BIOMÉRIEUX

BIONUMERICS®

version 8 - PLUGINS

WGS tools local plugin

Contents

NOTES

SUPPORT BY APPLIED MATHS, A BIOMÉRIEUX COMPANY

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, a bioMérieux company, 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: BE-DAU-INFO@biomerieux.com
URL: https://www.bionumerics.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: US-DAU-INFO@biomerieux.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.

BIONUMERICS uses following third-party software tools and libraries:

- Python 3.8 release from the Python Software Foundation, https://www.python.org/
- Xerces library for XML input and output from the Apache Software Foundation, https://xerces.apache.org/
- NCBI toolkit version 2.11.0, https://www.ncbi.nlm.nih.gov/BLAST/
- SRA Toolkit, https://ncbi.github.io/sra-tools/
- Boost c++ libraries, https://www.boost.org/
- Samtools for interacting with SAM / BAM files, https://www.htslib.org/download/
- 7-Zip (7za.exe), https://www.7-zip.org/
- Zlib library, https://zlib.net/
- Pigz for parallel gzip compression, https://zlib.net/pigz/
- Cairo 2D graphics library version 1.12.14, https://cairographics.org/
- Crypto++ library version 5.5.2, https://www.cryptopp.com/
- OpenSSL library, https://www.openssl.org/
- libSVM library for Support Vector Machines, https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- SQLite version 3.7.17, https://www.sqlite.org/
- pymzML Python module version 2.4.7, https://github.com/pymzml/pymzML
- NumPy Python library version 1.19.1, https://www.numpy.org/
- BioPython Python library version 1.78, https://www.biopython.org/
- pyodbc Python module version 4.0.30, https://pypi.org/project/pyodbc/
- jinja2 Python library version 2.11.2, https://pypi.org/project/Jinja2/
- MarkupSafe Python library version 1.1.1, https://pypi.org/project/MarkupSafe/
- regex Python library version 2.5.91, https://pypi.org/project/regex/
- Chromium Embedded Framework, https://bitbucket.org/chromiumembedded/cef/wiki/Home
- SPAdes genome assembler version 3.15.3, https://bioinf.spbau.ru/spades *
- SKESA version 2.3.0, https://github.com/ncbi/SKESA/releases
- Unicycler version 0.5.0, https://github.com/rrwick/Unicycler/releases *
- Velvet for Windows, source code can be downloaded from https://www.bionumerics.com/download/open-source
- Bowtie2 version 2.2.5 (https://bowtie-bio.sourceforge.net/bowtie2/index.shtml)*
- SNAP version 2.0.0, https://www.microsoft.com/en-us/research/project/snap/
- RAxML version 8.2.11, https://github.com/stamatak/standard-RAxML/releases

- FastTree version 2.1.10, https://www.microbesonline.org/fasttree/
- CFSAN SNP pipeline version 2.2.0, https://github.com/CFSAN-Biostatistics/snp-pipeline *
- Prokka version 1.14.5, https://github.com/tseemann/prokka *
- sourmash version 4.1.0, https://github.com/dib-lab/sourmash **
- SeqSero2 for Windows, source code can be downloaded from https://www.bionumerics.com/download/open-source
- Fastp version 0.22.0, https://github.com/OpenGene/fastp

*: On Calculation Engine only **: See license conditions below

Sourmash license conditions:

Chapter 1

Purpose of the WGS tools local plugin

The Calculation Engine is a server application that provides access from the BIONUMERICS desktop client application to a High-Performance Computing (HPC) environment for doing calculation-intensive tasks in Whole Genome Sequencing (WGS) data analysis. A nomenclature service for whole genome Multi-Locus Sequence Typing (wgMLST) is integrated into the Calculation Engine. The Calculation Engine is installed on a powerful computer cluster, either on physical servers present on premises or in the cloud. The default Calculation Engine instance is the Applied Maths Cloud Calculation Engine, which is hosted on Amazon Web Services (AWS) servers in the USA and accessible worldwide.

In October 2021, bioMérieux announced the phasing out of the BIONUMERICS software. This phasing out entails that version 8.1.1 (released on October 6th, 2022) is the final BIONUMERICS version released and that customer support for the software will stop on December 31st, 2024. At the same time, the Applied Maths Cloud Calculation Engine will be taken offline.

The ability to run assembly-based wgMLST analysis locally was already introduced, but the *WGS tools plugin* still relies on the Calculation Engine's nomenclature services, notably to get allele IDs for new alleles. The purpose of the *WGS tools local plugin* is to allow BIONUMERICS version 8.1.1 users to continue running local assembly-based wgMLST analysis after the Applied Maths Cloud Calculation Engine is shut down.

Hash-based allele calling is a viable option to perform wgMLST without the need for a centrally maintained nomenclature. In computer science, a hash function is mathematical function that maps data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values or hashes. Specific for the wgMLST allele calling process, allele sequences that fulfill the nomenclature acceptance criteria are converted by a hash function into 17-digit integer numbers. Since hash functions are deterministic and because obviously the *WGS tools local plugin* always uses the same hash function implementation, a given allele sequence will always result into the same hash value. However, hash values as such cannot be used directly, because the character experiment type in BIONUMERICS is not designed to reliably store and compare 17-digit integers. In addition, hash values look very different from the simple integer allele IDs that wgMLST users are familiar with.

A possible concern with hash functions are so-called *collisions*, where different inputs result in the same hash value being generated. The *WGS tools local plugin* calculates hash values in the same way as the hash values that are sent to and stored in the Calculation Engine nomenclature databases. It is therefore reassuring to know that even in large allele databases such as those of *Salmonella* and *E. coli*, no collisions were observed in the nine years that the latter databases are in use.

The *WGS tools local plugin* sets up a local allele nomenclature in which the hashes are converted into simple integer IDs. This local nomenclature is stored as a file in the source files directory (see

the Reference manual, Chapter The BIONUMERICS relational database) to ensure that all users of the same database have access to the same nomenclature. The nomenclature assigns incremental integer IDs to unique sequence hash values per locus. The simple integer allele IDs from the local nomenclature are stored in the **wgMLST_Local** experiment type, which is automatically created and synchronized with the **wgMLST** experiment type by the *WGS tools local plugin*. The **wgMLST_Local** experiment type only considers assembly-based wgMLST allele calls, assembly-free allele calling is not taken into account.

To allow exchange of wgMLST data between different labs and/or databases, the *WGS tools local plugin* provides functionality to export local wgMLST profiles as hash values, and to import local wgMLST profiles from hash values (see 5).

Chapter 2

Starting and setting up BIONUMERICS

This guide is designed as a manual for the *WGS tools local plugin* of BIONUMERICS. The *WGS tools local plugin* can be considered an add-on for the *WGS tools plugin* that allows the latter to work without an external Calculation Engine by providing a local nomenclature for wgMLST (see 1).

The *WGS tools local plugin* is supported in the **BIONUMERICS-SEQ** and **BIONUMERICS-SUITE** configurations and can only be installed on top of a *WGS tools plugin* installation.

Make sure the latest version of BIONUMERICS is installed (https://www.bionumerics.com/download/software). The installation manual can be downloaded from https://www.bionumerics.com/download/manuals.

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 2.1).

A new BIONUMERICS database is created from the Startup program by pressing the button.

An existing database is opened in BIONUMERICS with or by simply double-clicking on a database name in the list.

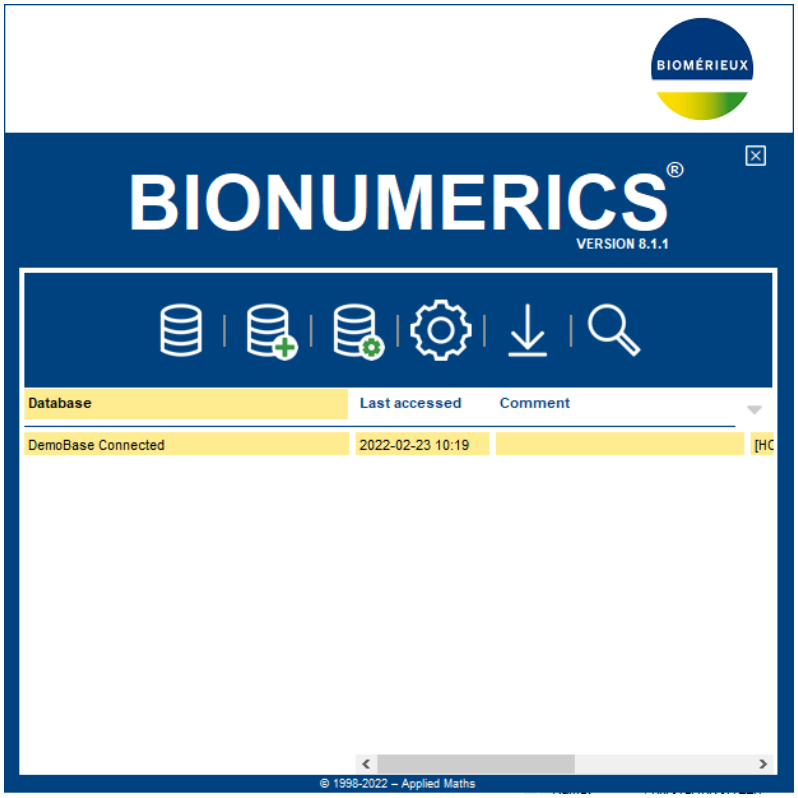

Figure 2.1: The *BIONUMERICS Startup* window.

Chapter 3

Installing the WGS tools local plugin

3.1 Prerequisites

The *WGS tools local plugin* is designed to work with the latest BIONUMERICS version. Earlier versions are not tested an might have compatibility issues. Please check our software download page (https://www.bionumerics.com/download/software) to see if an update to your current version is available.

The *WGS tools local plugin* should be installed in a BIONUMERICS database in which the *WGS tools plugin* is already installed and connected to a Calculation Engine instance via a CE project and password.

Installation of the *WGS tools local plugin* should be done before December 31th, 2024 while the Applied Maths Cloud Calculation Engine is still up and running.

3.2 Installation procedure

The *WGS tools local plugin* is made available as an online plugin, which can be installed in the relational database. This has as an advantage that no Windows administrator rights are required for installation. Furthermore, in a multi-user database setup, this procedure ensures that all database users work with the same plugin version.

Proceed as follows to install the *WGS tools local plugin*:

2.1 Select ***File*** > ***Install / remove plugins...*** ( ) in the *Main* window to call the *Plugins and Scripts* dialog box.

2.2 Press the <***Manage database plugins***> button to open the *Manage database plugins* dialog box.

The *Manage database plugins* dialog box lists the plugins that are currently stored in the relational database. Likely, this list is initially empty.

2.3 Press the <***Add/Update***> button to open the *Add database plugins* dialog box (see Figure 3.1.

2.4 Check the check box in front of the *WGS tools local plugin* and click <***OK***>.

A message appears, indicating that the plugin will be loaded after the database is restarted.

2.5 Press <***Close***> in the *Add database plugins* dialog box and repeat the same action in the *Plugins and Scripts* dialog box.

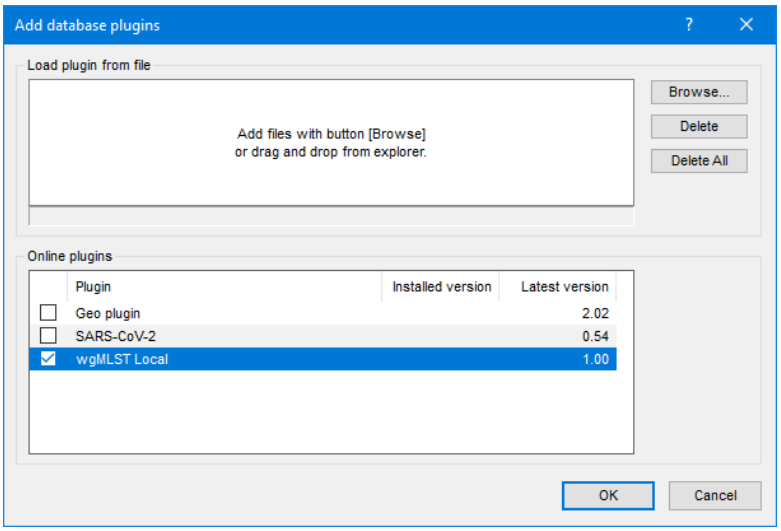

Figure 3.1: The *Add database plugins* dialog box, listing the *WGS tools local plugin*.

2.6 Close and restart the BIONUMERICS database.

2.7 Select ***File*** > ***Install / remove plugins...*** ( ) in the *Main* window to call the *Plugins and Scripts* dialog box again.

The *WGS tools local plugin* is now displayed in the *Plugins tab* of the *Plugins and Scripts* dialog box at the bottom of the list and is preceded by a database icon (see Figure 3.2).

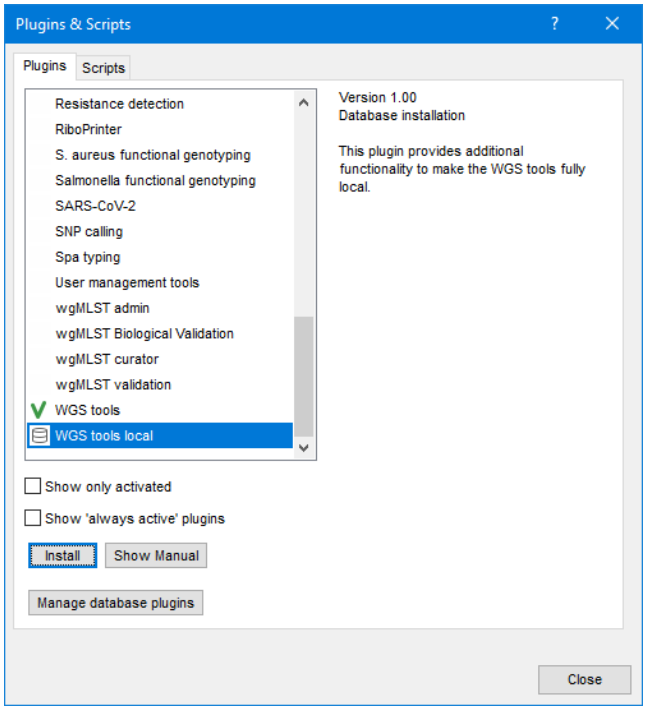

Figure 3.2: The *Plugins and Scripts* dialog box with the *WGS tools local plugin* highlighted.

2.8 Click on the *WGS tools local plugin* to highlight it, press <***Install***> and confirm the plugin installation.

The plugin automatically creates and initiates an experiment type **wgMLST_Local** by performing

a synchronization with the Calculation Engine.

The *WGS tools local plugin* will use the experiment type that is specified to contain the wgMLST allele calls in the *Experiment types panel* of the *Calculation engine settings* dialog box and will add the **_Local** suffix to the experiment name. Even though the actual name might be different, the experiment type will be referred to as **wgMLST_Local** for reasons of conciseness.

2.9 When the synchronization is complete, press <***OK***> to close the message box.

A message box pops up with the question ”Do you want to initialize the local allele nomenclature with the accepted alleles?”.

Answering “No” will create a local wgMLST nomenclature from scratch: allele IDs are assigned starting from 1 for each locus.

Answering “Yes” will start the local nomenclature from the current accepted alleles in the central nomenclature and new alleles will be added by increasing integer identifiers.

When the assembly-based accepted alleles are next updated, there might be newly accepted alleles with IDs that have meanwhile already been assigned to (other) local alleles. Inevitably, from the point of initialization on, the allele IDs will start to diverge!

If the assembly-based wgMLST search data are not up-to-date, an update of the search data will be performed first. This may take several minutes.

2.10 Select <***Yes***>.

This action will create a file `wgMLST_Local_Nomenclature.txt` in the source files directory with the local nomenclature.

A notification appears when the process is completed, indicating that the plugin is installed and prompting to restart the database.

2.11 Press <***Close***> in the *Plugins and Scripts* dialog box and close and restart the BIONUMERICS database.

Chapter 4

wgMLST allele calling using local nomenclature

4.1 Principle

At the start of each session, the *WGS tools local plugin* checks if a connection to a Calculation Engine instance is available and adjusts its functionality accordingly.

4.2 Prior to Calculation Engine shutdown

When a connection to a Calculation Engine instance is available, all functionality provided by the *WGS tools plugin* remains available. Additionally, each time an assembly-based allele calling result is retrieved (no matter if it is run locally or on the Calculation Engine), the assembly-based allele calls according to the local wgMLST nomenclature (see 1) will be imported in the **wgMLST_Local** experiment.

Hence, to populate the **wgMLST_Local** experiments for all entries in the database, it is required to re-run all assembly-based wgMLST calls. On the local calculation engine, this is a relatively quick process that does not require CE credits. For extremely large databases, we recommend submitting the jobs in batches of maximum 1000 entries.

Alleles (represented by their hash value) that are not yet known in the local nomenclature, will be assigned a new allele ID as incremental integers for each locus separately. Assembly-free allele calls are not considered for the **wgMLST_Local** experiment type, even if they are present or retrieved.

For each request made to the Calculation Engine, the response will be cached and stored in the DBSETTINGS table of the relational database.

To avoid unexpected error messages when the Calculation Engine is shut down, it is recommended to execute each command from the ***WGS tools*** menu at least once while a connection to the Calculation Engine is still available.

4.3 After Calculation Engine shutdown

When no connection to a Calculation Engine instance can be made, all functionality for job submission to the external Calculation Engine will be grayed out in the *Submit jobs* dialog box and the

Submit comparison jobs dialog box (see Figure 4.1).

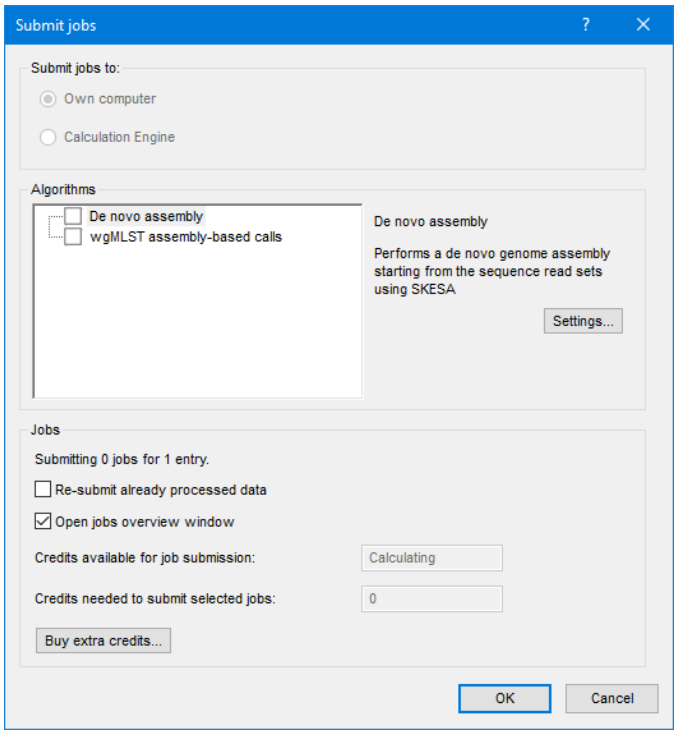

Figure 4.1: The *Submit jobs* dialog box, with the option to submit jobs to the external Calculation Engine grayed out.

As a consequence, jobs can only be submitted to the local calculation engine on your own computer.

When the results are retrieved for a local wgMLST assembly-based allele calling job, the **wgMLST** and **wgMLST_Local** experiments will both be updated. The **wgMLST** experiment will contain consensus calls in case an assembly-free allele calling job was run earlier for the same entry. In case new alleles are found, the allelic profile stored in **wgMLST** will be incomplete, since these new alleles cannot be submitted to the Calculation Engine nomenclature database. The allelic profile stored in the **wgMLST_Local** experiment (obtained through hash-based allele calling) will be complete and based on the local nomenclature maintained by the *WGS tools local plugin*.

For any request made to the Calculation Engine, a cached response will be served if available. If not, an error message will be generated. Some wgMLST functionality will not be available anymore because it relies on specific nomenclature services:

- ***WGS tools*** > ***Assign wgMLST sequence types...***: "The 'Assign sequence types' nomenclature service is no longer available". MLST sequence types can still be assigned using the *MLST for WGS plugin*.
- ***WGS tools*** > ***View wgMLST reports...*** shows only cached reports, otherwise: "The 'Update reports' nomenclature service is no longer available". A similar report can be generated by the *MLST for WGS plugin*.
- ***WGS tools*** > ***Store wgMLST locus sequences...***: "The 'Get allele sequences' nomenclature service is no longer available". The *MLST for WGS plugin* can optionally store sequences for MLST loci.
- ***WGS tools*** > ***Get alleles mapping***: "The 'Get allele mapping' nomenclature service is no longer available". For smaller schemes, the *MLST for WGS plugin* can again provide an

alternative.

- ***Alleles*** $>$ ***Open alignment...*** () in the *wgMLST quality assessment* window: ”The ’Get allele sequences’ nomenclature service is no longer available”. An alignment of allele sequences can only be generated manually, after import of the allele sequences from the accepted alleles file from the assembly-based allele calling search data.

Chapter 5

Exporting and importing hash-based allele profiles

5.1 Principle

Because a *local* nomenclature is used, **wgMLST_Local** allelic profiles from different BIONUMERICS databases cannot be directly compared with each other. In order to exchange and compare data with other databases or labs, the allele IDs need to be converted into something that can be uniquely referenced, i.e. the hash values obtained from the allele sequence.

The import and export wizard have additional methods in ***Fields and characters*** to import and export hashed wgMLST calls. The look-and-feel is almost identical to the default import and export, except that the character experiments are restricted to the **wgMLST_Local** experiment type, and all (local) allele IDs are converted to and from allele hashes.

When importing an allele hash that is not yet present in the local nomenclature, it will be added (i.e., assigned a new allele ID), and subsequent local detections of the same allele will be assigned this new allele ID.

5.2 Exporting profiles

Proceed as follows to export allele hashes for a set of entries:

2.1 Select the entries that you want to export and use ***File*** > ***Export...***.

2.2 In the *Export* dialog box, under the topic ***Character type data***, highlight ***Export fields and hashed wgMLST calls*** and press <***Next***>.

2.3 In the *Export* dialog box that appears, select the ***Fields*** and the subset of the **wgMLST_Local** experiment (under ***Characters***) that you wish to include in the export. Press <***Next***>.

2.4 In the *Settings* dialog box, specify how absent values are denoted and whether or not the export should be limited to the active characters only. Press <***Finish***>.

This action will create the `export.csv` in the database directory and will open the file in your computer's default editor for `*.csv` files.

MS Excel, which is the default CSV editor on most PCs, displays the allele hashes in scientific notation (e.g. 5,83E+16). For categorical data, this does in fact not make sense.

By specifying “No” for the preference ***Export table files in CSV format***, allele hashes files will be exported in tab-delimited text format (see the Reference manual, Chapter The BIONUMERICS user interface).

5.3 Importing profiles

Proceed as follows to import entries with their allele hashes from an exported text file:

3.1 Select ***File*** > ***Import...*** (, **Ctrl+I**).

3.2 Browse for the text file (`*.csv` or `*.txt`) exported earlier, highlight the option ***Import fields and hashed wgMLST calls (text file)*** and press <***Finish***>.

3.3 Press <***Next***>.

The remainder of the procedure is identical to importing fields and character data from text files (see the Reference manual, Chapter Setting up character type experiments), except that only the **wgMLST_Local** character type can be selected as target.

bioMérieux S.A. • 69280 Marcy l'Étoile • France • Tel.: + 33 (0)4 78 87 20 00 • Fax: +33 (0)4 78 87 20 90
www.biomerieux.com