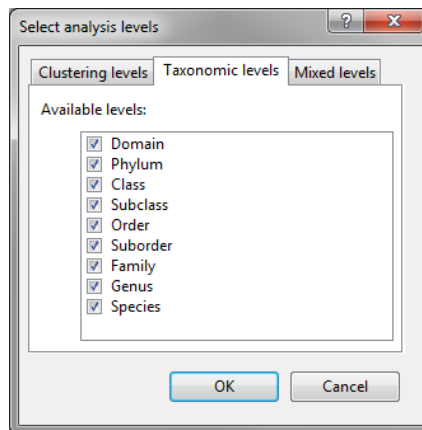


Figure 19.5.9: The *Select analysis levels* dialog box.

19.5.3 The Execution log panel

During the calculation of a metagenomics project, detailed information on each of the element steps is displayed in real time in the *Log* panel. This analysis log contains information on the time the computation started and stopped, the time elapsed and a message about the analysis details. Additionally, elements that include third party tools, e.g. parts of the metagenomics analysis performed by the *mothur* program, have the information provided by the software tool streamed to the execution log.

19.5.4 The Report list panel

The *Report list* panel gives an overview of the visualized predefined and custom generated reports within the metagenomics project. Once the project is calculated, report templates can be used to automatically generate the required analysis report.

To use any of these reports, select **Report > Add report from template...** (🔧) to create a the report by applying the report template to the data in the analysis at hand. This opens the *Create report* dialog box (see Figure 19.5.10).

In the *Create report* dialog box, the report template to create the report from can be selected. Depending on the data sources available within the project, different report templates will be displayed.

Within this example (see Figure 19.5.10), one existing report template for the exploratory data analysis is available: the **Input report**, and two templates for the diversity analysis are available: one reporting template on **OTU results**, and one on the **Taxonomy results**. If required, one can modify the display **Name** for the report on top of the dialog.

After selecting any of the report templates from the list and confirming by selecting **<Next>**, the report template is applied to the project data, and the generated report is added to the report list. From the moment the report is loaded, the report content is also displayed in the *Report* panel (see 19.5.6 for more information on this panel).

For some templates, no further information is required, but for both the OTU and the taxonomy report, the analysis levels to be included in the report need to be specified within the dialog that pops up when creating the report.

Once the reports are generated within the project, for each report, the report name (default name or custom name specified in the **Name** box) and a description are available from the *Create report* dialog box. By default, no descriptions are present, but, by clicking the description field, a report description can be entered.

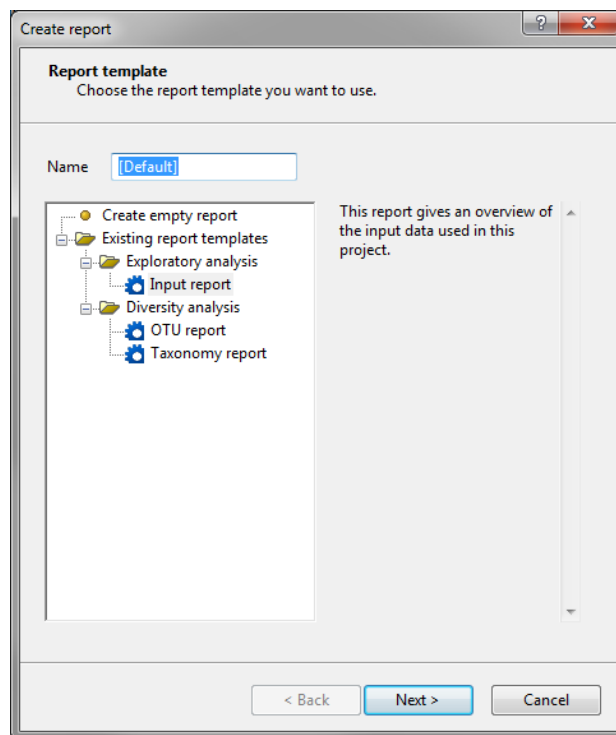


Figure 19.5.10: The *Create report* dialog box.

In case the existing templates do not contain the required information, one can always create a customized report. Thereto, select **Create empty report** from the *Create report* dialog box, fill in a template name, and select **<Next>**. A new report is now added to the report list. By selecting this report, the *Report* panel is updated. By now, no information is displayed and the panel remains empty. One can start creating the custom report by adding text and visualizations to it. See 19.5.5 for more information on adding data sources to the report and 19.5.4.1 to save the report layout as template.

A new, empty report can also be created by selecting **Report > Add report...** (+). This opens the *Add report* dialog box where the name of the new report can be entered. When selecting **<OK>**, a new report is added to the report list. Similar to the **Create empty report** functionality in the *Create report* dialog box, initially no information is displayed and one needs to modify the report by adding text and visualizations.

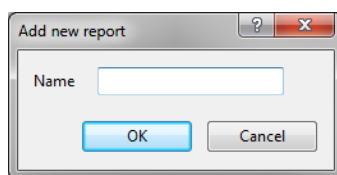


Figure 19.5.11: The *Add report* dialog box.

A selected report can be removed from the project by selecting **Report > Remove report** (x). Please note that this command removes the selected report without any warning and cannot be undone.

19.5.4.1 Working with report templates

Once you have created a customized report, the report layout can be saved to a report template which is saved to the database in order to be used in future metagenomics projects. Additionally, the exchange of metagenomics report templates between databases and even between different BioNumerics users is possible through the use of XML report templates.

To create a report template from the selected report in the *Report list* panel, select **Report** > **Save report as template...** (⚙️). This opens the *Save report template* dialog box (see Figure 19.5.12).

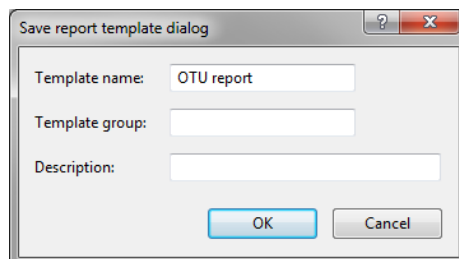


Figure 19.5.12: The *Save report template* dialog box.

In the *Save report template* dialog box, the “Template name”, “Template group” and “Description” can be specified. After selecting <OK>, the template is automatically saved to the database. The next time the *Create report* dialog box is launched, the project template becomes available under the *Existing report templates*.

Saved reports can be managed from the *Remove report templates* dialog box (see Figure 19.5.13). In this dialog all existing report templates within the database are listed and can be selected to be removed. Highlight the template to be removed and confirm by pressing <OK>. The report template is now erased from the database and is no longer listed when launching the *Create report* dialog box.

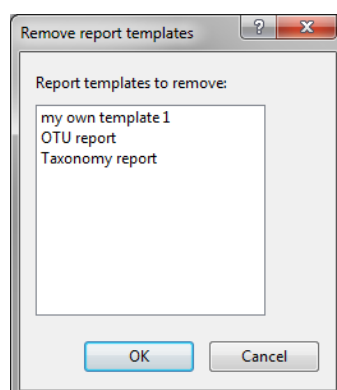


Figure 19.5.13: The *Remove report templates* dialog box.

To exchange metagenomics reports, the selected report can be exported to an XML file by selecting **Report** > **Export report template....** This opens the *Save report template to file* dialog box (see Figure 19.5.14).

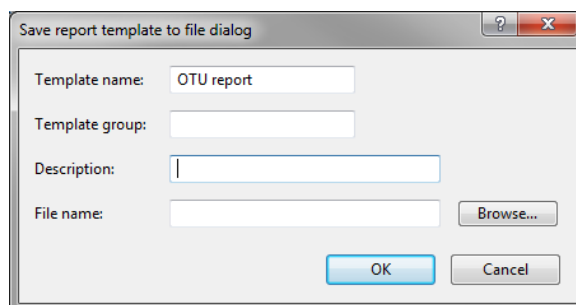


Figure 19.5.14: The *Save report template to file* dialog box.

In the *Save report template to file* dialog box, the “Template name”, “Template group” and “Description” can be specified. Select the <Browse...> button to open the *Report template file* dialog to browse to the

directory where the template will be saved to, and specify a “File name”. Press <**Open**> to save the path and file name to the *Save report template to file* dialog box. Press <**OK**> to actually create the XML template file.

A metagenomics report template can be imported into a project by selecting **Report > Import report template**. This will open the *Load report template from file* dialog box (see Figure 19.5.14).

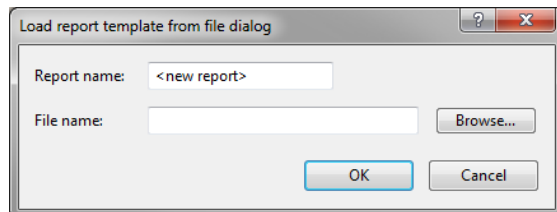


Figure 19.5.15: The *Load report template from file* dialog box.

In the *Load report template from file* dialog box, a different “Report name” can be specified for the report that will be generated by the report template. Next, use the <**Browse...**> button to navigate to the directory where the XML template file is saved, and select the corresponding file. Press <**Open**> to return to the *Load report template from file* dialog box. Press <**OK**> to create the report in the metagenomics project.

19.5.5 The Data source overview panel

The navigation tree in the *Data source overview* panel displays the generated data sources for the different elements in the analysis: visualizations, data sets and text tables. Under each analysis step, the generated visualizations, data sets and text tables are listed. The + and - signs before each category node can be used to navigate through the data listed in the tree.

- *Visualizations* include charts and taxonomic trees that are created from the data sets generated within the analysis. A variety of visualizations is automatically created upon calculation of the project. For this, chart templates are applied to the data at hand and the obtained visualization is added to the navigation tree. Additionally, custom charts can be created from the available data sets. Once a custom chart is created, the chart template can be saved, and applied in other analysis projects.
- *Data sets* are intermediate analysis results generated by one of the analysis steps. Typically, intermediate data is saved in individual files, consisting of multiple columns, each column holding character information on e.g. the sequences in the analysis, the taxonomic analysis results, a calculated statistic Data sets can be visualized in the *Data set grid* window (see 19.5.7.1).
- *Text tables* are static text elements that contain specific information on e.g. the sequence read set used. No changes can be made to the text table once imported in a report.

A selected data source can be visualized by selecting **Visualization > Show in dedicated window** (🔍). Depending of the data source type, the *Charts and statistics* window (Figure 19.5.22), *Data set grid* window (Figure 19.5.21) or *Taxonomic tree* window (Figure 19.5.25) will open, displaying the selected information.

When creating or modifying analysis reports, a selected data source can be added to the report by selecting **Visualization > Add to report** (📎). The report is immediately updated with the selected data source.

When the default visualizations do not meet your expectations, a custom visualization can be created. Thereto, first select the data set in the data source overview that contains the source information to create your graph from. Next, select **Visualization > Add visualization...** (➕) to call the *Create visualization* dialog box (see Figure 19.5.16.).

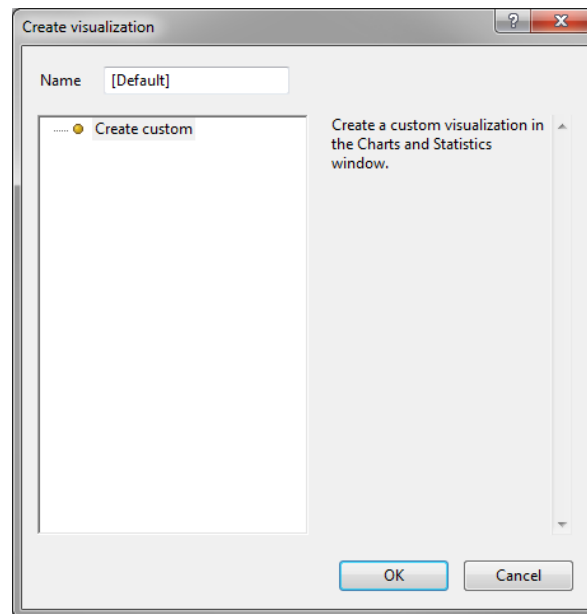


Figure 19.5.16: The *Create visualization* dialog box.

In this dialog, the “Name” of the chart can be specified. Upon installation, no custom visualization templates are present in the database. However, when the software detects a visualization template that was previously saved, it will be listed in this dialog. By default, only the option **Create custom** is present. With this option selected, and after confirming by <OK>, the *Charts and statistics* window opens.

In brief, a new chart can be started from **Plot > Create plot wizard...** (🔧). Once the requested chart is created, the *Charts and statistics* window can be closed and the chart is added to the node in the navigation tree and updated in the *Report* panel of the *Metagenomics* window. To create a chart template from the generated chart, select **File > Save as report template...** (⚙️) in the *Charts and statistics* window. Detailed information on the functionality of the *Charts and statistics* window can be found in 14.5.

When a template is saved to the database, the next time the *Create visualization* dialog box is started from a data set that is compatible with the chart template, the template will be listed under the **Existing chart templates** (see Figure 19.5.17).

For each component, visualization, data set ..., specific settings can be specified in the *Report* panel. When one component is added to the report multiple times, all these components share the same display and filter settings. This might not always be desirable e.g. when starting from the same visualization and one wants to highlight different lineages e.g. one taxonomic representation only displaying the Firmicutes and the other only displaying the Proteobacteria. In this case, a clone of the selected visualization present in the data source overview can be created, and both visualizations can be added to the report, each visualization having its own display and filter settings. To do so, select the visualization from the overview that needs to be cloned and select **Visualization > Clone visualizations**. This command creates a copy of the visualization in the data source overview and adds the clone to the active report.

Any selected visualization in the data source overview can be deleted by selecting **Visualization > Remove visualizations** (✖️). Please note that the visualization will be deleted without any warning and cannot be undone.

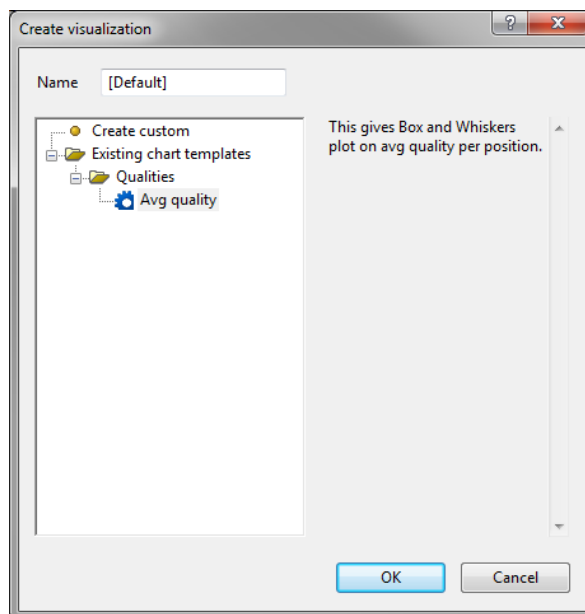


Figure 19.5.17: The *Create visualization* dialog box.

19.5.6 The Report panel

In the *Report* panel, the analysis result from the metagenomics projects are displayed. Each time a report is selected in the *Report list* panel (see 19.5.4), the content of the report is updated.

The report consists of multiple components, usually text components or data sources that have been added to the report from the *Data source overview* panel (see 19.5.5).

Dedicated analysis report are created by adding the components as defined in the template. However, the report can be edited and modified at any time. The following edit functions are general to all components:

- **Component > Zoom to fit** (🔍): changes the size of the selected component to fit the display window.
- **Component > Enlarge** (⏏): enlarges the display window of the selected component.
- **Component > Shrink** (⏏): reduces the display window of the selected component.
- **Component > Remove** (✖): removes the selected component from the report.
- **Component > Open the selected components in dedicated windows** (📄): opens the selected component in its dedicated window. Depending on the data source type, this is the *Charts and statistics* window (Figure 19.5.22, see also 19.5.7.2), the *Data set grid* window (Figure 19.5.21, see also 19.5.7.1), the *Taxonomic tree* window (Figure 19.5.21, see also 19.5.7.5) or the *Rich table* window (Figure 19.5.25, see also 19.5.7.3),
- **Component > Copy to clipboard** (📋): copies the selected component to the clipboard. To copy a visualization, the settings need to be defined in the *Copy bitmap to clipboard* dialog box (see Figure 19.5.18).

In this dialog, the export of a visualization as bitmap or as metafile can be specified. For each of the export options, use the check box to define whether or not the chart legend should be exported along with the chart itself. When exporting the chart as a bitmap, additionally, the image resolution can be set. Confirming by <OK> will copy the visualization to the clipboard.

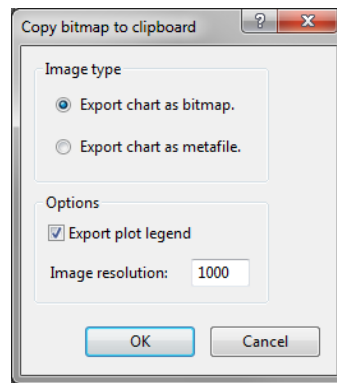



Figure 19.5.18: The *Copy bitmap to clipboard* dialog box.

For a selected *visualization*, the following commands can be found under the  button or under **Component** > **Selected component**:

- **Settings...**: Calls the *Chart options* dialog (see Figure 19.5.19).

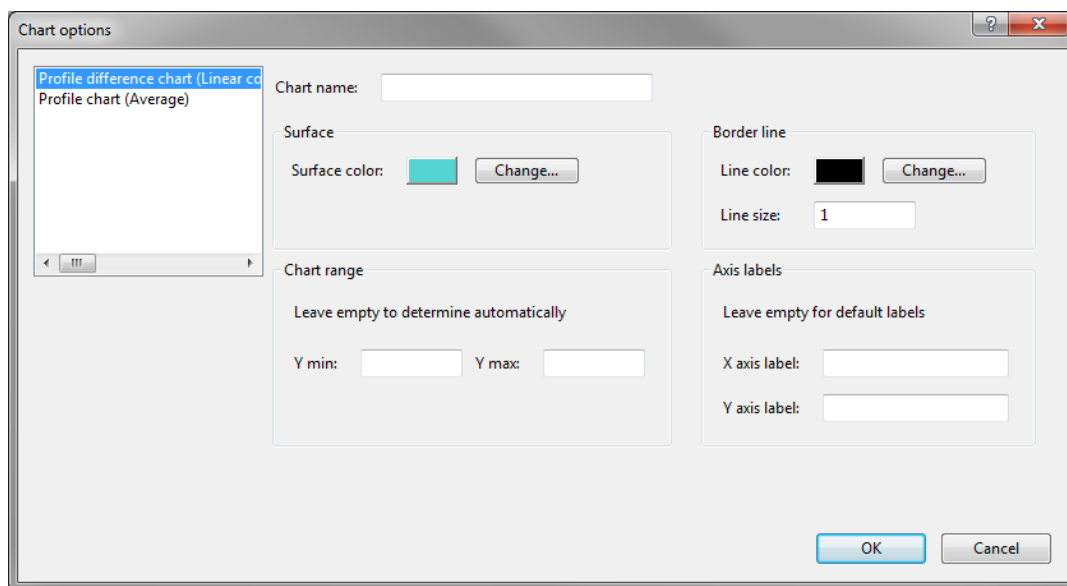



Figure 19.5.19: *Chart options* dialog.

In this dialog box, the chart properties can be altered. More information on chart options can be found under 14.5.4. After modification of the chart options, select <OK> to update the chart in the report.

- **Choose active plot...**: When multiple plots are defined in one chart, select this command to display the *Choose active plot* dialog box (see Figure 19.5.20).

This dialog lists the available plots. Select the plot to be displayed in the report and confirm by <OK>.

- **Print...**: Calls the *Print Setup* dialog box to adjust the printer settings and to start the printing job.

For a selected *data set*, the following commands can be found under the  button or under **Component** > **Selected component**:

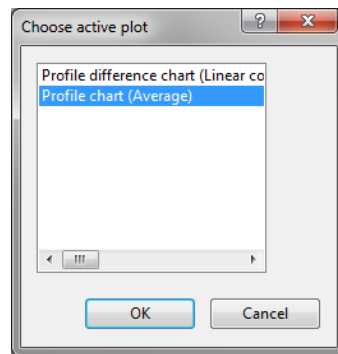





Figure 19.5.20: The *Choose active plot* dialog box.


- **Sort by highlighted column:** Sorts the data in increasing order according to the highlighted column.
- **Copy content to clipboard:** Copies the content of the data set to the clipboard.
- **Save content to file:** Saves the content of the data set to a comma-separated file which is then opened in the default program for csv files e.g. Excel.

For a selected *text table*, the following additional commands can be found under the  button or under **Component > Selected component**:

- **Zoom to fit:** Zooms the selected text table to fit the text.
- **Copy content to clipboard (formatted):** Copies the content of the text table to the clipboard and keeps the source formatting when pasting the content.
- **Copy content to clipboard (unformatted):** Copies the content of the text table to the clipboard without keeping the source formatting.

Next to the data sources from the *Data source overview* panel, also additional text information can be added to the reports by selecting **Component > Add text** (). This opens the *Rich edit* window. See 19.5.7.4 for more information on this rich text editor. Once the text information is entered in this dialog, select **File > Save** to save the modifications and to close the dialog. The text is now added to the report.

Once a report is customized, the report layout can be saved to the database by selecting **Report > Save report as template...** (). In this way, the report template can be used to create a similar report on other data from another calculated metagenomics project. See 19.5.4.1 for more information on working with report templates.

A chart that is displayed in the report is always attached to a data source from which it was created. As such, changing entry selections on a data set is instantly synchronized in the *Charts and statistics* window and vice versa. At any time, the entry selections can be changed. To quickly clear all selections from all charts and data sets, select **File > Clear selection** (.

19.5.7 Specific data visualization windows

19.5.7.1 Data set grid window

The *Data set grid* window (see Figure 19.5.21) is used to display the information contained in the intermediate data files in a structured way. The data set can be sorted based on each of the available columns.

Data set grid									
File Edit Window Help									
Content									
	Position	Num...	Average	Standard...	95% quant...	75% quant...	median	25% quant...	5% quant...
0	5886	38.184166	0.777811	40	40	39	39	30	
1	5886	38.331125	0.659225	40	40	39	39	30	
2	5886	38.059123	0.748674	40	40	39	39	30	
3	5886	38.061672	0.74671	40	40	39	39	30	
4	5886	38.073394	0.742248	40	40	39	39	30	
5	5886	39.711349	0.003878	40	40	40	40	40	
6	5886	39.72596	0.003334	40	40	40	40	40	
7	5886	39.748046	0.002094	40	40	40	40	40	
8	5886	39.890588	0.000194	40	40	40	40	40	
9	5886	39.93595	0.000045	40	40	40	40	40	
10	5886	39.948352	0.000028	40	40	40	40	40	
11	5886	39.955827	0.00002	40	40	40	40	40	
12	5886	39.955318	0.000021	40	40	40	40	40	
13	5886	39.948182	0.000038	40	40	40	40	40	
14	5886	39.947842	0.00004	40	40	40	40	40	
15	5886	39.819062	0.001767	40	40	40	40	39	
16	5886	39.668366	0.008457	40	40	40	40	39	
17	5886	37.102956	1.567133	40	39	39	39	26	
18	5886	37.137275	1.594006	40	40	39	39	26	
19	5886	37.1123	1.620603	40	40	39	39	26	
20	5886	39.767074	0.002567	40	40	40	40	40	
21	5886	39.779647	0.0026	40	40	40	40	39	
22	5886	39.864084	0.000612	40	40	40	40	40	
23	5886	39.681278	0.005377	40	40	40	40	39	
24	5886	39.554536	0.012162	40	40	40	40	37	
25	5886	35.699117	3.801442	40	40	40	30	21	

Figure 19.5.21: The *Data set grid* window.

There to, highlight a column and select **Edit > Sort by highlighted column**. This will sort the data in the column in increasing order and expand the sorting to all other columns in the data set. By clicking on the column properties button (⚙️), the general grid panel functionality can be accessed e.g. to export the data set information. Select **File > Exit** to close the *Data set grid* window.

19.5.7.2 Charts window

For more information on the specificities and the different features of the *Charts and statistics* window (see Figure 19.5.22), we refer to 14.5.1.

19.5.7.3 Rich text table window

The *Rich table* window (see Figure 19.5.23) contains tabular text information generated by the metagenomics project. The information is structured in a table and cannot be altered. By clicking on the column properties button (⚙️), the general grid panel functionality can be accessed e.g. to export the table information. Select **File > Exit** to close the window.

19.5.7.4 Rich text editor window

The *Rich edit* window (see Figure 19.5.24) is an interface for editing rich text within reports, which presents a "what-you-see-is-what-you-get" editing area. For more information on this window, see 3.2.13.

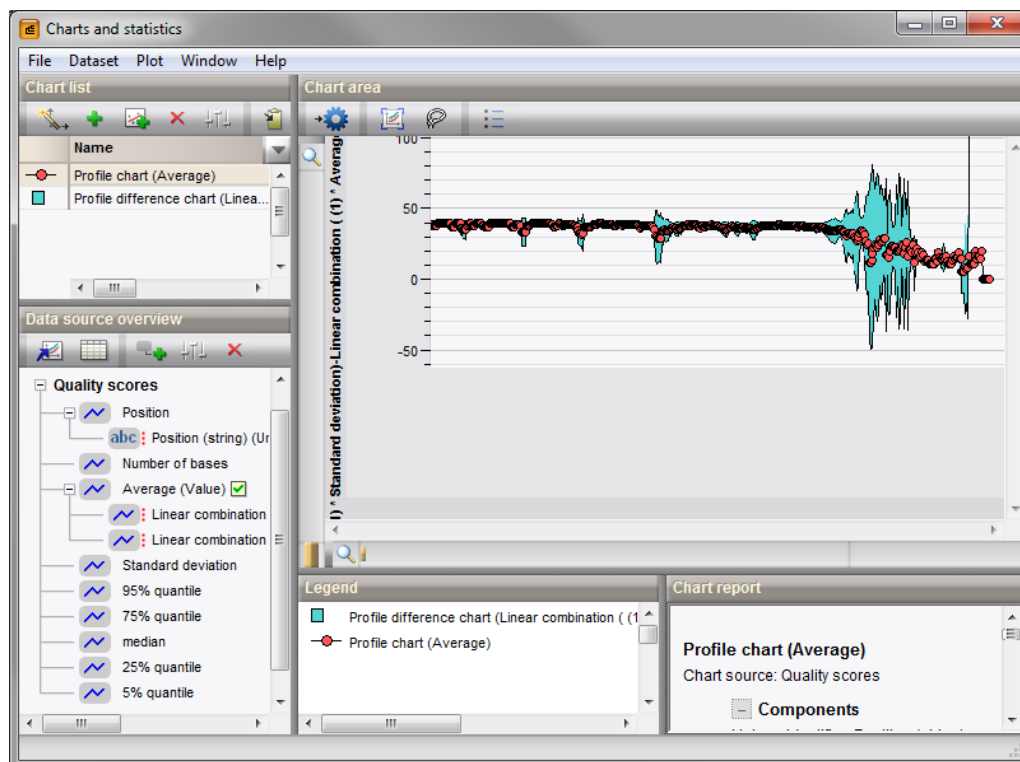



Figure 19.5.22: The *Charts and statistics* window.

Table	
Entry keys	STUDY_AMISH_TAKE20000002
Experiment type	meta raw
Number of sequences	5886
Quality scores	yes

Figure 19.5.23: The *Rich table* window.

19.5.7.5 Taxonomic tree window

Metagenomics projects that perform an identification of the reads against a taxonomic library result in a taxonomic assignment for each of the reads. These individual assignments are summarized in the taxonomy summary, which basically indicates how many reads were identified within each taxon of the reference taxonomy. These results can also be visualized on the taxonomic reference tree, where bar graphs or pie charts indicate how many reads were assigned to each taxon category. One of the default data sources calculated by the analysis template for the identification against a taxonomic database, is the taxonomic tree listed in the data source overview. When selecting the taxonomic tree and selecting **Visualization > Show in dedicated window** () , the *Taxonomic tree* window opens (see Figure 19.5.25).

The *Taxonomic tree* window opens with the taxonomic abundance information of the analyzed sample plotted on the taxonomic tree. By default, all OTUs detected are displayed for the complete taxonomic depth defined in the taxonomic reference database. When the *Taxonomic tree* window opens, a zoom to fit operation is executed, which displays the complete taxonomic tree in the window. When a large variety of

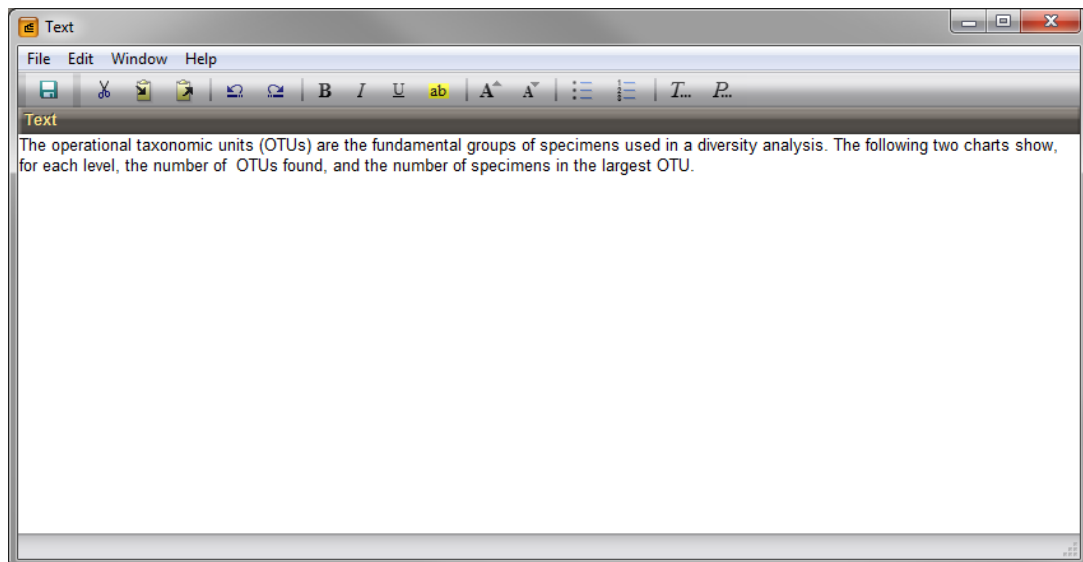
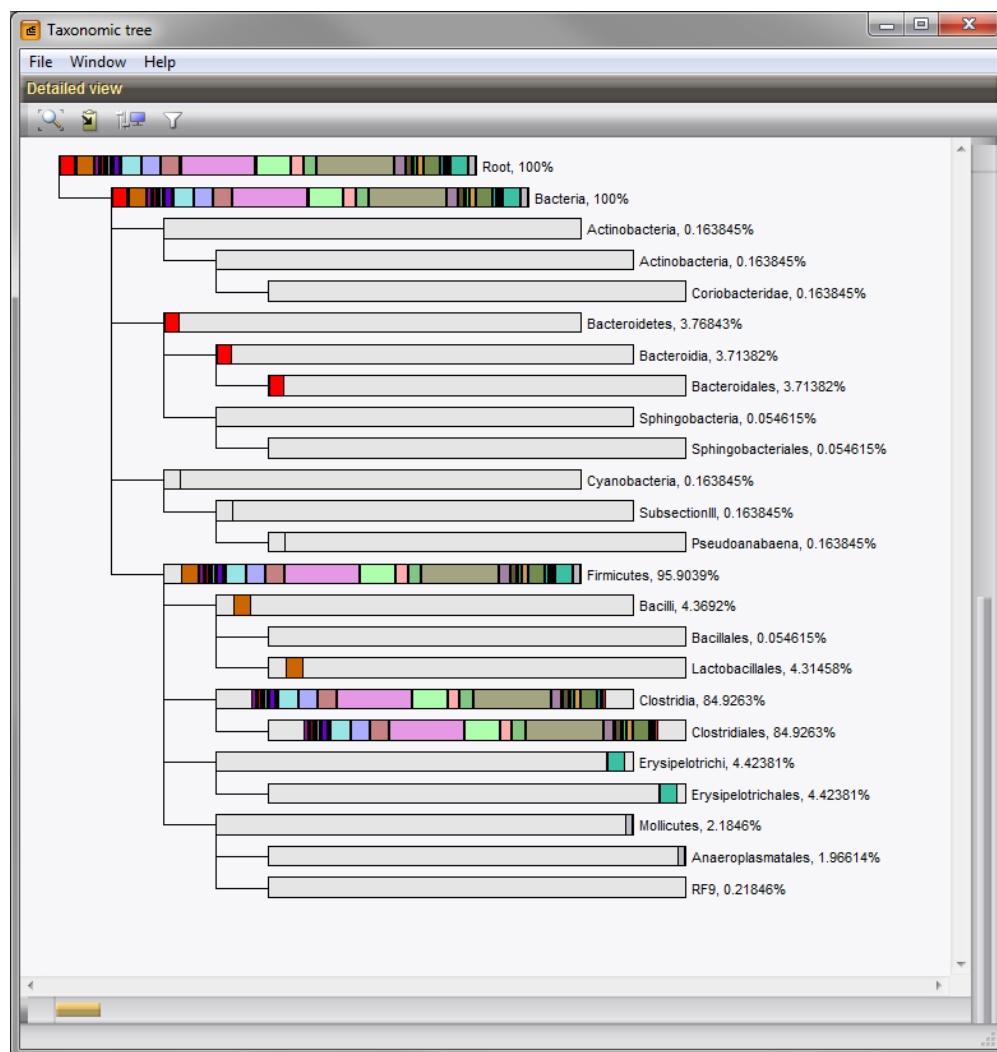


Figure 19.5.24: ?.

Figure 19.5.25: The *Taxonomic tree* window.

OTUs was detected, this results in a large tree where no OTU assignments are displayed. Thereto, different visualization settings can be applied to the tree.

The most simple way of exploring the tree is by pulling the zoom slider at the bottom of the window to change the horizontal zoom. When increasing the zoom level, the color blocks of the different OTUs within a taxonomic level become visible and the OTU assignments are displayed next to the bar graphs. Within this increased zoom level one can scroll through the tree by using the vertical slide bar.

More advanced visualization settings can be defined in the *Visualization settings* dialog box when selecting **File > Display settings...**

The label and abundance information displayed on the taxonomic tree and the sizes of the graphical elements can be defined from the tabs in the *Visualization settings* dialog box.

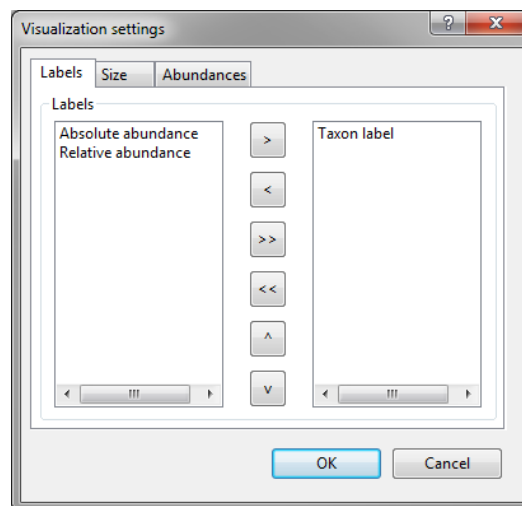


Figure 19.5.26: The *Visualization settings* dialog box.

- From the *Labels* tab, one can specify which labels will be displayed next to the bar graphs. Default, only the taxon label is displayed. Other possible labels include the absolute and relative taxon abundance values.
 - Use ">" to add the selected label in the left column to the active labels, displayed in the right column.
 - Use "<" to remove the selected active label in the right column to the inactive labels, displayed in the left column.
 - Use ">>" to add all labels as active labels.
 - Use "<<" to remove all active labels.
 - Use "v" and "^" to change the display order of the active labels in the right column.
- From the *Size* tab, the general layout of the visualization can be altered by changing the Size scale factors for the X- and Y-direction, as well as the font size by changing the Font scale factor.
- From the *Abundances* tab, one can choose whether or not to show the abundances. When no abundances are displayed, only the labels remain visible in the tree. When the abundances are displayed, by default, they are displayed relative to the total number of reads in the analysis. When unchecking this option, the abundances are scaled to the number of reads present in the taxon node at hand. By changing the Plot type of the visualization, one can choose to have the abundances displayed in bar graphs or in pie charts by selecting the item from the drop down list. The size of the pie charts is fixed, whereas the size of the bar graph can be altered by increasing or decreasing the Abundance plot size (expressed in percentage relative to the default bar graph size).

Press **<OK>** to activate the new settings or **<Cancel>** to close the dialog without saving any changes in the visualization.

By default, all the taxon nodes of the reference taxonomy that have at least one read identified as belonging to the taxon, are displayed in the taxonomic tree. This implies that for samples having a high species diversity e.g. soil samples, the taxonomic tree may rapidly seem difficult to visualize at once. Thereto, different visualization filters can be defined from the *Manage tree visualization filters* dialog box.

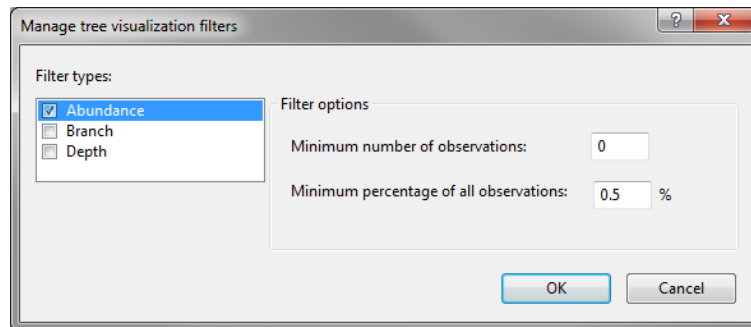


Figure 19.5.27: The *Manage tree visualization filters* dialog box.

Three filter types can be defined on the visualization: the abundance filter, the taxonomic branch filter and the filter on taxonomic depth. To activate one of the filters, simply check the check box in front of the filter type. Multiple filters can be combined in one visualization.

- The *Abundance* filter allows to restrict the visualization to only these taxa which have a minimum number of observations (i.e. the number of reads identified as belonging to the taxon), and/or the minimum percentage of all observations (i.e. the ratio of the the number of reads identified as belonging to the taxon, to the total number of reads in the analysis (expressed as percentage)). When both abundance filters are defined, the most stringent one is applied to the data set.
- The *Branch* filter allows to restrict the visualization to one taxon in particular. First, the taxonomic level of the taxon can be selected from the drop down, and second, the name of the branch needs to be entered in the dialog box.
- The *Depth* filter allows to limit the visualization within specific taxonomic levels, not focusing on one taxon in particular. To set the filter options, select the taxonomic levels from the drop down lists that should be used as start and end levels in the visualization. These taxonomic levels are the same as the ones identified in the taxonomic reference file used in the current project.

Press **<OK>** to filter the data set and update the taxonomic tree or **<Cancel>** to close the dialog without altering the visualization.

File > Zoom to fit will resize the updated taxonomic tree to fit the visualization window.

Selecting **File > Copy to clipboard** allows to export the taxonomic tree as a bitmap or as metafile. When exporting the image as bitmap, the image resolution can be specified in the *Copy taxonomic tree to clipboard* dialog box.

Selecting **File > Print...** pops up the *Print Setup* dialog box where the printer and paper settings can be specified to print the taxonomic tree image.

When filters have been applied to the data and updated the taxonomic tree visualization, the source data can be visualized in a *Data set grid* window by selecting **File > Show filtered data source...**

Select **File > Exit** to close the *Taxonomic tree* window. This will update the taxonomic tree displayed in the report.

Part 20

Matrix mining

Chapter 20.1

An introduction to the Matrix mining window

20.1.1 Introduction

The flexibility of handling and clustering large matrices, as well as some sophisticated statistical functions are provided in a separate window, the *Matrix Mining* window.

Datasets that can be analyzed in the *Matrix Mining* window are character data sets, composite data sets, trend parameter values, and aligned sequences. In case of fingerprint type data, a band matching needs to be performed first; in case of spectral type data, a peak matching is required.

20.1.2 The Matrix mining window

The *Matrix Mining* window consists of a menu, a toolbar for quick access to some important functions, and multiple panels. You can drag the horizontal and vertical separator lines between the panels, in order to divide the space among the panels optimally.

The *Layers* panel, *Subsets* panel and *Groups* panel are located on the left-hand side of the *Matrix Mining* window and have their own toolbar. The panels are dockable, which enables the user to customize the layout according to personal preference.

The *Main view*, located on the right-hand side of the *Matrix Mining* window presents the data matrix with its information fields, the dendrograms and the profiles if calculated. Initially the characters (characters, band classes, peak classes, ...) are displayed as rows and the entries are displayed as columns (see Figure 20.1.1 for an example). See 20.2.5 for different ways of organizing the data.

The calculation of a PCA, SOM or Partitioning opens gateways to other views, with for each of them other menus, toolbars and specific commands.

The command *Statistics > Matrix mining...* in the *Comparison* window launches the *Matrix Mining* window. The values of the selected experiment are loaded into the *Matrix Mining* window (see Figure 20.1.1). An error is generated if the data cannot be loaded into the *Matrix Mining* window.

20.1.3 Discovering the Main view

The *Main view* is the default view, displaying the data set together with following surrounding information panels (see Figure 20.1.1):

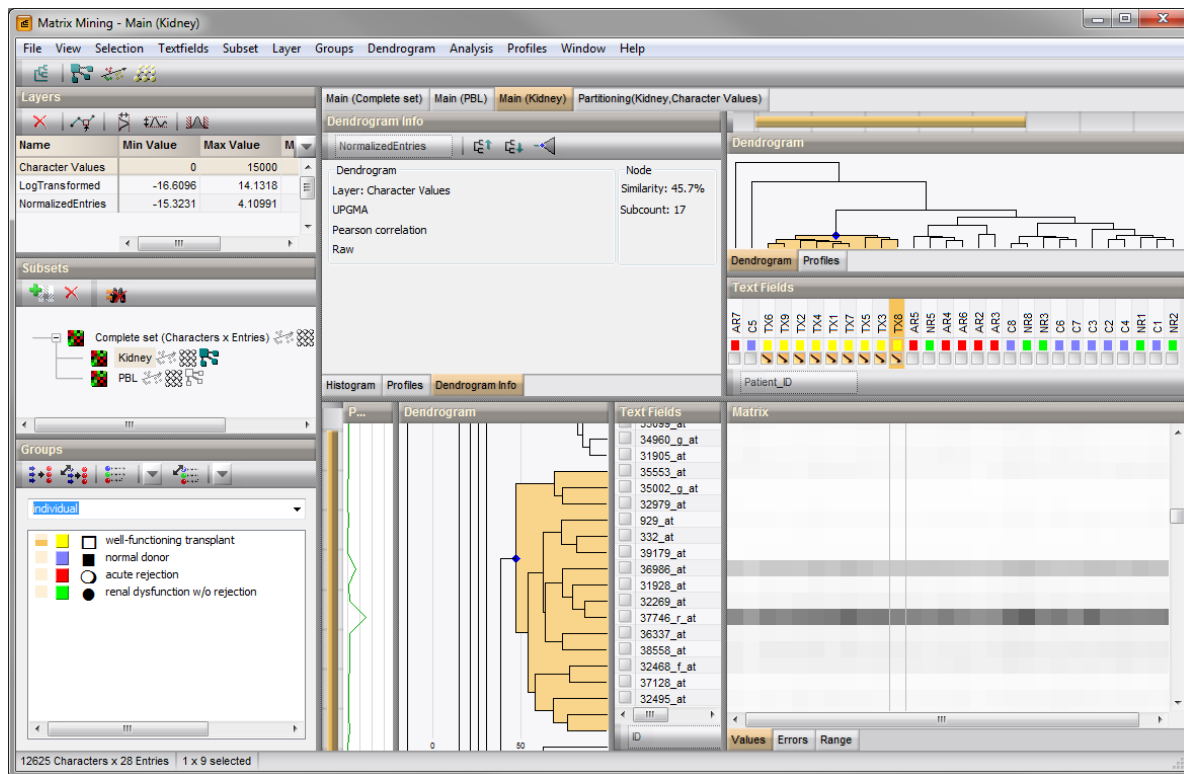


Figure 20.1.1: The *Matrix Mining* window with example data displayed in the *Main view*: characters are displayed as rows, entries are displayed as columns.

- *Matrix panel*: This central panel displays the data in the selected layer by means of colored blocks. The *Values* tab shows the values in the selected layer in color code. The *Errors* tab shows the error flags on the values in color code. If no errors are present this will result in a uniform field. In the *Range* view, each cell in the matrix shows the range between the lowest and maximum value. These values are derived from the value and error flag on that value. If there are no error flags present, this option offers the same view as the *Values* tab.
- *Row/Column Text Fields panel*: These panels show the information fields of the rows/columns. The drop down lists allow the displayed information in these panels to be changed. If another information field is selected from the list, the information in the panel is updated. When a row or column is selected in the panel, its information is highlighted. Upon pressing **Ctrl+C**, the contents of the selected row/column is copied to the clipboard. The **Home** and **End** buttons can be used to go to the first respectively last row/column listed in these panels.
- *Row/Column Dendrogram panel*: These panels display the dendrogram in case a clustering of the rows/columns is calculated. There are a number of tools available to edit the outlook of a dendrogram.
- *List Profiles panel*: Lists all row/column profiles that have been calculated in the *Matrix Mining* window.
- *Dendrogram Info panel*: Displays information on the method used to calculate the clustering of the rows and some numerical information on the selected node in the dendrogram.
- *Histogram panel*: Shows a histogram of the data set. Zooming in or out on the graphical representation can be done with the zoom slider on the right.
- *Row/Column Profiles panel*: Shows the row/column profiles. Different layout options are available.

The zoom sliders can be used to zoom selectively in the horizontal or vertical direction, respectively. If you zoom in horizontally, the values in the matrix are displayed.


Chapter 20.2

General functionality

20.2.1 Views in the matrix mining

The *Row/Column Text Fields panels* show the information fields of the rows/columns. The drop down lists located at the bottom of these panels allow the displayed information in these panels to be changed. If another information field is selected from the list, the information in the panel is updated.

Alternatively, changing the displayed text fields can also be done with the *Text fields* options listed under the menu-item *View*.

With *View > Flip* () one can switch the rows in the expression matrix into columns and back.

The content of a highlighted column entry can be changed with *Textfields > Edit column....* This action calls the *Entry* window. The same window is called when double-clicking on the entry. See [3.3.4](#) for more information about the *Entry* window.

The content of one or more selected row entries can be changed with *Textfields > Edit row....* This calls the *Edit Character Field* dialog box. The same window is called when double-clicking on a character.



Since the content of the character name field cannot be changed, make sure to select another character field from the drop down list in the *Row Text Fields panel*.

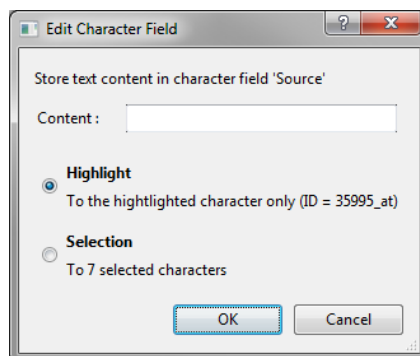


Figure 20.2.1: Change the content of one or more row entries.

Changes will be stored in the row field selected in the *Row Text Fields panel*.

The new *Content* can be saved for the highlighted row entry (*Highlight*) or for all selected row entries (*Selection*).

20.2.2 Selections in the matrix mining

The selection of row and column entries is very similar except that one selection will act on the rows of the data set and the other selection will act on the columns of the data set. Selecting entries can be done in any view in the *Matrix Mining* window and changing selections is instantly synchronized with every view in the *Matrix Mining* window and with the BioNumerics *Main* window and *Comparison* window.

20.2.2.1 Manual selection functions

In the *Main view* of the *Matrix Mining* window, rows/columns are selected manually by holding the **Ctrl**-key and left-clicking on the rows/columns in the *Text Fields panels*. Check boxes for selected rows/columns are indicated as ☒. Selected rows/columns are unselected in the same way.

In order to select a range of rows/columns, select the first row/column of the range and select the last row/column of the range while holding down the **Shift**-key.

The complete selection is cleared with **Selection > Clear selection** (🗑️, **F4**).

The selection of the rows is cleared with **Selection > Clear row selection** (**F5**).

The selection of the columns is cleared with **Selection > Clear column selection** (**Shift+F5**).

A selection in rows/columns is inverted with **Selection > Invert row selection** and **Selection > Invert column selection** respectively. All rows/columns that are currently selected will be unselected and all rows/columns that are currently not selected will be selected.

A double-click in the check box bar in the *Row Text Fields panel*/*Column Text Fields panel* will select all rows/columns at once.

20.2.2.2 Automatic selection functions

In addition to manually selecting rows and columns, rows and columns can be selected automatically using a simple and intuitive query tool.

To launch the tool for a query on the rows select **Textfields > Query row fields...** (**Ctrl+Q**). To launch the tool for a query on the columns select **Textfields > Query column fields...** (**Ctrl+Shift+Q**). These actions will open the *Find Text* dialog box (see Figure 20.2.2).

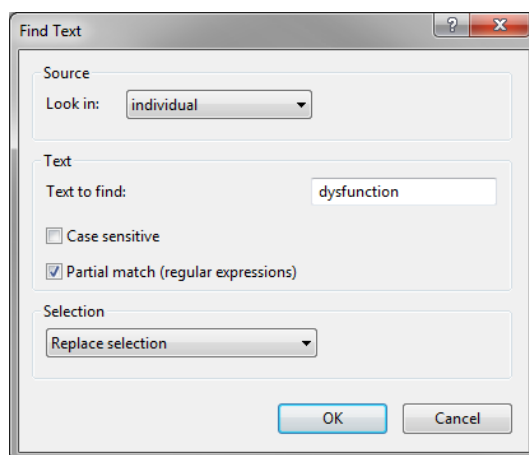


Figure 20.2.2: The *Find Text* dialog box.

All information fields will default be included in the search. The search can be restricted to one text field,

by using the **Source** drop down list.

When checking the option **Case sensitive**, a distinction is made between upper- and lowercase.

Check the option **Partial match (regular expressions)** if you wish to search for a part of a word.

It is possible to replace the current selection (**Replace selection**), to add the results of the query to the current selection (**Add to selection**), or to search within the current selection (**Find in selection**).

Pressing <OK> starts the query.

20.2.3 Groups in the matrix mining

Groups can make a visual distinction between rows/columns that belong together and rows/columns that do not belong together. Groups are the necessary input for multidimensional scaling and for a number of statistic tools like e.g. ANOVA. Deciding which rows/columns belong to a group can be done in several ways based on analyses or queries. Groups can be made and handled from every view. If entry groups have been defined in the *Comparison* window, these groups are automatically transferred to the *Matrix Mining* window.

If you want to create/edit row groups, the command **Groups > Edit row groups...** (🎨, Ctrl+G) will open the *Groups* dialog box (see Figure 20.2.3). If you want to create/edit column groups, the command **Groups > Edit column groups...** (🎨, Ctrl+Shift+G) will open the same dialog. Initially no groups are defined, and an empty dialog box will pop up.

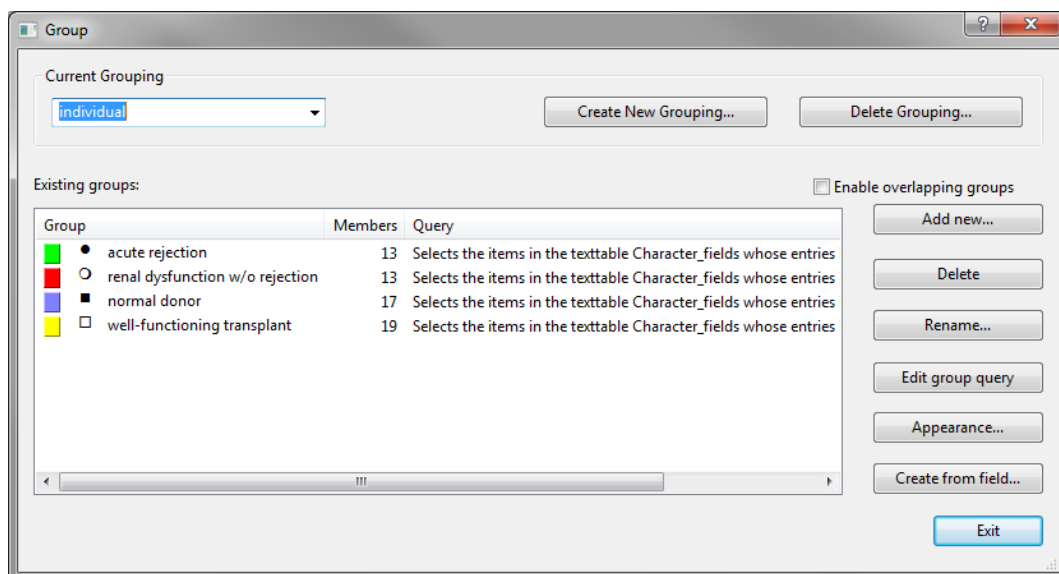


Figure 20.2.3: The *Groups* dialog box.

With the <**Create New Grouping**> button a new grouping can be defined.

A **Name** can be entered or an information field name can be selected from the pull down menu. Optionally a **Comment** can be entered.

When selecting an information field name from the pull down menu, the *Generate group from field* dialog box pops up (see Figure 20.2.5). This dialog box can also be called from the dialog box with the <**Create from field**> button in the *Groups* dialog box.

In the *Generate group from field* dialog box, groups related to an information field can be created.

The name of the information field that should be the source for the groups can be selected with the **Text field**

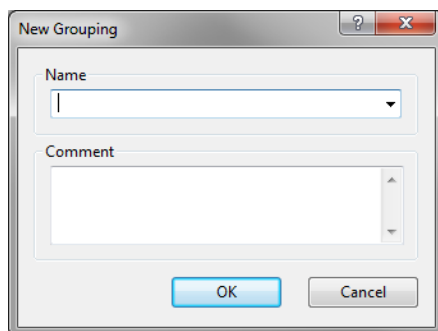


Figure 20.2.4: Define a new grouping.

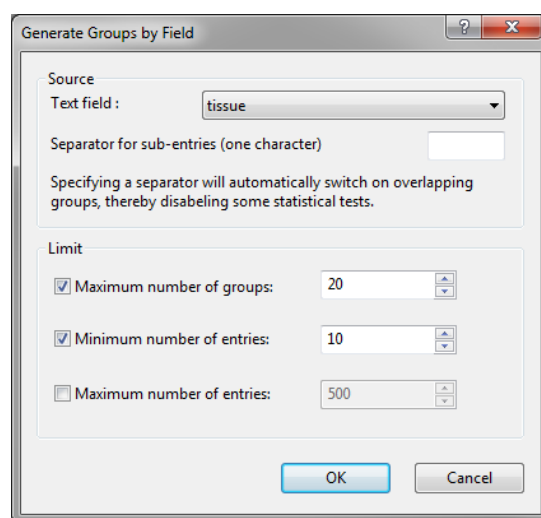


Figure 20.2.5: Create groups based on a field.

pull down menu in the *Source panel*.

When creating groups from fields with multiple values, you can enter the delimiter for the values in the field **Separator for sub-entries**. When using a delimiter, the option **Enable Overlapping Groups** is turned on in the *Groups* dialog box.

In the *Limit panel*, a limit to the number of groups can be set by specifying a **Maximum number of groups** (standard setting: 20), with a **Minimum number of entries** (standard setting: 10), and optionally a **Maximum number of entries** (standard setting: disabled).

After confirmation of the creation of the groups, the groups are listed in the *Existing groups panel* together with their color code, symbol, number of members and query.

It is possible to work with overlapping groups by selecting **Enable overlapping groups**. This option offers the possibility to assign a row or column to more than one group. Please note that checking this option will disable some statistical tests.

With the option **<Delete Grouping>** the currently selected grouping can be removed.

By clicking **<Add New>**, a new group marker will be created. A name for the group will be prompted for, and a color code and symbol will be suggested by the program. The groups will be added to the list of existing groups.

If a new group is created with the **<Add New>** button, and a number of entries are selected in the main window, the selected entries are automatically assigned to the new group, by answering **<Yes>** in the confirmation box that pops up.

The option **<Delete>** removes the currently selected group in the *Existing groups panel*.

With the **<Rename>** button, a new name can be given to the currently selected group.

Clicking the **<Appearance>** button will open the *Group Appearance* dialog box (see Figure 20.2.6).

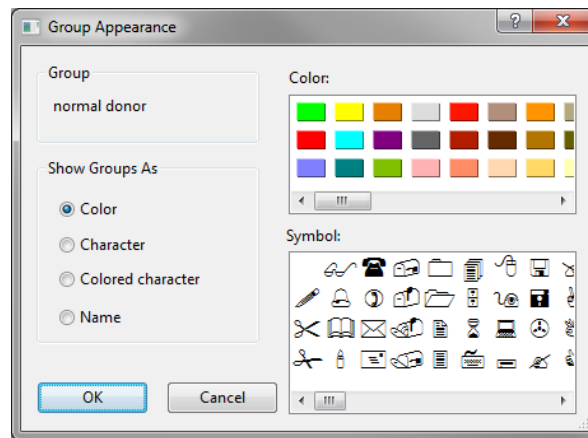


Figure 20.2.6: Changing the layout of a group.

The appearance of the groups in the views is set in the *Show Groups As panel*. The default view is set to **Color**. For the selected group (indicated in the *Group panel*), the color code can be changed in the *Color panel* and the symbol in the *Symbol panel*.

Once groupings and their group markers are defined, the pull down arrow next to the group buttons in the Group bar of the *Groups panel* will show the available row and column groupings.

A selection of columns is pasted into a group of the currently selected grouping with the command **Groups > From column selection...** (🔗). A selection of rows is pasted into a group of the currently selected grouping with the command **Groups > From row selection...** (🔗). If no group markers are defined, this will open the *Groups* dialog box (see Figure 20.2.3). If group markers are already defined, this will open the *Selection to group* dialog box (see Figure 20.2.7).

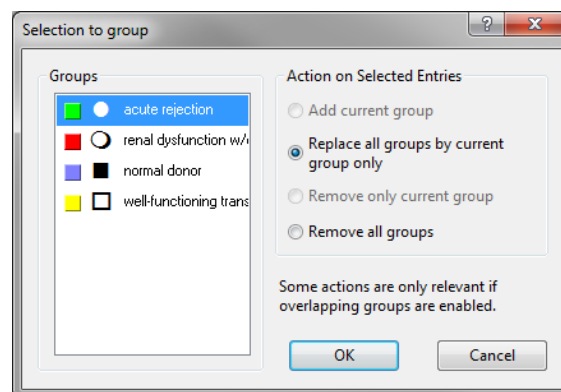


Figure 20.2.7: Perform an action on the selected entries.

In the *Groups panel* the groups are listed.

If overlapping groups are enabled, it is possible to add the selected entries to the selected group in the *Groups panel* (check **Add current group**), or to replace the group for the selected entries with the currently selected one (check **Replace all groups by current group only**). It is also possible to remove the selected group (check **Remove only current group**), or to remove all groups (check **Remove all groups**).

If overlapping groups are disabled, you can replace the group for the selected entries with the currently

selected one (check **Replace all groups by current group only**), or remove the group marker (check **Remove all groups**).

It is also possible to create groups based on the outcome of an analysis. This is described together with the analyses.

In default configuration, the *Groups* panel appears as tabbed view together with the *Layers* panel. The drop down menu in the *Groups* panel lists all groupings defined in the database. When a grouping is selected from the drop down menu, the groups defined for that grouping are listed in the panel below.

Rows/columns belonging to a group can be selected from within this window. To select all members belonging to a group, **Ctrl** +click on the square next to the group color. The square is highlighted and the members of the group are selected. Use the **Shift**-key to select more than one group at a time.

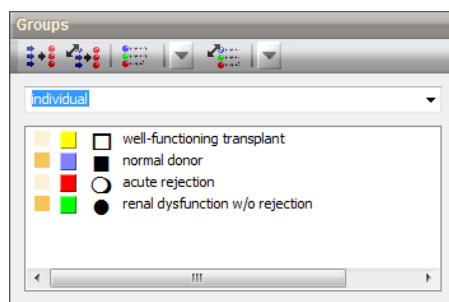


Figure 20.2.8: The *Groups* panel.

20.2.4 Subsets in the matrix mining

When dealing with a lot of data, one will meet the need to concentrate on a part of the available data. The *Matrix Mining* window therefore supports the concept of subsets. Just like groups, subsets can be made and handled from every view.

The created subsets are displayed in the *Subsets* panel. The tree shows the dependencies between the different subsets. The selected subset is the active set, i.e. the set that is shown in the selected view.

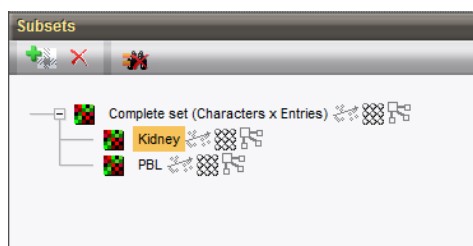


Figure 20.2.9: The *Subsets* panel.

With the command **Subset > Selection to subset...** (icon) a new subset can be created containing the rows and columns that are selected. This selection can be a result of a query, manual selection, or an analysis. The *Create Subset* dialog box will pop up, prompting for some settings (see Figure 20.2.10).

A very general name for the new subset is proposed, but a more meaningful name can be entered in the **Name** field if desired. In the **Comment** field some optional information on the subset can be entered.

The new subset should get an appropriate place in the *Subset tree* of the *Subsets* panel. It is standard considered as **Child of the complete set**, but it can also be created as **Child of the current subset** if this option is checked.

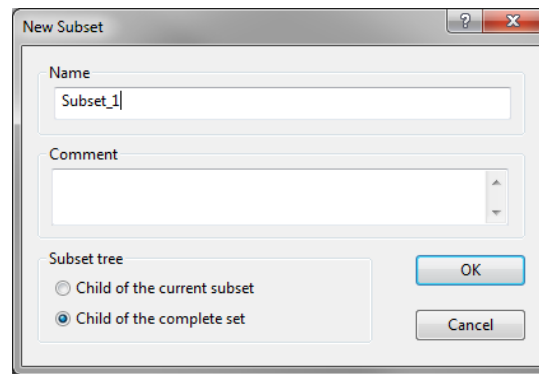


Figure 20.2.10: The *Create Subset* dialog box.

If only a selection of rows is made, all columns are by default included in the new subset. Likewise, if only a selection of columns is made, all rows are by default included in the new subset.

A subset selected in the *Subsets* panel is deleted with **Subset > Delete subset** (✖).

20.2.5 Scopes and aspects in the matrix mining

20.2.5.1 Introduction

The data matrix in the *Matrix Mining* window can be organized in such a way that each row or column has more than one replicate. In some cases, you might want to work with the combined information of these replicates. These different ways of looking at the data are what we call *aspects*. A view of a data matrix consists of two aspects: a row aspect and a column aspect. Two aspects that make up one data matrix is called the *scope* of the data. The standard scope is *Characters x Entries*. The aspects are shown in the *Subsets* panel between brackets next to the name of the subset (see Figure 20.2.9).

20.2.5.2 Collapse by field

Rows and columns can be collapsed based on the content of an annotation text field with **Subset > Collapse Aspect > By field...** A new dialog is displayed (see Figure 20.2.11).

In the *Source panel*, you need to indicate which subset and aspect (**Row** or **Column**) you want to use to derive your new aspect from. The selected text field in the *Text field panel* will be used to collapse the rows or columns. All rows/columns that have exactly the same content in this annotation field are collapsed to one row/column in the new scope. The way the values of the collapsed rows/columns in the parent aspect are summarized to one value in the new aspect can be specified in the *Collapse algorithm panel* of the dialog box. Three options are available:

- **Average:** The values in the new aspect are the average values of all rows/columns in the parent aspect that were collapsed to the same row/column.
- **Average weighted by errors:** The values in the new aspect are the weighted averages (by errors) of the rows/columns in the parent aspect that were collapsed to the same row/column.
- **Median:** The values in the new aspect are the median values of the rows/columns in the parent aspect that were collapsed to the same row/column.

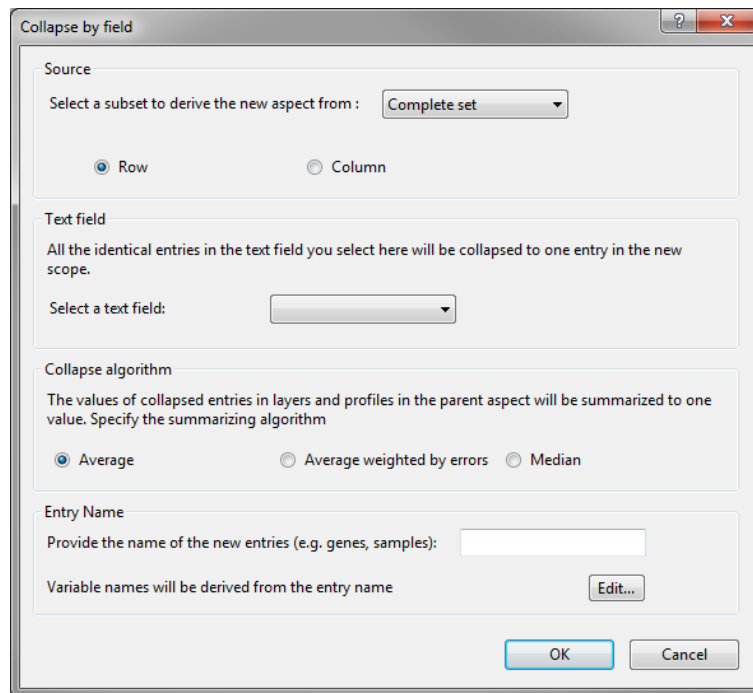


Figure 20.2.11: The *Collapse by field* dialog box.

In the lower panel, the new **Entry Name** needs to be provided. By default, all variable names (Scope, Aspect, Text Fields, Grouping, Selection, and Complete set) will be derived from this name. Each variable name can be changed by pressing the <Edit> button.

20.2.5.3 Collapse by group

All rows/columns belonging to (overlapping) groups can be collapsed to one row/column with the command **Subset > Collapse Aspect > By group...**. This action calls a new dialog (see Figure 20.2.12).

In the *Source panel*, you need to indicate which subset and aspect (**Row** or **Column** aspect) you want to use to derive your new aspect from. In the *Grouping panel*, select the grouping that will be used to collapse the rows/columns. All rows/columns that belong to the same group will be collapsed to one row/column in the new scope. The way the values of the collapsed rows/columns in the parent aspect are summarized to one value in the new aspect can be specified in the *Collapse algorithm panel* of the dialog box. Three options are available:

- **Average:** The values in the new aspect are the average values of the rows/columns in the parent aspect that were collapsed to the same row/column.
- **Average weighted by errors:** The values in the new aspect are the weighted averages (by errors) of the rows/columns in the parent aspect that were collapsed to the same row/column.
- **Median:** The values in the new aspect are the median values of the rows/columns in the parent aspect that were collapsed to the same row/column.

In the lower panel, the new **Entry Name** needs to be provided. By default, all variable names (Scope, Aspect, Text Fields, Grouping, Selection, and Complete set) will be derived from this name. The variable name can be changed by pressing the <Edit> button.

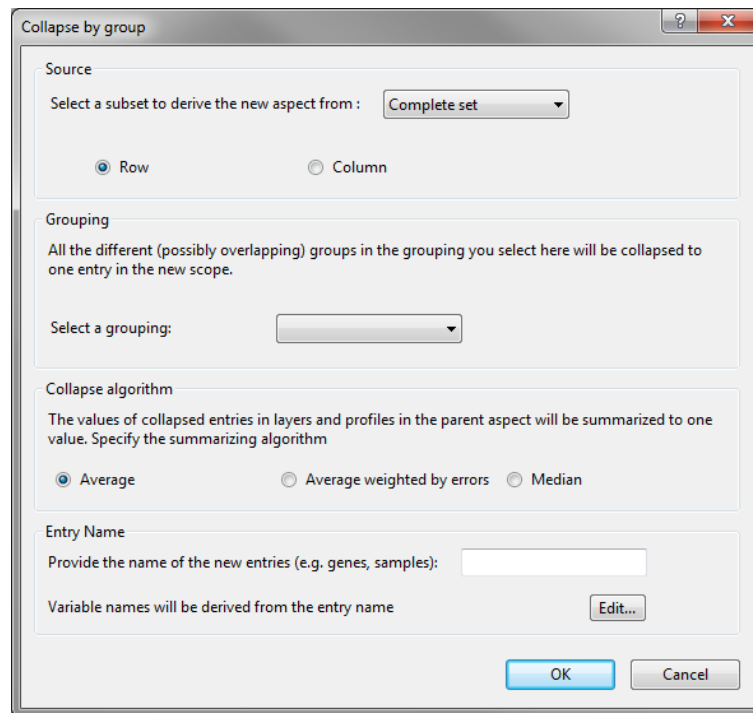


Figure 20.2.12: The *Collapse by group* dialog box.

20.2.5.4 Transfer text fields

Each aspect has its own textfields. When a new aspect is derived, it only contains those annotation textfields for which the information content of the annotation text field is the same for all rows/columns that are collapsed to the same row/column. The content of the textfields is automatically transferred to the new aspect. Besides this automated transfer of data information, the software also offers the possibility to manually transfer an identifier and its content from one aspect to another.

Manually transferring an identifier and its content from one aspect to another is done with *Textfields > Transfer text fields....* This action calls a new dialog (see Figure 20.2.13).

In the first panel of the dialog, the direction needs to be selected (*Rows* or *Columns*). In the *Transfer panel* the source and destination aspects need to be specified. The drop down lists are updated according to the selected direction. In the *Text fields panel*, select the identifier that holds the information you want to transfer. The content of the identifier will be transferred to the new aspect according to the option selected in the *Rules panel*:

- **Majority:** Only if the majority of the rows/columns that were collapsed to the same row/column have the same content, the text field information will be transferred to the destination row/column.
- **All:** Only if all rows/columns that were collapsed to the same row/column have the same content, the text field information will be transferred to the destination row/column.
- **All non empty:** Only if all the non-empty rows/columns that were collapsed to the same row/column are identical, the text field information will be transferred to the destination row/column.

20.2.5.5 Transfer selection

A selection in an aspect is by default only displayed in each scope that contains this aspect. The *Matrix Mining* window offers the possibility to transfer a selection between different aspects.

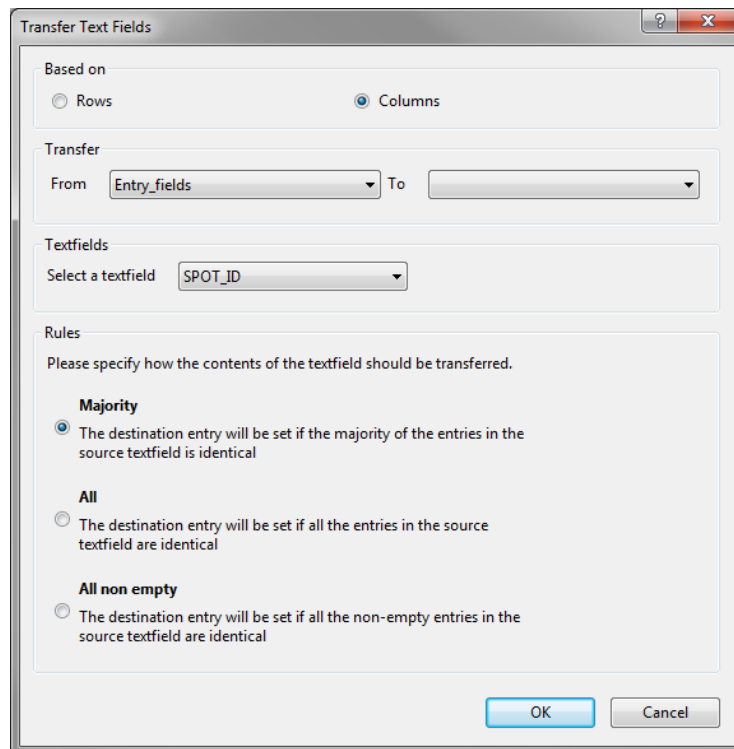


Figure 20.2.13: The *Transfer Text Fields* dialog box.

Transferring a selection between different aspects is done with *Selection > Transfer selection....* This action calls a new dialog (see Figure 20.2.14).

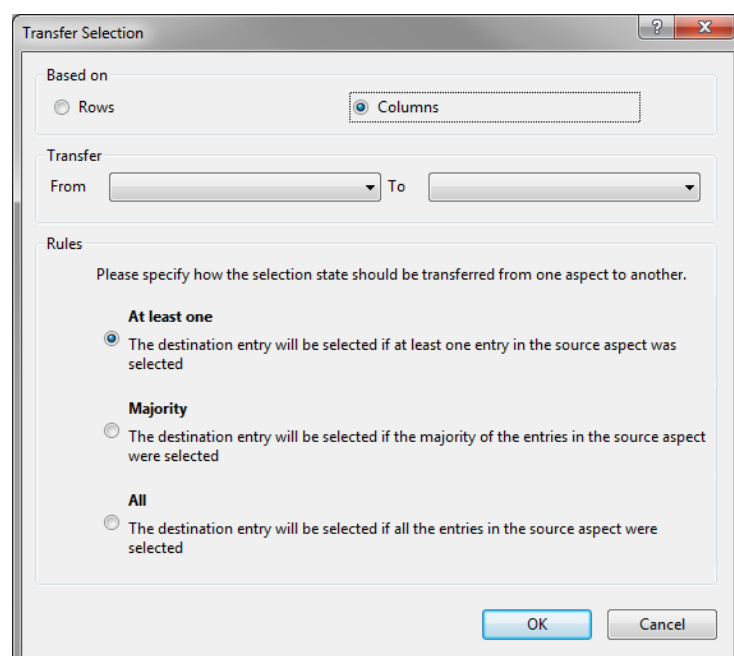


Figure 20.2.14: The *Transfer Selection* dialog box.

In the upper part of the window, the direction needs to be selected (**Rows** or **Columns**). In the *Transfer panel* the source and destination aspects need to be selected. The drop down lists are updated according to the selected direction. The destination rows/columns are selected according to the option selected in the *Rules panel*:

- ***At least one:*** If at least one row/column in the source aspect is selected, the destination row/column will be selected in the destination aspect.
- ***Majority:*** Only if the majority of the rows/columns in the source aspect are selected, the destination row/column will be selected.
- ***All:*** The destination row/column will be selected if all the rows/columns in the source aspect are selected.


Chapter 20.3

Layers

20.3.1 Introduction

Layers contain the actual values identified with combinations of rows and columns. In the *Matrix Mining* window, several layers can be defined containing e.g. different ways of expressing the actual data (e.g. log transformed or not), or different ways of normalizing the data (e.g. centered around the mean or centered around the median). Layers can be made and handled from any view. The commands that act on the layers can be found under the **Layer** menu.

The *Layers* panel, located in the upper left corner of the *Matrix Mining* window displays the layers. The layer that is selected is the active layer, i.e. the layer that is shown in the *Main view* and that is standard used in calculations and analyses.

Clicking the column properties button () located on the right hand side in *Layers* panel gives access to functions to hide, freeze, or move information fields.

20.3.2 General

A layer selected in the *Layers* panel is deleted with **Layer > Delete layer...** (.



If only one layer is present in the *Layers* panel, this layer cannot be deleted.

20.3.3 Data Transformation

20.3.3.1 Log Transformation

The option **Layer > Data Transformation > Log Transform...** transforms the values in a layer to logarithmic values. A new dialog opens (see Figure 20.3.1).

Following settings need to be specified:

- The **Source** layer.
- The **Destination** layer: The user can choose to overwrite the source layer, or to specify a new destination layer to store the log values.
- The **Scope**.

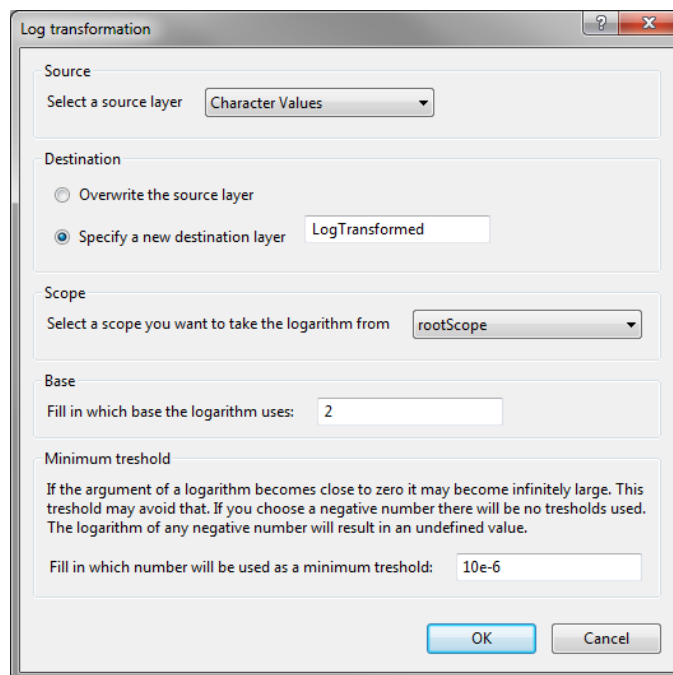


Figure 20.3.1: The *Log transformation* dialog box to transform the values to logarithmic values.

- The **Base**: The default option is base 2, but the user can change this if desired.
- The **Minimum threshold**: A minimum threshold may avoid infinitely small results.

20.3.3.2 Average values

The option **Layer > Data Transformation > Average...** can be used to calculate the average of a number of layers. A new dialog opens (see Figure 20.3.2).

The **Scope** and the **Source layers** are prompted for and the results will be stored in the **Destination** layer. In the *Average type panel*, the average type can be set:

- **Unweighted**: This option gives the same weight to all the measurements, regardless of their quality.
- **Weighted by errors**: In this case, a weight function is used based on the error bars to favor measurements with a better quality. The inverse square of the errors on the expression values is used as weight.

20.3.4 Normalization

20.3.4.1 Normalize columns/rows

With **Layer > Normalization > Normalize columns...** (🔧) and **Layer > Normalization > Normalize rows...** (🔧) the settings for the global normalization of the columns and rows respectively can be entered (see Figure 20.3.4 and Figure 20.3.3).

- The operation will be performed on the layer indicated in the *Source panel*.

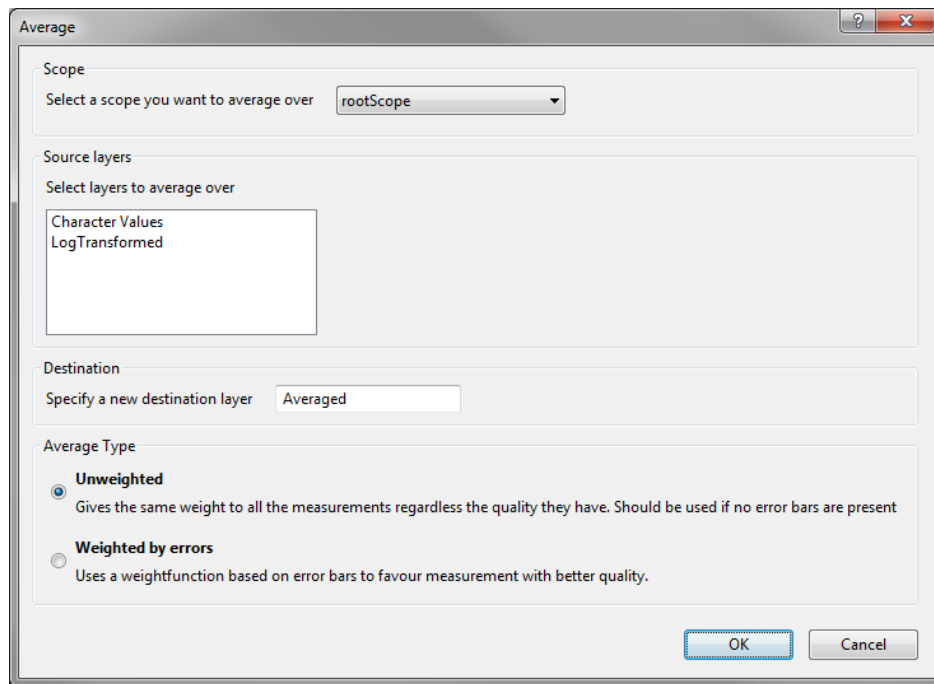


Figure 20.3.2: Average values.

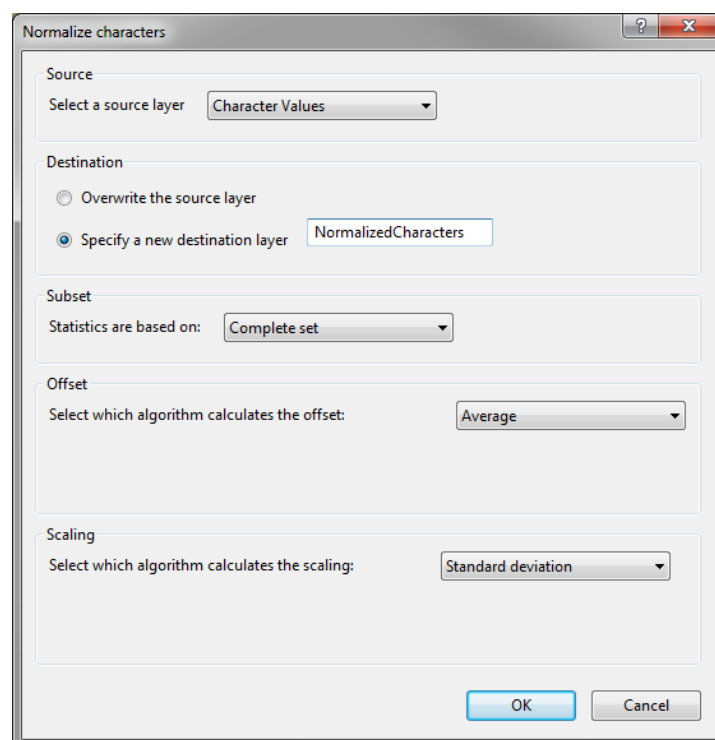


Figure 20.3.3: Normalize rows.

- The results can be stored in the source layer or a new destination layer can be specified.
- In the *Subset panel*, the subset can be selected from the list of available subsets present in the session.
- All expression values (x) in the layer will be replaced by a new value calculated as $(x - \text{offset})/\text{scale}$. Following algorithms are available for the calculation of the *offset*: **Average**, **Median**, **Percentile**, **Fixed array**, **Constant value**. For the calculation of the *scale*, following algorithms are available:

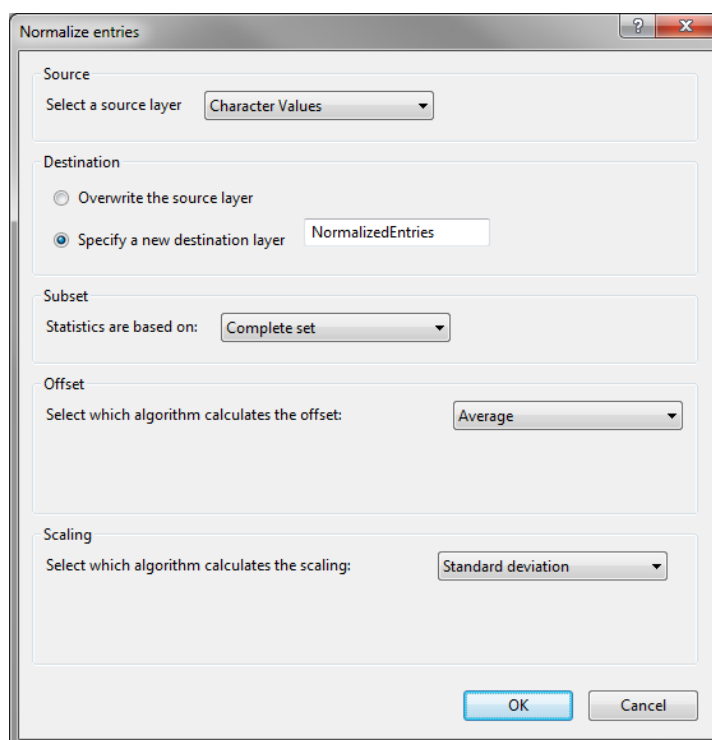


Figure 20.3.4: Normalize columns.

Standard deviation, Root mean square, Mean absolute deviation, Median absolute deviation, Fixed array, Constant value.



Scaling a layer can create missing values if this comes down to a division by zero. This is possible if e.g. the standard deviation (or another factor) is used for the scaling and if the value of the standard deviation (or the other factor) for a certain row or column happens to be zero.

20.3.4.2 Quantile Normalization

This non-parametric normalization method assumes that the distribution of row abundances is nearly the same in all columns. For convenience the pooled distribution of the rows on all columns is taken. Then to normalize each column, for each value, the quantile of that value in the distribution of the row values is calculated. Next, the original value is transformed into that quantile's value.

With the command **Layer > Normalization > Quantile normalization...** the settings can be specified (Figure 20.3.5).

- The operation will be performed on the layer indicated in the *Source panel*.
- The results can be stored in the source layer or a new destination layer can be specified.
- In the *Subset panel*, the subset can be selected from the list of available subsets.

20.3.4.3 Remove effect

With the option **Layer > Normalization > Remove effect...** biases connected to a group effect as a result from an ANOVA model can be removed. The command calls a new window (Figure 20.3.6).

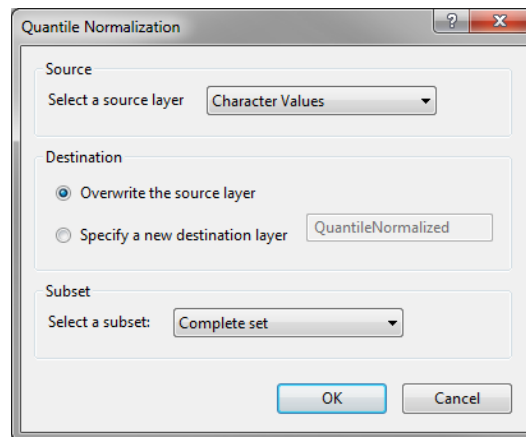


Figure 20.3.5: The *Quantile Normalization* dialog box.

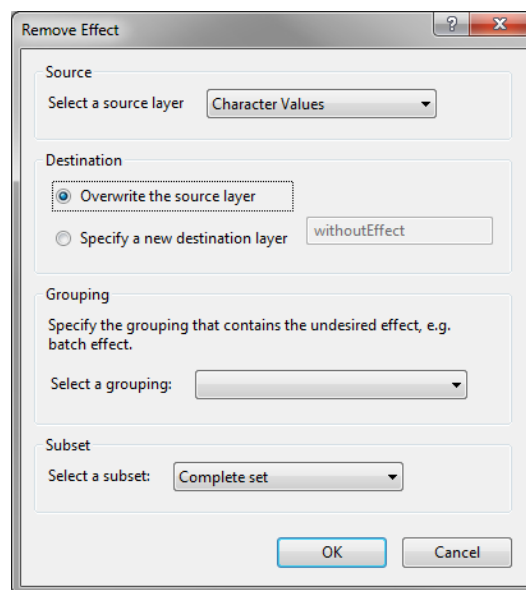


Figure 20.3.6: The *Remove Effect* dialog box.

- The operation will be performed on the layer indicated in the *Source panel*.
- The results can be stored in the source layer or a new destination layer can be specified.
- In the *Grouping panel*, the grouping that contains the undesired effect needs to be specified.
- In the *Subset panel*, select the subset from the list of available subsets.

20.3.5 Filtering

20.3.5.1 Clip internal

The command **Layer > Filtering > Clip internal...** (📏) sets the values of a layer to absent when the values of this layer are within certain boundaries. A new dialog is called with this command, prompting for some settings (see Figure 20.3.7).

- In the upper panel, the *Source layer* needs to be specified.

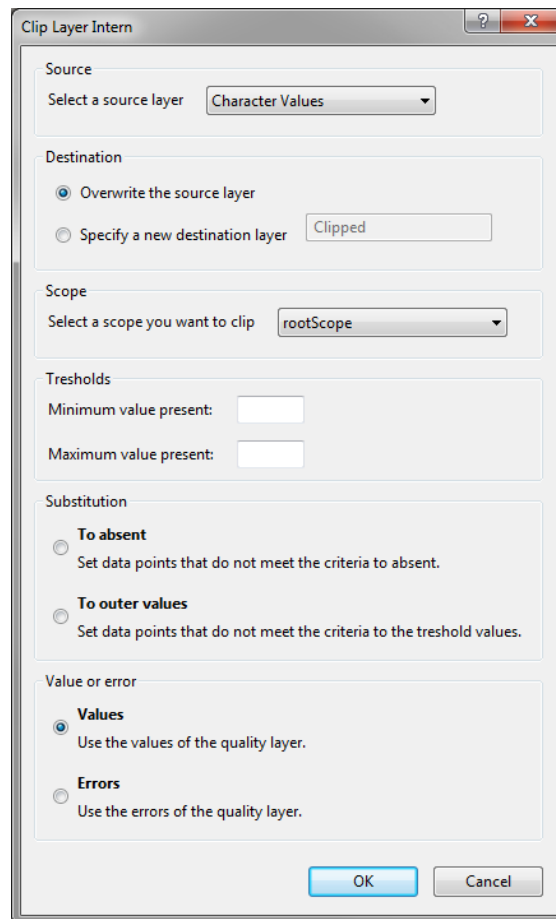


Figure 20.3.7: Clip if values are within certain boundaries.

- The results can be stored in the source layer or a new destination layer can be specified.
- In the *Scope panel*, the scope is selected from the list of available subsets.
- The boundaries can be specified in the *Thresholds panel*.
- The substitution method can be selected in the *Substitution panel*: data points that do not meet the criteria can be set **To absent**, or the data points that do not meet the criteria can be set to the threshold values (= **To outer values**).
- In the lower panel, the user can choose to **Use the values of the quality layer**, or **Use the errors of the quality layer**.

20.3.5.2 Clip external

The command **Layer > Filtering > Clip external...** transfers values of a source layer to a destination layer depending on whether the values of the errors of a quality layer on that position is within certain boundaries. Otherwise the destination layer will get an absent value. A new dialog is called with this command, prompting for some settings (see Figure 20.3.8).

- In the upper panel, the **Source layer** and **Quality layer** need to be specified.
- The results can be stored in the source layer or a new destination layer can be specified.
- In the *Scope panel*, the scope is selected from the list of available subsets.

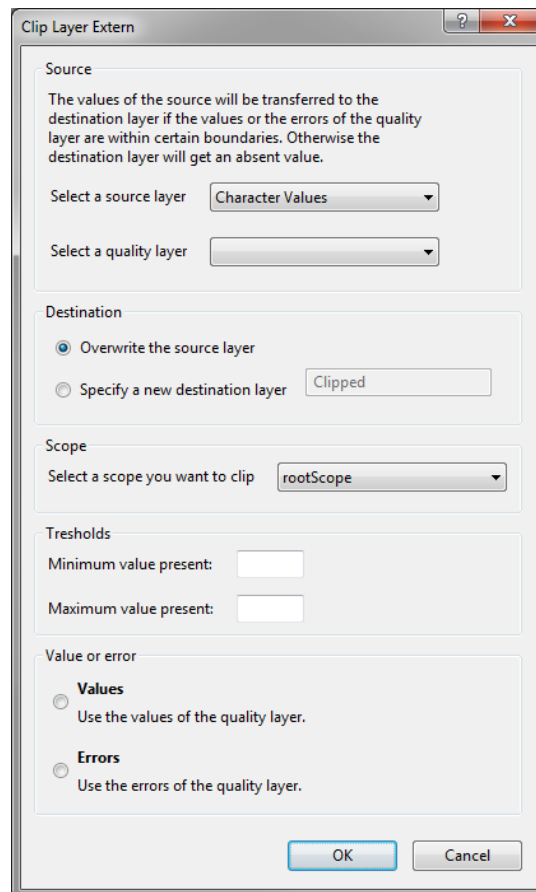


Figure 20.3.8: Clip if values are within certain boundaries.

- The boundaries can be specified in the *Thresholds* panel.
- In the lower panel, the user can choose to use the **Values** of the quality layer, or use the **Errors** of the quality layer.

20.3.5.3 Absent values

With the option **Layer > Filtering > Copy absent values...** absent values are copied to a destination layer. The command calls a new dialog prompting for some settings (see Figure 20.3.9).

- In the upper panel, the **Source layer** needs to be specified. Absent values in this layer will be absent in the destination layer.
- The **Presence Layer** will be used to copy the absent values from.
- The results can be stored in an existing layer or a new destination layer can be specified.
- In the *Scope* panel, the scope is selected from the list of available subsets.

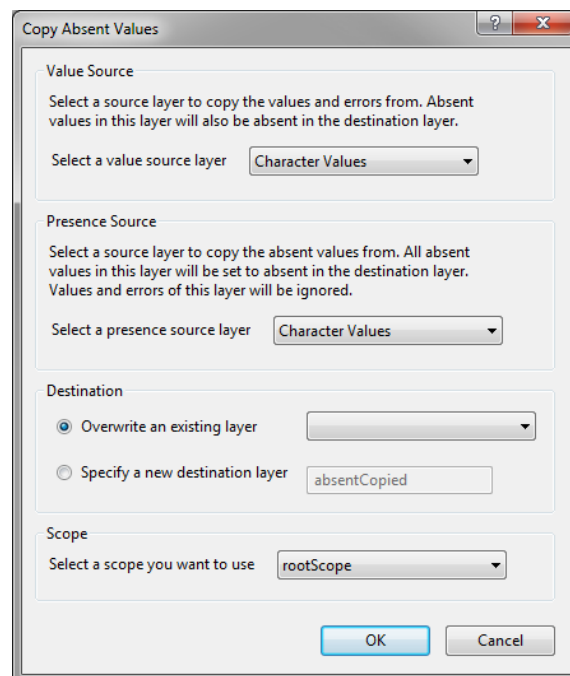


Figure 20.3.9: Copy absent values.

Chapter 20.4

Profiles

20.4.1 Introduction

A *row profile* contains values over several columns, whereas a *column profile* contains values over several rows. Profiles can have various origins: profiles extracted directly from the data matrix, principal components derived from a PCA, discriminants derived from a discriminant analysis, an average profile from a dendrogram branch, an average profile from a cell in a SOM or partitioning, etc.

In general terms profiles can be seen as one-dimensional functions that have column or row variables as an argument. This number of arguments determines their size. Profiles can be created and shown in every view of the *Matrix Mining* window. Several types of plots are available and there are also a number of statistics to apply on profiles.

20.4.2 Creating profiles

A selection of columns or rows are turned into separate profiles with the command **Profiles > Add selection as profiles** (📊). The profiles are graphically displayed in the row/column *Profiles panel* and are summarized in the *Profiles panel*.

A selection of columns or rows is turned into an averaged profile with **Profiles > Plot averaged selection** (📊).

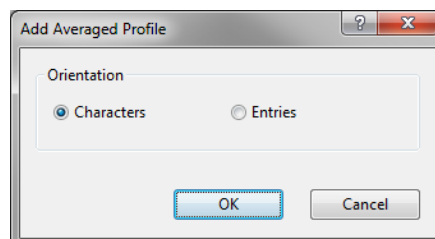


Figure 20.4.1: Adding an averaged row or column profile.

An averaged profile can be made for a selection of rows or columns.

Column or row profiles containing a statistic are created with the *Statistics Wizard* dialog box. This wizard is opened with the command **Profiles > Statistics wizard...** (📊). See 20.4.4 for more information.

20.4.3 Displaying and handling profiles

A column profile is displayed in the *Profiles panel*, situated left from the *Dendrogram*, *Text fields* and *Matrix panel*.

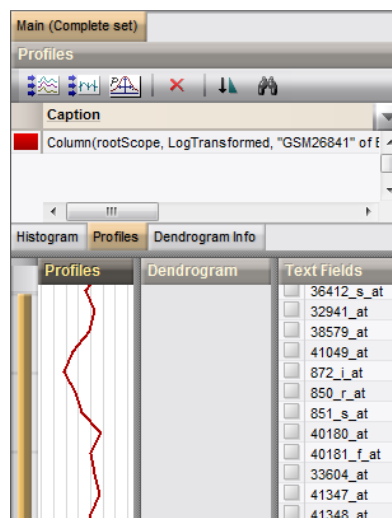


Figure 20.4.2: Display of a column profile.

A row profile is displayed in the *Profiles panel*, appearing in default configuration as tabbed view together with the *Dendrogram panel* in the upper right corner of the *Main view*.

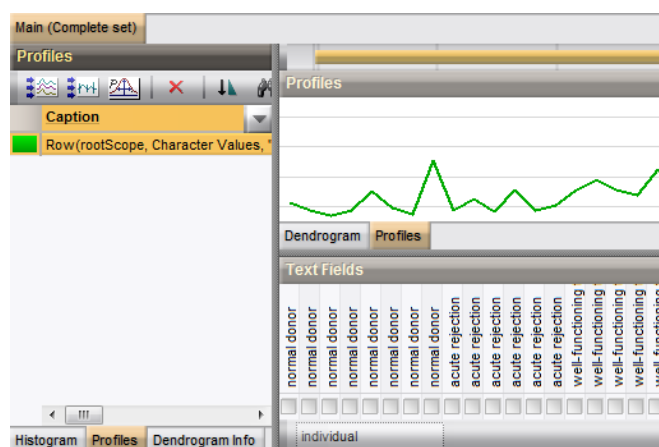


Figure 20.4.3: Display of a row profile.

The *List Profiles tab*, appearing in default configuration as tabbed view together with the *Histogram* and *Dendrogram panel*, lists the profile(s) shown in the row/column *Profiles*, their color code and a brief description of their contents.

A profile is selected by clicking on its name in the left upper *Profiles panel* or on its graphical representation. The name is then highlighted and the profile itself appears in a thick line.

When right clicking the mouse in the profile area, you can choose from the options shown in Figure 20.4.4.

- With the **Show as Numbers** option checked, the graphical representation becomes a numerical one, showing all the values.
- If the option **Show errors** is checked, the error bars (if present) are shown.

- With **Show grid**, you can toggle on/off grid lines.

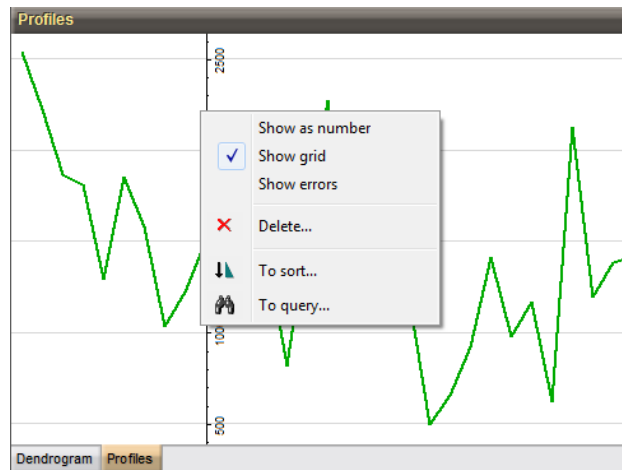


Figure 20.4.4: Profile options.

A selected profile is deleted with the command **Profiles > Selected profile > Delete...** (🗑️). The option **Delete** in Figure 20.4.4 does the same.

With **Profiles > Selected profile > To sort...** (📊), the currently selected profile is sorted with monotonically increasing values. The same is achieved when selecting **To sort** in Figure 20.4.4.

The *Profile To Query* dialog box is launched with **Profiles > Selected profile > To query...** (🔍) (see Figure 20.4.5).

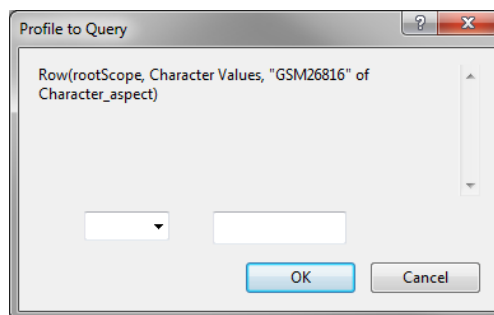


Figure 20.4.5: Selecting entries based on a query.

Entries for which the value of the selected row/column profile is smaller, (not) equal to or higher than a predefined value can be selected.

20.4.4 Statistics wizard

Creating profiles with statistical information on a selected data set is done by means of the *Statistics Wizard*.

The *Statistics Wizard* is found behind the command **Profiles > Statistics wizard...** (🔧).

The first window of the *Statistics Wizard*, the user needs to indicate which **Orientation** should be used:

- **Calculate a statistic for each Row:** This means that the columns will be used as population.
- **Calculate a statistic for each Column:** This means that the rows will be used as population.

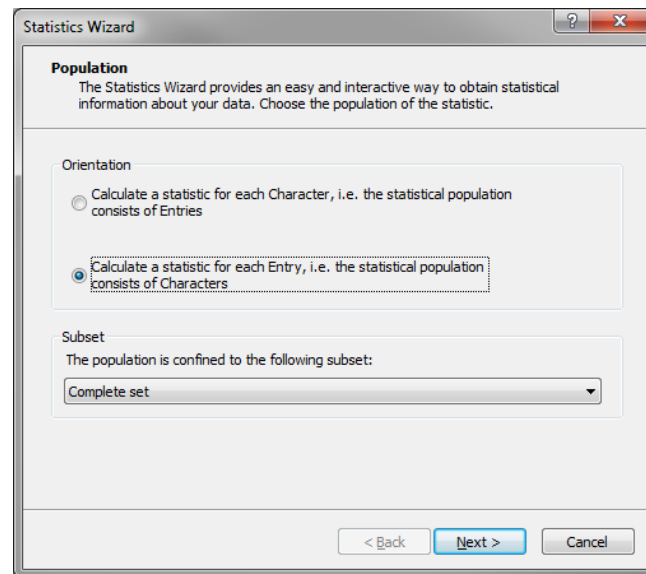


Figure 20.4.6: The *Statistics Wizard*: step 1.

The *Subset* from which the data has to be taken is can be selected from a drop down list.

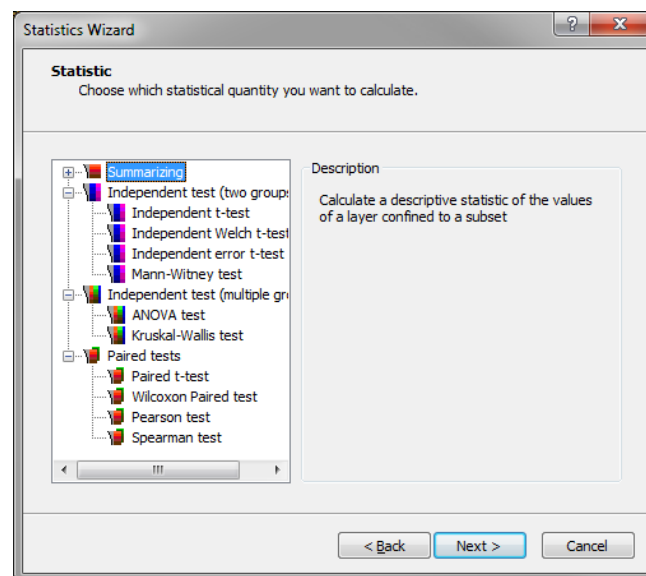


Figure 20.4.7: The *Statistics Wizard*: step 2.

In the second step of the *Statistics wizard*, a statistic needs to be selected:

1. SUMMARIZING

- **Mean:** The mean of the population.
- **Median:** The median of the population.
- **Percentile:** A given percentile of the population. The percentile is entered in step three.
- **Standard deviation:** The standard deviation of the population, this is the square root of the variance.
- **Root mean square (RMS):** The root mean square of the population, this is the square root of the mean of the squares of a population.

- **Coefficient of Variation (CV):** The coefficient of variation of a population: i.e. the standard deviation divided by the mean.
- **Mean absolute deviation (MAD):** The mean absolute deviation of a population.
- **Mean Error:** The mean of the errors of a population.
- **Fraction absent values:** The fraction of absent values in a population.
- **Median error:** The median of the errors of the population.
- **Error Percentile:** A given percentile of the errors of a population. The percentile is entered in step three.
- **Kolmogorov test of normality:** The p-value associated with the Kolmogorov-Smirnov test for normality of the members of a population.
- **Minimum Value:** Minimum value of a population.
- **Maximum Value:** Maximum value of a population.
- **Highest fold change:** Highest fold change of a population.
- **Absolute expression value:** Absolute expression value of a population.
- **Median absolute deviation (MAD):** The median absolute deviation of a population, this is sometimes used as alternative for the standard deviation. This is the median of the deviation between each data point and the overall median of the population.

2. INDEPENDENT TEST (2 GROUPS)

- **Independent t-test:** Tests whether the mean of the two samples are equal or not. The test assumes normality of the data.
- **Independent Welch t-test:** The p-value associated with the T-test for equal means of independent samples. The independent samples are indicated by different non-overlapping groups and are taken from the same layer.
- **Independent error t-test:** This test is very similar to the independent T-test, but in this case the error bars are taken into account when calculating the variances. It is also possible to use weighted means in the test.
- **Mann-Witney test:** The p-value associated with the Mann-Witney test (or Wilcoxon rank sum test) for equal means of independent samples. The independent samples are indicated by different non-overlapping groups and are taken from the same layer.

3. INDEPENDENT TEST (MULTIPLE GROUPS)

- **ANOVA test:** The p-value associated with the ANOVA test (analysis by variance) for equal means of independent samples. The test assumes normality of data.
- **Kruskal-Wallis test:** The p-value associated with the Kruskal-Wallis test for equal means of independent samples.

4. PAIRED TESTS

- **Paired t-test:** The p-value associated with the t-test for equal means of paired samples. The paired samples are taken from two separate layers. The test assumes normality of the data.

- **Wilcoxon paired test:** The p-value associated with the Wilcoxon test for equal means of paired samples. The paired samples are taken from two separate layers.
- **Pearson test:** The p-value associated with the Pearson test for correlation between paired samples. The paired samples are taken from two separate layers. The test assumes normality of the data.
- **Spearman test:** The p-value associated with the Spearman test for correlation between paired samples. The paired samples are taken from two separate layers.

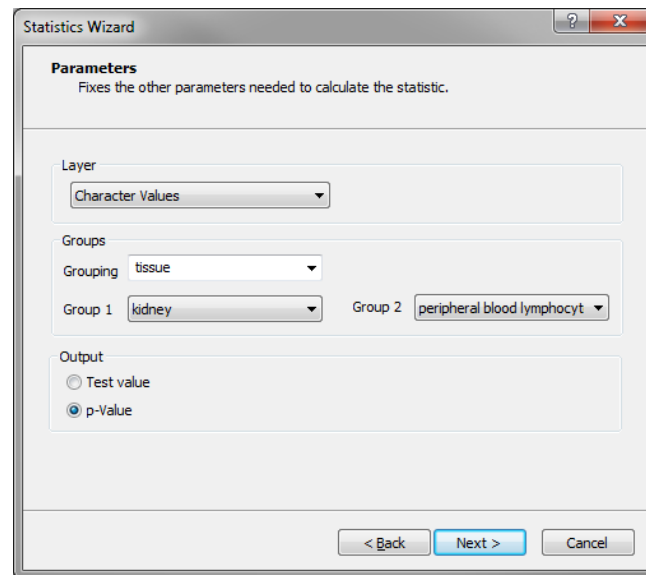


Figure 20.4.8: The *Statistics Wizard*: step 3.

The third step fixes the parameters needed to calculate the statistic selected in step two. The **Layer** has to be indicated in any case. More parameters may be required.

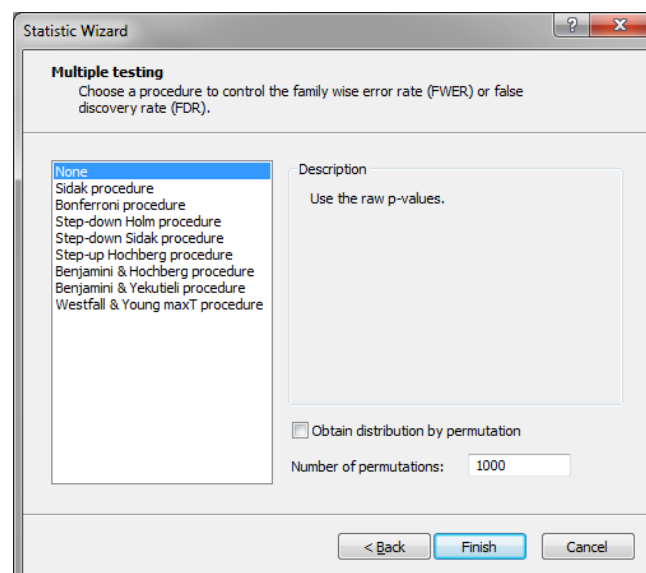


Figure 20.4.9: The *Statistics Wizard*: final step.

In the final step, a multiple testing procedure can be applied. Each entry is considered independently for hypothesis testing. This means that the selected test is executed on each entry separately. When testing many entries at the same time, the chance for a false positive increases by increasing the number of entries

tested simultaneously. The purpose of applying a multiple testing procedure is to apply a correction on the test in order to control the false positive rate or the false negative rate. It is possible to apply a correction for multiple testing, in order to control the family wise error rate (FWER) or false discovery rate (FDR).

The available procedures are:

- Sidak procedure (FWER)
- Bonferroni procedure (FWER)
- Step-down Holm procedure (FWER)
- Step-down Sidak procedure (FWER)
- Step-up Hochberg procedure (FWER)
- Benjamini & Hochberg procedure (FDR)
- Benjamini & Yekutieli procedure (FDR)
- Westfall & Young max T procedure (FDR)

The statistic profile will be shown in the view from which the *Statistics Wizard* was opened.

20.4.5 Plot wizard

A plot is created using the *Create chart* dialog box that is called with **Profiles > Charts and statistics...** (**F7**). More information about the *Create chart* dialog box and the *Charts and statistics* window can be found in [14](#).

Chapter 20.5

Hierarchical clustering

20.5.1 Introduction


Hierarchical clustering is an *unsupervised* grouping technique. It is based on the pairwise distances between all profiles that are to be clustered. It is called unsupervised clustering because it uses only the information that is contained in the data set itself and no prior available information on possible groupings.

Hierarchical clustering techniques are based on a similarity coefficient, used to quantify how similar the profiles are, and on a clustering criterion, to determine distances between the different clusters.

The result of a hierarchical clustering is usually visualized as a tree in which more closely related entries are in denser groups than less closely related entries. Such a tree is called a *dendrogram*.

In the *Matrix Mining* window it is possible to cluster the rows as well as the columns and there are a number of tools to edit the outlook of a dendrogram.

20.5.2 Calculating a cluster analysis

A hierarchical clustering is calculated with *Analysis > Cluster analysis...* ().

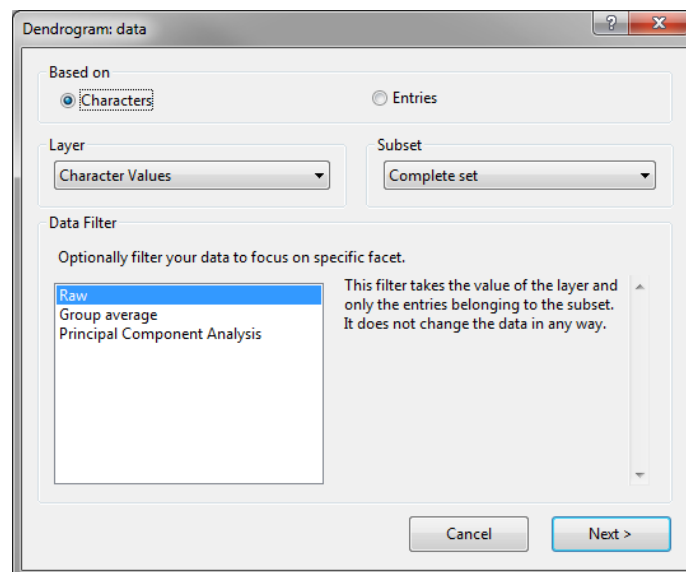


Figure 20.5.1: The *Dendrogram Wizard*: step 1.

In the first step you need to indicate which information the clustering should be based on: based on the **Rows** or **Columns**.

The **Layer** and **Subset** that are to be used for the calculation need to be selected in the first step of the wizard. There are drop down lists available to do so.

In the lower part of the window, a **Data Filter** can be selected to focus on a specific facet:

- **Raw:** This option (default) does not change the data in any way.
- **Group average:** All the rows/columns belonging to the same group are averaged. When this option is selected from the list, the grouping needs to be selected in the next window.
- **Principal Component Analysis:** With this option, the dimension of the data is reduced, and only a fixed set of principal components are retained. In the next window, the number of components needs to be specified.

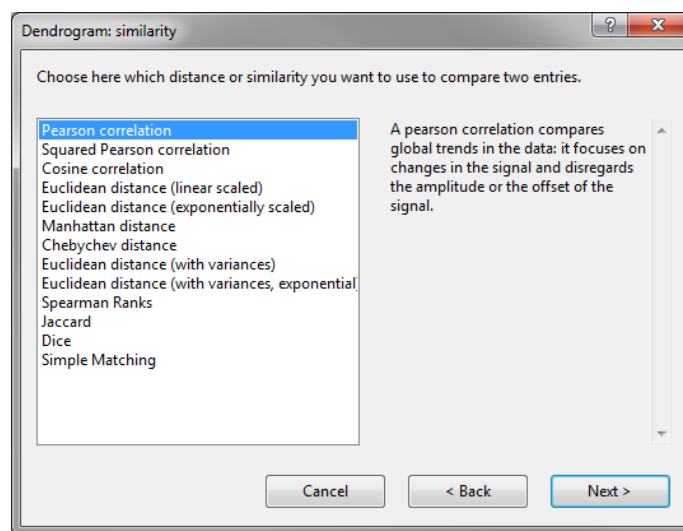


Figure 20.5.2: The *Dendrogram Wizard*: step 2.

In the second step, the similarity or distance coefficient to compare the rows/columns needs to be specified:

- **Pearson correlation:** A Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the scaling or offset of the profiles.
- **Squared Pearson correlation:** A Squared Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the sign, the scaling or offset of the profiles. Compared to a normal Pearson correlation, this distance gives profiles which are anti-symmetric a high similarity.
- **Cosine correlation:** The Cosine correlation focuses on changes in the signal and disregards the scaling of the profiles.
- **Euclidean distance:** An Euclidean distance calculates the distance like one would measure it in the real world. Profiles are regarded as similar if all the entries are nearly identical. **Linear scaled:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **Exponentially scaled:** The distance measure is transformed into a similarity by scaling linear with the standard deviation of all entries and taking the exponential. This gives rise to more realistic similarity values if the dimension is higher, compared to the linear scaled euclidean distance.

- **Euclidean variance distance:** An Euclidean variance distance calculates the distance like one would measure it in the real world, increased with the error bars of both entries. Profiles are regarded as similar if all the entries are nearly identical. Compared to the usual Euclidean distance, this distance takes the uncertainty on measurements into account. **With variances:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **With variances, exponential:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries and taking the exponential.
- **Manhattan distance:** A Manhattan distance calculates the distance by measuring it along the lines of a coordinate grid. Profiles are regarded as similar if all entries are nearly identical. Compared to the Euclidean distance the Manhattan distance is more tolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the MAD of all coordinates of all entries and taking the exponential.
- **Chebyshev distance:** A Chebyshev distance calculates the distance by taking the maximum difference over all coordinates. Profiles are regarded as similar if all the entries are exactly identical. Compared to the Euclidean distance, this distance is very intolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the maximum coordinate difference over all entries and taking the exponential.
- **Spearman Ranks:** A Spearman similarity is the non-parametric version of the Pearson correlation.
- **Jaccard:** In case of two binary row (resp. column) profiles the Jaccard coefficient J gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles. The Jaccard distance is defined as $1 - J$.
- **Dice:** The Dice coefficient Di is similar to the Jaccard coefficient but assigning a double weight to the data points that are present on both profiles, it gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles but relative to the number of points that are labeled as present. The Dice distance is defined as $1 - Di$.
- **Simple matching:** The simple matching coefficient S gives the degree of overlap between the two binary profiles while equal weight is given to present and absent data points. The simple matching distance is $1 - S$.

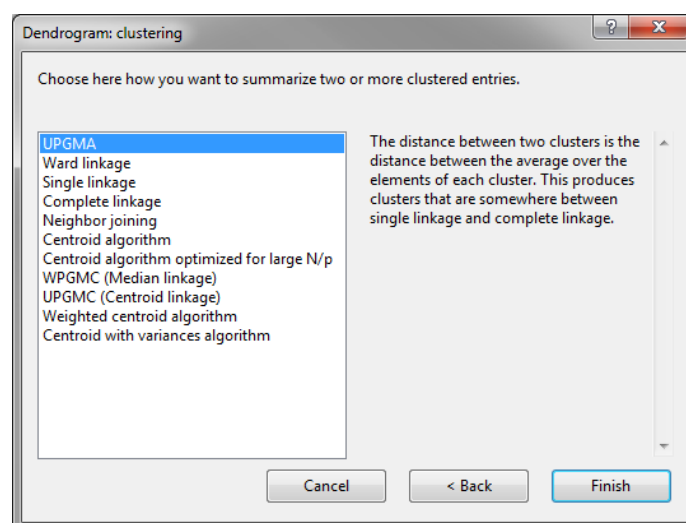


Figure 20.5.3: The *Dendrogram Wizard*: final step.

In the last dialog box, the clustering algorithms are listed:

- **UPGMA:** UPGMA stands for Unweighted Pair-Group Method using Arithmetic averages. The distance between two clusters is the average of the distances between all pairs of profiles that can be considered with one profile from the first cluster and the other profile from the second cluster. It is advisory to use this method if you have no idea about the distribution of the data points in advance. If the data are expected to contain naturally distinct groups, a complete linkage can be used. If the data is expected to have a more chained structure, a single linkage can be used.
- **Ward linkage:** For every cluster, the variance between the profiles is calculated and these variances are summed. The merging of two clusters that yields the smallest increase in the summed variance of all clusters is considered as the next step in the clustering. This linkage criterion requires the use of the Euclidean distance as metric. This method tends to produce approximately equally but rather small sized clusters.
- **Single linkage:** The distance between two clusters is the minimum of the distances between all pairs of profiles that can be considered with one profile from the first cluster and the other profile from the second cluster. This criterion works well if there are multiple equal minimum distances between clusters but tends to produce a chained clustering. This method is also called *nearest neighbor* clustering.
- **Complete linkage:** The distance between two clusters is the maximum of the distances between all pairs of profiles that can be considered with one profile from the first cluster and the other profile from the second cluster. This criterion works well if the data contains naturally distinct groups and tends to produce compact clusters. This method is also called *furthest neighbor* clustering.
- **Neighbor joining:** This method subsequently joins clusters to minimize the sum of branch lengths of the whole tree and continues until two clusters are left. The result is an unrooted tree.
- **Centroid algorithm:** The distance between two clusters is the distance between the unweighted average of the elements of each cluster. The approximation uses the elements in each cluster and does not calculate the full similarity matrix. It is therefore fit for a number of large entries.
- **Centroid algorithm optimized from large N/p :** The distance between two clusters is the distance between the weighted average of the elements of each cluster. This approximation uses the elements in each cluster and does not calculate the full similarity matrix. The method is therefore fit for a number of large entries. Moreover this method is optimized for datasets with a large number of entries (N) and a relatively small number of components (p).
- **WPGMC (Median linkage):** WPGMC stands for Weighted Pair-Group Method using Centroids. The distance between two clusters is the distance between the weighted average of the elements of each cluster. Choose this option if you expect the size of the clusters to be very different. This approximation uses the similarity matrix and requires the use of the Euclidean similarity coefficient as metric.
- **UPGMC (Centroid linkage):** UPGMC stands for Unweighted Pair-Group Method using Centroids. The distance between two clusters is the distance between the unweighted average of the elements of each cluster. This approximation uses the similarity matrix and requires the Euclidean similarity coefficient as metric.
- **Weighted centroid algorithm:** The distance between two clusters is the distance between the weighted average of the elements of each cluster. This approximation uses the elements in each cluster and does not calculate the full similarity matrix. It is therefore fit for a number of large entries.
- **Centroid with variances algorithm:** This method calculates the average of each cluster with error bars and is therefore fit for similarity coefficients that use the error bars in their calculations.

20.5.3 Displaying and handling dendrograms

Most but not all functionality is present for row and column dendrograms. A few functionalities, like e.g. clipping the dendrogram, are not available for column dendrograms, but this is overcome by flipping the matrix.

When a dendrogram node is clicked on, a blue diamond-shaped cursor appears on that position and the branch is highlighted. In case of a selection of a node in a row dendrogram, the similarity or distance of the selected node and the numbers of row entries that are contained in the node are displayed in the *Dendrogram info* panel.

A branch can be moved up (*Dendrogram* > *Move branch up* (↕)) or down (*Dendrogram* > *Move branch down* (↕)) to improve the layout of a dendrogram or to make its description easier.

To simplify the representation of large and complex row dendrograms, it is possible to simplify branches by abridging them as a triangle with *Dendrogram* > *Abridge branch* (◀). Repeating this action will undo the abridge operation.

An automated way to reduce the number of entries on display is to select *Dendrogram* > *Clip dendrogram*....

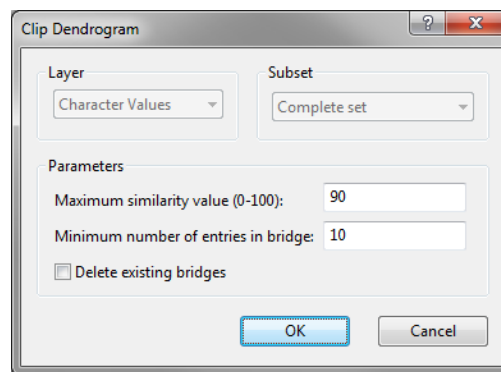


Figure 20.5.4: Clipping branches in a dendrogram.

Rows/Columns are grouped if their similarity is below a certain threshold and if there are a minimum number of entries in a clipped branch. The layer and subset for the currently displayed dendrogram are indicated. The Parameters for clipping are:

- **Maximum similarity value:** This sets an upper value for the similarity of the branches that will be clipped.
- **Minimum number of entries in the bridge:** Sets a minimum to the number of rows/columns that are replaced by a bridge when the branch is clipped.
- **Deleting existing bridges:** Replaces an existing clipping with a new one.

Rows/columns contained in a selected branch are selected with *Dendrogram* > *Branch to selection*..

A subset is created containing all rows/columns from the selected branch with *Dendrogram* > *Branch to subset*.... If a selection in the other direction is present, these rows/columns will be contained in the new subset. If there is no selection present in the other direction, all rows/columns of the parent subset will be contained in the new subset. A name is automatically assigned to the subset.

With *Dendrogram* > *Branch to group*... a group is created containing all rows/columns from the selected branch. Markers are automatically created. Groups are edited following the description in 20.2.3.

With ***Dendrogram > Branch to profile...*** a profile is generated for the rows/columns contained in the selected branch.

With ***Dendrogram > Branch to statistics report...*** a statistics report is generated for the selected branch.

Chapter 20.6

Partitioning

20.6.1 Introduction

A partitioning method is used to divide a set of profiles in a number of predefined groups. It starts with a random assignment of the profiles to the groups. The next step is to obtain a measure for the relatedness between the profiles and the separate groups. Then the profiles are shuffled so that they belong to the group they are closest related to. These last two steps form an iterative process and are repeated until all profiles belong to the closest group.

20.6.2 Partitioning methods

Different partitioning methods use different measures for the relatedness between the profiles and the groups. The two methods that are implemented in the *Matrix Mining* window: the *manual* partitioning and the *automatic* partitioning.

20.6.2.1 Manual partitioning

The k-means clustering algorithm is an often used type of partitioning. The distance of the individual data points to the group is based on the similarity of the data point and the cluster centre. At initialization, this cluster centre coincides with a randomly chosen data point in the cluster. For further iterations, a cluster centre is determined based on the similarities of the cluster profiles. This partitioning method requires to know the number of groups in advance, hence calculating a partitioning may be a recursive operation.

20.6.2.2 Automatic partitioning

It is possible to perform a partitioning that decides automatically on the number of groups. This is based on a number of qualitative criteria for the obtained partitioning.

- The maximum number of groups in the partition.
- The minimum number of members in a group.
- A compactification ratio.
- A separation ratio.

The first step is to calculate a partition with the **Maximum number of groups**. A first correction is done by checking the actual number of profiles in each group against the **Minimum number of members** in a group. Groups that have less than the minimum number of members are placed in the *trash bin* group. A trash bin group contains the profiles that can not be properly placed into another group.

Two additional parameters are required for the calculation:

The **Compactification ratio** is an upper limit for the internal group deviation between the profiles relative to the deviation between all profiles of the parent. If one aims at obtaining more compact groups, the compactification ratio should be increased. As a result of this, the groups will generally be tighter.

The **Separation ratio** is a lower limit for the mean distance between two groups. If one aims at obtaining better separated groups, the separation ratio should be increased. As a result of this, the groups will generally be smaller.

20.6.3 Calculating a Partitioning

A partitioning is calculated with **Analysis > Partitioning...** (🔍).

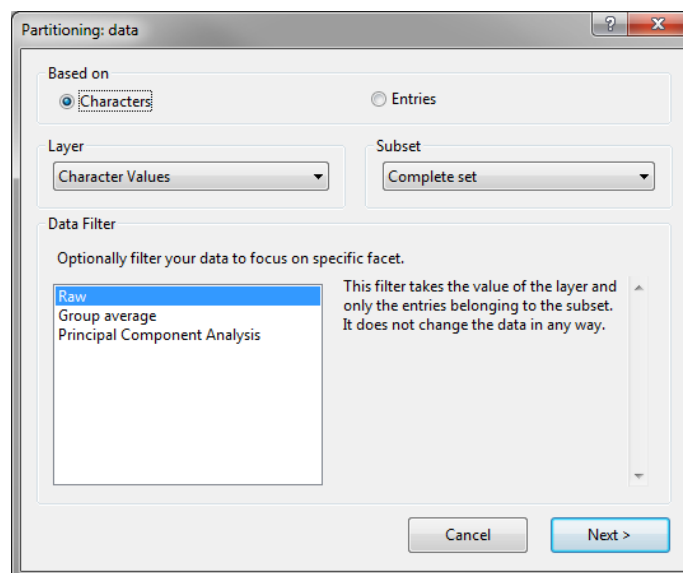


Figure 20.6.1: The *Partitioning* wizard: step 1.

In the first step you need to indicate which information the clustering should be based on: based on the **Rows** or **Columns**.

The **Layer** and **Subset** that are to be used for the calculation need to be selected in the first step of the wizard. There are drop down lists available to do so.

In the lower part of the window, a **Data Filter** can be selected to focus on a specific facet:

- **Raw:** This option (default) does not change the data in any way.
- **Group average:** All the rows/columns belonging to the same group are averaged. When this option is selected from the list, the grouping needs to be selected in the next window.
- **Principal Component Analysis:** With this option, the dimension of the data is reduced, and only a fixed set of principal components are retained. In the next window, the number of components needs to be specified.

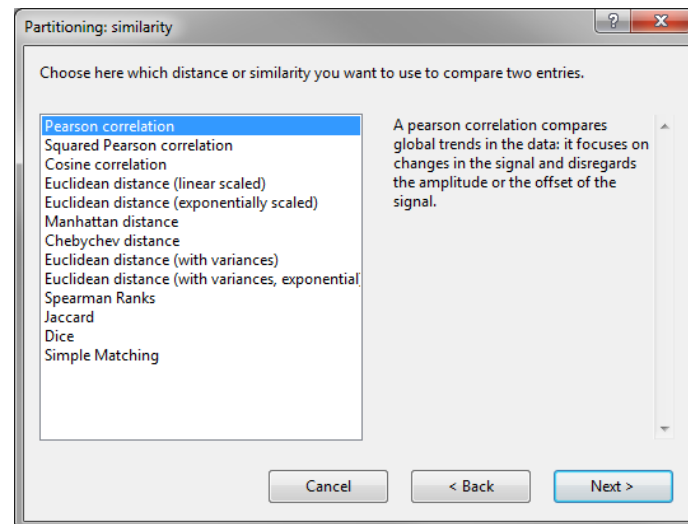


Figure 20.6.2: The *Partitioning* wizard: step 2.

In the second step, the similarity or distance coefficient to compare the rows/columns needs to be specified:

- **Pearson correlation:** A Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the scaling or offset of the profiles.
- **Squared Pearson correlation:** A Squared Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the sign, the scaling or offset of the profiles. Compared to a normal Pearson correlation, this distance gives profiles which are anti-symmetric a high similarity.
- **Cosine correlation:** The Cosine correlation focuses on changes in the signal and disregards the scaling of the profiles.
- **Euclidean distance:** An Euclidean distance calculates the distance like one would measure it in the real world. Profiles are regarded as similar if all the entries are nearly identical. **Linear scaled:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **Exponentially scaled:** The distance measure is transformed into a similarity by scaling linear with the standard deviation of all entries and taking the exponential. This gives rise to more realistic similarity values if the dimension is higher, compared to the linear scaled euclidean distance.
- **Euclidean variance distance:** An Euclidean variance distance calculates the distance like one would measure it in the real world, increased with the error bars of both entries. Profiles are regarded as similar if all the entries are nearly identical. Compared to the usual Euclidean distance, this distance takes the uncertainty on measurements into account. **With variances:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **With variances, exponential:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries and taking the exponential.
- **Manhattan distance:** A Manhattan distance calculates the distance by measuring it along the lines of a coordinate grid. Profiles are regarded as similar if all entries are nearly identical. Compared to the Euclidean distance the Manhattan distance is more tolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the MAD of all coordinates of all entries and taking the exponential.

- **Chebyshev distance:** A Chebyshev distance calculates the distance by taking the maximum difference over all coordinates. Profiles are regarded as similar if all the entries are exactly identical. Compared to the Euclidean distance, this distance is very intolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the maximum coordinate difference over all entries and taking the exponential.
- **Spearman Ranks:** A Spearman similarity is the non-parametric version of the Pearson correlation.
- **Jaccard:** In case of two binary row (resp. column) profiles the Jaccard coefficient J gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles. The Jaccard distance is defined as $1 - J$.
- **Dice:** The Dice coefficient Di is similar to the Jaccard coefficient but assigning a double weight to the data points that are present on both profiles, it gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles but relative to the number of points that are labeled as present. The Dice distance is defined as $1 - Di$.
- **Simple matching:** The simple matching coefficient S gives the degree of overlap between the two binary profiles while equal weight is given to present and absent data points. The simple matching distance is $1 - S$.

The result is shown in the *Partitioning View*.

20.6.4 Partitioning view

20.6.4.1 Discovering the Partitioning view

The *Partitioning view* contains its own menu bar, toolbars and information panels (see Figure 20.6.3).

At the start the dataset is shown in one cell in the *Partition panel*. A cell is selected by clicking in the cell. A selected cell appears within a red border.

The *Members panel* lists the keys of the components in the selected cell, ordered by decreasing similarity with the mean cell profile.

The *Partition info panel* contains some information on the currently selected cell. This information includes the level of the cell in the partitioning, with the zeroth level corresponding to the complete dataset. Also the number of **Members** in the cell is given, together with a measure for the compactness of the cell (**Compact. ratio**) and the value for the standard deviation of the cell profiles with respect to the mean cell profile (**Stddev.**).

It is also possible to show a profile on top of the partitioning. These profiles are entered in the same way as described in 20.4.2. The profile that is inserted is indicated in the *Profile panel*.

20.6.4.2 Selections and view

A cell is selected by clicking in the cell. The selected cell appears within a red border.

To change the current grouping(s) displayed in the *Partitioning view*, select **Groups > Edit row groups...** (🎨, **Ctrl+G**) or **Groups > Edit column groups...** (🎨, **Ctrl+Shift+G**) and select the grouping from the list. Alternatively, use the pull down arrow next to the row and column group buttons in the Group bar of the *Groups panel*.

In default configuration, the *Groups panel* appears as tabbed view together with the *Layers panel*. The drop down menu in the *Groups panel* lists all groupings defined in the database. When a grouping is selected from

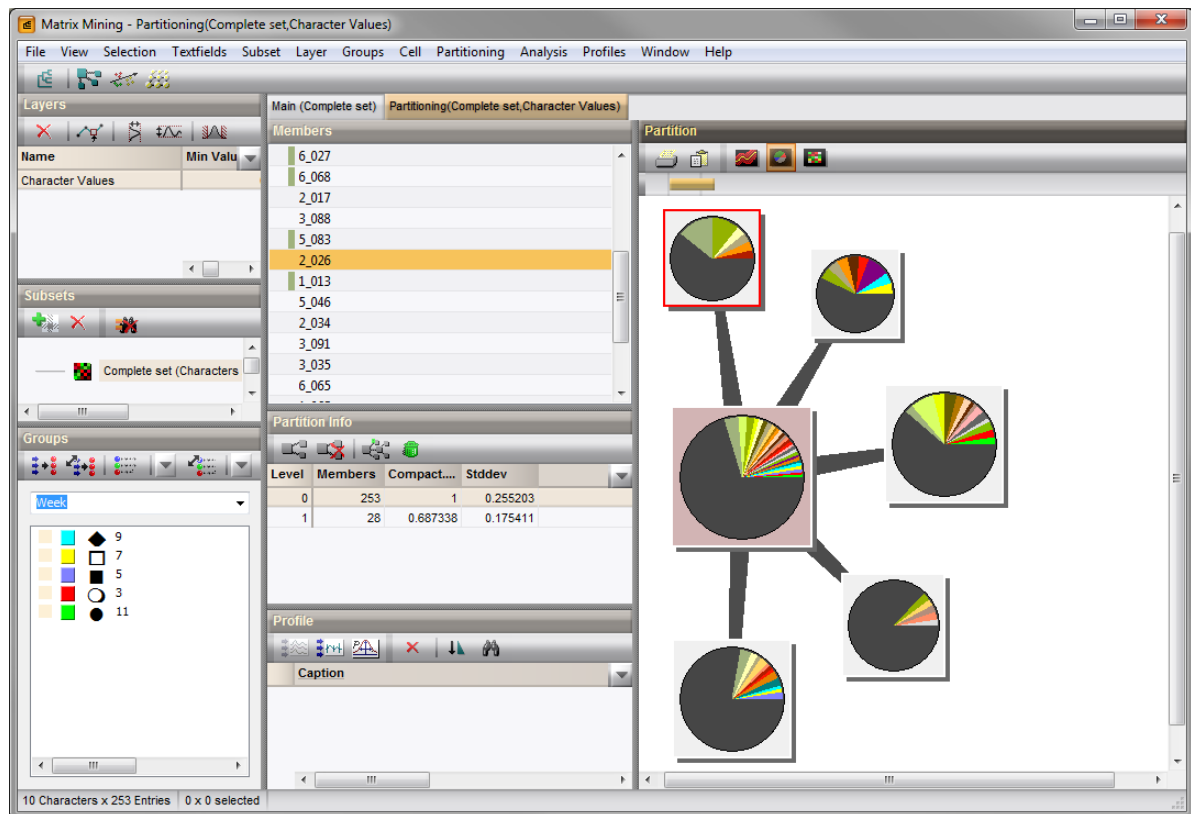


Figure 20.6.3: The *Partitioning* view.

the drop down menu, the groups defined for that grouping are listed in the panel below. Group members can be selected from within this window. To select all members belonging to a group, **Ctrl** +click on the square next to the group color. The square is highlighted and the members of the group are selected. Use the **Shift**-key to select more than one group at a time.

Zooming in or out on the selected cell can be done with the zoom slider.

There are a number of different representations for the *Partitioning* View:

- The *cell profile representation* is the standard settings. In this representation the cells are represented by a profile that is the average of all row or column profiles contained in the cell. The error bars on the cell profiles are calculated from the standard deviation of the profiles in the cell from the average cell profile. This representation is obtained with **View > Profile view** (📊).
- In the *expression values representation* profiles are represented by their expression values. This representation is obtained with **View > Expression Values**.
- In the *group pie chart representation* the cells are represented by a circle where the groups are indicated as colored pie charts. Entries that do not belong to a group are presented in gray. This representation is obtained with **View > Group Pie Chart**.

Profiles can be entered in the *Partitioning* View as described in 20.4.2. The profile that is inserted is indicated in the *Profiles* panel.

- In the expression values representation, the profiles are presented above the cells and separated from the cells with a thin line.
- In the group pie chart representation the Pearson correlation coefficient between the inserted profile and the mean profile for the group is indicated in the center of the pie chart.

- In the cell profile representation the inserted profile is plotted (in green) on top of the profiles in the group.

20.6.4.3 Splitting in partitions

As a first step of a partitioning, the complete data set is shown as one cell. Splitting a cell manually in partitions is done with **Partitioning > Split cell...** (🗑️).

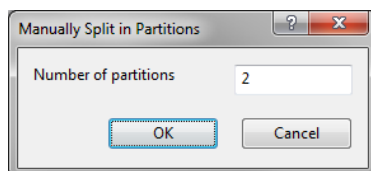


Figure 20.6.4: Manually split in a number of partitions

The numbers of partitions is prompted for.

An automatic partitioning is obtained with **Partitioning > Auto split...** (🗑️).

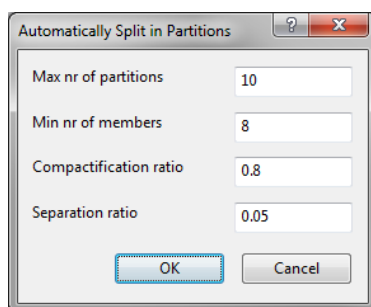


Figure 20.6.5: Automatically split in partitions.

In this dialog box the **Maximum number of partitions** and **Minimum number of members** for each cell should be entered.

Two additional parameters are required for the calculation:

The **Compactification ratio** is an upper limit for the internal group deviation between the profiles relative to the deviation between all profiles of the parent. If one aims at obtaining more compact groups, the compactification ratio should be increased. As a result of this, the groups will generally be tighter.

The **Separation ratio** is a lower limit for the mean distance between two groups. If one aims at obtaining better separated groups, the separation ratio should be increased. As a result of this, the groups will generally be smaller.

After the automatic partitioning a number of cells may have the image of a garbage bin in their upper left corner. This indicates that these cells contain the entries that could not be classified properly.

It is to collect all garbage entries into one cell with **Partitioning > Collect garbage** (🗑️).

Multiple cells are selected by clicking on them while holding the **Ctrl**-key. If multiple cells are selected they can be combined with **Partitioning > Merge selected cells**.

A partitioning is deleted by selecting the parent cell and selecting **Partitioning > Delete children** (🗑️).

20.6.4.4 Handling cells

A selected cell in the *Partition panel* appears within a red border.

Members in the selected cell are selected with **Cell > Cell to selection...** by holding the **Shift-key** while clicking on the cell with the left mouse button.


A subset containing the members in the cell is created with **Cell > Cell to subset...**


A group containing the members in the cell is created with **Cell > Cell to group...**

With **Cell > To profile** a selected cell is added as a profile to the *Profile panel*.

With **Cell > Cell to statistics report** a statistics report is generated for the selected cell.

20.6.4.5 Printing and exporting

The contents of the *Partitioning View* is printed with **File > Print Image** (.

The contents of the *Partitioning View* is copied to the clipboard with **File > Copy Image to Clipboard...** (.

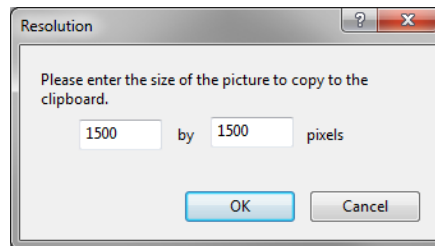


Figure 20.6.6: Enter the size of the picture to copy to the clipboard.

The user is prompted to enter the size of the picture in pixels.

Chapter 20.7

Dimensioning techniques

20.7.1 Introduction

Principal Components Analysis (PCA) and *Discriminant Analysis* are grouping techniques that can be classified as *dimensioning techniques*. In contrast to dendrogram inferring methods, they do not produce hierarchical structures like dendrograms. Instead, these techniques produce two- or three-dimensional plots in which the entries are spread according to their relatedness.

Both techniques are essentially a linear transformation that is a combination of a rotation and a scaling. Due to the rotation, the axes of the two- or three-dimensional representation are aligned with the directions that represent the largest variation in the original data set. The scaling is applied to make distances along the axis comparable in a statistical sense.

Discriminant analysis is very similar to PCA. The major difference is that PCA calculates the best discriminating components from the dataset as a whole, without foreknowledge about groups, whereas discriminant analysis calculates the best discriminating components for groups that have been defined by the user. In case of discriminant analysis, the principal components are called discriminants.

20.7.2 Calculating a PCA

A PCA is calculated with *Analysis > Principal components analysis...* .

The analysis can be based on the *Rows* or *Columns*.

The *Layer* and *Subset* need to be specified from a drop down list.

It is possible to perform a scaling for the rows and columns by *Subtracting the average* and by *Dividing by the RMS* ('root mean square'):

- Subtraction of the *averages* over the *columns* results in a PCA plot arranged around the origin, and therefore, it is recommended for general purposes.
- Division by the *variances* over the *columns* results in an analysis in which each column entry is equally important. Enabling this option can be interesting in a study containing column entries of unequal occurrence. Dividing by the variance for each column normalizes for range differences, making each column entry contributing equally to the total separation of the system.
- Subtraction of the *averages* over the *rows* results in column entry sets of which the sum of the columns equals zero for each row entry. This feature has little meaning for general purposes.

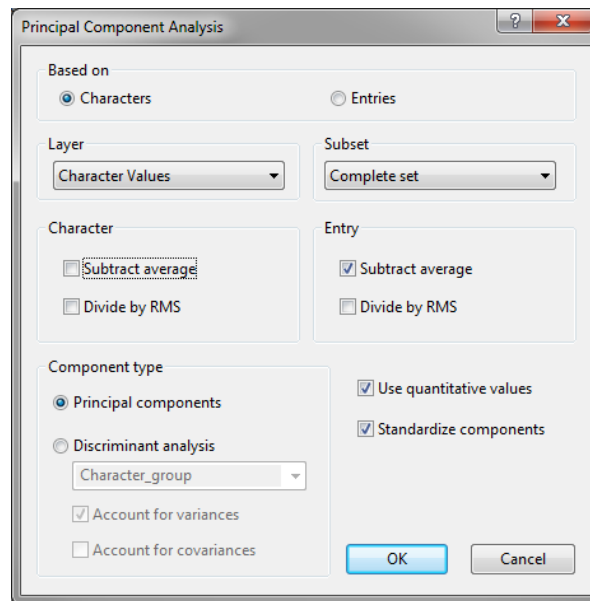


Figure 20.7.1: The *Calculate PCA* dialog box.

- Division by the *variances* over the *rows* results in column entry sets for which the intensity is normalized for all row entries. Dividing by the variances normalizes the column sets for irrelevant differences, making column entry sets with different overall developments fall together as long as the relative reactions of the column entries are the same.

There are two *Component types* that can be calculated: *Principal components* and *Discriminant Analysis*. If *Discriminant Analysis* is checked, the grouping can be selected from a drop-down menu and two additional options can be checked:

- *Account for variances*: accounting for the fact that row entries can show systematically more variance on some column entries (or vice versa). In this case, the number of row profiles should be significantly larger than the number of column profiles.
- *Account for covariances*: taking possible correlations between different row entries on different column entries (or vice versa) into account.

You can decide to *Use quantitative values* and/or *Use Standardize components*. There are check boxes available for this input.

The result of the calculations is shown in the *PCA View* (see Figure 20.7.2).

20.7.3 PCA view

20.7.3.1 Discovering the 2D view

The *PCA view* contains its own menu bar, toolbars and information panels (see Figure 20.7.2).

The *2D plot view* is divided in two separate panels:

- If the PCA was calculated based on the rows, the projection of the row entries on the principal components is shown in the left panel, the projection of the column entries on the principal components is presented in the right panel. The view is labeled as "PCA (Subset,Layer)".

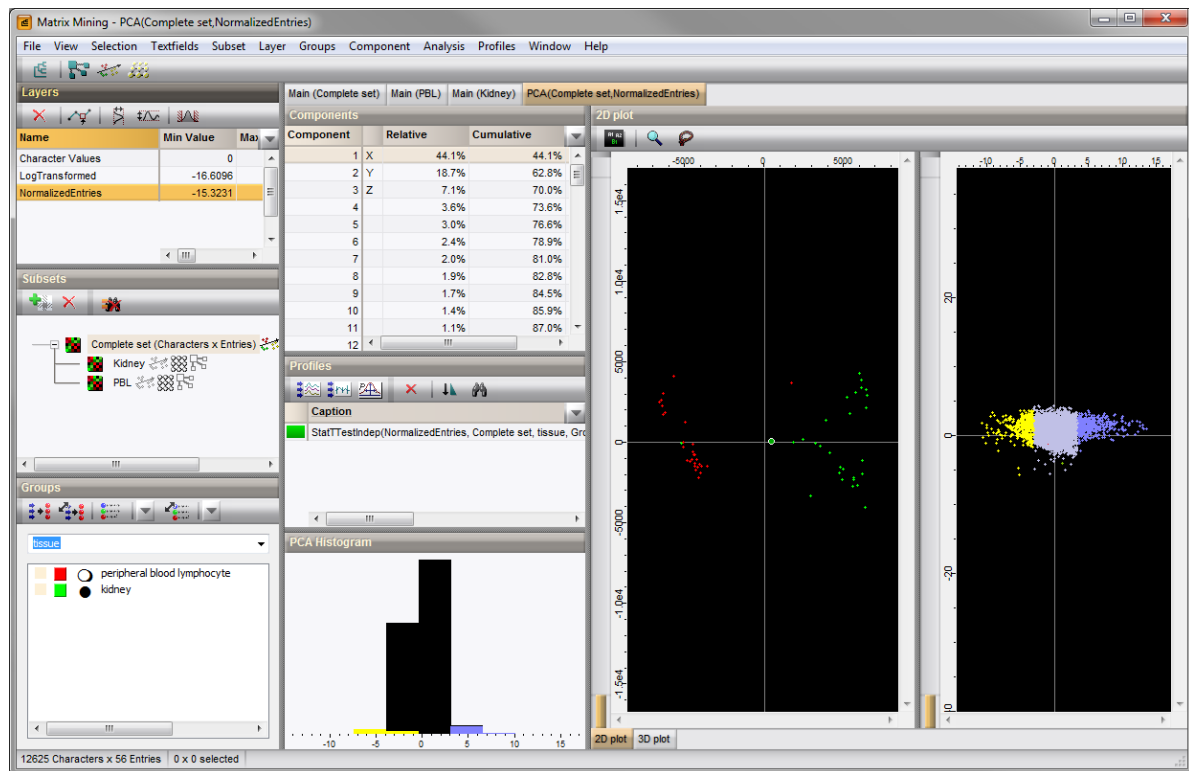


Figure 20.7.2: The PCA view.

- If the PCA was calculated based on the columns, the projection of the column entries on the principal components is shown in the left panel, the projection of the row entries on the principal components is presented in the right panel. The view is labeled as "PCA (Subset,Layer)".

The *Components panel* displays a list of the components with the X, Y and Z principal components listed on top. The relative and cumulative contributions of the components are given. The components labeled with X and Y are used as X and Y axis for the plots. The component labeled with Z is used as Z axis for the representation in 3D.

The *PCA Histogram panel* shows a histogram of the PCA representation. This may give additional information on the compactness of the cloud of points. If groups are defined within the data set, these are shown in color in the histogram.

It is possible to show profiles on top of the PCA representations. These profiles are entered in the same way as described in 20.4.2 and are listed in the *Profiles panel*. Profiles are displayed on the plots with the option **View > Show curves** (default checked). The profiles are presented on the plots by means of large colored spots (see Figure 20.7.3).

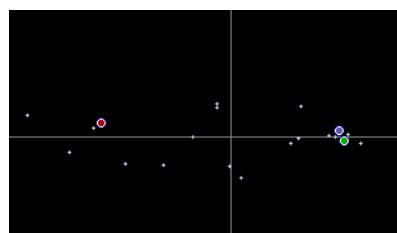


Figure 20.7.3: Show profiles on the plot.

20.7.3.2 Selections and views

To change the current grouping(s) displayed in the *PCA view*, select **Groups > Edit row groups...** (🔍), **Ctrl+G** or **Groups > Edit column groups...** (🔍), **Ctrl+Shift+G** and select the grouping from the list. Alternatively, use the pull down arrow next to the row and column group buttons in the Group bar of the *Groups panel*.

In default configuration, the *Groups panel* appears as tabbed view together with the *Layers panel*. The drop down menu in the *Groups panel* lists all groupings defined in the database. When a grouping is selected from the drop down menu, the groups defined for that grouping are listed in the panel below. Group members can be selected from within this window. To select all members belonging to a group, **Ctrl** +click on the square next to the group color. The square is highlighted and the members of the group are selected. Use the **Shift**-key to select more than one group at a time.

Row and column labels are shown in the PCA representation with the commands **View > Settings 2D > Show row labels** (🔍) and **View > Settings 2D > Show column labels** respectively.

If you want to use other components than the ones suggested by the program as axes in the plot, select the component you want to use by clicking on it in the *Components panel* and select one of the following commands: **View > Use component as x-axis**, **View > Use component as y-axis**, **View > Use component as z-axis**.

One may wish to preserve the aspect ratio for the plots or not. Switching between these two states is done with **View > Preserve aspect ratio**. If the axis ratio is preserved, the command is flagged.

One may wish to display error bars or not. Switching between these two states is done with **View > Show error bars**. If the error bars are shown, the command is flagged.

The zoom functionality is switched on with **View > Settings 2D > Zoom mode** (🔍). By drawing a rectangle in the *Plot panels* while pushing the left mouse button, the selected region is enlarged. As long as in zooming mode, the zoom button remains lowered. To return to normal mode, press the button again. To return to the original scale while in zooming mode, double click with the left mouse button in the *Plot panel*.

A tool for manual selection is available in the form of a lasso tool (**View > Settings 2D > Lasso mode** (🔍)). By clicking in a *Plot panel* and dragging the mouse while holding the left button, points are selected. After selection, the entries are surrounded by a blue square. As long as in lasso mode, the lasso button remains lowered. To return to normal mode, press the button again.

20.7.3.3 Handling components

With **Component > Selection from row component...** or **Component > Selection from column component...** a selection is made based on the PCA representation. These commands open a new dialog.

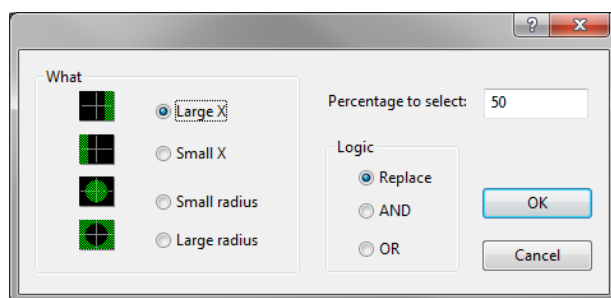


Figure 20.7.4: Component selection tools.

A number of selection criteria that determine what to select are available:

- The option **Large X** should be selected in case you want to make a selection of columns or rows that are represented by large values on the currently selected X-axis.
- The option **Small X** should be selected in case you want to make a selection of columns or rows that are represented by small values on the currently selected X-axis.
- The option **Small radius** should be selected in case you want to make a selection of columns or rows that are located close to the current center of the coordinate system in which the PCA analysis is represented.
- The option **Large radius** should be selected in case you want to make a selection of columns or rows that are located far from the current center of the coordinate system in which the PCA analysis is represented.

In the right column, fill in the fraction of columns/rows you want to select, as **Percentage to select**.

As for the **Logic** to apply, it is possible to perform the selection and **Replace** the current selection, or to add it to the current result (**OR**), or to search in the current selection (**AND**).

With **Component > To row profile** and **Component > To column profile** a selected component is added as a profile to the *Profiles panel* in the *PCA View*.

With **Component > Arrange rows by component** and **Component > Arrange columns by component**, the row/column entries are arranged in the *Main view* based on the selected component.

20.7.3.4 Printing and exporting

The result of the PCA for the rows can be printed with **File > Print image (row)**.

To print the result of the PCA for the columns select **File > Print image (column)**.

The contents of the *PCA view* for the rows can be copied to the clipboard with **File > Copy image to clipboard (row)**.

To copy the contents of the *PCA view* for the columns to the clipboard select **File > Copy image to clipboard (column)**.

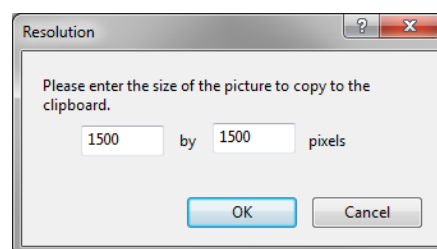


Figure 20.7.5: Enter the size of the picture to copy to the clipboard.

The user is prompted to enter the size of the picture in pixels. All row coordinates are exported at once in a tab-delimited format with **File > Export > Row coordinates**. To export all column coordinates at once, select **File > Export > Column coordinates**.

The user is prompted for an output file name, which can be entered in the text box or browsed for with **<Browse>**.

20.7.3.5 Discovering the 3D view

The *3D plot panel* appears as tabbed view together with the *2D plot panel* and has its own toolbar.

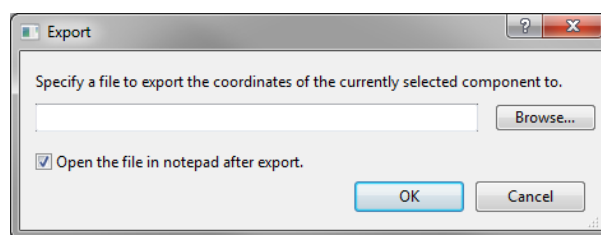


Figure 20.7.6: Export the coordinates to a file.

Zooming in and out on the plot is done with **View > Settings 3D > Zoom in** (🔍, Page Up) and **View > Settings 3D > Zoom out** (🔍+, Page Down) respectively.

The data points can be displayed in pixels (**View > Settings 3D > Pixel** (🖨)), in small points (**View > Settings 3D > Smooth Pixel**), in 3D diamonds (**View > Settings 3D > Diamond**), or in 3D spheres (**View > Settings 3D > Sphere**).

The 3D objects in the plot are shrunk with **View > Settings 3D > Shrink object** (📏, Del), the 3D objects in the plot are enlarged with **View > Settings 3D > Enlarge object** (📏, Ins).

The opacity of the error bars is changed with **View > Settings 3D > Increase opacity** (Home) and **View > Settings 3D > Decrease opacity** (End).

To include or exclude the effect of fading with distance on the objects, choose **View > Settings 3D > Fade with distance** (🌑). The command is checked if the option is turned on. The button remains lowered as long as the feature is turned on.

To show or hide the grid, choose **View > Settings 3D > Background grid** (🌐). The command is checked if the option is turned on. This button remains lowered as long as the feature is turned on.

The plot is shown black on white or white on black, this is changed with **View > Settings 3D > Black background**.

Chapter 20.8

Self-organizing map

20.8.1 Introduction

A Self-Organizing Map (SOM, also called Kohonen map) is a neural network that classifies entries in a two-dimensional space (map) according to their likeliness. The technique which is used for grouping, i.e. the training of a neural network, is completely different from all previously described methods. SOMs therefore provide an interesting addition to conventional grouping methods such as cluster analysis, principal component analysis and related techniques. Unlike PCA, the distance between entries on the map is not in proportion to the taxonomic distance between the entries. Rather, a SOM contains areas of high distance and areas of high similarity. Such areas can be visualized by different shading, for example when a darker shading is used in proportion to the distance in the SOM.

20.8.2 Calculating a SOM

A SOM is calculated with *Analysis > Self-Organizing map...* (🗺️).

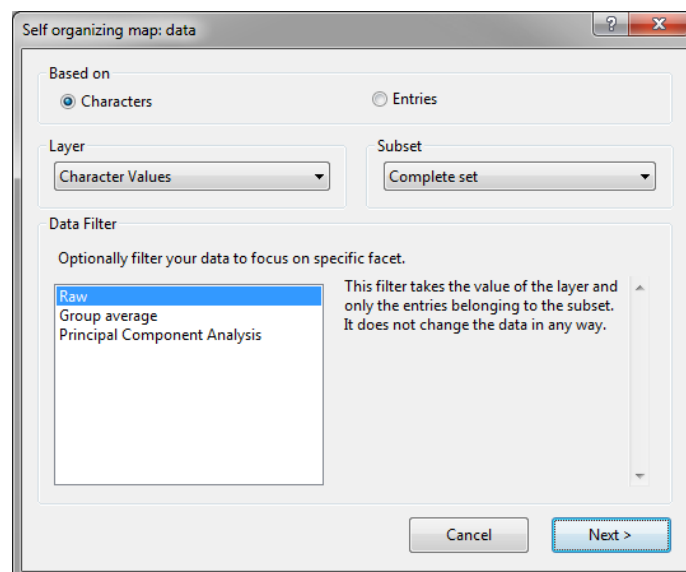


Figure 20.8.1: The SOM wizard: step 1.

In the first step you need to indicate which information the clustering should be based on: based on the **Rows** or **Columns**.

The **Layer** and **Subset** that are to be used for the calculation need to be selected in the first step of the wizard. There are drop down lists available to do so.

In the lower part of the window, a **Data Filter** can be selected to focus on a specific facet:

- **Raw:** This option (default) does not change the data in any way.
- **Group average:** All the rows/columns belonging to the same group are averaged. When this option is selected from the list, the grouping needs to be selected in the next window.
- **Principal Component Analysis:** With this option, the dimension of the data is reduced, and only a fixed set of principal components are retained. In the next window, the number of components needs to be specified.

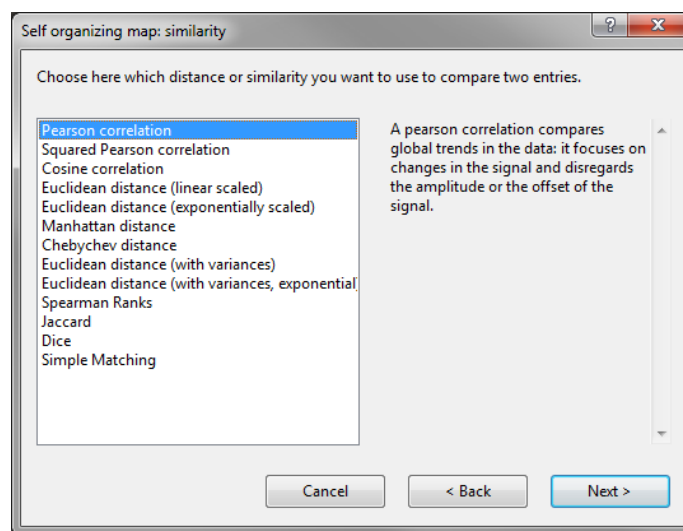


Figure 20.8.2: The SOM wizard: step 2.

In the second step, the similarity or distance coefficient to compare the rows/columns needs to be specified:

- **Pearson correlation:** A Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the scaling or offset of the profiles.
- **Squared Pearson correlation:** A Squared Pearson correlation compares the global trends in the data: it focuses on changes in the profiles and disregards the sign, the scaling or offset of the profiles. Compared to a normal Pearson correlation, this distance gives profiles which are anti-symmetric a high similarity.
- **Cosine correlation:** The Cosine correlation focuses on changes in the signal and disregards the scaling of the profiles.
- **Euclidean distance:** An Euclidean distance calculates the distance like one would measure it in the real world. Profiles are regarded as similar if all the entries are nearly identical. **Linear scaled:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **Exponentially scaled:** The distance measure is transformed into a similarity by scaling linear with the standard deviation of all entries and taking the exponential. This gives rise to more realistic similarity values if the dimension is higher, compared to the linear scaled euclidean distance.

- **Euclidean variance distance:** An Euclidean variance distance calculates the distance like one would measure it in the real world, increased with the error bars of both entries. Profiles are regarded as similar if all the entries are nearly identical. Compared to the usual Euclidean distance, this distance takes the uncertainty on measurements into account. **With variances:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries. This gives rise to very high similarity values if the dimension is high. **With variances, exponential:** The distance measure is transformed into a similarity by scaling linear with the maximum coordinates of all entries and taking the exponential.
- **Manhattan distance:** A Manhattan distance calculates the distance by measuring it along the lines of a coordinate grid. Profiles are regarded as similar if all entries are nearly identical. Compared to the Euclidean distance the Manhattan distance is more tolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the MAD of all coordinates of all entries and taking the exponential.
- **Chebyshev distance:** A Chebyshev distance calculates the distance by taking the maximum difference over all coordinates. Profiles are regarded as similar if all the entries are exactly identical. Compared to the Euclidean distance, this distance is very intolerant towards an entry that is very dissimilar. The distance is transformed into a similarity by linear scaling with the maximum coordinate difference over all entries and taking the exponential.
- **Spearman Ranks:** A Spearman similarity is the non-parametric version of the Pearson correlation.
- **Jaccard:** In case of two binary row (resp. column) profiles the Jaccard coefficient J gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles. The Jaccard distance is defined as $1 - J$.
- **Dice:** The Dice coefficient Di is similar to the Jaccard coefficient but assigning a double weight to the data points that are present on both profiles, it gives the degree of overlap between the two binary profiles, ignoring the points that are absent on both profiles but relative to the number of points that are labeled as present. The Dice distance is defined as $1 - Di$.
- **Simple matching:** The simple matching coefficient S gives the degree of overlap between the two binary profiles while equal weight is given to present and absent data points. The simple matching distance is $1 - S$.

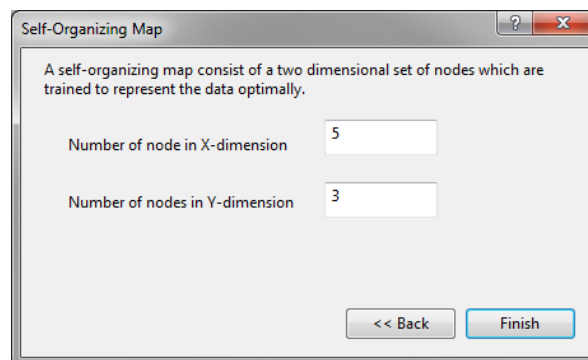


Figure 20.8.3: The SOM wizard: final step.

In the last window, the number of nodes need to specified which are trained to represent the data optimally. The result of the calculations is displayed in the *SOM View*.

20.8.3 SOM view

20.8.3.1 Discovering the SOM view

This view contains its own menu bar, toolbars and information panels.

The panel on the right contains the SOM. The rows/columns are presented on top of the shaded map.

The *Cell panel* contains a schematic view of the SOM, where the rows/columns are not shown individually but just the cells are presented by a dot.



A cell is selected in the *Cell panel* or in the main panel with the large map by clicking in the cell. The selected cell appears within a yellow border.

The *Components panel* lists the keys of the components in the selected cell. The colors corresponding to the currently selected grouping are shown next to the keys.



It is also possible to show a profile on top of the SOM representation. The profile is entered in the same way as described in 20.4.2. The profile that is inserted is indicated in the *Profiles panel*.

20.8.3.2 Selections and views


A cell is selected in the *Cell panel* or in the main panel with the large map by clicking in the cell. The selected cell appears within a yellow border.

To change the current grouping(s) displayed in the *SOM view*, select **Groups > Edit row groups...** , **Ctrl+G** or **Groups > Edit column groups...** , **Ctrl+Shift+G** and select the grouping from the list. Alternatively, use the pull down arrow next to the row and column group buttons in the Group bar of the *Groups panel*.

In default configuration, the *Groups panel* appears as tabbed view together with the *Layers panel*. The drop down menu in the *Groups panel* lists all groupings defined in the database. When a grouping is selected from the drop down menu, the groups defined for that grouping are listed in the panel below. Group members can be selected from within this window. To select all members belonging to a group, **Ctrl**+click on the square next to the group color. The square is highlighted and the members of the group are selected. Use the **Shift**-key to select more than one group at a time.

The zoom functionality is switched on with **View > Zoom in** , for enlarging the plots, or with **View > Zoom out** , for a smaller plot. Zooming in focuses on the selected cell.

There are a number of different representations for the *SOM View*:

- The *cell profile representation* is the standard settings. In this representation the cells are represented by a profile that is the average of all row or column profiles contained in the cell. The error bars on the cell profiles are calculated from the standard deviation of the profiles in the cell from the average cell profile (see Figure 20.8.4). This representation is obtained with **View > Profile view** .
- In the *expression values representation* profiles are represented by their expression values (see Figure 20.8.5). This representation is obtained with **View > Expression Values**.
- In the *group pie chart representation* the cells are represented by a circle where the groups are indicated as colored pie charts. Row/column profiles that do not belong to a group are presented in gray (see Figure 20.8.6). This representation is obtained with **View > Group Pie Chart**.

Profiles can be entered in the *SOM View* as described in 20.4.2. The profile that is inserted is indicated in the *Profiles panel*.

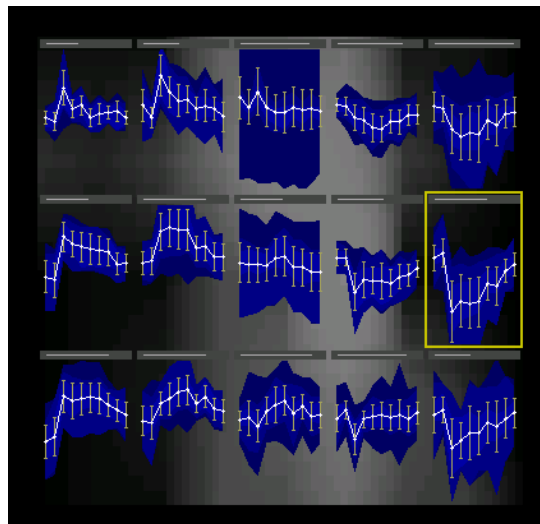


Figure 20.8.4: The cell profile representation.

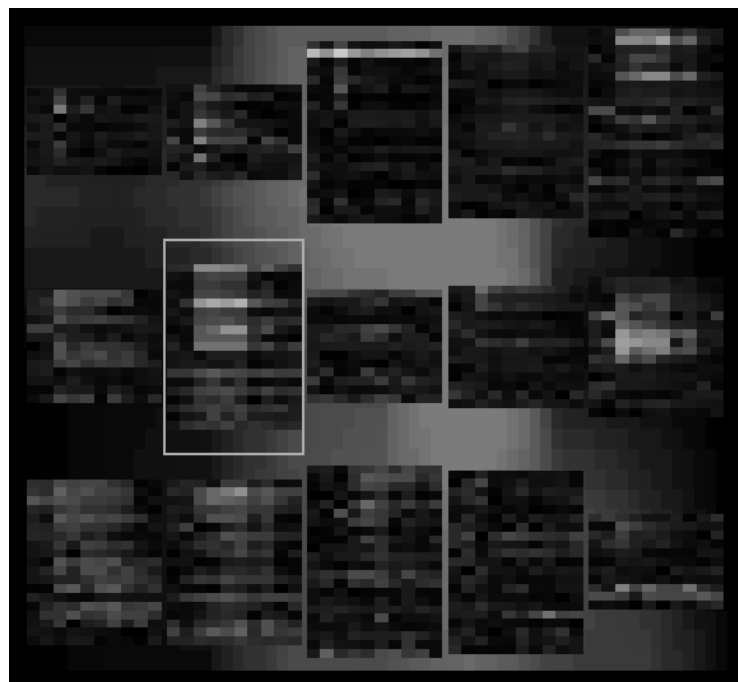


Figure 20.8.5: The expression values representation.

- In the cell profile representation the inserted profile is plotted (in green) on top of the profiles in the group (see Figure 20.8.7).
- In the group pie chart representation the Pearson correlation coefficient between the inserted profile and the mean profile for the group is indicated in the center of the pie chart (see Figure 20.8.8).
- In the expression values representation, the profiles are presented above the cells and separated from the cells with a thin line (see Figure 20.8.9).

20.8.3.3 Handling cells

A selected cell in the *Cell panel* or main panel appears within a yellow border.

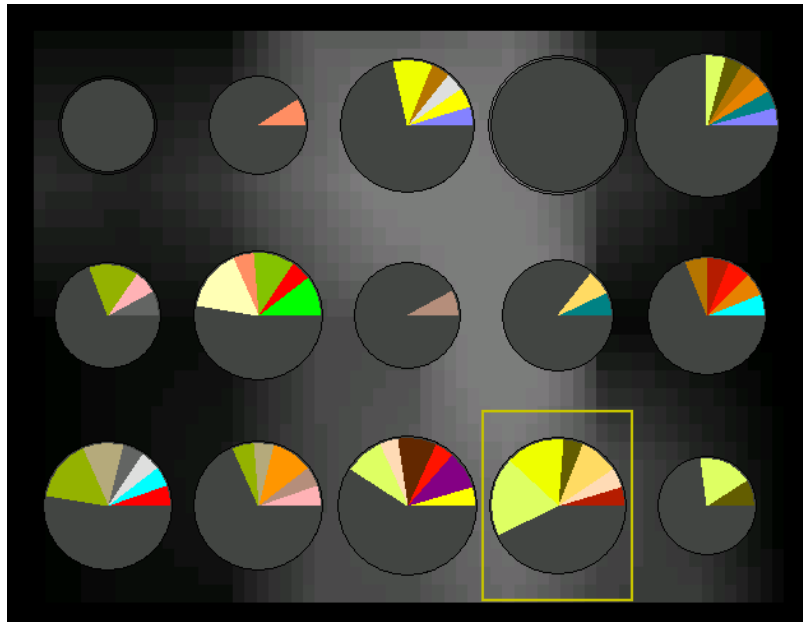


Figure 20.8.6: The group pie chart representation.

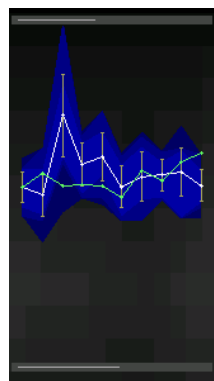


Figure 20.8.7: Profile added to a SOM in the cell profile representation.

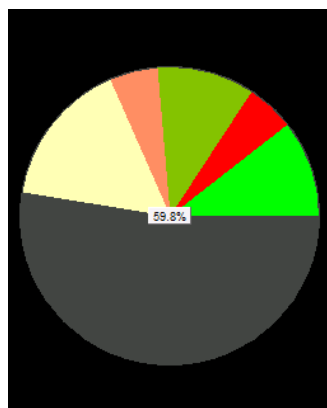


Figure 20.8.8: Profile added to a SOM in the pie chart representation.

All members in the selected cell are selected with *Cell > Cell to selection...* by holding the **Shift-key** while clicking on the cell with the left mouse button.

A subset containing the members in the cell is created with *Cell > Cell to subset...*



Figure 20.8.9: Profile added to a SOM in the expression values representation.

A group containing the members in the cell is created with **Cell > Cell to group...**

With **Cell > To profile** a selected cell is added as a profile to the *Profiles panel*.

With **Cell > Cell to statistics report** a statistics report is generated for the selected cell.

20.8.3.4 Printing and exporting

The contents of the *SOM View* is printed with **File > Print Image** (🖨️). The shading of the background map on paper will be the reverse of the shading of the background map on the screen.

The contents of the *SOM View* is copied to the clipboard with **File > Copy Image to Clipboard...** (📋).

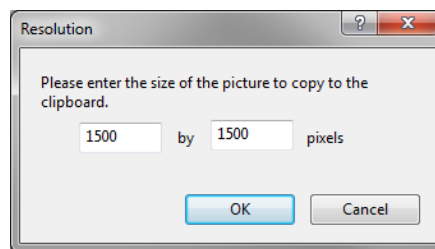


Figure 20.8.10: Enter the size of the picture to copy to the clipboard.

The user is prompted to enter the size of the picture in pixels.

Part 21

Appendix

Chapter 21.1

Relational database table structure

21.1.1 Introduction

In the description below, the structures of the tables required by BioNumerics are given (see [3.7.2](#)). The tables are indicated with their default names. It is possible to use different names for these tables or views in an actual database, which are recorded in the connection description file (.xdb). The names of the columns within the tables, however, are fixed.

In several of the tables that are described below, additional fields might be present that correspond to the custom object fields of the corresponding object. The name of the relational database field will be the name of the custom object field as entered by the user, with a prefix (e.g. "FIELD_") added.

The object "CLOB" means a large text field. This may be described differently depending on the database management software used (e.g., the Microsoft Access equivalent is "memo"). NULL values should be allowed for all fields.

21.1.2 Table ACTIONS

This table holds a record for every action undertaken by a *user* (a single user action can affect several objects).

- ACTIONID (NUMBER): Unique serial number of the user action.
- TMSTAMP (VARCHAR(40)): A time stamp, i.e. the time at which the action occurred.
- APUSER (VARCHAR(80)): The database user who performed the action.
- OSUSER (VARCHAR(80)): The Windows user that was logged on when the action was performed.
- COMMENT (VARCHAR(150)): Description of the action.
- STATUS (NUMBER): Whether or not an action has been successful (0 = successful; ERRORID when unsuccessful).

21.1.3 Table ALIGNPROJ

Contains a record for every alignment project created in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an alignment object.

- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- OBJCNAME (VARCHAR(80)): Name of the alignment project.
- OBJCDATECREATED (VARCHAR(80)): The date the alignment project was created.
- OBJCDATEMODIFIED (VARCHAR(80)): The date of the last modifications made to the alignment project.
- OBJCDATA (CLOB): Alignment data.

21.1.4 Table ANNOTATION

Contains a record for every annotation project created in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an annotation object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- OBJCNAME (VARCHAR(80)): Name of the annotation project.
- OBJCDATECREATED (VARCHAR(80)): The date the annotation project was created.
- OBJCDATEMODIFIED (VARCHAR(80)): The date of the last modifications made to the annotation project.
- OBJCDATA (CLOB): Annotation data.

21.1.5 Table ANTEMPLATES

This table holds information about analysis templates.

- OBJACTIONID (NUMBER): Unique serial number of an action on an analysis template object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- ANTYPE (VARCHAR(80)): The type of analysis template.
- TPNAME (VARCHAR(80)): The name of the analysis template.
- TPCOMMENT (VARCHAR(250)): Optional description that can be entered when an analysis template is created.
- TPCONTENT (CLOB): XML string with the analysis template content.

21.1.6 Table ATTACHMENTS

Contains a record for every attachment present in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an attachment object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- KEY (VARCHAR(80)): Key of the entry the attachment belongs to.
- IDN (VARCHAR(10)): Identifier attachment.
- CLSS (VARCHAR(20)): The data type of the attachment (1 = Text file, 2 = Bitmap image, 3 = HTML document, 4 = Word document, 5 = Excel document, 6 = PDF document).
- DESCRIPT (VARCHAR(80)): The description of the attachment.
- FILENAME (VARCHAR(250)): The path where the attachment file is stored.
- CONTENT (CLOB): The content of a text file.

21.1.7 Table AUTOSQNUMBERS

This table keeps track of the highest identifiers that were used to uniquely identify entries in the ACTIONS, ATTACHMENTS, ERRORS, OBJACTIONS, OLIGOSEQ, and USERLOG tables.

- SQNAME (VARCHAR(80)): Holds the table name and the name of the column containing the unique identifiers, separated by a ”_” sign.
- SQVALUE (NUMBER): Contains the highest identifier that was used to uniquely identify an entry.

21.1.8 Character Values table

Each character type has its own table holding character value information for the database entries. The default name of this table is the name of the character type, although it is possible to specify any table name (the exact name is contained in the TABLES column of the EXPERIMENTS table). Each record in the table corresponds to a single character value belonging to a single entry in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a character values object.
- KEY (VARCHAR(80)): Key of the entry this character value belongs to.
- CHARACTER (VARCHAR(80)): Key of the character.
- VALUE (FLOAT): Numerical value.

21.1.9 Character Experiments table

Each record in the table corresponds to a character experiment belonging to a single entry in the database. The default table name is the name of the character type, padded with "EXPER", but it is possible to specify any other name (the exact name is contained in the TABLES column of the EXPERIMENTS table).

- OBJACTIONID (NUMBER): Unique serial number of an action on a character experiment object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- KEY (VARCHAR(80)): The key of the entry, to which the character experiment belongs to.
- CREATIONDATE (VARCHAR(40)): The time at which the character experiment was created.
- MODIFIEDDATE (VARCHAR(40)): The time at which the character experiment was last modified.

21.1.10 Character Fields table

Contains information about the additional information fields that can be stored together with characters in a character type. The default table name is the name of the character type, padded with "FIELDS", but it is possible to specify any other name (the exact name is contained in the TABLES column of the EXPERIMENTS table). Every record in this table corresponds to a single field for a single character.

- CHARACTER (VARCHAR(80)): Name of the character this information field belongs to.
- FIELD (VARCHAR(80)): Name of the field.
- CONTENT (VARCHAR(150)): Content of the field.

21.1.11 Table CHROMOCOMP

This table holds information about chromosome comparisons in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a chromosome comparison object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- OBJCNAME (VARCHAR(80)): The name of the chromosome comparison.
- OBJCDATECREATED (VARCHAR(80)): The date the chromosome comparison was created.
- OBJCDATEMODIFIED (VARCHAR(80)): The date of the last modifications made to the chromosome comparison.
- OBJCDATA (CLOB): String holding the structure of the chromosome comparison.

21.1.12 Table CLASSIFIERDATA

Table holding the meta data of all *classifiers*. Classifiers are algorithms that are used to speed up the search when comparing curve data.

- CLASSIFIERID (NUMBER): The unique ID of a classifier.
- NAME (VARCHAR(80)): Holds the name of the classifier.
- CLASSIFIERDATA (CLOB): Contains the meta data of a classifier.

21.1.13 Table CLASSIFIERSETTINGS

Table holding the settings of all *classifiers*. Classifiers are algorithms that are used to speed up the search when comparing curve data.

- CLASSIFIERID (NUMBER): The unique ID of a classifier.
- NAME (VARCHAR(80)): Holds the name of the classifier.
- SUBSET (VARCHAR(80)): The name of the subset the classifier is based on.
- EXPERIMENTTYPE (VARCHAR(80)): Name of the experiment the classifier is based on.
- SETTINGS (CLOB): XML string that holds the classifier settings.

21.1.14 Table COMPAREXTS

This table holds information about comparison components , e.g. partition mappings, MANOVA analyses, network analyses, etc..

- OBJACTIONID (NUMBER): Unique serial number of an action on a comparison component object.
- COMPNAME (VARCHAR(200)): The name of the comparison that the comparison component belongs to.
- EXTTYPE (VARCHAR(80)): The type of comparison component.
- EXTID (VARCHAR(200)): A serial number for the comparison component.
- EXTNAME (VARCHAR(200)): The name for the comparison component.
- CONTENT (CLOB): The content of the comparison component.

21.1.15 Table COMPARISONS

The table contains information about all comparisons saved in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a comparison object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).

- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- NAME (VARCHAR(200)): The name of the comparison.
- CMPTPE (VARCHAR(80)): The type of comparison (comparison, library, library unit or neural network).
- CMPCLSS (VARCHAR(200)): The comparison class (only in case of library units: the name of the library the unit belongs to).
- CMPOwner (VARCHAR(80)): The comparison "owner".
- CMPDATECREATED (VARCHAR(80)): The date the comparison was created.
- CMPDATEMODIFIED (VARCHAR(80)): The date of the last modifications made to the comparison.
- CMPDATA (CLOB): Comparison data.

21.1.16 Table DBSCHEMAS

This table holds information about the database table structure required by the current version of the software and any installed plugins.

- NAME (VARCHAR(80)): Name of the software and the installed plugins for which new tables were added to the table structure.
- SCHVERSION (VARCHAR(80)): Version number of the software and the installed plugins for which new tables were added to the table structure.
- SCHDEF (CLOB): XML information on the database table structure.

21.1.17 Table DBSETTINGS

This table holds the installed plugins and the active information fields of the *Database entries* panel.

- NAME (VARCHAR(200)): "ActivePlugins", "DEFAULTLEVELSETTINGS".
- CONTENT (CLOB): The string of the "ActivePlugins" holds the installed plugins, the "DEFAULTLEVELSETTINGS" holds the active information fields of the *Database entries* panel.

21.1.18 Table DECISNTW

The table contains information about all decision networks made in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a decision network object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).

- OBJCNAME (VARCHAR(80)): The name of the decision network.
- OBJCDATECREATED (VARCHAR(80)): The date the decision network was created.
- OBJCDATEMODIFIED (VARCHAR(80)): The date of the last modifications made to the decision network.
- OBJCDATA (CLOB): String holding the structure of the decision network.

21.1.19 Table ENLEVELS

Contains information about the levels defined in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an entry level object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- LEVELID (NUMBER): Holds the level ID of the defined levels.
- LEVELNAME (VARCHAR(80)): Name of the level.
- SETTINGS (CLOB): String that holds the active fields of each level.

21.1.20 Table ENRELATIONS

This table contains all entries belonging to a relation.

- OBJACTIONID (NUMBER): Unique serial number of an action on an entry relation object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- RLID (NUMBER): The unique ID for each defined relation in the database.
- RELTYPEID (NUMBER): The identifier for each defined relation type.
- KEY1 (VARCHAR(80)): The key of the entry belonging to the forward relation.
- KEY2 (VARCHAR(80)): The key of the entry belonging to the reverse relation.

21.1.21 Table ENRELATIONTYPES

Contains information about the defined relation types.

- OBJACTIONID (NUMBER): Unique serial number of an action on a relation type object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).

- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- RELATID (NUMBER): The unique identifiers for each defined relation type.
- RELATFORWNAME (VARCHAR(200)): The name of the forward relation.
- RELATBACKNAME (VARCHAR(200)): Name of the reverse relation.
- LEVELID1 (NUMBER): The levelID of the forward relation.
- LEVELID2 (NUMBER): The levelID of the reverse relation.
- RELTYPE1 (NUMBER): The type of forward relation (0 = many; 1 = one).
- RELTYPE2 (NUMBER): The type of reverse relation (0 = many; 1 = one).

21.1.22 Table ENTRYFLD

This table contains a record for each value stored in a flexible database field.

- OBJACTIONID (NUMBER): Unique serial number of an action on a value in a flexible field.
- KEY (VARCHAR(80)): The key of the database entry the information belongs to.
- FIELDID (NUMBER): The identifier of the flexible field the information belongs to.
- CONTENT (VARCHAR(150)): The information (= value) stored in the flexible field for a certain entry.

21.1.23 Table ENTRYINFOFIELDS

This table contains a record for each information field for which properties (i.e. field states, colors, data types, flexible fields) were defined.

- OBJACTIONID (NUMBER): Unique serial number of an action on an entry fields settings object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- NAME (VARCHAR(80)): Name of the information field.
- SETTINGS (CLOB): XML string containing the information field settings.
- FIELDID (NUMBER): A unique identifier for the information field.
- FIELDTYPE (NUMBER): The type of information field (0 = fixed field; 1 = flexible field).
- SLEVELID (NUMBER): The database level to which the information field belongs.

21.1.24 Table ENTRYTABLE

This table contains a record for every entry in the database.

- KEY (VARCHAR(80)): The unique identifier for every entry in the database (e.g. isolate number).
- LEVELID (NUMBER): The levelID for each entry in the database.
- ENDTCREATED (VARCHAR(80)): The time at which the entry object was created.
- ENDTMODIF (VARCHAR(80)): The time at which the entry object was last modified.
- OBJACTIONID (NUMBER): Unique serial number of an action on a entry object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).

Other fields in the table are user-defined database information fields (VARCHAR(80), but other data types and computed fields are accepted if the field is set read only; see [3.3.6](#)).

21.1.25 Table ERRORS

This table contains a record for every unsuccessful action.

- ERRORID (NUMBER): A unique serial number for the error.
- TMSTAMP (VARCHAR(40)): A time stamp, i.e. the time at which the error occurred.
- CONTENT (VARCHAR(200)): The content of the error message.

21.1.26 Table EVENTLOG

This table maintains a history list of events that were generated during the manipulation of the database.

- DATETIME (VARCHAR(80)): Recording date and time of the event.
- LOGIN (VARCHAR(50)): Windows login at the moment the event was generated.
- TYPE (VARCHAR(10)): Event type.
- SUBJECT (VARCHAR(50)): Database component for which this event was generated.
- DESCRIPTION (VARCHAR(500)): Description of the event.

21.1.27 Table EXPERATTACH

This table contains descriptive information for any specific key-experiment combination. For example, the error reports generated in the *Spa*, *MLST* and *Batch sequence assembly plugin* are stored in this table.

- OBJACTIONID (NUMBER): Unique serial number of an action on an experiment attachment object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- EXPRATTACHID (NUMBER): The unique identifier for each key-experiment combination.
- KEY (VARCHAR(80)): Key of the database entry the information relates to.
- EXPERIMENT (VARCHAR(80)): Name of the experiment the information relates to.
- NAME (VARCHAR(80)): Names assigned to groups of key-experiment combinations.
- CONTENT (CLOB): Descriptive information specific for each key-experiment combination, e.g. error report.

21.1.28 Table EXPERIMENTS

This table contains a record for every experiment type present in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an experiment object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- EXPERIMENT (VARCHAR(80)): Holds the name of the experiment (should be unique through the whole database).
- TYPE (VARCHAR(80)): Can be "Fingerprint", "Character", "Sequence", "Matrix", "Curve" or "2DGel".
- SETTINGS (CLOB): XML string that holds the processing, visualization and analysis settings of the experiment type.
- TABLES (VARCHAR(160)): Used for character experiments only: holds the name of the tables that hold character values and additional character fields (separated by a comma).

In addition, the table will contain a column for each user-defined character information field, indicated with FIELD_*fieldname*.

21.1.29 Table FPRBNDCLS

This table contains a record for each band class defined in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a band class object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- CLSID (NUMBER): The unique identifier for each band class defined in the database.
- CLSEXPER (VARCHAR(80)): Holds the name of the experiment type.
- CLSNAME (VARCHAR(80)): Name of the band class.
- CLSPOSIT (FLOAT): The position (metrics) of each band class.
- CLSTOL (FLOAT): Tolerance of a band class.

21.1.30 Table FPRINT

This table contains a record for every fingerprint that is entered in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a fingerprint object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- KEY (VARCHAR(80)): The unique identification key of the sample to which this fingerprint belongs.
- EXPERIMENT (VARCHAR(80)): The name of the experiment type to which this fingerprint belongs.
- FILENAME (VARCHAR(80)): The name of the batch to which this fingerprint belongs.
- FILEIDX (NUMBER): The number of the fingerprint inside the fingerprint file.
- SPLINE (VARCHAR(200)): Holds the exact positioning and size of the gelstrip on the image.
- CURVESPLINE (VARCHAR(200)): Describes what part of the gelstrip is used for calculation of the densitometric curve.
- GELSTRIPINFO (VARCHAR(50)): Contains resolution information about the gelstrip image info.
- GELSTRIP (CLOB): This field holds the bitmap values of the gelstrip.
- DENSURVEINFO (VARCHAR(50)): Holds the resolution of the densitometric curve.
- DENSURVE (CLOB): Holds the densitometric curve data.
- DENSERR (CLOB): Holds error information of the curve data.
- BANDS (CLOB): Holds information about the bands assigned on the fingerprint.

- BANDCONC (CLOB): Holds information about 2D concentration estimates.
- BANDCONCINFO (CLOB): Holds information about 2D concentration estimates.
- REFPOS (VARCHAR(250)): Contains the reference positions assigned to this fingerprint.
- MAPFORWARD (CLOB): Contains a forward normalization vector.
- MAPBACK (CLOB): Contains the reverse normalization vector.
- REFSYSTEM (CLOB): Holds the reference system of the fingerprint.
- TONECURVE (VARCHAR(250)): Contains the tone curve.
- CHPTRN (VARCHAR(250)): Contains cached pattern information on the band positions for a fingerprint type with "Fast band matching" enabled.

Other information fields: additional information fields added in the *Fingerprint information panel* in the *Fingerprint* window.

21.1.31 Table FPRINTFILES

This table contains a record for every "batch" of fingerprints that is entered in the database. A batch may correspond to fingerprints that should be normalized simultaneously: e.g. they were run on the same electrophoresis gel, run in the same batch on a sequencer, etc.

- OBJACTIONID (NUMBER): Unique serial number of an action on a fingerprint file object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- FILENAME (VARCHAR(80)): The name of the batch (should be unique for every batch). In case of scanned electrophoresis gels, this corresponds to the name of the TIFF image file.
- EXPERIMENT (VARCHAR(80)): Name of the experiment type to which this fingerprint batch belongs.
- LOCKED (VARCHAR(10)): Whether or not this batch is locked (Yes or No).
- INLINELINK (VARCHAR(80)): If this batch is linked to another batch (for normalization purposes), this specifies the name of the batch that contains normalization info.
- CREATIONDATE (VARCHAR(40)): The time at which the batch was created, i.e. imported in the database.
- MODIFIEDDATE (VARCHAR(40)): The time at which the batch was last modified.
- BOUNDINGBOX (VARCHAR(200)): Specifies the bounding box of the lanes on a 2D fingerprint image.
- SETTINGS (VARCHAR(250)): Data processing settings.
- TONECURVE (VARCHAR(200)): Specifies how bitmap pixel values are mapped to gray shades on the screen.

- REFSYSTEM (CLOB): Specifies the reference system that is used to normalize the batch.
- MARKERS (VARCHAR(200)): Holds marker points that may be used to align linked fingerprint images to each other.
- IMATYPE (VARCHAR(80)): The type of image file.
- IMA (CLOB): Image data.

21.1.32 Table FPRINTREGION

- OBJACTIONID (NUMBER): Unique serial number of an action on a fingerprint file object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- ID (NUMBER): Fingerprint region identifier.
- FILENAME (VARCHAR(80)): Fingerprint file name on which the region is defined.
- FILEIDX (NUMBER): Fingerprint lane number on which the region is defined.
- RAWPOS1 (NUMBER): Region start position (defined on the raw lane).
- RAWPOS2 (NUMBER): Region stop position (defined on the raw lane).
- RNAME (VARCHAR(80)): Name of the region.
- RLABEL (VARCHAR(80)): Additional label of the region.
- RCOLOR (VARCHAR(80)): Color of the region.
- LP1 (NUMBER): Transverse start position of the region on a displayed fingerprint.
- LP2 (NUMBER): Transverse stop position of the region on a displayed fingerprint.

21.1.33 Table GROUPRIGHTS

Contains a record for each allow/deny rule (privilege) defined and the user group to which the rule was assigned.

- OBJACTIONID (NUMBER): Unique serial number of an action on a user group right object.
- GROUPNAME (VARCHAR(80)): The name of the user group.
- RIGHTID (VARCHAR(80)): The rule that was defined (privilege).
- RTYPE (NUMBER): The rule type (1 = allow rule; 0 = deny rule).
- RIDX (NUMBER): The order in which the rules should be checked for the user group.

21.1.34 Table MATRIXVALS

Holds pairwise similarity values. Each record in this table represents a single similarity value between two database entries.

- EXPERIMENT (VARCHAR(80)): Name of the experiment type this similarity value belongs to.
- KEY1 (VARCHAR(80)): Key of the first database entry.
- KEY2 (VARCHAR(80)): Key of the second database entry.
- VALUE (FLOAT): Similarity value.

21.1.35 Table OBJACTIONS

This table holds a record for each action that is performed on an object the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an object.
- ACTIONID (NUMBER): Unique serial number of the user action.
- OBJTYPE (VARCHAR(80)): The object type on which the action was performed.
- OBJID (VARCHAR(250)): The object on which the action was performed.
- PRIORID (NUMBER): The object action that precedes the current one in the audit trail.
- NEXTID (NUMBER): The next object action in the audit trail.

21.1.36 Table OBJQUERIES

The table holds a record for each object query created.

- OBJACTIONID (NUMBER): Unique serial number of an action on an object query.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- NAME (VARCHAR(80)): Name of the object query.
- CONTENT (CLOB): Content of the object query.
- CREATIONDATE (VARCHAR(40)): Time at which the object query was created.
- MODIFIEDDATE (VARCHAR(40)): Time at which the object query was last modified.

21.1.37 Table OBJSETTINGS

This table contains the audit trail settings for each object that is included in the audit trail.

- OBJID (VARCHAR(80)): The name of the object.
- TRACKSETT (CLOB): The audit trail settings for the object.

21.1.38 Table OLIGOSEQ

This table holds information about the oligo nucleotide entries in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on an oligo sequence object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- ID (NUMBER): Unique ID of the oligo nucleotide entry.
- SQNAME (VARCHAR(80)): The name of the oligo nucleotide entry.
- SQTYPE (VARCHAR(80)): Type of oligo nucleotide sequence.
- SQCONTENT (VARCHAR(250)): Contains the sequence of the oligo nucleotide entry.
- SSEQKEY (VARCHAR(80)): Holds the key of the linked database entry.
- SSEQTYPE (VARCHAR(80)): Holds the name of the linked sequence type.

21.1.39 Table PAPIPL

This table holds information about actions in power assembly projects.

- OBJACTIONID (NUMBER): Unique serial number of an action on a Power Assembler action object.
- PAPIPLID (NUMBER): Identifier of the power assembly action.
- PASMBLID (NUMBER): Identifier of the power assembly project the power assembly action belongs to.
- PIPIDX (NUMBER)
- NAME (VARCHAR(80)): Name of the power assembly action.
- DESCR (VARCHAR(250)): Description of the power assembly action.
- PIPL (CLOB): XML string containing the operator flowchart of the power assembly action.
- PRPTY (CLOB): XML string containing the properties of the power assembly action.

21.1.40 Table PARECSET

This table holds information about sequence record data sets in power assembly projects.

- OBJACTIONID (NUMBER): Unique serial number of an action on a Power Assembler action object.
- PARECSETID (NUMBER): Identifier of the sequence record data set.
- PASMBLID (NUMBER): Identifier of the power assembly project the sequence record data set belongs to.

- PAPIPLID (NUMBER): Identifier of the power assembly action the sequence record data set belongs to.
- NAME (VARCHAR(80)): Name of the sequence record data set.
- DESCR (VARCHAR(250)): Description of the sequence record data set.
- STORETYPE (VARCHAR(80)): Storage type of the sequence record data set.
- FILENAME (VARCHAR(250)): Name of the location where the sequence record data set is stored (if stored externally).
- CONTENT (CLOB): Content of the sequence record data set (if stored in the database).
- PRPTY (CLOB): XML string containing the properties of the sequence record data set.

21.1.41 Table PASMBL

This table holds information about power assembly projects.

- OBJACTIONID (NUMBER): Unique serial number of an action on a power assembly object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- PASMBLID (NUMBER): Identifier of the power assembly project.
- NAME (VARCHAR(80)): Name of the power assembly project.
- DESCR (VARCHAR(250)): Description of the power assembly project.
- PRPTY (CLOB): XML string containing the properties of the power assembly project.
- GRPHCS (CLOB): XML string containing the summary graphs of the power assembly project.
- CRVS (CLOB): XML string containing the description of the sequence curves of the power assembly project.
- SMPLSHEET (CLOB): Content of the sample record data set.
- ASMBLS (CLOB): XML string containing the description of the assemblies of the power assembly project.
- CREATIONDATE (VARCHAR(40)): The time at which the power assembly was created.
- MODIFIEDDATE (VARCHAR(40)): The time at which the power assembly was last modified.

21.1.42 Table REPTEMPLATES

- OBJACTIONID (NUMBER): Unique serial number of an action on a report.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.

- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- RPTYPE (VARCHAR(80)): Report type.
- RPTNAME (VARCHAR(80)): Report name.
- RPTGROUP (VARCHAR(80)): Report group.
- RPTCOMMENT (VARCHAR(250)): Comment associated to a report.
- RPTCONTENT (CLOB): Content of a report.
- INFPPRVID (VARCHAR(80)): ID of the info provider that provides a report.
- INFPPRVNAME (VARCHAR(80)): Name of the info provider that provides a report.

21.1.43 Table SEARCHDATA

Table holding the search data of all *classifiers*. Classifiers are algorithms that are used to speed up the search when comparing curve data.

- CLASSIFIERID (NUMBER): The unique ID of a classifier.
- CURVEID (VARCHAR(80)): Holds the entry key of the curve.
- SEARCHDATA (CLOB): Contains the search data of a classifier.

21.1.44 Table SEQTRACEFILES

This table holds information about the sequence trace files (four-channel chromatogram files from automated sequencers).

- OBJACTIONID (NUMBER): Unique serial number of an action on a sequence trace file object.
- KEY (VARCHAR(80)): For use with the Kodon software.
- CONTIGFILE (VARCHAR(80)): Unique ID of the contig that is associated to this sequence trace file.
- TRACEID (VARCHAR(80)): Unique ID of the trace file.
- DATA (CLOB): Holds the full trace information including sequence and the chromatogram files in case the trace files are stored in the database. Otherwise, it stores a link to the path of the trace file.
- INFO (CLOB): Contains the full editing information of the sequence trace file.

21.1.45 Table SEQUENCES

This table holds the sequence information stored in the database. Note that the columns designed for contig files have changed with respect to earlier versions of the software.

- OBJACTIONID (NUMBER): Unique serial number of an action on a sequence object.

- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- KEY (VARCHAR(80)): Key of the database entry this sequence belongs to.
- EXPERIMENT (CHARCHAR(80)): Experiment type of the sequence.
- SEQUENCE (CLOB): Sequence data.
- HEADER (CLOB): Holds the header description information.
- FORMAT (CLOB): Holds the feature description information.
- SEQUENCEQUAL (CLOB): Quality coefficient for each base in the sequence.
- CONTIGFILE (VARCHAR(80)): Unique ID of the contig file that is associated to this sequence (if any).
- CONTIG (CLOB): Holds the contig sequence and its full editing history.
- CONTIGSTATUS (VARCHAR(10)): Contains the status of the contig file, i.e. confirmed or not.
- CREATIONDATE (VARCHAR(40)): Time at which the sequence was created.
- MODIFIEDDATE (VARCHAR(40)): Time at which the sequence was last modified.

21.1.46 Table SIGNATURES

The table contains a record for every signature placed, i.e. every time an object version is digitally signed ("sign action").

- SIGNID (NUMBER): Unique serial number for the signature.
- ONOBJACTION (NUMBER): The object action (i.e. object version) that was signed.
- USERID (VARCHAR(80)): The user ID of the user who signed.
- USERNAME (VARCHAR(80)): The name of the user who signed.
- TMSTAMP (VARCHAR(80)): The time at which the signature was placed.
- COMMENT (VARCHAR(250)): The comment entered by the signer.
- SDOC (CLOB): XML string with information about the object version that was signed.
- SIGNAT (CLOB): Encrypted signature.
- PUBKEY (CLOB): Public key of the signer.

21.1.47 Table STSETTINGS

This table holds the database system settings. Each record specifies a single setting.

- SETTID (VARCHAR(80)): The name of the database system setting.
- SETTYPE (VARCHAR(80)): The type of database system setting.
- SETTCONTENT (CLOB): The content of the database system setting.

21.1.48 Table SUBSETMEMBERS

This table contains information about the subsets that were defined in the database. Each record specifies the membership of a single entry to a single subset.

- OBJACTIONID (NUMBER): Unique serial number of an action on a subset member object.
- KEY (VARCHAR(80)): The key of the database entry.
- SUBSET (VARCHAR(80)): The name of the subset to which this key belongs.

21.1.49 Table SUBSETS

This table contains information about the subsets that were defined in the database. Each record corresponds to a subset that was created in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a subset object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- SUBSET (VARCHAR(80)): The name of the subset.

21.1.50 Table TRENDATA

Holds information about the trend data types.

- OBJACTIONID (NUMBER): Unique serial number of an action on a trend data object.
- KEY (VARCHAR(80)): The key of the database entry.
- EXPERIMENT (VARCHAR(80)): Name of the trend data type.
- CURVE (VARCHAR (80)): Name of the trend curve.
- DATA (CLOB): XML string that holds the data.
- PARAMS (CLOB): Lists the parameter(s) defined for the trend data type.

21.1.51 Table TRENDEXPERS

Each record in the table corresponds to a trend data experiment belonging to a single entry in the database.

- OBJACTIONID (NUMBER): Unique serial number of an action on a trend data experiment object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.

- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- KEY (VARCHAR(80)): The key of the entry to which the trend data experiment belongs to.
- EXPERIMENT (VARCHAR(80)): The name of the trend data type experiment.
- CREATIONDATE (VARCHAR(40)): The time at which the trend data experiment was created.
- MODIFIEDDATE (VARCHAR(40)): The time at which the trend data experiment was last modified.

21.1.52 Table USERGROUPS

This table holds information on user groups: the three default user groups (Administrators, Powerusers and Users) and any additional user group defined.

- OBJACTIONID (NUMBER): Unique serial number of an action on a user group object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- GROUPNAME (VARCHAR(80)): The user group name.
- DESCRIPTION (VARCHAR(200)): An optional description of the user group.

21.1.53 Table USERKEYS

The table contains user keys, that are used to digitally sign object versions.

- OBJACTIONID (NUMBER): Unique serial number of an action on a user key object.
- KEYID (NUMBER): A unique serial number of the user key.
- USERID (VARCHAR(80)): The user ID of the user to whom the key belongs to.
- PRVKEY (CLOB): The private key used for encryption of the signature.
- PUBKEY (CLOB): The public key, for verification of the private key.
- VALIDSTART (VARCHAR(80)): The time from which the user key is valid and can be used to sign object versions.
- VALIDSTOP (VARCHAR(80)): The time until which the user key is valid and can be used to sign object versions.

21.1.54 Table USERLOG

Contains a record for each user action, tracked in the user activity log (e.g. logins, failed authentication attempts, failed sign actions and changes made to database system parameters).

- USERLOGID (NUMBER): A unique serial number for the user action.

- APUSER (VARCHAR(80)): The database user who performed the action.
- OSUSER (VARCHAR(80)): The Windows user that was logged on when the action was performed.
- LOGTYPE (VARCHAR(20)): The type of action (LOGIN, INFO, ERROR or SYSTEM).
- LOGCOMMENT (VARCHAR(80)): Description of the user action.
- DATETIMESTART (VARCHAR(50)): Time at which the action started.
- DATETIMESTOP (VARCHAR(50)): Time at which the action stopped.

21.1.55 Table USERMEMB

The information in this table keeps track of which user belongs to which user group(s).

- OBJACTIONID (NUMBER): Unique serial number of an action on a user group membership object.
- GROUPNAME (VARCHAR(80)): The user group name.
- USERID (VARCHAR(80)): The user ID of the database user.

21.1.56 Table USERS

This table contains a record for each database user.

- OBJACTIONID (NUMBER): Unique serial number of an action on a user object.
- OBJLCK (NUMBER): Whether or not the object is locked (0 = not locked; 1 = locked).
- OBJOWNER (VARCHAR(80)): The database user who has the ownership of the object.
- OBJSHARED (NUMBER): Whether or not the object is shared (0 = not shared; 1 = shared).
- USERID (VARCHAR(80)): The user ID of the database user.
- NAME (VARCHAR(200)): The name of the database user.
- PSWRD (VARCHAR(250)): Encrypted password of the database user.
- PSWRDDATE (VARCHAR(250)): The time at which the password was created.

21.1.57 Audit trail tables

Most tables, discussed in previous paragraphs, have an audit trail table counterpart. These are indicated with a "Z_" prefix. The audit trail tables are used to store historic versions of the corresponding database objects. The tables are required by the software, even if the audit trail and versioning tool is disabled.

21.1.58 Indices in the database

21.1.58.1 Using indices

In order to obtain sufficient speed for larger databases, it is absolutely necessary that a number of indices are present. This section contains a list of advised indices. However, depending on the purpose of the database (emphasis on read or write, database size ...), it may be preferable to modify, add or remove indices. For larger databases where speed becomes critical, it is strongly advised to use the tuning tools provided with the database in order to optimize the various settings and indices.

21.1.58.2 ENTRYTABLE

- KEY: May be defined as primary key.

21.1.58.3 EXPERIMENTS

- EXPERIMENT: May be defined as primary key. This usually won't attribute to the performance, since the number of records in this table is usually very limited.

21.1.58.4 FPRINTFILES

- FILENAME: May be defined as primary key.

21.1.58.5 FPRINT

- KEY: It should not be unique or primary key, since some lanes on a gel image may not be added to the database and will have an empty key (e.g. reference lanes).
- FILENAME: Note that this field should not be required, because some databases may contain fingerprints that are not associated with any batch (file).
- FILENAME
- FILEIDX

21.1.58.6 Character values table

- CHARACTER
- KEY

21.1.58.7 Character fields table

- CHARACTER
- FIELD

21.1.58.8 SEQUENCES

- KEY

21.1.58.9 MATRIXVALS

- EXPERIMENT
- KEY1
- KEY2

21.1.58.10 SUBSETMEMBERS

- KEY
- SUBSET

Chapter 21.2

Regular expressions

21.2.1 Understanding regular expressions

A *regular expression* is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions. "grep" understands two different versions of regular expression syntax: "basic" and "extended". In "GNU grep", there is no difference in available functionality using either syntax. In other implementations, basic regular expressions are less powerful. The following description applies to extended regular expressions; differences for basic regular expressions are summarized afterwards.

The fundamental building blocks are the regular expressions that match a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any meta character with special meaning may be quoted by preceding it with a backslash. A list of characters enclosed by "[" and "]" matches any single character in that list; if the first character of the list is the caret "^", then it matches any character **not** in the list. For example, the regular expression "[0123456789]" matches any single digit. A range of ASCII characters may be specified by giving the first and last characters, separated by a hyphen.

Finally, certain named classes of characters are predefined, as follows. Their interpretation depends on the "LC_CTYPE" locale; the interpretation below is that of the POSIX locale, which is the default if no "LC_CTYPE" locale is specified.

- "[[:alnum:]]": Any of "[[:digit:]]" or "[[:alpha:]]".
- "[[:alpha:]]": Any letter: "a b c d e f g h i j k l m n o p q r s t u v w x y z", "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z".
- "[[:blank:]]": Space or tab.
- "[[:cntrl:]]": Any character with octal codes 000 through 037, or "DEL" (octal code 177).
- "[[:digit:]]": Any one of "0 1 2 3 4 5 6 7 8 9".
- "[[:graph:]]": Anything that is not a "[[:alnum:]]" or "[[:punct:]]".
- "[[:lower:]]": Any one of "a b c d e f g h i j k l m n o p q r s t u v w x y z".
- "[[:print:]]": Any character from the "[[:space:]]" class, and any character that is **not** in the "[[:graph:]]" class.
- "[[:punct:]]": Any one of "! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { | } ~".
- "[[:space:]]": Any one of "CR FF HT NL VT SPACE".

- "[upper:]": Any one of "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z".
- "[xdigit:]": Any one of "a b c d e f A B C D E F 0 1 2 3 4 5 6 7 8 9".

For example, "[[:alnum:]]" means "[0-9A-Za-z]", except the latter form is dependent upon the ASCII character encoding, whereas the former is portable. (Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket list.) Most meta characters lose their special meaning inside lists. To include a literal "]", place it first in the list. Similarly, to include a literal "'", place it anywhere but first. Finally, to include a literal "-", place it last.

The period "." matches any single character. The symbol "\w" is a synonym for "[[:alnum:]]" and "\W" is a synonym for "[^[:alnum:]]".

The caret "^" and the dollar sign "\$" are meta characters that respectively match the empty string at the beginning and end of a line. The symbols "<" and ">" respectively match the empty string at the beginning and end of a word. The symbol "\b" matches the empty string at the edge of a word, and "\B" matches the empty string provided it is not at the edge of a word.

A regular expression may be followed by one of several repetition operators:

- "?": The preceding item is optional and will be matched at most once.
- "*": The preceding item will be matched zero or more times.
- "+": The preceding item will be matched one or more times.
- "{N}": The preceding item is matched exactly N times.
- "{N,}": The preceding item is matched N or more times.
- "{N,M}": The preceding item is matched at least N times, but not more than M times.

Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated subexpressions.

Two regular expressions may be joined by the infix operator "|"; the resulting regular expression matches any string matching either subexpression.

Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole sub expression may be enclosed in parentheses to override these precedence rules.

The backreference "\N", where N is a single digit, matches the substring previously matched by the N^{th} parenthesized sub expression of the regular expression.

Bibliography

- [1] S.G. Acinas, R. Sarma-Rupavtarm, V. Klepac-Ceraj, and M.F. Polz. Pcr-induced sequence artifacts and bias: insights from comparison of two 16s rna clone libraries constructed from the same sample. *Applied and Environmental Microbiology*, 71(12):8966–8969, 2005.
- [2] H.T. Allawi and J. SantaLucia Jr. Thermodynamics and nmr of internal g-t mismatches in dna. *Biochemistry*, 36(34):10581–10594, 1997.
- [3] H.T. Allawi and J. SantaLucia Jr. Nearest neighbor thermodynamic parameters for internal g-a mismatches in dna. *Biochemistry*, 37(8):2170–2179, 1998.
- [4] H.T. Allawi and J. SantaLucia Jr. Nearest-neighbor thermodynamics of internal a-c mismatches in dna: Sequence dependence and ph effects. *Biochemistry*, 37(26):9435–9444, 1998.
- [5] H.T. Allawi and J. SantaLucia Jr. Thermodynamics of internal c-t mismatches in dna. *Nucleic acids research*, 26(11):2694, 1998.
- [6] S.F. Altschul et al. Amino acid substitution matrices from an information theoretic perspective. *Journal of molecular biology*, 219(3):555–565, 1991.
- [7] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.
- [8] K.E. Ashelford, N.A. Chuzhanova, J.C. Fry, A.J. Jones, and A.J. Weightman. At least 1 in 20 16s rna sequence records currently held in public repositories is estimated to contain substantial anomalies. *Applied and Environmental Microbiology*, 71(12):7724–7736, 2005.
- [9] S. Boisvert, F. Laviolette, and J. Corbeil. Ray: Simultaneous assembly of reads from a mix of high-throughput sequencing technologies. *Journal of Computational Biology*, 17(11):1519–1533, 2010.
- [10] S. Bommarito, N. Peyret, et al. Thermodynamic parameters for dna sequences with dangling ends. *Nucleic Acids Research*, 28(9):1929, 2000.
- [11] I. Borg and P.J.F. Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.
- [12] K.P. Choi, F. Zeng, and L. Zhang. Good spaced seeds for homology search. 2004.
- [13] P. Du, Kibbe W.A., and Lin S.M. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006.
- [14] B. Efron. Bootstrap methods: another look at the jackknife. *The annals of Statistics*, pages 1–26, 1979.
- [15] B. Ewing, L.D. Hillier, M.C. Wendl, and P. Green. Base-calling of automated sequencer traces using-phred. i. accuracy assessment. *Genome research*, 8(3):175, 1998.
- [16] J. Felsenstein. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, pages 783–791, 1985.

- [17] B.J. Haas, D. Gevers, A.M. Earl, M. Feldgarden, D.V. Ward, G. Giannoukos, D. Ciulla, D. Tabbaa, S.K. Highlander, E. Sodergren, et al. Chimeric 16s rRNA sequence formation and detection in sanger and 454-pyrosequenced PCR amplicons. *Genome research*, 21(3):494–504, 2011.
- [18] E. Hellinger. *Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen*. PhD thesis, G. Reimer, 1909.
- [19] H.S. Horn. Measurement of "overlap" in comparative ecological studies. *American naturalist*, pages 419–424, 1966.
- [20] L. Ilie and S. Ilie. Multiple spaced seeds for homology search. *Bioinformatics*, 23(22):2969, 2007.
- [21] T.H. Jukes and C.R. Cantor. Evolution of protein molecules. 1969.
- [22] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of molecular evolution*, 16(2):111–120, 1980.
- [23] I. Korf, M. Yandell, and J. Bedell. *Blast*. O'Reilly & Associates, Inc., 2003.
- [24] B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440, 2002.
- [25] T. Madden. The blast sequence analysis tool. *The NCBI Handbook [Internet]*. National Library of Medicine (US), National Center for Biotechnology Information, Bethesda, MD, 2002.
- [26] C.D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. 2008.
- [27] M. Morisita. Measuring of the dispersion of individuals and analysis of the distributional patterns. *Mem. Fac. Sci. Kyushu Univ. Ser. E*, 2(21):5–23, 1959.
- [28] M. Morisita. I σ -index, a measure of dispersion of individuals. *Researches on population ecology*, 4(1):1–7, 1962.
- [29] NCBI. Blast basic local alignment search tool: Blast program selection guide. *The NCBI Handbook [Internet]*. National Library of Medicine (US), National Center for Biotechnology Information, Bethesda, MD, 2009.
- [30] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
- [31] M. Nei and T. Gojobori. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. *Molecular biology and evolution*, 3(5):418, 1986.
- [32] MS Nikulin. Hellinger distance. *Encyclopaedia of Mathematics*. Kluwer Academic Publishers, 2002.
- [33] M.S. Rajeevan, I.M. Dimulescu, E.R. Unger, and S.D. Vernon. Chemiluminescent analysis of gene expression on high-density filter arrays. *Journal of Histochemistry & Cytochemistry*, 47(3):337, 1999.
- [34] J. SantaLucia Jr. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proceedings of the National Academy of Sciences*, 95(4):1460, 1998.
- [35] P.D. Schloss, S.L. Westcott, T. Ryabin, J.R. Hall, M. Hartmann, E.B. Hollister, R.A. Lesniewski, B.B. Oakley, D.H. Parks, C.J. Robinson, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and environmental microbiology*, 75(23):7537–7541, 2009.
- [36] W. Smith, A.R. Solow, and P.E. Preston. An estimator of species overlap using a modified beta-binomial model. *Biometrics*, pages 1472–1477, 1996.

- [37] D.J. States, W. Gish, and Altschul S.F. Improved sensitivity of nucleic acid database searches using application-specific scoring matrices. *Methods: A companion to Methods in Enzymology*, (3):66–70, 1991.
- [38] GC Wang and Y. Wang. Frequency of formation of chimeric molecules as a consequence of pcr coamplification of 16s rna genes from mixed bacterial genomes. *Applied and Environmental Microbiology*, 63(12):4645–4650, 1997.
- [39] G.C.Y. Wang and Y. Wang. The frequency of chimeric molecules as a consequence of pcr coamplification of 16s rna genes from different bacterial species. *Microbiology*, 142(5):1107–1114, 1996.
- [40] Q. Wang, G.M. Garrity, J.M. Tiedje, and J.R. Cole. Naive bayesian classifier for rapid assignment of rna sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, 73(16):5261–5267, 2007.
- [41] W.J. Wilbur and D.J. Lipman. Rapid similarity searches of nucleic acid and protein data banks. *Proceedings of the National Academy of Sciences*, 80(3):726, 1983.
- [42] J.C. Yue, M.K. Clayton, and F.C. Lin. A nonparametric estimator of species overlap. *Biometrics*, 57(3):743–749, 2004.
- [43] D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821, 2008.



A B I O M É R I E U X C O M P A N Y

Copyright 1998-2018, Applied Maths NV. All rights reserved.

Please contact us for any additional information you might require, we will gladly help you!

Headquarters

📍 Keistraat 120 • 9830 Sint-Martens-Latem • Belgium
☎ +32 922 22 100 ✉ info@applied-maths.com

USA and Canada

📍 11940 Jollyville Rd., Suite 115N • Austin, TX 78750 USA
☎ +1 512 482 9700 ✉ info-us@applied-maths.com