BIOMÉRIEUX

BIONUMERICS®

version 8 - PLUGINS

Whole Genome Sequence tools plugin

Contents

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: BE-DAU-INFO@biomerieux.com
URL: https://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: 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 2.7.4 release from the Python Software Foundation, http://www.python.org/
- Xerces library for XML input and output from the Apache Software Foundation, https://xerces.apache.org/
- NCBI toolkit version 2.2.28, http://www.ncbi.nlm.nih.gov/BLAST/
- SRA Toolkit, https://ncbi.github.io/sra-tools/
- Boost c++ libraries, http://www.boost.org/
- Samtools for interacting with SAM / BAM files, http://www.htslib.org/download/
- 7-Zip (7za.exe), http://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, http://cairographics.org/
- Crypto++ library version 5.5.2, http://www.cryptopp.com/
- OpenSSL library, https://www.openssl.org/
- libSVM library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- SQLite version 3.7.17, http://www.sqlite.org/
- 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/
- Chromium Embedded Framework, https://bitbucket.org/chromiumembedded/cef/wiki/Home
- SPAdes genome assembler version 3.13.1, http://bioinf.spbau.ru/spades *
- SKESA version 2.3.0, https://github.com/ncbi/SKESA/releases
- Unicycler version 0.4.8, https://github.com/rrwick/Unicycler/releases *
- Velvet for Windows, source code can be downloaded from https://www.applied-maths.com/download/open-source
- Ray for Windows, source code can be downloaded from https://www.applied-maths.com/download/open-source
- Bowtie2 version 2.2.5 (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml)*
- SNAP version 1.0.18, http://snap.cs.berkeley.edu/
- RAxML version 8.2.11, https://github.com/stamatak/standard-RAxML/releases
- FastTree version 2.1.10, http://www.microbesonline.org/fasttree/

- CFSAN SNP pipeline version 0.8.2, https://github.com/CFSAN-Biostatistics/snp-pipeline *
- Prokka version 1.12, https://github.com/tseemann/prokka *

*: On Calculation Engine only

Chapter 1

Starting and setting up BIONUMERICS

1.1 Introduction

This guide is designed as a manual for the *Whole Genome Sequence tools plugin* (or *WGS tools plugin*) of BIONUMERICS. The *WGS tools plugin* is virtually indispensable when working with whole genome sequences and connects the BIONUMERICS client desktop software to an external Calculation Engine (see 5). Not only does the Calculation Engine provide access to an high performance computing (HPC) environment, to which heavy calculations can be outsourced on demand, it also functions as nomenclature server for **whole genome Multi Locus Sequence Typing (wgMLST)** (see 4).

All communication between the BIONUMERICS client and the Calculation Engine service is handled by the *WGS tools plugin*. Authentication is done through a project name and password, linked to your BIONUMERICS license.

The *WGS tools plugin* is supported in the **BIONUMERICS-SEQ** and **BIONUMERICS-SUITE** configurations.

1.2 Startup program

Make sure the latest version of BIONUMERICS is installed (https://www.applied-maths.com/download/software). The installation manual can be downloaded from https://www.applied-maths.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 1.1).

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.

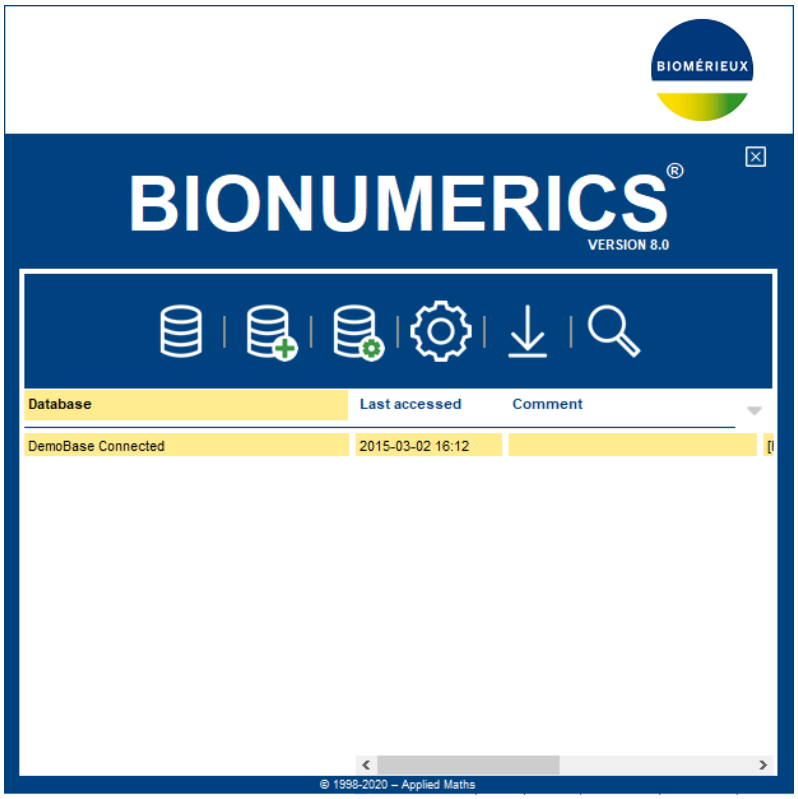

Figure 1.1: The *BIONUMERICS Startup* window.

1.3 Creating a new database

3.1 Press the  button in the BIONUMERICS *BIONUMERICS Startup* window to enter the *New database* wizard.

3.2 Enter a name for the database, and press <***Next***>.

A new dialog box pops up, prompting for the type of database (see Figure 1.2).

3.3 Leave the default option selected and press <***Next***>.

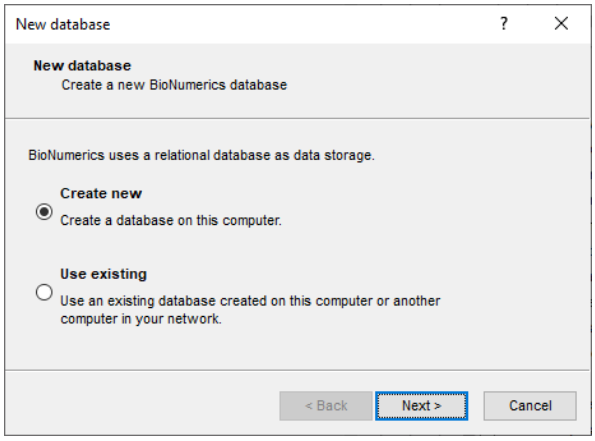

Figure 1.2: The *New database* wizard page.

A new dialog box pops up, prompting for the database engine (see Figure 1.3).

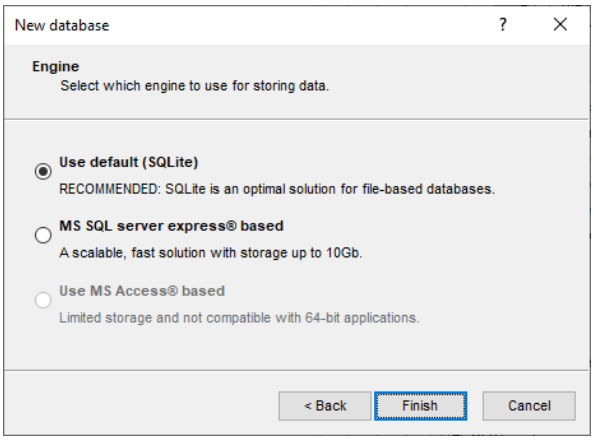

Figure 1.3: The *Database engine* wizard page.

3.4 Leave the default option selected and press <***Finish***> to complete the setup of the new database.

Chapter 2

Installing the WGS tools plugin

Installing a plugin in a BIONUMERICS database is done from the *Plugins* dialog box (see Figure 2.1), which can be called from the *Main* window by selecting ***File*** > ***Install / remove plugins...*** ( ).

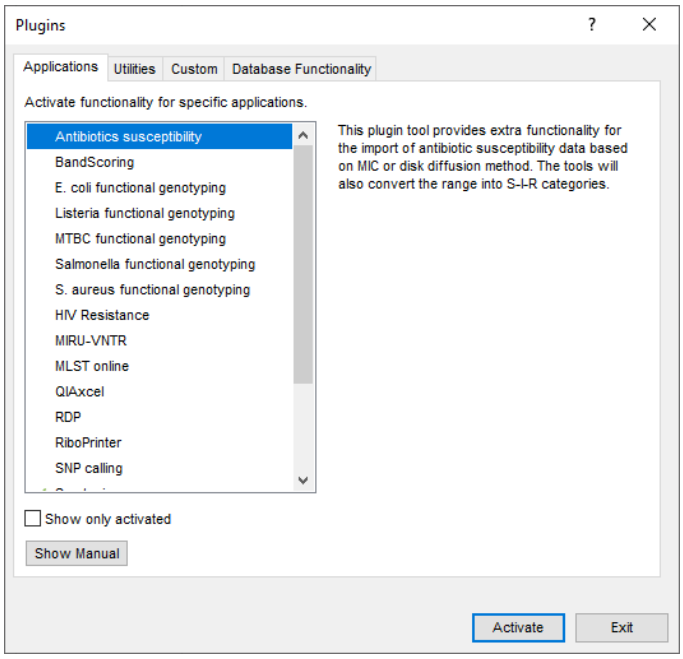

Figure 2.1: The *Plugins* dialog box.

When a particular plugin is selected from the list of plugins, a short description appears in the right panel.

A selected plugin can be installed with the <***Activate***> button. The software will ask for confirmation before installation. Some plugins are only supported in specific BIONUMERICS configurations. If the plugin is not supported by your BIONUMERICS configuration, it 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.

0.1 Select the *WGS tools plugin* from the list in the *Applications tab* and press the <***Activate***> button.

The software asks the user to confirm the installation of the *WGS tools plugin*. After confirmation,

the plugin installation starts and the *WGS tools installation* wizard is shown (Figure 2.2).

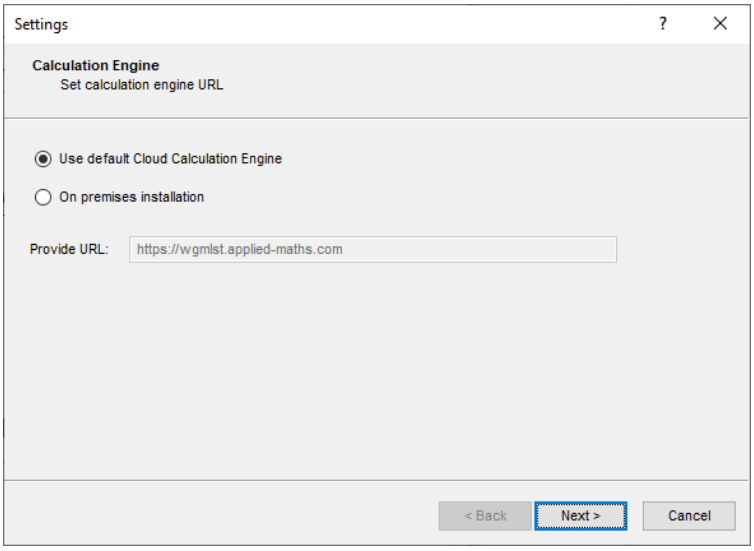

Figure 2.2: The *Calculation engine URL* wizard page in the *WGS tools installation* wizard.

The *Calculation engine URL* wizard page queries for the Uniform Resource Locator (URL) that uniquely identifies the calculation engine instance to connect to. Two options are available:

- ***Use default Cloud Calculation Engine***: Most clients will use the Applied Maths cloud instance (`https://wgmlst.applied-maths.com`), which is hosted on Amazon servers in the US. Note that his option should also be selected if you do not intend to run jobs on the calculation engine, but instead run all calculations on your own computer (see further).
- ***On premises installation***: Choose this option only if you want to connect to another instance, e.g. in case your institute or company has its own installation of the calculation engine server software. The calculation engine URL needs to be requested from your local IT administrators and entered in the corresponding text box.

Press <***Next***> to proceed to the next page in the *Calculation engine URL* wizard page.

If the default Cloud Calculation Engine was selected in the first page of the wizard, the *Organism and project* wizard page will pop up (see Figure 2.3).

Two options are available:

- Choose ***Local calculations only*** if you do not intend to run jobs on the calculation engine and instead wish to run all calculations on your own computer (see 5.3). With this option checked, an ***Organism*** should be selected from the drop-down list. By selecting an organism, credentials to a demo project will be filled in automatically. Demo projects are calculation engine projects to which no credits can be assigned, but that allow BIONUMERICS to download organism-specific settings and search data.
- Choose ***Enable running jobs on Cloud Calculation Engine*** to unlock the full potential of the default Cloud Calculation Engine. In this case, you will need credentials to your own calculation engine project, for which credits can be purchased. Your ***Project name*** and corresponding ***Password*** should be entered in the corresponding text boxes. Pressing <***Request a new CE project***> will direct you to a form on the Applied Maths website where a new calculation engine project can be requested.

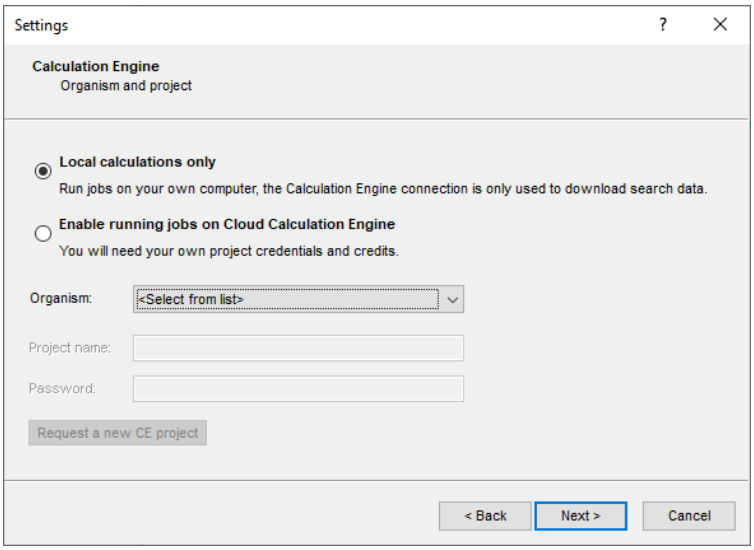

Figure 2.3: The *Organism and project* wizard page in the *WGS tools installation* wizard.

When a ***Project name*** and a ***Password*** are provided (implicitly by selecting an organism from the list or explicitly by entering this information in the corresponding text boxes), press <***Next***> to proceed with the installation.

If an on premises installation of the Calculation Engine was selected in the first page of the wizard, the *Project credentials* wizard page will pop up (see Figure 2.4).

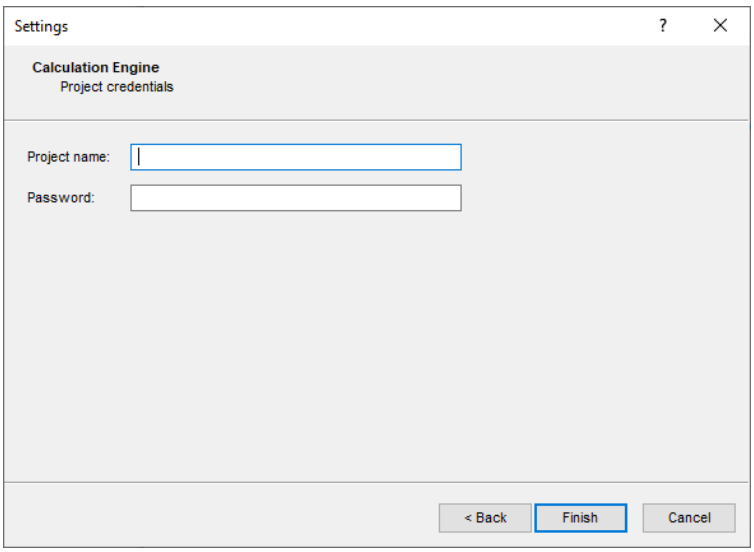

Figure 2.4: The *Project credentials* wizard page in the *WGS tools installation* wizard.

A ***Project name*** and corresponding ***Password*** should be provided, which can be requested from the administrator(s) of the on premises Calculation Engine installation.

Pressing <***Next***> in the *Organism and project* wizard page or *Project credentials* wizard page will download all required information from the calculation engine. Depending on the size of the wgMLST schema and your connection speed, this step may take up to several minutes.

A confirmation dialog is displayed when the synchronization has been completed (see Figure 2.5 for an example).

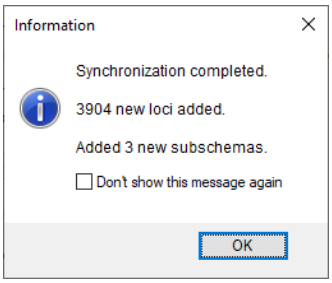

Figure 2.5: Synchronization completed message.

When the *WGS tools plugin* installation is complete, you will be prompted to restart the database. The *Plugins* dialog box can be closed by pressing the <***Exit***> button and the database via ***File*** > ***Exit***.

Open the database again from the *BIONUMERICS Startup* window.

A ***WGS tools*** menu item is available in the *Main* window (see Figure 2.6).

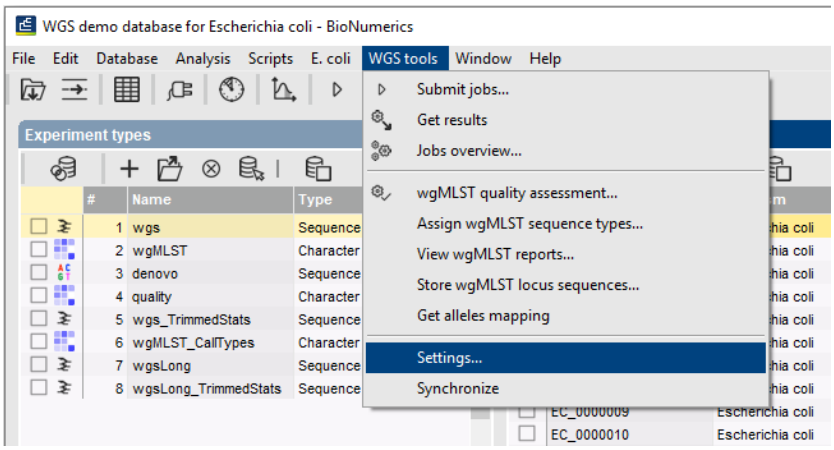

Figure 2.6: The ***WGS tools*** menu items.

Seven experiment types are created in the database (see Figure 2.7):

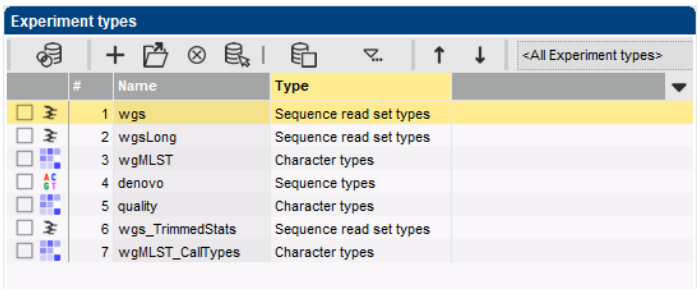

Figure 2.7: Experiment types created by the plugin.

- **wgs**: This sequence read set experiment type contains the links to short read sequence read data (typically Illumina paired-end or single-end datasets).
- **wgsLong**: This sequence read set experiment type optionally contains the links to long read sequence read data (typically PacBio or MinION datasets).

- **wgMLST**: This character experiment type will contain the results from the wgMLST analysis, i.e. the consensus allele calls for all loci.
- **denovo**: This sequence experiment type will contain the results from the de novo assembly, i.e. the concatenated de novo contigs.
- **quality**: This character experiment type will be used to save quality statistics on the read set, the de novo assembly, the allele identification, ...
- **wgs_TrimmedStats**: This sequence read set experiment type will contain some data statistics about the reads retained after trimming.
- **wgMLST_CallTypes**: This character experiment will hold the details on the call types.

Depending on the organism, one or more entry information fields might be created to stored assigned sequence types (see 14 for more information).

During installation of the plugin, the **wgMLST** character experiment is synchronized with the organism-specific wgMLST scheme. All detected loci and subschemes (see Figure 2.5 for an example) are added to this experiment.

For additional settings of the *WGS tools plugin*, see 3.

Chapter 3

WGS tools settings

After installation, settings for the *WGS tools plugin* can be accessed via ***WGS tools*** > ***Settings...***.

The settings in the *Calculation engine settings* dialog box are grouped in four separate tabs:

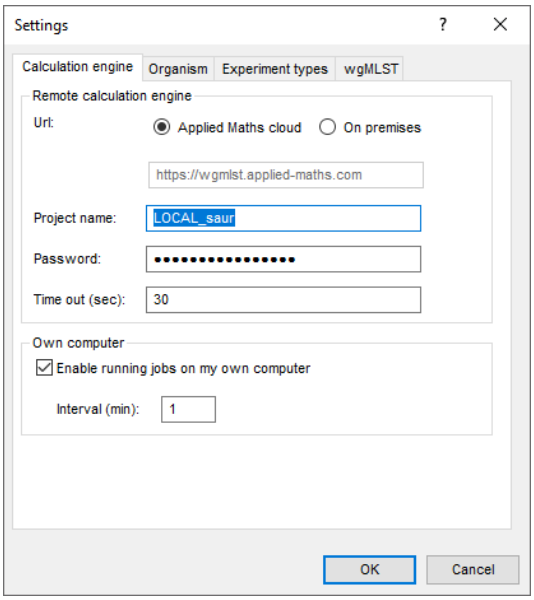

Figure 3.1: The *Calculation engine* tab of the *Calculation engine settings* dialog box.

The calculation engine ***URL*** and credentials to your project (i.e. ***Project name*** and ***Password***) were entered during installation of the *WGS tools plugin* (see 2) and normally do not need to be edited later on.

The ***Time out (sec)*** is the maximum wait time between sending a request to the calculation engine and receiving a response, expressed in seconds. If timeouts are encountered frequently, try increasing the timeout setting. Once a timeout is encountered in a session, it is recommended to close and re-open the database.

The option ***Enable running jobs on my own computer*** should be checked to enable the local calculation engine, i.e. to run jobs on your own computer in the same way as running jobs on the calculation engine (see 5.3 for more information). The ***Interval (min)*** is the time interval expressed in minutes when the software will check if new local jobs are available.

The ***Organism*** that is shown in the corresponding drop-down list cannot be changed, since a calculation engine project is linked to an organism-specific allele database. The number of loci available in this allele database is also indicated in this tab.

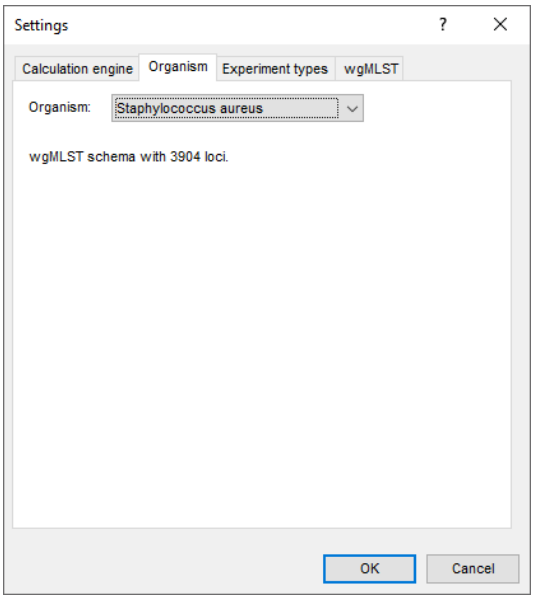

Figure 3.2: The *Organism* tab of the *Calculation engine settings* dialog box.

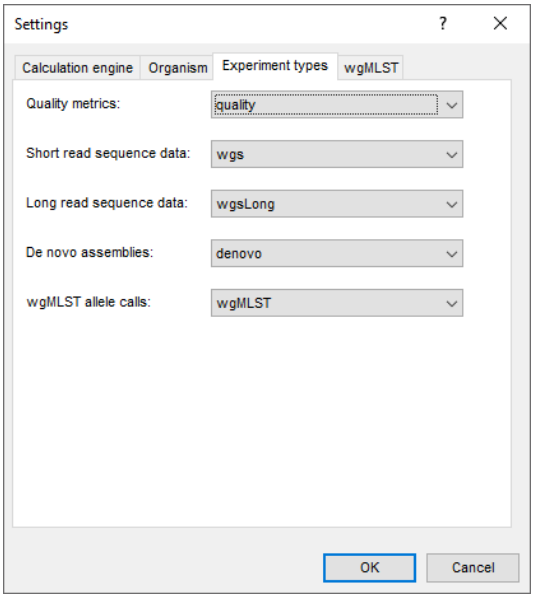

Figure 3.3: The *Experiment types* tab of the *Calculation engine settings* dialog box.

Five experiment types created during installation of the plugin (see 2) are automatically linked to the datasets used for wgMLST analysis (see Figure 3.3). Using the drop-down lists, other experiment types can be selected e.g. in case of pre-existing databases in which experiment types were named differently.

The ***Lab ID*** is used as identification tag when submitting new alleles to the centralized reference allele database. By default the name of the project is used, but this can be changed by the user.

New alleles are automatically submitted when the ***Submit new alleles automatically*** check box is checked. The automatic submission criteria are specified in the *Auto submission criteria* dialog box (see Figure 3.5).

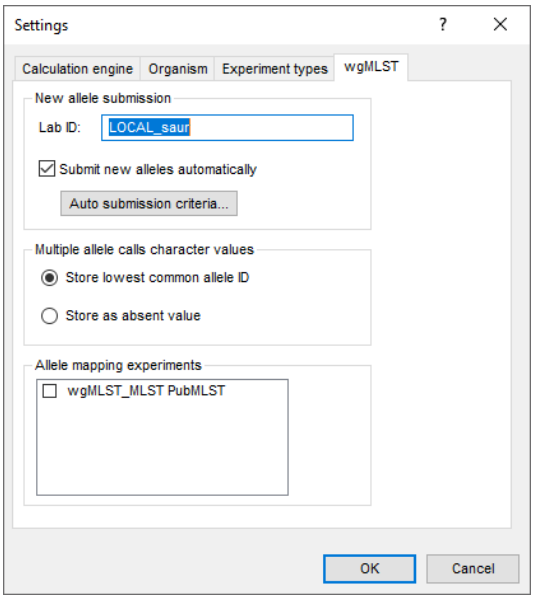

Figure 3.4: The *wgMLST* tab of the *Calculation engine settings* dialog box.

If ***Submit new alleles automatically*** is switched off, the assembly-based algorithm will only display a positive hit with alleles marked as 'Reference' or 'Accepted' in the allele database. Matches with 'Tentative' alleles are only made upon submission of the alleles to the allele database (see 13).

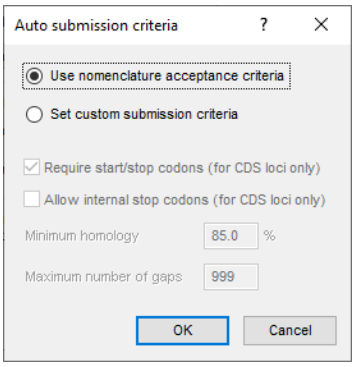

Figure 3.5: The *Auto submission criteria* dialog box.

By default, the ***Use nomenclature acceptance criteria*** option will be checked, meaning that the automatic submission settings are defined by the curator of the allele database. However, automatic submission settings can be changed by the user when ***Set custom submission criteria*** is checked. In the latter case, following parameters can be specified:

- ***Require start/stop codons (for CDS loci only)***: only submit sequences that correspond to a protein coding region (CDS), i.e. the sequence starts with a start codon and ends with a stop codon. This does not apply for non-CDS loci such as "classical" MLST loci.
- ***Allow internal stop codons (for CDS loci only)***: sequences are submitted even if they contain internal (i.e. premature) stop codons.
- The ***Minimum homology*** towards one of the reference allele sequences within the same locus.

- The ***Maximum number of gaps*** in the pairwise sequence alignment towards the closest allele sequence assigned the same locus.

In case multiple allele calls are made and different calls obtained for the same locus, two options are available as to what information is stored in the final allelic profile (i.e. the **wgMLST** character type experiment):

- ***Lowest common allele ID***: among the allele calls that the assembly-based and the assembly-free method have in common for a given locus, the one with the lowest allele ID is retained. This is the default option.
- ***Store as absent value***: no consensus call is retained in the allelic profile for these loci.

The ***Allele mapping experiments*** list shows the allele mappings that were set up by the wgMLST allele database curators. Allele mappings are used to synchronize against other wgMLST services (see 18). Via the check boxes, allele mapping experiments can be activated or inactivated. When an allele mapping experiment is activated, a character experiment type with the same name will be created and used to store the external allelic profiles in.

Chapter 4

An introduction to wgMLST

4.1 What is wgMLST?

Whole genome Multi Locus Sequence Typing (wgMLST) uses whole genome sequencing data to perform multi-locus sequence typing on a genome-wide scale. For each sample, locus presence is analyzed, and if present, the allele variant is determined. If the sequence is different from the known alleles for that locus, it is considered to be a new allele and is assigned a unique allele number. Starting from the complete wgMLST scheme, different subschemes can be defined as a fixed set of loci leading to typing schemes on different levels of resolution or function, e.g. MLST, extended MLST, ribosomal MLST, virome, resistome and much more.

Using the wgMLST method, one looks at the total sequence similarity of coding regions between strains. wgMLST is based on the concept of allelic variation, meaning that recombinations and deletions or insertions of multiple positions are counted as single evolutionary events. This approach might be biologically more relevant than approaches that consider only point mutations. Moreover, the wgMLST analysis strategy naturally incorporates not only the core genome but also the accessory genome, and therefore supersedes the single reference issue when performing reference-based SNP detection.

4.2 wgMLST in BIONUMERICS

Within the BIONUMERICS software, two procedures are in place for allele identification (see Figure 4.1). The **assembly-based method** identifies the alleles from de novo assembled genomes using BLAST. This is a computationally intensive method if your de novo assembly was not calculated already outside of BIONUMERICS, but is required for extrinsic validation of the allele calls. The de novo approach implies that some loci can be missed due to the multiple contigs from the assembly. In addition, de novo assembly has undefined behavior for the reconstruction of multi-copy loci, and therefore multi-copy loci are not very well detected from de novo contigs. The **assembly-free method** is computationally less intensive, and is designed to be exhaustive. Missing loci are now missing from the reads rather than from the de novo assembly. Moreover, multi-copy loci are picked up as separate allele calls.

wgMLST processing is based on two separate entities. On the one hand, the BIONUMERICS client has full control over the *sample database*. All meta data remains local and within BIONUMERICS, storage of the data and wgMLST analysis is very user-friendly and results are easily accessible. The *WGS tools plugin* enables BIONUMERICS to link to batches of sequence read sets from public and private online repositories and from local file servers (see 7). From the BION-

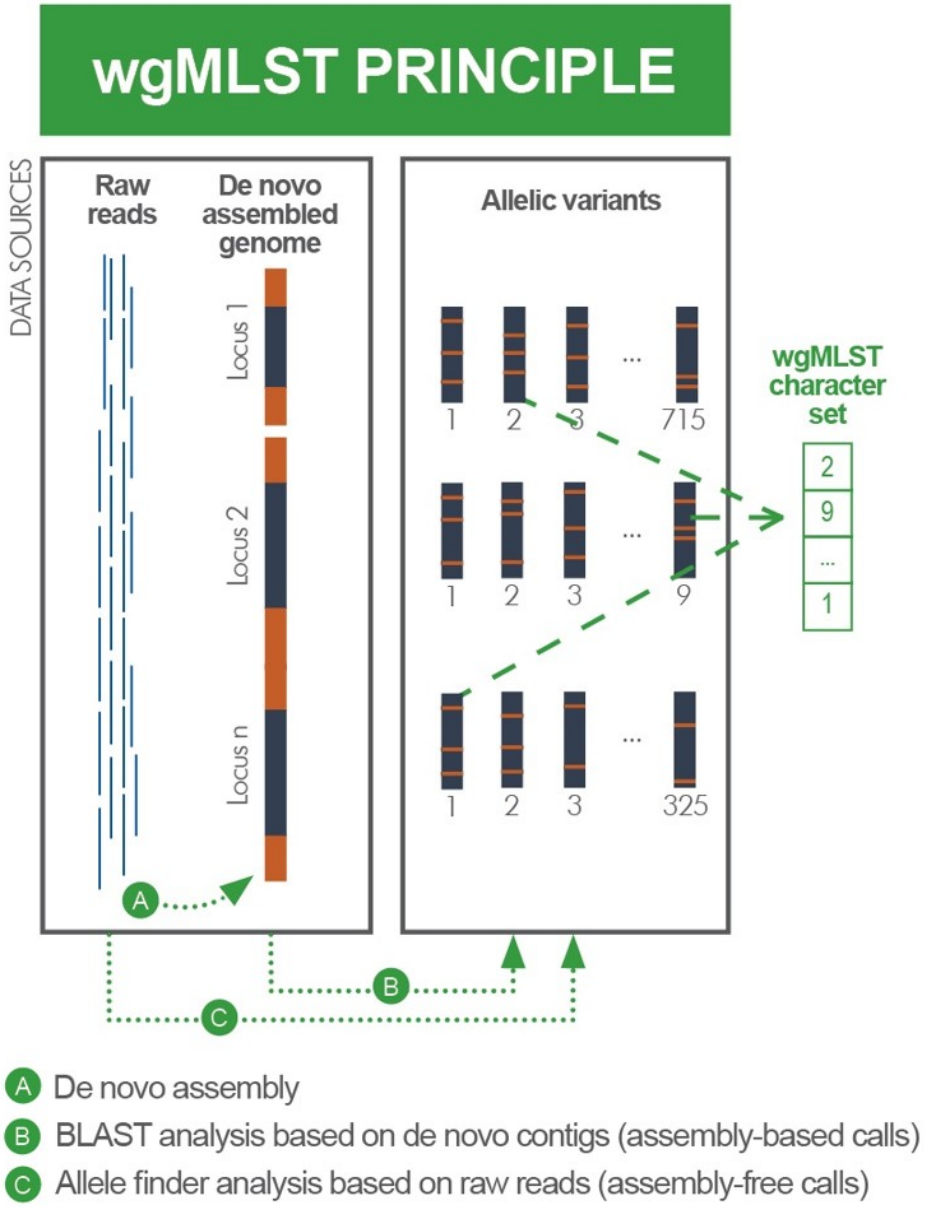

Figure 4.1: The wgMLST principle.

UMERICS client, jobs can be launched on a calculation engine (see below), and the results can be imported back into the BIONUMERICS database with a single click. The jobs currently offered include de novo assemblies, and assembly-based or assembly-free wgMLST allele detection. Results are stored in the database and are available for statistical and population analysis, clustering and calculation of minimum spanning trees, partitioning, and identification using BIONUMERICS' impressive set of analysis tools.

On the other hand, there is the curated nomenclature server which hosts the organism-specific locus and allele information (further referred to as the *allele database*). Alleles are semi-automatically named and validated through a range of criteria. In addition, automated synchronization procedures with public nomenclature servers and sample reporting tools are in place. In absence of suitable automated tools, it would be a daunting task to maintain a consistent allele assignment for thousands of loci. To accommodate for this, the *wgMLST curator plugin* provides automated curation tools needed to set up and maintain a wgMLST scheme and derived subschemes for any

organism of choice.

Demanding calculations such as de novo assemblies can be performed on an external *calculation engine*. The choice here is offered between pay-per-use cloud solutions or a local deployment e.g. on a computer cluster. The Applied Maths cloud-based calculation engine is designed to process hundreds of isolates within the hour, providing extremely fast turnaround times for the primary analysis. From within BIONUMERICS, jobs can be posted on the calculation engine and the results from such calculations retrieved. An extensive quality control environment allows you to look at and interact with the results from a genome-wide view up to base level. Only the wgMLST allelic profiles are stored in the BIONUMERICS database as character sets, resulting in a lightweight and responsive strain database. This also means that hardware requirements for the desktop or laptop computer running BIONUMERICS are kept modest.

4.3 wgMLST definitions

Some wgMLST definitions used throughout the software:

- *Imperfect match*: An allele that resembles closely to one of the approved alleles in the curator database but is not 100% identical to one of these alleles. The imperfect match results from the assembly-based algorithm that did not find an approved allele whose sequence is 100% identical to the query allele sequence.
- *New match*: This is an imperfect match eligible for submission or an allele hit that has already been submitted (and was an imperfect match in the past until it was submitted). Not all imperfect matches meet the criteria set for submission e.g. due to degenerate IUPAC code.
- *New call that can be submitted*: This is a new match that has not been submitted yet.
- *Known allele hit*: Allele hit for which an allele identification algorithm found a matching allele in the curator database (in case of the assembly-based algorithm, the sequence identity does not have to be 100%).
- *Unknown allele hit*: These kind of hits can be found by the assembly-free algorithm. It is used for cases where the algorithm is sure that the locus is present, but the algorithm was unable to find a 100% matching allele for this locus.
- *Summary calls*: After each round of allele identification, all available data from the two allele identification algorithms are combined and condensed into a single set of allele assignments. If only one of the two algorithms was run, this set contains all known and unknown allele hits as found by that algorithm. If both algorithms were run, the outcome for each locus depends on the data available for that locus. If only one of the two algorithms found one or more alleles for a locus, the allele hits of the one algorithm will be included in the summary calls. In case the assembly-free algorithm found a so far unknown allele and the assembly-based algorithm found at least one hit, only the known hits with a sequence identity of 100% are included. If both algorithms found hits for a locus (of type known), only those hits found by both with a sequence identity of 100% are included. If there is no overlap, the summary calls will have no results as the allele calls were discrepant for that locus.
- *SI (assembly-free)*: The fraction of the number of k-mers of the allele sequence found in the sample versus the number of k-mers present in the matching allele sequence in the allele database.
- *SI (assembly-based)*: The sequence identity between the allele sequence from the assembled genome and a matching allele sequence in the allele database, as determined by BLAST.

Chapter 5

The Calculation Engine

5.1 What is the Calculation Engine?

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 WGS data analysis. These include de novo ssequence assembly, genome annotation, reference mapping, SNP analysis, whole genome multi-locus sequence typing (wgMLST) allele calling and maximum likelihood (ML) clustering methods. ”Outsourcing” these demanding calculations means that your own desktop or laptop computer remains responsive and available for other tasks. Additionally, the Calculation Engine allows you to run popular and well-accepted open source tools (which are typically only available for Linux) on data stored in a BIONUMERICS database via a familiar user interface.

The Calculation Engine is installed on a powerful computer cluster, either on physical servers present on premises or in the cloud. A nomenclature service for wgMLST is integrated with the Calculation Engine.

The interaction between the Calculation Engine and the *WGS tools plugin* in BIONUMERICS occurs through a *job queue*. Via the *WGS tools plugin* jobs can be posted on the Calculation Engine and the results of these jobs retrieved when the calculations are finished. In between, job status and execution logs can be consulted from the job queue. Any part of the job queue that corresponds to the jobs of a particular user is visualized in the *Job overview* window (see 10).

Behind the scenes, jobs get distributed to calculation nodes. Different job types may require calculation nodes with different specifications. For example, de novo assembly jobs require nodes with more computer memory than other jobs do.

5.2 The Applied Maths Cloud Calculation Engine

The default Calculation Engine instance in the *WGS tools plugin* is the Applied Maths Cloud Calculation Engine, which is hosted on Amazon Web Services (AWS) servers in the USA and accessible worldwide. With up to a hundred calculation nodes, it is designed to process several hundreds of jobs within the hour, ensuring short turnaround times for the user.

Analyses on the Applied Maths Cloud Calculation Engine are charged at a pay-per-use system through Calculation Engine credits. One Calculation Engine credit costs one Euro. Table 5.1 provides an overview of all job types currently available on the Applied Maths Cloud Calculation Engine and the associated cost in credits.

Job name	Job type	Description	Job cost (credits)
Reference mapping	Entry	Reference mapping using SNAP [10] or Bowtie 2 [4]	1
De novo assembly	Entry	De novo genome assembly from short reads using SPAdes [1], SKESA [7], Unicycler [9] or Velvet [11]	1
De novo assembly	Entry	Hybrid de novo genome assembly from short and long reads using Unicycler [9]	10
wgMLST assembly-based calls	Entry	BLAST-based allele calling on assembled genomes for wgMLST analysis	3
wgMLST assembly-free calls	Entry	k-mer based allele calling on sequence read sets for wgMLST analysis	3
Annotation by Prokka	Entry	Genome annotation by Prokka [6]	1
CFSAN SNP pipeline	Comparison	SNP analysis on sequence read sets via the pipeline created by the FDA Center for Food Safety and Applied Nutrition (CFSAN) [2]	5
RAxML pipeline	Comparison	A RAxML maximum likelihood clustering [8] on aligned sequence data	5
FastTree pipeline	Comparison	A FastTree maximum likelihood clustering [5] on aligned sequence data	3
MTBC genotyping	Entry	Spoligo typing, resistance/lineage prediction and species prediction for *Mycobacterium tuberculosis* complex	2
SeqSero	Entry	*Salmonella* serotype prediction by SeqSero [12]	1

Table 5.1: An overview of available jobs on the Applied Maths Cloud Calculation Engine and the associated job costs.

5.3 Local calculation engine

The local calculation engine allows users to launch jobs on their own computer, much in the same way as launching jobs on the calculation engine (see 5.1). The same principle of a *job queue* is used. Jobs from the job queue run as separate processes and hence do not block the use of BIONUMERICS while they are being executed.

Since all calculations for the local calculation engine are performed on the same computer on which BIONUMERICS is installed, i.e. a Windows desktop or laptop computer, not all calculation engine algorithms could be made available locally. While some algorithms are only available for Linux, others require more memory than an average laptop computer has to offer. For example, SKESA is the only de novo assembly algorithm offered on the local calculation engine because it is the only assembler with a sufficiently small memory footprint.

For reasons of performance, only two jobs are allowed to run simultaneously on the local calcu-

lation engine. The *Main* window of the BIONUMERICS database should be open for new jobs to start. Once a job is running, it will keep running even if BIONUMERICS is closed. However, your computer should remain switched on while jobs is running.

De novo assembly and reference mapping jobs on the local calculation engine require that the sequence read set data are made available on your computer. Except for e.g. fastq.gz files stored on a local or network drive, this implies download of the sequence read set data. Just as on the calculation engine, data download is possible from SRA, ENA, Amazon S3 buckets, BaseSpace and Aliyun OSS buckets (see 7). In practice, a download job is automatically launched with a de novo assembly or reference mapping job and the actual job is only started after the download job successfully completes. Downloaded sequence read set data will be kept for ten days, after which they will be automatically removed.

A notable difference with the Calculation Engine is that data download jobs on the local calculation engine will appear in the *Job overview* window. Download jobs can be removed once they are finished. No actual results will be imported in the database when results of a download job are retrieved.

Since sequence read sets will be downloaded to the Windows TEMP folder (`%USERPROFILE%\AppData\Local`), make sure that sufficient space is available on your C: drive when using the local calculation engine feature.

Assembly-based wgMLST allele calling jobs can be launched on the local calculation engine. This requires that the correct allele database is first downloaded from the calculation engine, since the latter acts as a nomenclature service for wgMLST. Allele databases can only be downloaded from the calculation engine (typically the Applied Maths Cloud Calculation Engine; see 5.2) if a valid project name and password is provided (see 2). To reliably download these relatively large files over an internet connection with limited bandwidth, the files are split up in segments and the integrity of each file segment is verified through its md5 checksum.

Table 5.2 shows a comparison of the Applied Maths Cloud Calculation Engine and the local calculation engine in use.

	Applied Maths Cloud Calculation Engine	Local calculation engine
Cost per sample	Yes (see 5.2 for details)	No (except electricity / internet use)
Performance	Powerful calculation nodes	Depending on computer specifications
Time to result	Fairly predictable	Variable, depending on job queue
Job concurrency	Scalable, max. 100	Max. 2
Algorithms available	All	Limited subset
Dependency with BIONUMERICS instance	Jobs start even if BIONUMERICS is closed	*Main* window should be open for new jobs to start
Dependency with client computer	Jobs keep running if computer is shut down	Computer should stay on for jobs to run

Table 5.2: A comparison of the Applied Maths Cloud Calculation Engine and the local calculation engine in use.

The *WGS tools plugin* requires a permanent connection to the calculation engine. This means that the local calculation engine cannot be used offline.

Chapter 6

Synchronization with the allele database

The organism-specific set of loci and typing schemes are defined in the curated wgMLST allele database. Upon installation of the *WGS tools plugin* (see 2), the client database is automatically synchronized with the allele database.

When the reference loci for the wgMLST analysis are updated or new subschemes are added by the curator, these changes should also be reflected in the sample database. For such situations, the curator can request a synchronization of all client databases with the allele database. The same action can be triggered manually via ***WGS tools*** $>$ ***Synchronize***.

A synchronization action will download the latest scheme definitions from the allele database and import them in the sample database as character views on the loci. It will automatically create an entry information field to store sequence types in for each subscheme that has sequence types (see 14). In case additional loci were added to the scheme by the curator, these new loci will be added to the **wgMLST** character experiment type. Moreover, new loci may be incorporated in one or more wgMLST subschemes, so the subschemes will be updated as well. After synchronization, some feedback on the number of new loci and subschemes that were added is displayed in a message box.

Chapter 7

Importing sequence read sets for the Calculation Engine

7.1 Importing sequence read sets as links

To initiate wgMLST analysis on your samples, the sequence reads should preferably be imported as data links. This import option is only available after installation of the *WGS tools plugin*.

Select ***File*** > ***Import...*** ( , **Ctrl+I**) to open the *Import* dialog box and to start the import of the sequence read sets. Under ***Sequence read sets data***, the option ***Import sequence read set data as links*** became available after installation of the *WGS tools plugin*. One should use this option to import the read files as data links to the sequence read set type **wgs**, the experiment type used to initiate the wgMLST analyses from (as defined in the ***Sequence read set data*** from the *Import* dialog box). This starts the *Import sequence read sets as links* wizard.

If a job is submitted to the calculation engine with a sequence read set imported *as file*, BIONUMERICS first exports the sequence read set from the database to a `*.fastq.gz` file and then sends the latter to the calculation engine. The relatively slow export step can be avoided with sequence reads sets imported as links.

7.2 Importing sequence read sets: Data source

Sequence read sets can be imported as links from multiple data sources, including online and offline data repositories such as:

- ***NCBI (SRA)***: Defines a link to data from the Sequence Read Archive (SRA) repository, based on the NCBI run accession number (see 7.3).
- ***EMBL-EBI (ENA)***: Defines a link to data from the ENA repository, based on the EMBL-EBI run accession number (see 7.3).
- ***Amazon (S3)***: Defines a link to data uploaded to a client-specific data bucket hosted at the Amazon S3 storage repository. For this import, specific Amazon S3 credentials including the bucket name, the access key ID and the secret access key of the Amazon S3 user need to be completed before access is granted (see 7.4).
- ***BaseSpace***: Defines a link to data uploaded to a data folder hosted on Illumina BaseSpace. For this import, specific BaseSpace credentials need to be filled before access is granted (see 7.5).

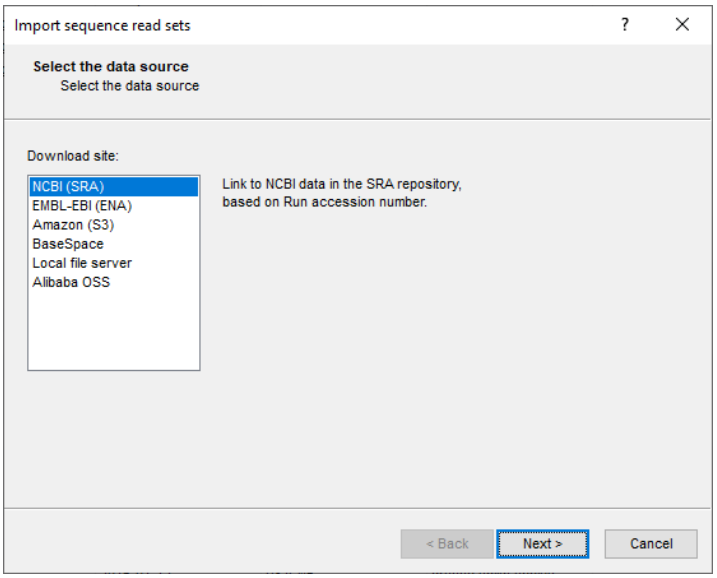

Figure 7.1: The first page of the *Import sequence read sets as links* wizard: the *Data source* wizard page.

- ***Local file server***: Defines a link to `*.fastq` or `*.fastq.gz` files on your computer or on a local data storage server (see 7.6).

- ***Alibaba OSS***: Defines a link to data uploaded to a client-specific data bucket hosted at the Alibaba Cloud Object Storage Service (OSS). For this import, specific Alibaba OSS credentials including the bucket name, the access key ID and the secret access key of the Alibaba OSS user need to be completed before access is granted (see 7.7).

Depending on the choice of import, different parameters may be queried in the *Data source* wizard page (see next paragraphs).

7.3 Importing sequence read sets from NCBI (SRA) or EMBL-EBI (ENA)

The only required information when importing data from NCBI or EMBL-EBI, are the run accession numbers for the read data.

When fetching multiple runs 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.

Continue the import by pressing <***Next***>. This opens the *Import template* wizard page (see 7.8).

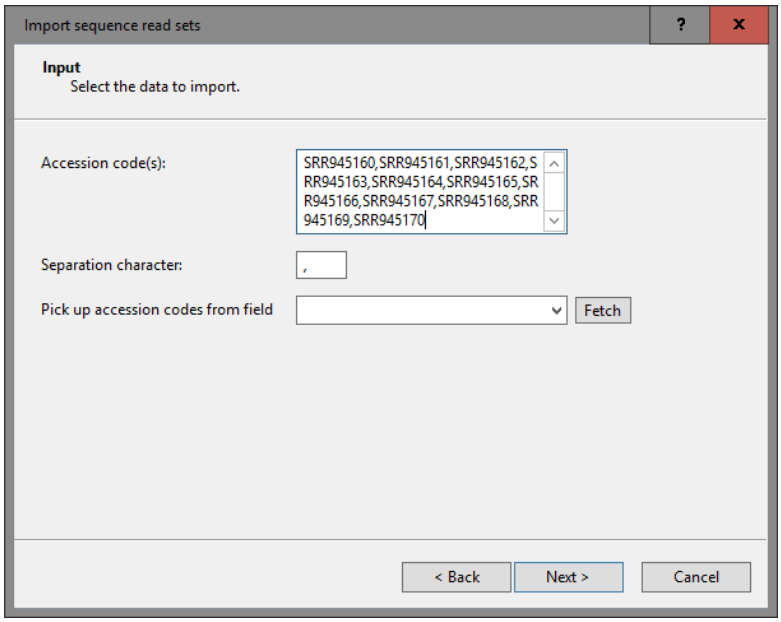

Figure 7.2: The *Input* wizard page: Import from NCBI.

7.4 Importing sequence read sets from Amazon (S3)

Upon the import of sequence read sets as data links from Amazon S3, the specific credentials are requested in the *Amazon S3 credentials* dialog box (see Figure 7.3).

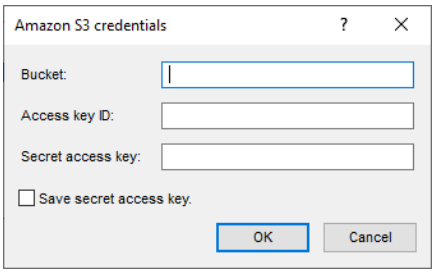

Figure 7.3: The *Import sequence read sets as links* wizard: *Amazon S3 credentials* dialog box.

After entering the ***Bucket name***, the ***Access key ID*** and the ***Secret access key***, one can proceed to the *Input* wizard page where the read files for import can be selected. Navigate through the bucket structure by selecting the + and - signs and check the folders and/or files that need to be imported as data links. Press <***Next***> to continue the import. This opens the *Import rules* dialog box (see 7.8).

7.5 Importing sequence read sets from BaseSpace

For the import of sequence read sets as data links from BaseSpace, browse access is requested upon import (see Figure 7.4).

After confirmation, a browser window opens which links to the Illumina Account Login page. Once logged in to your Illumina Account, browse access is requested for the wgMLST application (see

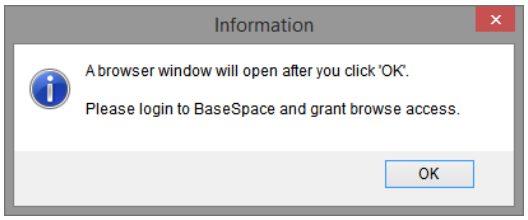

Figure 7.4: Information dialog to grant browse access to BaseSpace.

Figure 7.5).

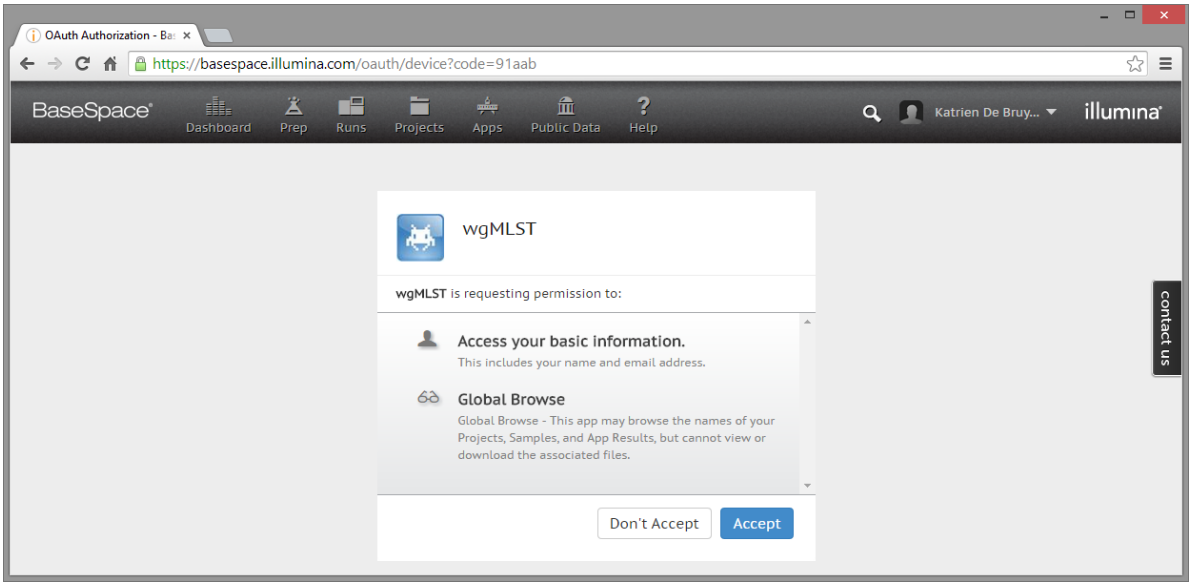

Figure 7.5: wgMLST access request on BaseSpace.

After acceptance, the browser window can be closed and project and sample information from the BaseSpace account is updated in the *Input* dialog page of the *Import sequence read sets as links* wizard (see Figure 7.6). After selecting the project and sample information that will be imported, the import template for both the project and sample name can be defined in the next dialog page.

The workflow for creating the import template is described under 7.8. It is very similar to the sequence read set import from NCBI SRA or EMBL ENA except that the accession code information is replaced by project and sample name as defined in the BaseSpace account.

In contrast to the other import methods, BaseSpace read access will additionally be questioned at the moment any of the wgMLST analyses are launched. After confirmation, a browser window will open where you can log in to your personal BaseSpace account and accept the permission for the wgMLST application to use a specific dataset (see Figure 7.7).

7.6 Importing sequence read sets from a local file server

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 their

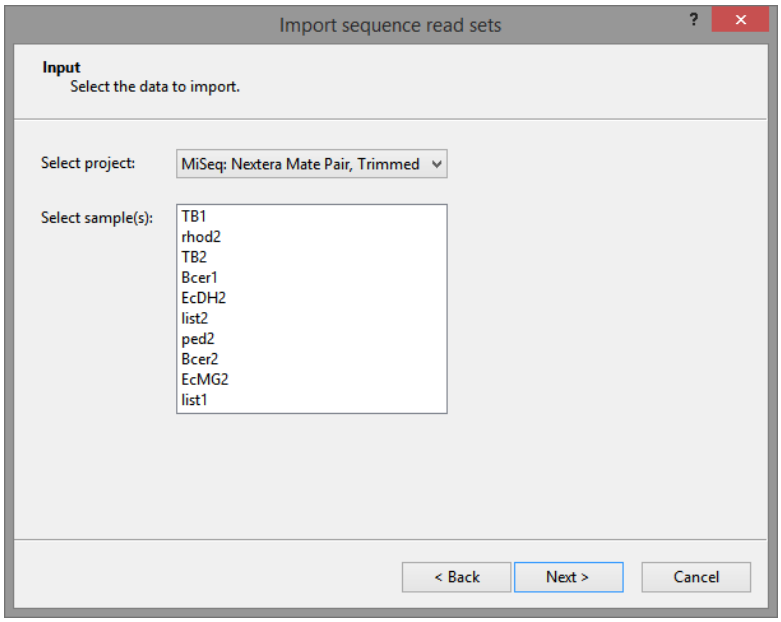

Figure 7.6: The *Input* dialog page of the *Import sequence read sets as links* wizard for BaseSpace.

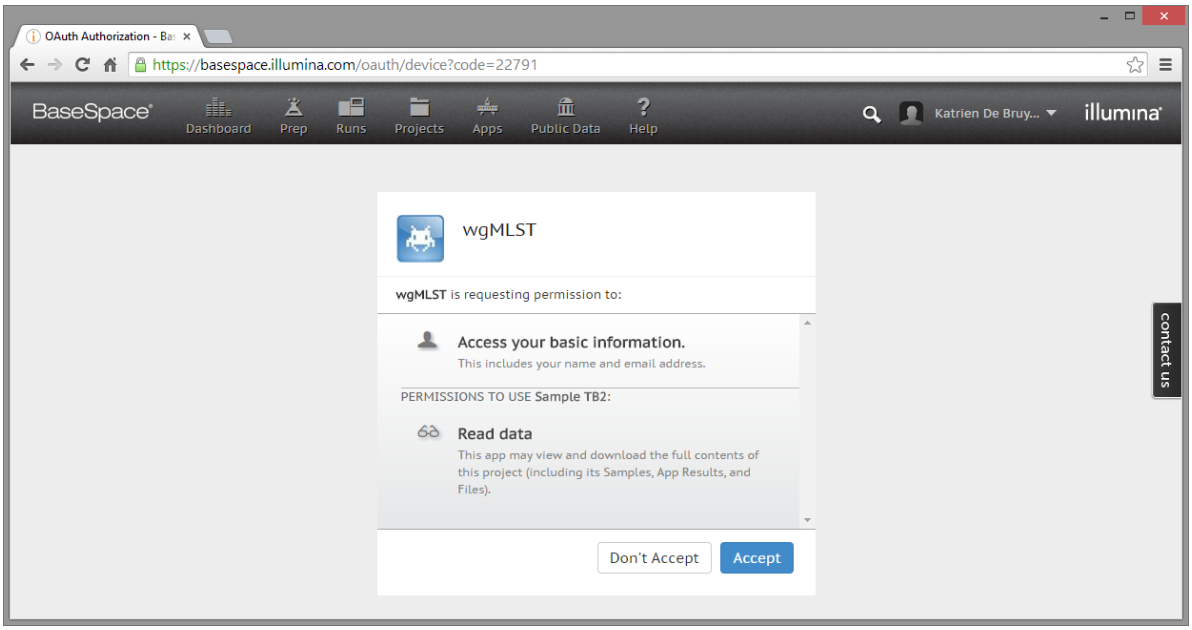

Figure 7.7: wgMLST read data request on BaseSpace.

total file size.

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 ***Auto-detect paired-end files*** 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 paired-end specific characters. 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. In the next step, the import template can be defined (see 7.8).

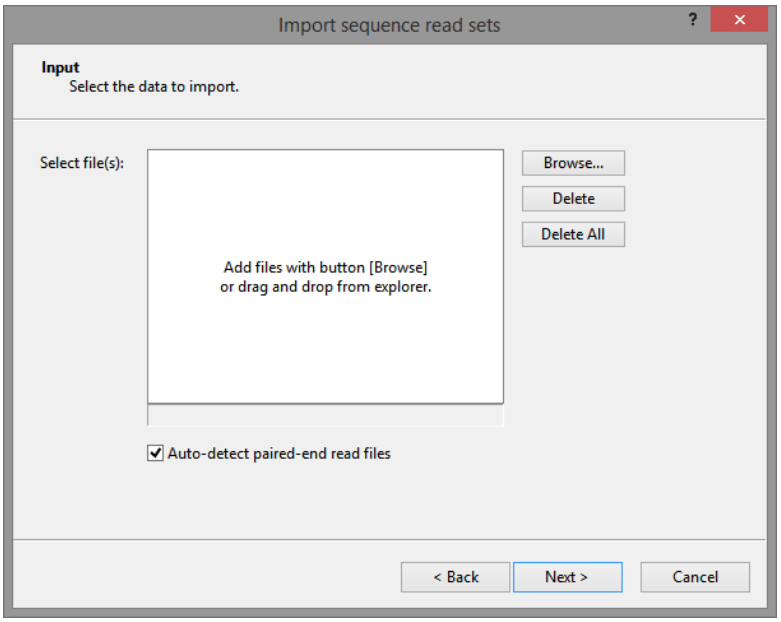

Figure 7.8: The *Input* wizard page: Select files from local file server.

7.7 Importing sequence read sets from Alibaba OSS

Upon the import of sequence read sets as data links from Alibaba OSS, credentials are requested in the *Alibaba OSS credentials* dialog box (see Figure 7.9).

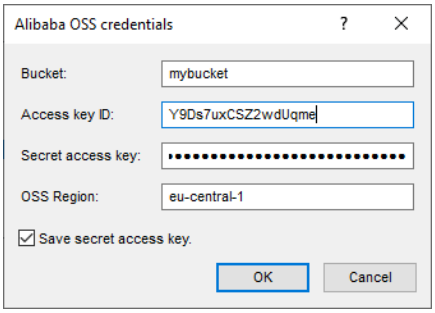

Figure 7.9: The *Import sequence read sets as links* wizard: *Alibaba OSS credentials* dialog box.

Credentials should be entered as ***Bucket*** name, ***Access key ID*** and ***Secret access key***. In addition, the ***OSS region*** where the bucket is located need to be provided. Enter the OSS Region ID without `oss-` prefix. For example, use “eu-central-1” for Germany (Frankfurt) or “cn-beijing” for China (Beijing).

Check the option ***Save secret access key*** to avoid having to enter the credentials for each import action or job.

Pres <***OK***> to proceed to the *Input* wizard page where the read files for import can be selected. Navigate through the bucket structure by selecting the + and - signs and check the folders and/or files that need to be imported as data links. Press <***Next***> to continue the import. This opens the *Import rules* dialog box (see 7.8).

7.8 Importing sequence read sets as links: import template

To specify which sample data and meta data 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.

In this part, we will only briefly discuss the creation of the import template, typically used for sequence read sets which are imported as links. Typically, all rows in the grid can be associated with a new or existing entry information field. Initially the rows are not linked to any information in the database, i.e. the ***Destination type*** and ***Destination*** for all rows is set to <***None***> (see Figure 7.10).

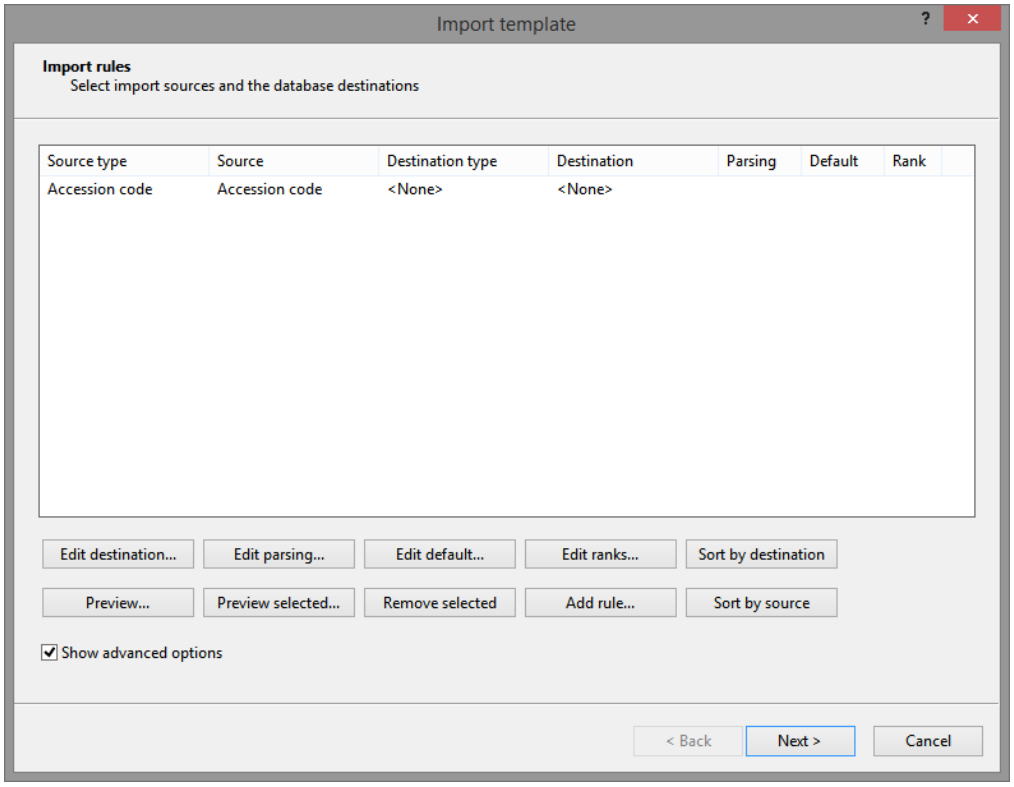

Figure 7.10: The *Import sequence read sets as links* wizard: *Import rules* dialog box.

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 displays a new dialog box prompting for the new destination for the selected row(s) (see Figure 7.11).

The information of the selected rows can be linked to:

- A ***Sequence read set data type***.
- 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 after confirmation by pressing the <***OK***> button. This new dialog box prompts for the entry information field name.

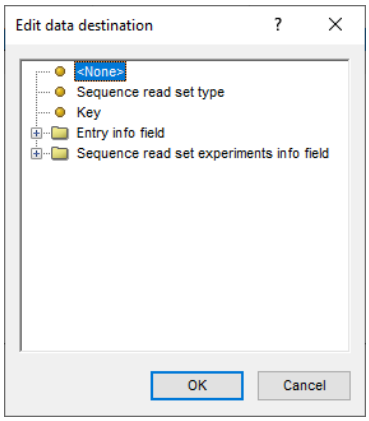

Figure 7.11: Import sequence read sets: Import template rules: Edit data destination.

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 and the link field is defined, the template can be saved and is displayed in the *Import template* wizard page (see Figure 7.12).

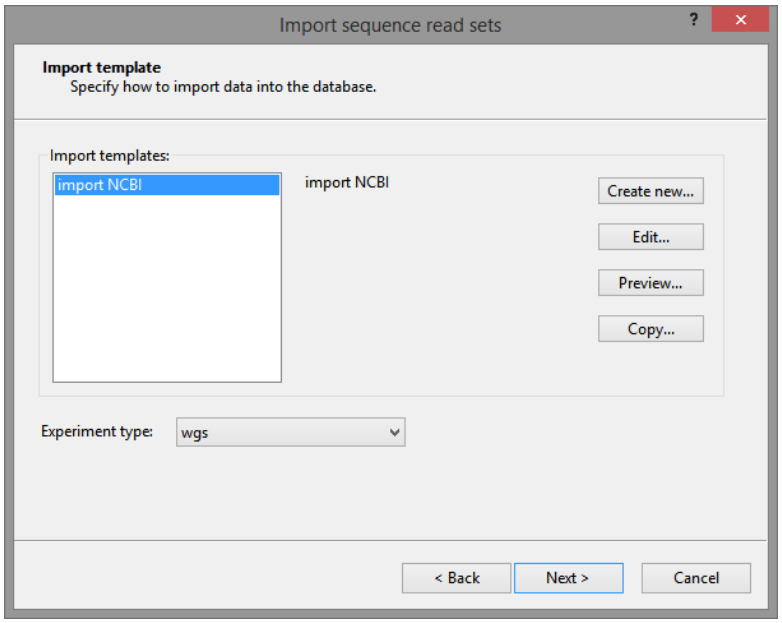

Figure 7.12: The *Import sequence read sets* wizard: *Import template* wizard page.

The experiment type where the data should be saved to also needs to be defined. All existing sequence read sets are displayed in the drop-down menu. Before continuing, make sure the experiment type **wgs** is selected for wgMLST applications (or other as defined in the ***Sequence read set data*** from the *Calculation engine settings* dialog box).

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 correspond-

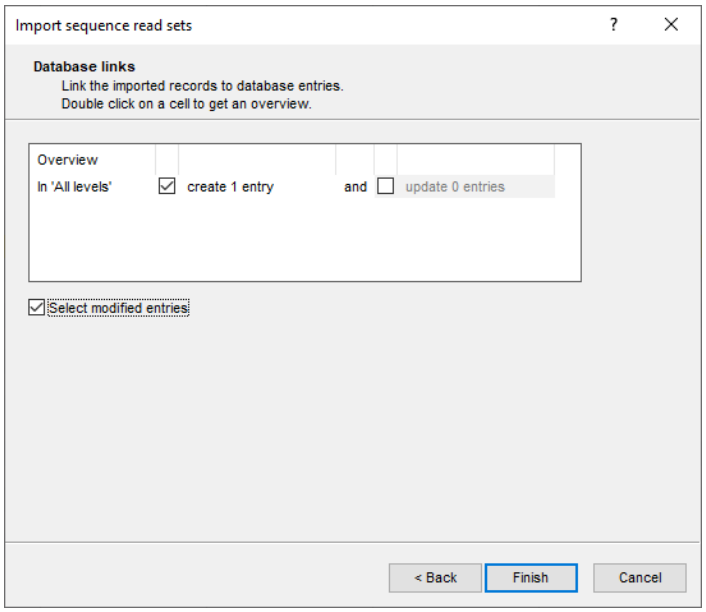

Figure 7.13: The *Import sequence read sets* wizard: *Database links* wizard page.

ing *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. Press <***Next***> to proceed to the *Processing* wizard page (see Figure 7.14).

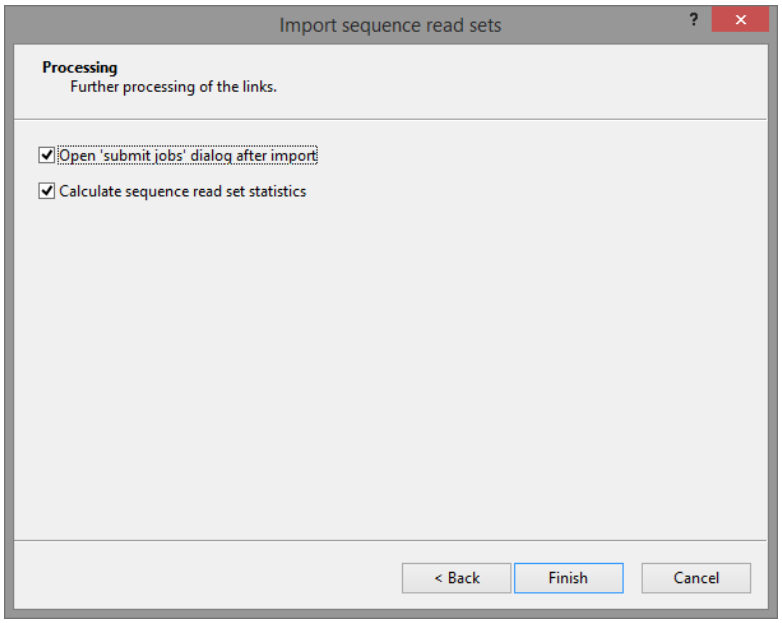

Figure 7.14: The *Import sequence read sets* wizard: *Processing* wizard page.

When the option ***Open 'submit jobs' dialog after import*** is checked, the *Submit jobs* dialog box will be opened after import.

The option ***Calculate sequence read set statistics*** only appears when ***Local file server*** was selected in the *Data source* wizard page (see 7.2) and allows to create sequence read set statistics during import, i.e. prior to running any jobs on the calculation engine.

Select <***Finish***> to start the actual import of the data into sequence read set experiments.

Chapter 8

Entry job management

8.1 Launching jobs

Launching entry jobs on the calculation engine for annotation, wgSNP and/or wgMLST is an easy process: In the *Main* window, select the entries that need to be analyzed and use ***WGS tools*** > ***Submit jobs...*** (▷). This action opens the *Submit jobs* dialog box (see Figure 8.1).

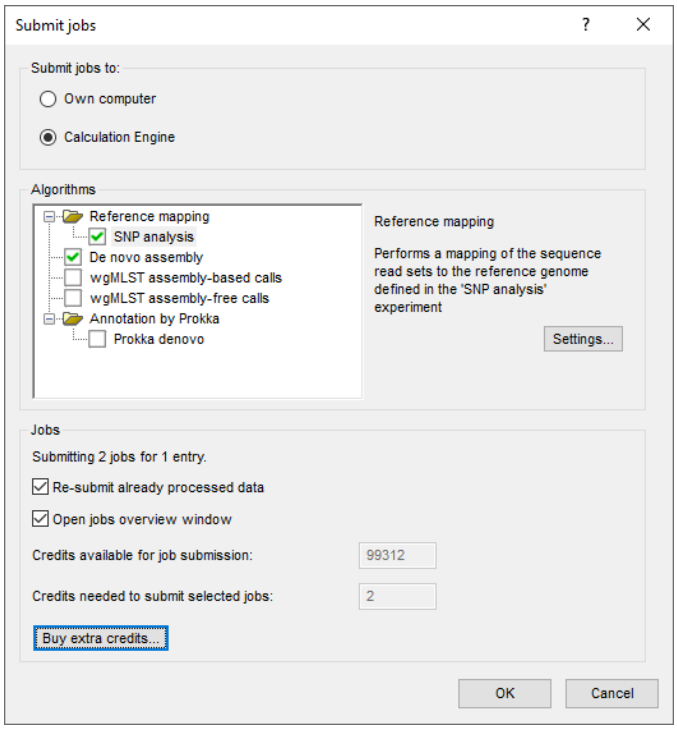

Figure 8.1: The *Submit jobs* dialog box.

From the *Submit jobs* dialog box, one can define which algorithms need to be run on the samples, and as such, define and launch the related jobs on the calculation engine.

The ***Submit jobs to*** option deals with the location where the jobs will be sent to for execution: either to your ***Own computer*** (i.e. the local calculation engine, see 5.3) or to the ***Calculation Engine*** to which the *WGS tools plugin* was connected (see 3), often the default Applied Maths cloud calculation engine (see 5.2). The option is important because credits are required for running jobs on the default Applied Maths cloud calculation engine (see 5.2), while the local calculation engine never requires credits. All entry jobs in a single submission should be sent either to the

calculation engine or to the local calculation engine, they cannot be mixed.

From the ***Algorithms*** part, select the analyses that need to be run on the selected entries.

- For use in **wgSNP** analysis, ***Reference mapping*** jobs can be launched on any reference-mapped sequence type (see 8.3).
- ***De novo assembly*** of sequence reads (see 8.4)
- ***wgMLST assembly-based calls*** to define the alleles in wgMLST based on a BLAST analysis on the de novo assembled contigs (see 8.5).
- ***wgMLST assembly-free calls*** to define the alleles in wgMLST directly from the reads (see 8.6),
- An ***Annotation by Prokka*** can be launched on any sequence type that is *not* reference-mapped (see 8.7).

In addition, raw data statistics (see 8.8) are automatically calculated with any job that acts on the sequence reads (i.e. with a ***Reference mapping***, ***De novo assembly*** or ***Assembly-free calls*** job).

Jobs that already have been submitted and have been imported successfully, will not be re-launched for analysis, unless the check box in front of ***Re-submit already processed data*** in the ***Jobs*** part is checked.

By default, the *Job overview* window will be opened after submission of the jobs. However, this can be changed by unchecking the option ***Open jobs overview window***.

The ***Credits needed to submit selected jobs*** are determined by the number of jobs and their respective credit costs. The ***Credits available for job submission*** are the number of credits currently available on the project. With the <***Buy extra credits...***> button, credits for the default Applied Maths cloud calculation engine can be purchased online. Your software serial number and wgMLST project name will be filled in automatically.

When the <***OK***> button is pressed in the *Submit jobs* dialog box, the jobs are launched to either the local calculation engine or the calculation engine.

With jobs submitted to the calculation engine and in case one or more of the sequence read sets are stored as **links to a local file server** (and hence not available to the calculation engine), the message ”Some local SRS links need to be exported to the CE Store. An external upload application will start. Please do not close this application until all files have been uploaded.” After confirmation, the CE Store Uploader will start (see 8.2).

In case one or more of the sequence read sets are stored as **files**, the message will additionally read ”Some local SRS data needs to be exported to fastq.gz files first. This may take a while and will block you from working with BN in the meantime.”. In this scenario, BIONUMERICS will first export the sequence read set files from the database to `*.fastq.gz` files in a temporary location. This export is a relatively slow process, during which BIONUMERICS will be unavailable for other commands. When the export of `*.fastq.gz` files is complete, the CE Store Uploader (see 8.2) will start and upload the `*.fastq.gz` files to the CE Store, in the same way as for local links.

8.2 The CE Store uploader

The **CE Store Uploader** is a separate executable included in the BIONUMERICS installation. This tool uploads `*.fastq.gz` files for sequence read sets stored as local links to an Amazon S3

temporary storage (called **CE Store**), which the calculation engine can access. The CE Store is managed by the Data Manager service on the calculation engine, ensuring that:

- Files on the CE Store can only be used in the context of the calculation engine project and the BIONUMERICS database from which they were uploaded.
- Only the calculation engine has read access to the files.
- Uploaded files are automatically removed after one week.

When submitting a job to the calculation engine for a sequence read set stored as local link, BIONUMERICS first checks if the `*.fastq.gz` files are already present on the CE Store. This will be the case when a job for the sequence read set was submitted earlier and the caching time has not exceeded yet. If the files are already available on the CE Store and the data link has not changed, a new upload is not needed and the files on the CE Store will be used. When one or more files actually should be uploaded to the CE Store, BIONUMERICS launches the CE Store Uploader (see Figure 8.2).

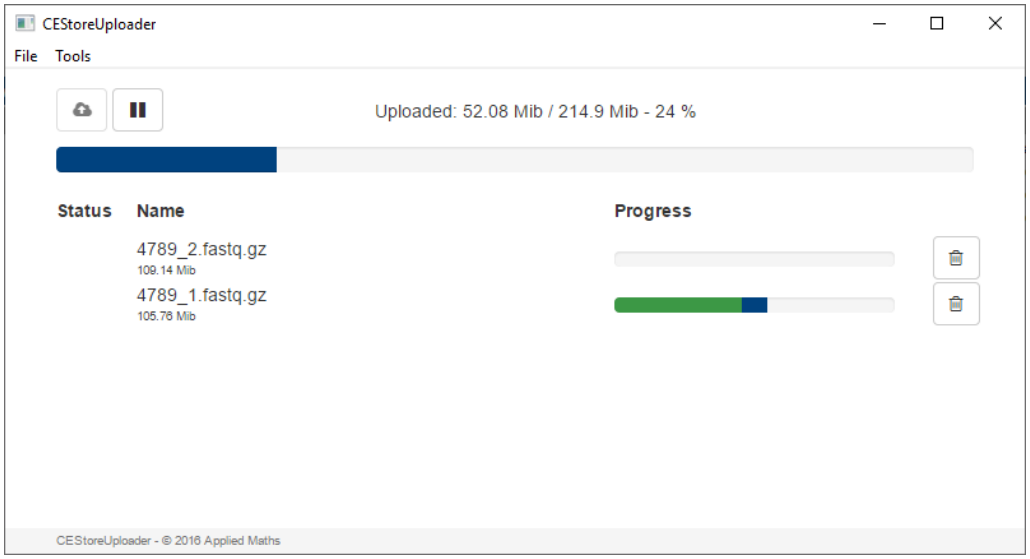

Figure 8.2: The CE Store Uploader.

For each file to be uploaded, a progress bar is displayed. Files will be split into multiple parts and uploaded in parallel. If the upload of a file part fails, the CE Store Uploader will try again (maximum three attempts). An orange segment in the progress bar means that the part is being uploaded. Green means that the upload is completed and red means that the file part could not be uploaded even after three attempts.

Pressing ⏸ will pause the uploads, pressing ☁ resumes the uploads again. If there are file parts for which the upload failed, another attempt to upload these file parts will be made when the ☁ button is pressed.

When attempting to close the CE Store Uploader (e.g. via ***File*** > ***Quit***) when uploads are still in progress, the confirmation message ”One or more files are still being uploaded. Quit anyway?” appears. If the CE Store Uploader is closed and started again, it will resume the uploads from where it left off.

⚠ An upload should never be stalled for an extended period of time because the corresponding job on the calculation engine will result in an error if the data are not uploaded within 24 hours after the job was launched.

File > ***Hide*** minimizes the CE Store Uploader to the Windows notification tray. With ***Tools*** > ***Show log***, a log file is displayed for the current session.

In case a file is overwritten under the same name (i.e. the file path has stayed the same, but the file content is different), the corresponding sequence read set should be re-imported as link in your BIONUMERICS database so that the CE Store Uploader recognizes that this file has changed. If not, the file will be served from the CE Store cache if it is available there.

8.3 Reference mapping

The ***Reference mapping*** option will launch a mapping of the sequence reads against a reference sequence using either SNAP [10] or Bowtie 2 [4] for each of the checked sequence types in the list below.

This list is limited to sequence types that are reference mapped. Hence, the template sequence to map against will be the reference sequence as specified in the sequence type settings (see the Sequence types, Chapter Setting up sequence type experiments).

The list below the ***Reference mapping*** option in the *Submit jobs* dialog box only shows experiment types from the currently active view in the *Experiment types* panel. To limit this list to the relevant ones (useful in case many reference mapped sequence types are defined), first select the sequence types for which to perform a mapping in the *Experiment types* panel and then switch to the <Selected Experiment types> view via the corresponding drop-down list, prior to calling the *Submit jobs* dialog box with ***WGS tools*** > ***Submit jobs...*** ( ).

The settings for this type of job can be defined by highlighting the job type and pressing <***Settings...***>. This action displays the *Reference mapping settings* dialog box (see Figure 8.3).

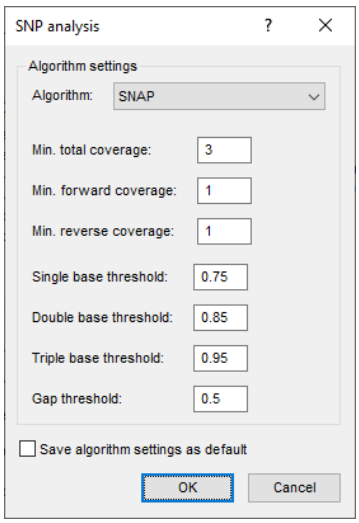

Figure 8.3: The *Reference mapping settings* dialog box.

Under ***Algorithm***, one of the two available reference mapping algorithms needs to be selected:

- ***SNAP***: the SNAP sequence alignment algorithm [10].
- ***Bowtie***: the Bowtie 2 gapped-read alignment algorithm [4].

On the local calculation engine, only SNAP is available.

Both algorithms have the same set of parameters:

- ***Min. total coverage***: Minimum total coverage of a position to be considered for consensus base calling. If the coverage is too low, the position will be called `N` in the consensus sequence.
- ***Min. forward coverage***: Minimum forward coverage of a position to be considered for consensus base calling. If the coverage is too low, the position will be called `N` in the consensus sequence.
- ***Min. reverse coverage***: Minimum reverse coverage of a base to be considered for consensus base calling. If the coverage is too low, the position will be called `N` in 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 altering these settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

The sequence created by the mapping algorithm will be imported in the BIONUMERICS database and saved in the corresponding reference mapped sequence type.

8.4 De novo assembly

The ***de novo assembly*** option launches the SPAdes [1], SKESA [7], Unicycler [9] or Velvet [11] algorithm to calculate the de novo sequence assembly and additionally, performs a correction on the base calling by mapping back the reads onto the generated contigs and performing a consensus calling on this mapping.

The settings for this algorithm can be defined highlighting the de novo assembly job and pressing <***Settings...***>. This action displays the *Perform de novo assembly* dialog box (see Figure 8.4).

In the *Perform de novo assembly* dialog box, the minimum coverage (***Min. coverage***) of the de novo contigs can be defined, the ***Expected coverage***, the minimum contig length to be retained (***Min. contig length***) and details on the assembler algorithm to use. A ***Min. coverage*** of "-1" implies that the minimum coverage is automatically defined from the de novo coverage.

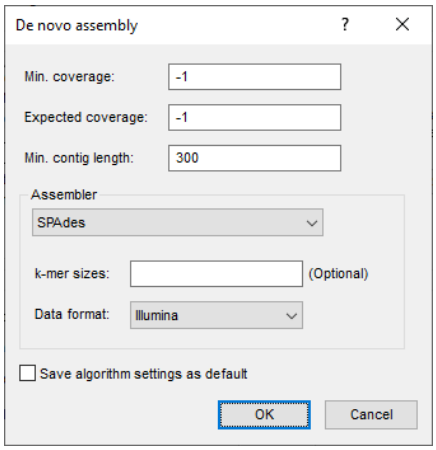

Figure 8.4: The *Perform de novo assembly* dialog box.

By default, the ***SPAdes*** genome assembly algorithm [1] is be used. Optionally, ***k-mer sizes*** and the ***Data format*** can be specified.

Six assembler options are available:

- ***Velvet*** [11] uses a fixed ***k-mer size***, which needs to be specified.
- ***Velvet optimizer (absolute k-mer sizes)*** is an implementation of Velvet Optimiser [3] that uses a range of absolute k-mer sizes, which needs to be specified as ***Absolute min. k-mer size*** and ***Absolute max. k-mer size***.
- ***Velvet optimizer (relative k-mer sizes)*** is an implementation of Velvet Optimiser [3] that uses a range of relative k-mer sizes, expressed as a fraction of the average read length (***Relative min. k-mer size*** and ***Relative max. k-mer size***).
- For the ***SPAdes*** assembler [1], ***k-mer sizes*** can optionally be provided. If left blank, SPAdes will determine the optimal k-mer size automatically. For the ***Data format***, a choice should be made between “Illumina” and “IonTorrent”.
- The ***SKESA*** assembler [7] does not have any additional parameters.
- The ***Unicycler*** assembler [9] can be run on short read set data like the other assemblers. However, if data is available in the long read experiment type (by default **wgsLong**), a ***Hybrid assembly*** can be performed. Hybrid genome assembly combines output of different sequencing technologies to obtain the best possible result. Unicycler creates a short-read assembly graph (typically using Illumina data) and then uses long reads data (PacBio or NanoPore) to build bridges, often resulting in a complete genome assembly. Note that a Unicycler hybrid assembly requires much more calculation time compared to a short read assembly and hence the job cost is also higher.

For performance reasons, only ***SKESA*** is available on the local calculation engine.

After the de novo assembly, read mapping is performed on the de novo contigs to correct for erroneous base calls. For the three Velvet flavors and for SPAdes, Bowtie 2 [4] is used as reference mapper. Final base calling for SKESA assemblies is done via SNAP [10] to ensure consistent results between assemblies on the calculation engine and local calculation engine. For Unicycler, final polishing is done via Pilon and/or Racon; no additional reference mapping is performed.

A downsampling is performed for all de novo assemblies when coverage is above 200. This coverage is assessed based on the genome size specified in the curator settings of the allele database. If the coverage is lower than or equal to 200, no downsampling is performed and the de novo assembly proceeds with the original data set. No downsampling is performed on long read data sets.

When altering these settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

The de novo contigs created by the assembly algorithm will be imported in the **denovo** sequence type in the BIONUMERICS database.

8.5 Assembly-based allele calling

The ***Assembly-based calls*** wgMLST algorithm launches the BLAST-based allele detection on the de novo assembled contigs. The algorithm will check which loci are present, and if present, the allele number for the loci in the contig sequences will be determined.

The settings for this algorithm can be defined by highlighting the assembly-based allele calling job and pressing <***Settings...***>. This action displays the *Perform BLAST on assemblies* dialog box (see Figure 8.5).

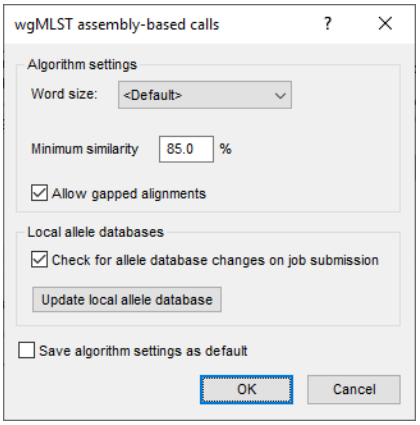

Figure 8.5: The *Perform BLAST on assemblies* dialog box.

In this dialog, the ***Word size*** for the BLAST search can be defined. The option <Default> hereby uses the default word size as specified in the allele database (i.e. by the allele database curator).

Furthermore, the ***Minimum similarity*** for an allele to be retained as a tentative match can be specified and whether or not to ***Allow gapped alignments***.

Assembly-based allele calling in wgMLST can be performed on the local calculation engine, in which case the BLAST search databases need to be downloaded first from the calculation engine to the local computer. A manual update can be performed by pressing <***Update local allele database***>. Depending on the size of the allele database and your connection speed, the initial download action might take up to several minutes. With the option ***Check for allele database changes on job submission***, one can ensure that the BLAST search databases are always up-to-date. The ***Local allele databases*** settings are not relevant for assembly-based allele calling jobs performed on the calculation engine and are therefore grayed out in this case.

When altering these settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

The results of this algorithm, i.e. the allele calls for the different loci, will be imported as **wgMLST** character information, and where applicable, combined with the results of the k-mer based allele detection (see 8.6 and 12.4).

8.6 Assembly-free allele calling

Starting from the sequence read set data, the ***Assembly-free calls*** wgMLST algorithm uses a k-mer based approach to check which loci are present from the organism-specific wgMLST scheme, and if present, identifies the allele number(s) of the present loci.

wgMLST assembly-free allele calls are not available on the local calculation engine.

The settings for this algorithm can be defined by by highlighting the assembly-free allele calling job and pressing <***Settings...***>. This action displays the *Find alleles* dialog box (see Figure 8.6).

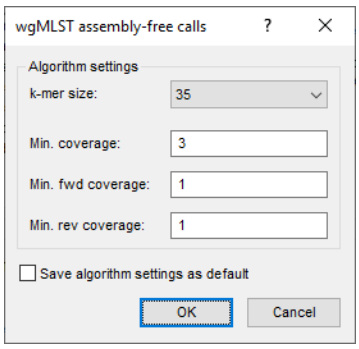

Figure 8.6: The *Find alleles* dialog box.

In the *Find alleles* dialog box, the ***k-mer size*** for the lookup table can be selected from the drop-down list and the minimum total coverage (***Min. coverage***), minimum forward coverage (***Min. fwd coverage***) and minimum reverse coverage (***Min. rev coverage***) for a locus to be called present can be defined. When altering these settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

The results of this algorithm, i.e. the allele calls for the different loci, will be imported as **wgMLST** character information, and where applicable, combined with the results of the BLAST-based allele detection (see 8.5 and 12.4).

8.7 Prokka annotation

The ***Annotation by Prokka*** option will launch a genome annotation by Prokka [6] for each of the checked sequence types in the list below.

Prokka annotations are not available on the local calculation engine.

To avoid issues with gaps in reference mapped (i.e. aligned) sequences, this list is limited to sequence types that are *not* reference mapped.

The settings for this type of job can be defined by highlighting the job type and pressing <***Settings...***>. This action displays the *Prokka settings* dialog box (see Figure 8.7).

Checking ***Force GenBank/ENA/DDJB compliance*** will make the annotations compliant with submission criteria from the GenBank, ENA and DDJB online repositories: add 'gene' features for

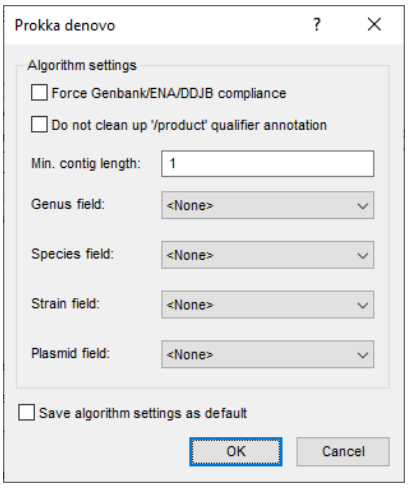

Figure 8.7: The *Prokka settings* dialog box.

each 'CDS' feature, a minimum contig length of 200 bp and adding a sequencing center ID (empty by default).

By default, Prokka tries to clean up the '/product' names to ensure they are compliant with GenBank/ENA conventions. Checking ***Do not clean up '/product' qualifier annotation*** will prevent this behavior.

The minimum size of a contig to be considered for annotation (***Min. contig length***) can be entered in bp.

In case the corresponding information is already present in the BIONUMERICS database, optionally a ***Genus field***, ***Species field***, ***Strain field*** and/or ***Plasmid field*** can be specified. Entry information contained in these fields will then be included in the annotation, which facilitates later submission to online repositories.

When altering these settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

The annotation will be imported in the BIONUMERICS database and saved with the corresponding sequence experiment. A newer Prokka annotation will replace any earlier Prokka annotation on the same sequence, but manually created features or annotation feature from other tools will not be overwritten: Prokka features will be added, even if they are defined on exactly the same positions.

8.8 Raw data statistics

The calculation of *Raw data statistics* is included in any job that works directly on the sequence reads. *Raw data statistics* calculates the number of reads available in the sequence read set, the sequence length statistics, the quality statistics and the base statistics that will be displayed in the *Sequence read set experiment* window.

Chapter 9

Comparison job management

9.1 Launching comparison jobs

In addition to entry jobs, calculation engine jobs can also be launched on *comparisons*. Following the same principles as cluster analyses in comparisons, comparison jobs apply on the whole comparison and the data stored in the *active* experiment and (where applicable) the active aspect (see the Basic cluster analysis, Chapter Comparisons in BIONUMERICS) in the *Experiments* panel (see Figure 9.1 for an example).

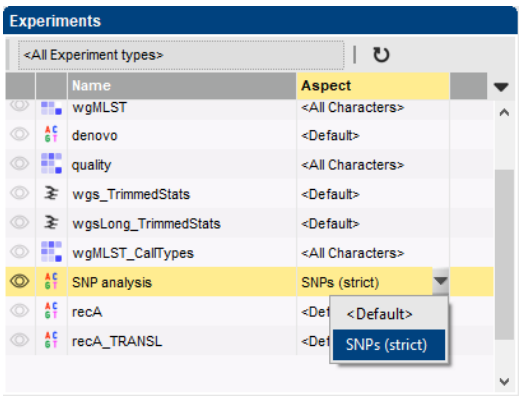

Figure 9.1: The *Experiments* panel in the *Comparison* window. The experiment type highlighted in yellow (**SNP analysis** is the active experiment, the **SNPs (strict)** is the active aspect.

To launch one or more comparison jobs, select ***File*** > ***Launch comparison jobs...*** ( ) in the *Comparison* window.

In case the comparison was not saved to the database yet, you will be prompted to save first (see the Basic cluster analysis, Chapter Comparisons in BIONUMERICS). Comparisons should be saved before any jobs can be launched.

Subsequently, the *Submit comparison jobs* dialog box will open (see Figure 9.2).

If the active experiment does not support any comparison jobs, an error message is shown and the *Submit comparison jobs* dialog box will not appear.

The ***Submit jobs to*** option deals with the location where the jobs will be sent to for execution: either to your ***Own computer*** (i.e. the local calculation engine, see 5.3) or to the ***Calculation Engine*** to which the *WGS tools plugin* was connected (see 3), often the default Applied Maths cloud calculation engine (see 5.2). The option is important because credits are required for running

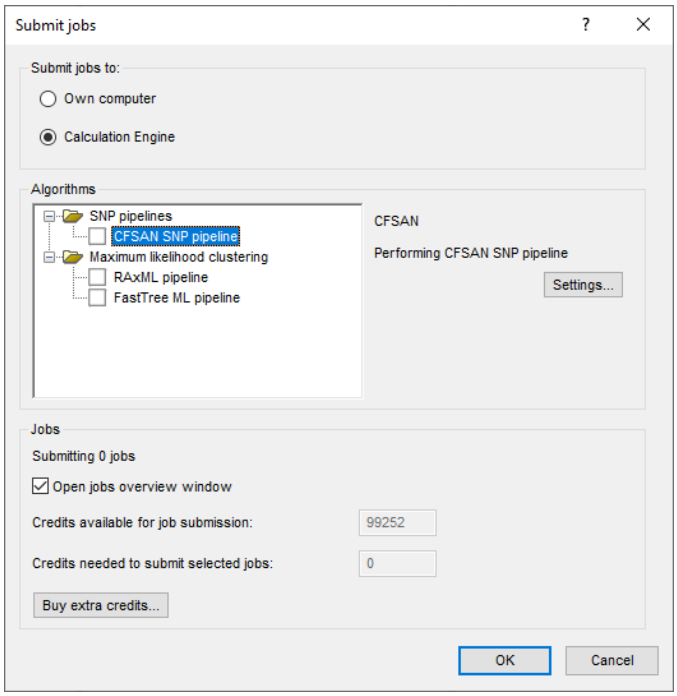

Figure 9.2: The *Submit comparison jobs* dialog box.

jobs on the default Applied Maths cloud calculation engine, while the local calculation engine never requires credits. All comparison jobs in a single submission (if more than one) should be sent either to the calculation engine or to the local calculation engine, they cannot be mixed.

From the ***Algorithms*** part, select the analyses to run on the active experiment in this comparison.

- ***CFSAN SNP pipeline*** jobs can be launched on any sequence read set type (see 9.2).
- ***Maximum likelihood clustering*** algorithms such as the ***RAxML pipeline*** (see 9.3) or ***FastTree ML pipeline*** (see 9.4) can only be applied on aligned sequences.

By default, the *Job overview* window will be opened after submission of the jobs. However, this can be changed by unchecking the option ***Open jobs overview window***.

The ***Credits needed to submit selected jobs*** are determined by the number of jobs and their respective credit costs. The ***Credits available for job submission*** are the number of credits currently available on the project. With the <***Buy extra credits...***> button, credits for the default Applied Maths cloud calculation engine can be purchased online. Your software serial number and wgMLST project name will be filled in automatically.

Comparison jobs can only be run on closed data matrices, i.e. the active experiment should contain data for all entries in the comparison.

9.2 CFSAN SNP pipeline

The ***CFSAN SNP pipeline*** option will launch a SNP pipeline created by the FDA Center for Food Safety and Applied Nutrition (CFSAN) [2] on sequence read set experiments.

The CFSAN SNP pipeline is not available on the local calculation engine.

The reference sequence(s) and other settings for this type of job can be defined by highlighting the job type and pressing <***Settings...***>. This action displays the *CFSAN SNP pipeline settings* dialog box (see Figure 9.3).

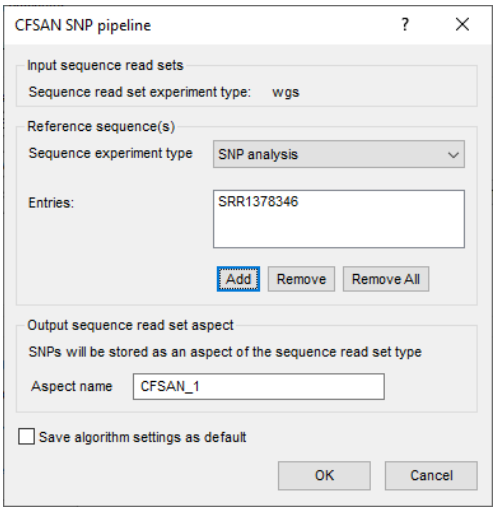

Figure 9.3: The *CFSAN SNP pipeline settings* dialog box.

The ***Sequence read set experiment type*** selected as ***Input sequence reads*** is read-only because it was set by choosing the active experiment prior to calling the *Submit comparison jobs* dialog box.

Under ***Reference sequence(s)***, the reference genome that should be used for the mapping needs to be specified. Select the ***Sequence experiment type*** that contains the assembled sequence of the reference genome from the corresponding drop-down list. Press the <***Add***> button and specify the entry that contains the reference genome. If needed, multiple reference genomes can be chosen by repeating this step.

The resulting SNP matrix (i.e. the output from the CFSAN SNP pipeline) will be stored as an aspect of the input sequence read set experiment type. An ***Aspect name*** can be entered manually or the default name accepted.

To avoid having to re-enter the above settings for a subsequent analysis, one can save them as defaults to the database with ***Save algorithm settings as default***.

9.3 RAxML ML clustering

The ***RAxML pipeline*** option will launch a RAxML maximum likelihood clustering [8] on aligned sequence data. This includes:

- Nucleic acid or amino acid sequences in the *Comparison* window on which a multiple alignment was calculated (see the Sequence types, Chapter Multiple alignment and cluster analysis of sequences).
- A SNP matrix generated by a wgSNP analysis (see the Sequence types, Chapter Whole genome single nucleotide polymorphism analysis), stored as an aspect of a reference mapped sequence type.
- A SNP matrix generated by the CFSAN SNP pipeline (see 9.2), stored as an aspect of a sequence read set.

The settings for this type of job can be defined by highlighting the job type and pressing <***Settings...***>. This action displays the *RAxML pipeline settings* dialog box (see Figure 9.4).

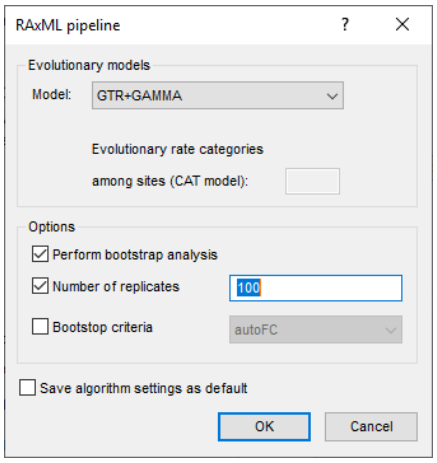

Figure 9.4: The *RAxML pipeline settings* dialog box for nucleic acid sequences.

In the ***Evolutionary models*** panel an evolutionary model for maximum likelihood analysis can be selected. The available models in the drop-down list next to the ***Model*** option consist of a combination of a basic evolutionary model (e.g. GTR, DAYHOFF,...) with a model for rate heterogeneity among sites (+GAMMA or +CAT) and an allowance for the presence of invariant sites (+I).

The GTR model is the most common substitution model for nucleotide sequence analysis and the only model available through RAxML. For amino acid sequence analysis twelve general amino acid substitution models can be selected, i.e. BLOSUM62, DAYHOFF, DCMUT, GTR, JTT, JTTDCMUT, LG, LG4M, LG4X, PMB, VT and WAG.

The GAMMA and CAT models account for variable rates of evolution across sites. The GAMMA model assumes a gamma distribution of rates across sites with four discrete rate categories. The CAT model optimizes the individual per-site substitution rates and classifies these individual rates into the number of rate categories specified by the ***Evolutionary rate categories among sites*** option. The default number of rate categories is set to 25. Note that this option is specific for the CAT model and is therefore not available when the GAMMA model has been chosen from the drop-down list.

The ***Options*** panel allows you to set the criteria for bootstrap analysis. When the ***Perform bootstrap analysis*** option is checked, bootstrap analysis is enabled and two options for bootstrap analysis become available:

- ***Number of replicates***: This option allows the user to set the number of bootstrap replicates. Default the number of bootstrap replicates is set to 100.
- ***Bootstop criteria***: When this option is checked the optimal number of bootstrap replicates to obtain stable support values will be determined automatically. In the drop-down list a RAxML bootstop criterion (i.e. threshold to decide when enough replicates have been computed) can be selected to determine convergence of the bootstrap estimates. The computed set of replicates is split into two equal sets and statistics are computed on the two sets (bipartitions):
  - ***autoFC*** (frequency-based criterion): The determination of bootstrap convergence relies on the observed frequencies of occurrences of distinct bipartitions.
  - ***autoMR*** (majority-rule consensus tree criterion): The determination of bootstrap convergence relies on building majority rule consensus trees.

- ***autoMRE*** (extended majority-rule consensus tree criterion): The determination of bootstrap convergence relies on building extended majority rule consensus trees.
- ***autoMRE_IGN***: The determination of bootstrap convergence is similar to autoMRE, but include bipartitions under the threshold whether they are compatible or not.

The bootstrap support values are drawn on the best-scoring maximum likelihood tree.

When altering the RAxML settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

When the job result is imported, the maximum likelihood clustering will be listed as an analysis in the *Analyses* panel. If the dendrogram is not automatically displayed in the *Dendrogram* panel, double-click on the analysis or use ***File*** > ***Analysis components*** > ***Open*** ( ) to display.

9.4 FastTree ML clustering

The ***FastTree pipeline*** option will launch a FastTree maximum likelihood clustering [5] on aligned sequence data (see 9.3 for a listing of sources of aligned sequences).

The settings for this type of job can be defined by highlighting the job type and pressing <***Settings...***>. This action displays the *FastTree ML pipeline settings* dialog box (see Figure 9.5).

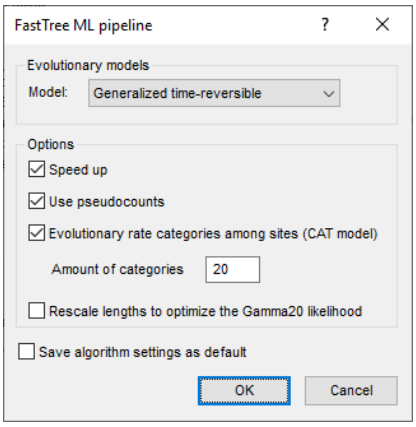

Figure 9.5: The *FastTree ML pipeline settings* dialog box for nucleic acid sequences.

In the ***Evolutionary models*** panel an evolutionary model for maximum likelihood analysis can be selected. Two models are available in the drop-down list for nucleotide sequence analysis (i.e. ***Generalized time-reversible*** and ***Jukes-Cantor***), while three models are available for amino acid analysis (i.e. ***Jones-Taylor-Thorton***, ***Le-Gascuel (2008)*** and ***Whelan-And-Goldman (2001)***).

In the ***Options*** panel four additional options for FastTree analysis can be enabled:

- ***Speed up***: This option can be checked to speed up the neighbor joining phase and to reduce memory usage.
- ***Use pseudocounts***: This option is recommended if the alignment has sequences with little or no overlap. A pseudocount (weight of 1.0) will be used to estimate the distances between these sequences.
- ***Evolutionary rate categories among sites (CAT model)***: If this option is checked the evolutionary model will be run under the CAT model for rate heterogeneity among sites. The

CAT model optimizes the individual per-site substitution rates and classifies these individual rates into the number of rate categories specified by ***Amount of categories***. The default amount of categories is set to 20.

- ***Rescale lengths to optimize the Gamma20 likelihood***: After the final round of optimizing branch lengths with the CAT model, report the likelihood under the discrete gamma model with the same number of categories. FastTree uses the same branch lengths but optimizes the gamma shape parameter and the scale of the lengths. The final tree will have rescaled lengths.

Local support values computed with the Shimodaira-Hasegawa test are drawn on the resulting maximum likelihood tree and provide an estimate of the reliability of each split in the tree. The FastTree support values range from 0 to 1 but are multiplied by 100 in BIONUMERICS.

When altering the FastTree settings, one can save the updated settings as defaults to the database with ***Save algorithm settings as default***.

When the job result is imported, the maximum likelihood clustering will be listed as an analysis in the *Analyses* panel. If the dendrogram is not automatically displayed in the *Dendrogram* panel, double-click on the analysis or use ***File*** $>$ ***Analysis components*** $>$ ***Open*** ( ) to display.

Chapter 10

Job overview window

In the *Job overview* window (see Figure 10.1), the job type, job name, time of submission, job status, a description of the job, its progress and much more can be monitored. The *Job overview* window can be opened from the *Main* window, in which case it lists all jobs for that BIONUMERICS database. When opened from a *Comparison* window, only the jobs for that comparison are shown.

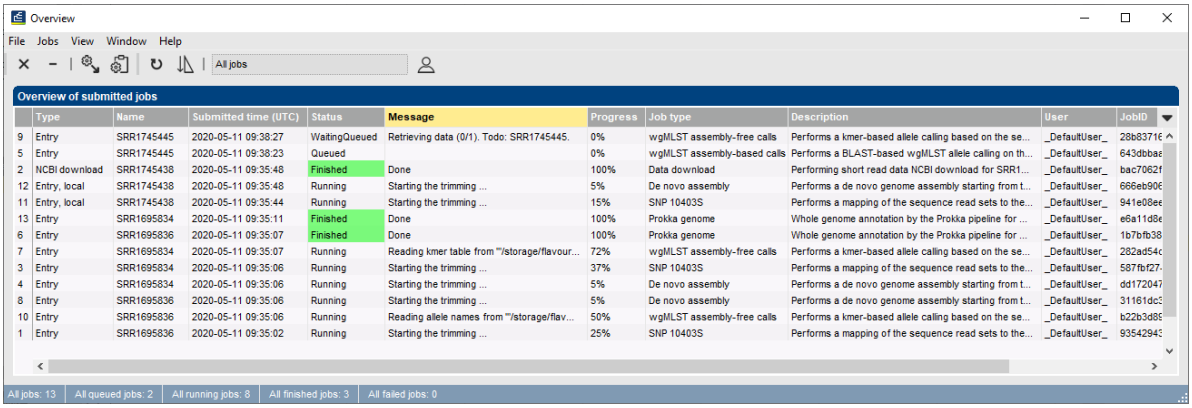

Figure 10.1: The *Job overview* window, called from the *Main* window.

In the 'Message' field, the run comments are displayed in real time which allows you to look into detail in the status of a specific job. One can have a look at the detailed log file for a selected job by selecting ***Jobs*** > ***Get logs...*** ( ) or double-clicking the job. This opens the log file for the job at hand. To refresh the overview, press ***View*** > ***Refresh*** (, **F5**). The *Job overview* window can be configured to update automatically, see below.

Jobs can be sorted based on the content of a selected column by ***View*** > ***Sort*** ( ). From the drop-down list in the toolbar, different job views can be used. By default, jobs for all users are displayed. By activating the drop-down list, one can filter the jobs and choose to display only the jobs that have been submitted or that are queued, running, finished or failed. By selecting ***View*** > ***My jobs*** ( ), the same job filters are applied, but this time only for the current user, i.e. ***My jobs*** are displayed instead of ***All jobs***.

A selected job can be canceled by selecting ***Jobs*** > ***Cancel*** ( ). This will only interrupt the calculation process, but the underlying data remains accessible on the calculation engine. Erroneous jobs and their related data can be deleted from the calculation engine by selecting ***Jobs*** > ***Cleanup*** ( ).

Once a job has been finished, the results can be imported in the database by selecting ***Jobs*** > ***Get results*** ( ) from the *Job overview* window. Multiple selected job results can be imported at once by selecting ***Jobs*** > ***Get results*** ( ).

Selected jobs can be resubmitted with ***Jobs*** > ***Resubmit***. A confirmation message will appear before the jobs are actually submitted again.

An automatic update can be defined from the *Settings* dialog box after selecting ***File*** > ***Settings*** (see Figure 10.2).

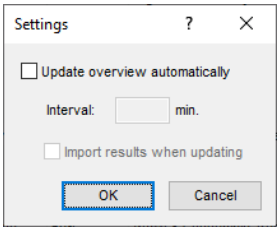

Figure 10.2: The *Settings* dialog box.

From this dialog, one can turn on the automatic update for the *Job overview* window and define the update interval (expressed in minutes). If the automatic update is enabled, there is the possibility to automatically import the results in the BIONUMERICS database upon completion of the jobs. Imported jobs are then removed from the job overview.

Results from comparison jobs (see 9) can not be automatically imported.

Although automatic retrieval of job results via the ***Import results when updating*** option is a useful feature if the *Job overview* window is left open e.g. overnight, it may interfere with user actions during an active session.

Close the *Job overview* window by selecting ***File*** > ***Exit*** (**Alt+F4**).

Chapter 11

Identification of allelic profiles

As already mentioned, job results can be imported from the *Job overview* window by selecting ***Jobs*** > ***Get results*** () or enabling the automatic update from the *Settings* dialog box.

As an alternative, the job results can also be imported starting from the entry selection in the *Main* window. Thereto, select ***WGS tools*** > ***Get results*** (). For the selected entries, all available job results will now be imported to the database and linked to their respective entry and experiment type. In addition, the log files from the calculation engine jobs are saved to the *Entry* window. All available log reports are displayed in the *Job log* panel. Once the results are imported, the corresponding jobs and their underlying data sets are automatically deleted from the calculation engine and as such, from the *Job overview* window.

For a selection of entries, all subscheme identification reports can be viewed by selecting ***WGS tools*** > ***View wgMLST reports...***. This opens the *Report* window (see Figure 11.1).

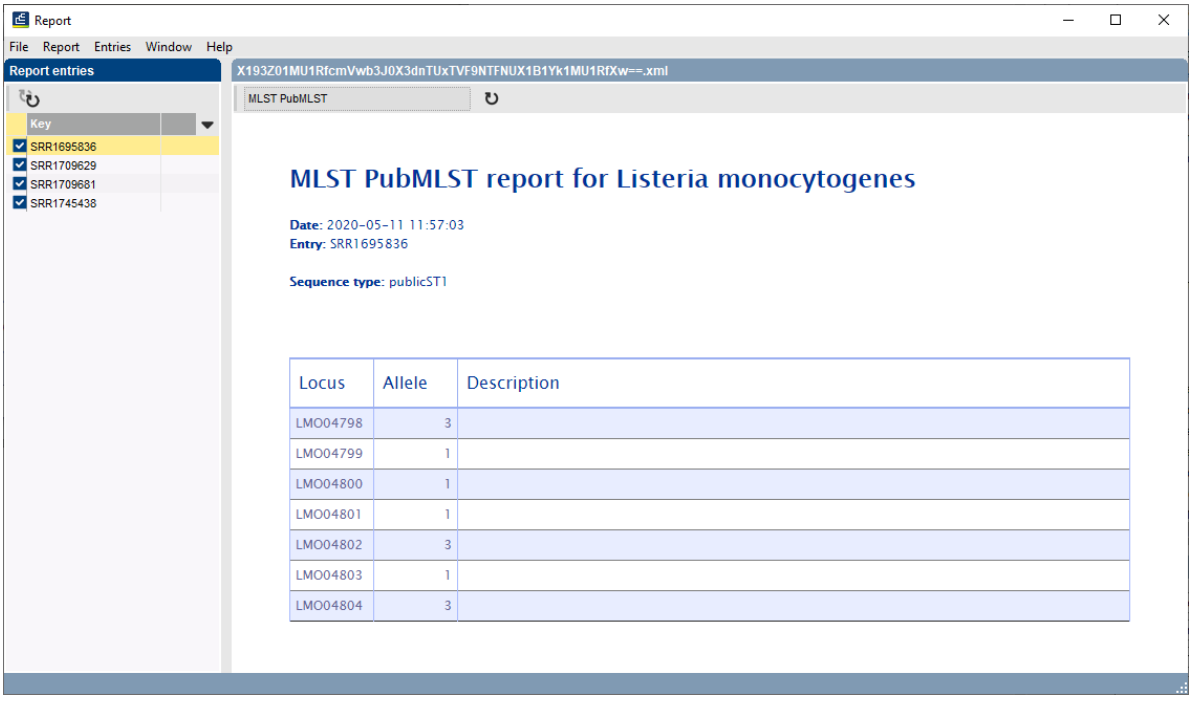

Figure 11.1: The *Report* window

At the left in the *Report* window, the different entries are listed and at the right, the detailed scheme report for the selected entry at hand is displayed. By default, the MLST subscheme is displayed but when different subschemes are defined in the curator database, one can navigate through the

subscheme reports by toggling between them. All reports can be updated one by one by selecting ***Report*** > ***Update current report*** (↻), or all at once by selecting ***Entries*** > ***Update all*** (↻).

When updating all reports, one gets the choice of updating only the selected subscheme or all subschemes defined in the curator database. Present reports can be updated by checking this option in the *Update reports* dialog box (see Figure 11.2).

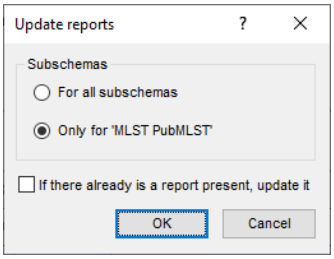

Figure 11.2: The *Update reports* dialog box.

From the *Entry* window, the wgMLST reports can be viewed by selecting the *Report* tab in the *Entry* window (see Figure 11.3). Also here, reports can be updated by selecting ***Report*** > ***Update current report*** (↻).

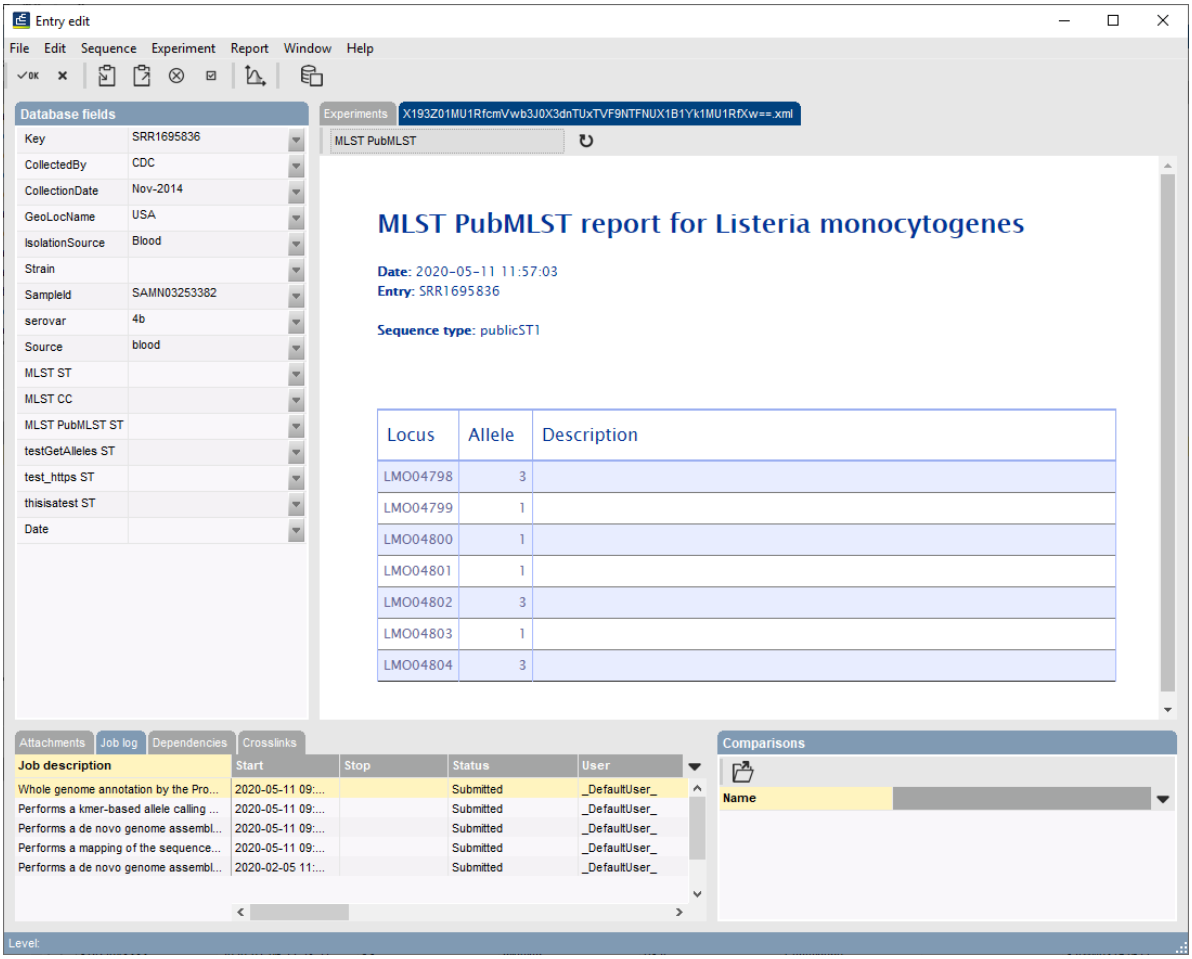

Figure 11.3: The wgMLST *Report* panel in the *Entry* window.

Chapter 12

Quality assessment of allelic profiles

12.1 Introduction

Detailed quality assessment of the allelic profiles and synchronization of the profiles with the allele database can be done from the *wgMLST quality assessment* window which can be opened from the *Main* window for the entry selection by selecting ***WGS tools*** > ***wgMLST quality assessment...*** (⚙) (see 12.2).

The quality parameters used in the *wgMLST quality assessment* window are stored in the character experiment type **quality** and can also be consulted in a quick and easy way in the *Comparison* window (see 12.3).

12.2 The wgMLST quality assessment window

12.2.1 Entries panel

The *wgMLST quality assessment* window can be opened from the *Main* window for the entry selection with ***WGS tools*** > ***wgMLST quality assessment...*** (⚙). The *Entries* panel contains the entry (i.e. sample or strain) information for which data is loaded and their quality parameters on each of the analyses is displayed (see Figure 12.1 for an example). When selecting a different entry, the circular graph and the allele information is updated with the pertaining information.

Entries

Entry	Raw data statistics	Raw data statistics after trimming	De novo assembly	Assembly-free calls	Assembly-based calls	Summary calls
SRR1566202	ok	ok	ok	ok	Not ok: Submitted alleles	ok
SRR1566203	Not ok: Average read quality	Not available	ok	ok	Not ok: Submitted alleles	ok
SRR1610008	Not ok: Average read quality	Not available	ok	ok	Not ok: Submitted alleles	Not ok: Confirme...
SRR1623013	Not ok: Average read quality	Not available	ok	ok	Not ok: Submitted alleles	ok
SRR1745438	ok	ok	ok	ok	Not ok: Multiple alleles	Not ok: Confirme...

Figure 12.1: The *Entries* panel.

The filter ***Entries*** > ***Show only entries with low-quality data*** (⚙) shows only those entries that have at least one quality score which is considered below acceptable, indicated in red. Threshold levels for which values are considered acceptable are managed by the curator of each wgMLST allele database individually. One can sort, based on a highlighted column in the *Entries* panel, by

selecting ***Entries*** > ***Sort entries*** ( ). Specific entry information can be queried after opening the *Entry* window by selecting ***Entries*** > ***Open highlighted entry...***.

The quality of the output from the calculations run on the calculation engine as well as the summary calls that are updated after each allele identification procedure are assessed for a number of criteria and is reflected by a single value (minimum of all scores for that algorithm or routine).

Detailed parameter values can be accessed from the *Quality control* dialog box that opens after double-clicking an entry (see Figure 12.2).

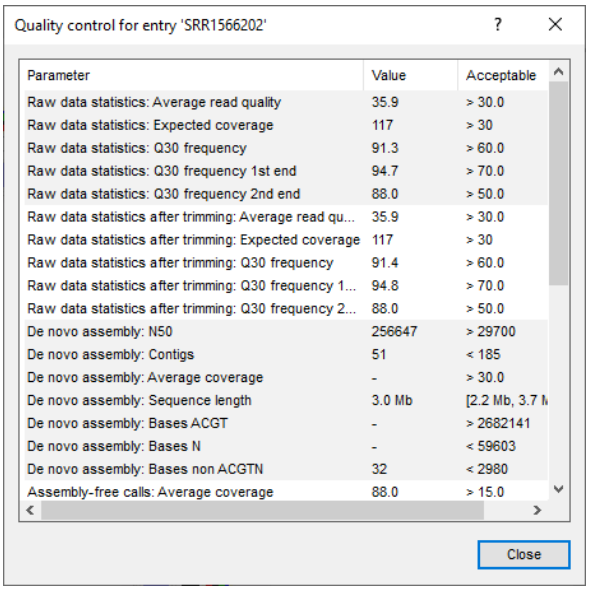

Figure 12.2: The *Quality control* dialog box.

For each criterion a reference value is set by the curator of the wgMLST allele database that serves as a

- ***Minimum threshold***: the quality values must be equal to or greater than the threshold to be considered as acceptable,
- ***Maximum threshold***: the quality values must be smaller than or equal to the threshold to be considered as acceptable,
- ***Centered value***: the quality values must lie in an interval around the centered value to be considered as acceptable. The range of the interval is determined by a tolerance factor.

These reference values are used for calculating a single quality score for each quality value.

Four different quality score calculations exist where a quality value must be (i) greater than a minimum threshold, (ii) greater than a minimum threshold but is bounded to a maximum, (iii) less than a maximum threshold, or (iv) close to a centered value.

For each algorithm, a number of criteria are evaluated. If all criteria are within acceptable bounds, 'OK' is printed. If this is not the case, the parameter which deviates most is the final value that is reported in the *Entries* panel of the *wgMLST quality assessment* window. The color is an indication of the magnitude of deviation.

A detailed explanation of each parameter can be found in 12.4.

12.2.2 Genome Viewer and Tracks panel

The *Genome* panel is a visualization tool for interactive exploration of the genome sequences and its features e.g. the wgMLST allele assignments, the de novo contigs, the GC content and the forward and reverse coverage. A zoom-able map is generated consisting of the different tracks over the genome.

The *Tracks* panel gives an overview of the information available that can be plotted on the circular graph. Depending on the track that is highlighted in the *Tracks* panel, the features (if any) of the selected track are displayed in the *Alleles* 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 12.3). The base numbers shown on top of the sequence correspond to the base numbering as used in the *Sequence editor* window.

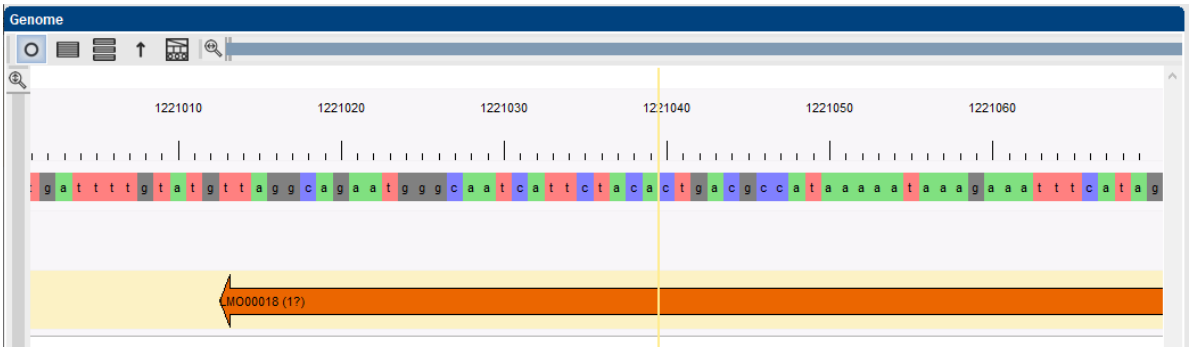

Figure 12.3: Zooming up to base level.

A zooming area can be specified with ***Graph*** > ***Set view range...***. This actions calls the *Set view range* dialog box (see Figure 12.4).

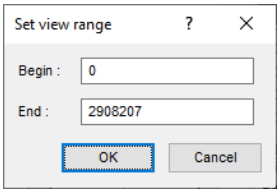

Figure 12.4: The *Set view range* dialog box.

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 12.3). The circular sequence can be rotated by holding down the left mouse button while dragging the mouse. With ↑ 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***.

The graphical representation of the sequence can be exported to the clipboard with ***File*** > ***Export...***. This calls the *Export image* dialog box.

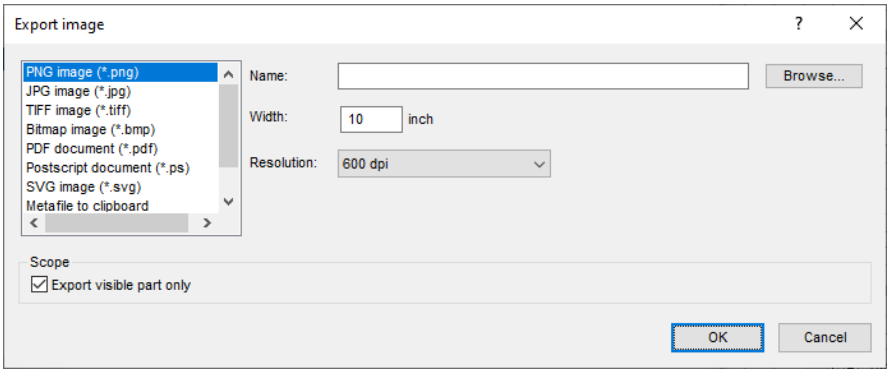

Figure 12.5: The *Export image* dialog box.

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.

The *Tracks* panel gives an overview of the information that can be displayed on the sequence in the *Genome* panel. The ***Sequence*** and ***Sequence Scale*** tracks are available for every sequence. The availability of the other tracks depends on the information present in the underlying database. From top to bottom, the default track order contains:

- The ***Sequence scale*** track contains the base pair indication of the sequence length (clockwise).
- The ***Sequence*** track contains the nucleotide calls for a specific nucleotide position. 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.
- The ***Summary calls*** track is listed if at least one locus is present in the summary loci obtained by combining the assembly-free and assembly-based calls. If both methods returned allele calls, the summary is defined as the alleles that are similar between both analyses. If for a specific loci, the allele call is only available from one algorithm, the allele call is also included in the summary. Selecting this track in the *Tracks* panel will display all alleles in the *Genome* panel and update the *Alleles* panel with their ***Locus*** name, ***Allele*** number, assembly-free and assembly-based sequence identity ***SI (assembly-free)*** and ***SI (assembly-based)***, their position on the sequence (***Start*** and ***Stop***), and the ***Contig*** information, if the allele was detected by the assembly-based method. Clicking on a locus in the *Alleles* panel will update the cursor selection on the map. The loci are plotted as colored arrows on the map, indicating the locus number and the allele sequence number between brackets. Sequence identity matches range over white (100% similarity), yellow, orange to red (lowest similarity).
- The ***Assembly-free calls*** track is listed if at least one locus is detected by the assembly-free algorithm. This trace contains the wgMLST allele calls obtained by k-mer analysis directly on the reads. Selecting this track will also update the *Genome* panel and the information in the *Alleles* panel. Clicking on a locus in the *Alleles* panel will update the cursor selection on the map, if the locus was also detected by the assembly-based approach. If not, no position information is present for that locus (as this cannot be derived from the assembly-free algorithm) and the locus is only present in the grid but omitted from the *Genome* panel.

- The ***Assembly-based calls*** track is listed if at least one locus is detected by the assembly-based algorithm. This trace contains the wgMLST allele calls obtained by BLAST on the de novo contigs against the reference alleles. Selecting this track will also update the *Genome* panel and the information in the *Alleles* panel. Clicking on a locus in the *Alleles* panel will update the cursor selection on the map, if the locus was also detected by the assembly-based approach (i.e. ***Start***, ***Stop*** and ***Contig*** information present for the locus). If not, the locus is omitted from the *Genome* panel.The loci are plotted as colored arrows on the map, indicating the locus number and the allele sequence number between brackets. Sequence identity matches range over white (100% similarity), yellow, orange to red (lowest similarity).
- The ***Contigs*** track shows the span i.e. the length of the contigs in alternating black and white.
- The ***GC content*** track contains the GC% over the genome (GC% calculated in a window of 10,000 bp).
- The ***fwd*** track and ***rev*** tracks contain the forward and reverse read coverage information, as calculated over the de novo contigs. When zooming in on the tracks, the bases are colored based on the coverage information: the bases have pale color when all reads contain the same base at that position, and a dark color when there is at least one read with a different base at that position (see Figure 12.6).

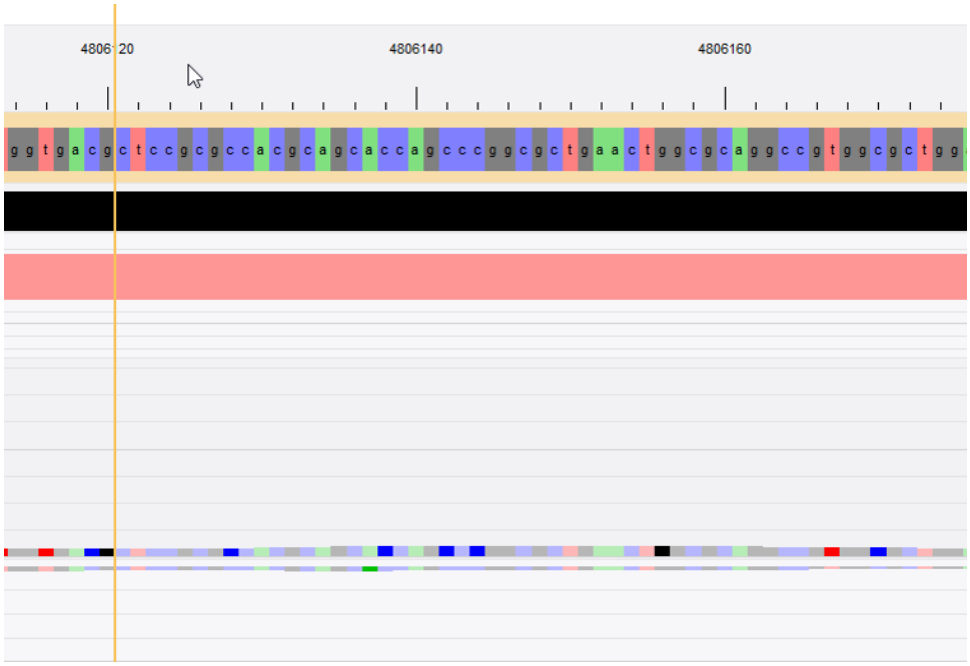

Figure 12.6: Forward and reverse read coverage information.

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 12.7). This makes it easy to detect the different tracks on the graphical representation at a glance.

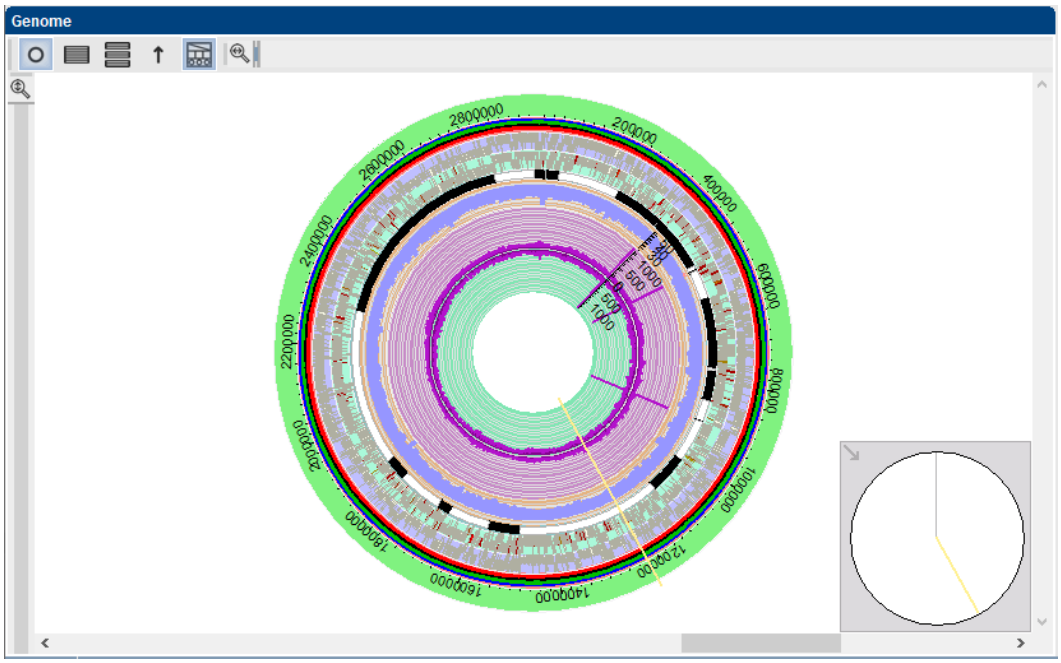

Figure 12.7: Tracks displayed in color.

12.2.3 Alleles and Details panel

12.2.3.1 Allele calls

In the *Alleles* panel the allelic assignments are listed for the entry selected in the *Entries* panel (see Figure 12.8 for an example). Details of the selected allelic assignment is shown in the *Details* panel below.

The different statuses of an allele, used in this section, are given below:

- ***Reference***: There is typically only one reference allele, though for loci with higher diversity and different use of frames, more than one reference can be defined. Only reference alleles are used to search for matches by the Blast Allele Finder.
- ***Accepted***: An accepted allele meets the quality criteria set by the curator. All accepted alleles are used for detection with the assembly free allele calling. The matching alleles found by the Blast allele Finder are compared with all accepted alleles.
- ***Tentative***: A tentative allele does not meet the quality criteria set by the curator. Tentative alleles are not part of the search data for the allele calling algorithms. They can only be assigned an allele ID after submission to the allele nomenclature database (either automatically if the user has lower quality parameters for submission than the criteria for acceptance, or manually).
- ***Revoked***: An allele that has been manually removed by the curator due to issues not picked up by the automated submission. Revoked alleles are very rare and not included in any of the search data.

For ease of interpretation the results of the assembly-free and assembly-based algorithms are split up in this section:

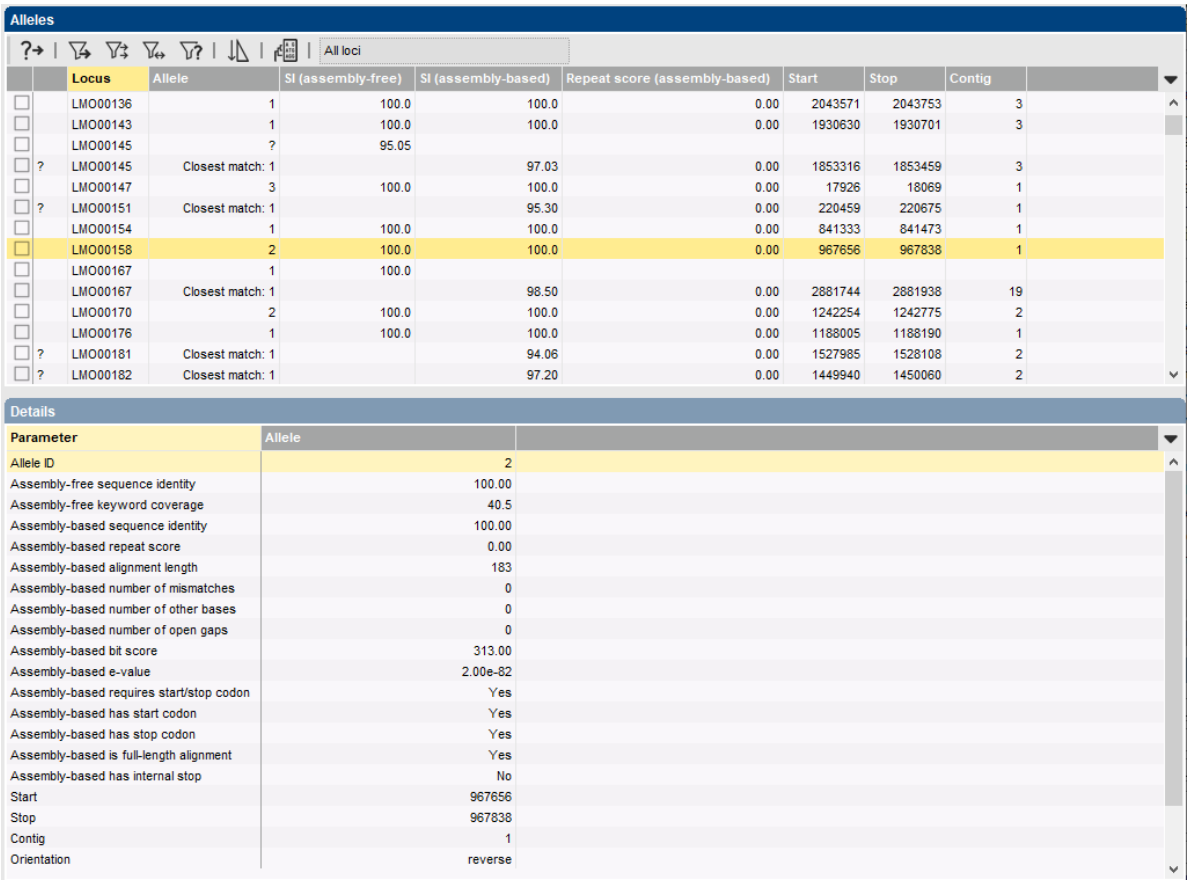

Figure 12.8: Allelic assignments.

ASSEMBLY-FREE CALLS

All loci that passed the assembly-free criteria (see Figure 8.6) are listed in the *Alleles* panel. The locus identifier is displayed in the ***Locus*** column. The result of the matching of the allelic sequences against the nomenclature allele database records are listed in the ***Allele*** and ***SI (assembly-free)*** columns:

- When a 100% match is found with an allele in the allele database, the allele number is indicated in the ***Allele*** column and the similarity value (100%) is indicated in the ***SI (assembly-free)*** column.
- Matches with a similarity below 100% are also listed, but are not further considered. A question mark is displayed in the ***Allele*** column and the similarity value with the best matching reference allele is indicated in the ***SI (assembly-free)*** column.

Details of the selected assembly-free calling are shown in the *Details* panel below: the ***Sequence identity*** between the allelic sequence and the best matching reference in the allele database and the ***keyword coverage*** are listed.

Loci that were only detected based on the assembly-free algorithm will not be plotted on the sequence in the *Genome* panel since no contig position information can be derived from the assembly-free algorithm. If the locus is also detected by the assembly-based approach, the locus will be plotted both on the ***Assembly-free calls*** and ***Assembly-based calls*** track.

ASSEMBLY-BASED CALLS

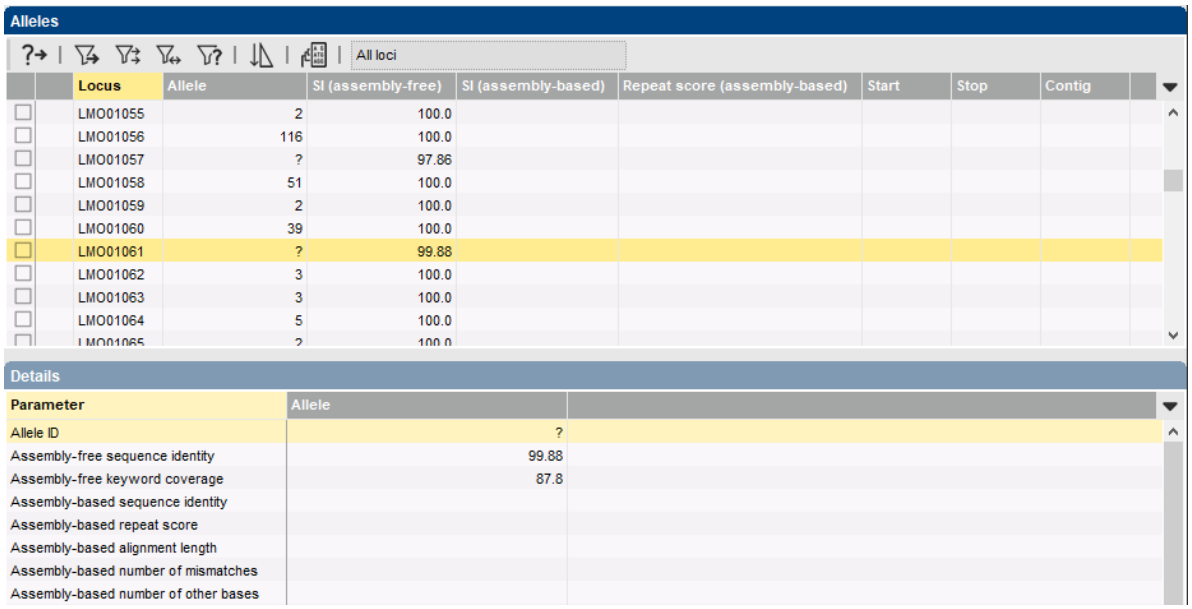

Figure 12.9: Assembly-free results: perfect and non-perfect matches.

Only the detected alleles that passed the ***Minimum similarity*** threshold (see Figure 8.5), i.e. the minimum BLAST similarity between the allele sequence and (one of) the reference sequence(s) in the allele database are retained and are listed in the *Alleles* panel. The locus identifier is displayed in the ***Locus*** column.

The results of the exact matching of the allelic sequence against the reference and accepted alleles in the allele database are listed in the ***Allele*** and ***SI (assembly-based)*** columns.

- When a 100% match (***SI (assembly-based)***) is found with a reference or accepted allele sequence for a locus, the allele number is indicated in the ***Allele*** column.
- Matches that do not have a 100% match with an allele in the allele database but fulfill all specified automatic submission criteria are automatically submitted and receive the ”tentative” status until approved by the curator. This is indicated with an ”!” in the first column. An automatic curation process is followed instantly: when the ”tentative” allele passes the curator settings, the status is automatically converted to ”accepted”. All accepted alleles are updated each night.
- When a 100% match (***SI (assembly-based)***) is found with a tentative allele sequence for a locus, an ”!” is indicated in the first column, the (tentative) allele number is indicated in the ***Allele*** column.
- Matches that do not have a 100% match with an allele in the allele database and that do not fulfill the automatic submission criteria are indicated with the text ***Closest match: x***. The best matching reference allele is listed (x) together with the similarity with this reference sequence (see ***SI (assembly-based)*** column). When the sequence consists of non-ambiguous bases a ”?” is indicated in the first column (eligible for manual submission); when IUPAC code is present, nothing is indicated in the first column.

The automatic submission criteria can be called with ***WGS tools*** > ***Settings...***: click the *wgMLST* tab and the <***Auto submission criteria***> button. By default, the ***Use nomenclature acceptance criteria*** option will be checked, meaning that the automatic submission settings are used that are defined by the curator of the allele database. By default a start and stop codon are required in

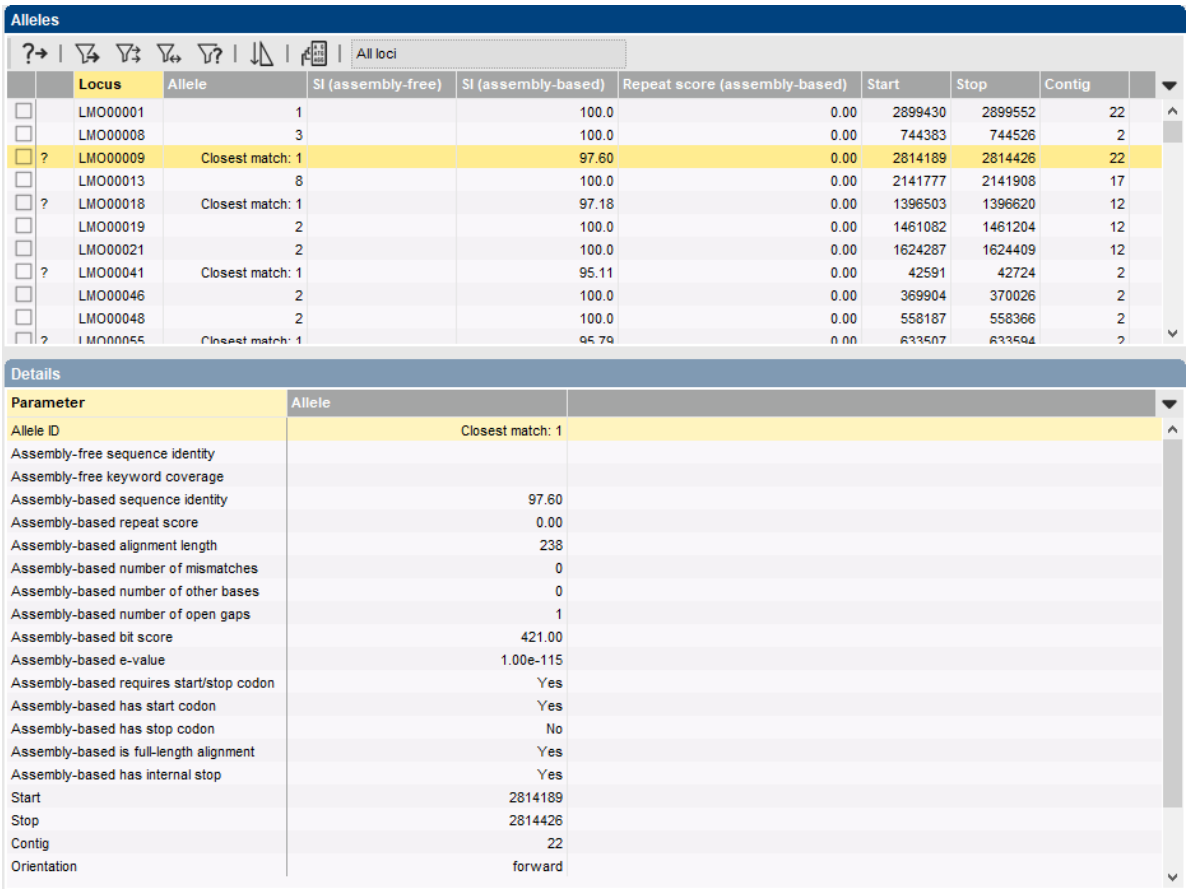

Figure 12.10: Assembly-based results.

case of CDS loci, internal stops are not allowed, and a minimum homology with the reference allele(s) is required for automatic submission (see 3 for more information).

Details of the selected assembly-based calling are shown in the *Details* panel below and the locus is selected and located in the upper area of the circular sequence in the *Genome* panel. The locus is plotted on the map (based on the ***Start***, ***Stop*** and ***Contig*** information of the locus) on the ***Assembly-based calls*** track. The locus identifier and allele sequence number (between brackets) are indicated. Matches that do not have a 100% match (see ***SI (assembly-based)*** column) are colored based on the similarity value: yellow over red (lowest similarity). When the locus was also detected by the assembly-free algorithm, the locus is also plotted on the ***Assembly-free calls*** track.

A highlighted allele identification result in the grid can be viewed in detail by double-clicking or selecting ***Alleles*** > ***Open alignment...*** ( ). This opens the *Sequence alignment* window with the query allele sequence (if a BLAST hit was found by the assembly-based algorithm) and all reference and accepted allele sequences for that specific locus (see Figure 12.11). This way, allele identification results can be verified within the locus setting. The *Sequence alignment* window is opened as a temporary analysis and modifications cannot be saved to the BIONUMERICS database.

SUMMARY CALLS

When both algorithms (assembly-free and assembly-based) were run, all available data from the two allele identification algorithms are ”summarized” into a single set of allele assignments and stored in the ***wgMLST*** character experiment. The way the data is ”summarized” depends on the calls that were obtained for each locus and on the settings defined in the *wgMLST tab* in the

Figure 12.11: Detailed alignment view for allele identification.

Calculation engine settings dialog box (see 3). Default, among the allele calls that the assembly-based and the assembly-free method have in common for a given locus, the one with the lowest allele ID is retained.

An overview of how the summary calls are obtained by combination of the assembly-free and assembly-based allele calls, is given in Table 12.1.

Assembly-based / Assembly-free	**X**	**unknown**	**1**	**2**	**1,2**	**2,3**
X	X	unknown	1	2	1,2	2,3
1	1	1	1	Discrepant	1	Discrepant
2	2	2	Discrepant	2	2	2
1,2	1,2	1,2	1	2	1,2	2
3,4	3,4	3,4	Discrepant	Discrepant	Discrepant	3
Closest match:2	unknown	unknown	1	2	1,2	2,3
New: 7	7	7	Discrepant	Discrepant	Discrepant	Discrepant

Table 12.1: Combination of the assembly-free and assembly-based resulting allele calls to summary calls.

Horizontal: Assembly-free calls

Vertical: Assembly-based calls

X = absent locus call

Unknown = unknown allele

Discrepant = discrepant allele

1 = locus called as allele sequence 1.

1,2 = locus called with multiple allele sequences, i.e. allele sequence 1 and 2. Both allele numbers

will be listed in the Alleles panel, but only the lowest allele number will be retained in the wgMLST experiment.

Closest match: No 100% match with an allele in the allele database is found and the automatic submission criteria are not fulfilled.

New: No 100% match with an allele in the allele database is found but all automatic submission criteria are fulfilled, or the sequence has a 100% match with a tentative allele.

12.2.3.2 Sorting and filtering options

In the *Alleles* panel one can filter allele results to subscheme-specific loci or to alleles that need to be submitted to the reference allele database, alleles that show imperfect or new matches, alleles with multiple matches or alleles for which no summary call was obtained.

The content of the ***Allele*** column can be sorted by selecting ***Alleles*** > ***Sort alleles*** ( ).

The different views on the *Alleles* panel allow to zoom in to specific subsets of loci:

- ***Alleles*** > ***Show imperfect and new matches only*** ( ) filters out the imperfect and new matches. These include all alleles which do not have a 100% match with one of the alleles from the database and the alleles with a 100% match with a tentative allele. Those without a 100% match, can either be already submitted (new matches) or not (imperfect matches).

- ***Alleles*** > ***Show multiple matches only*** ( ) filters out the loci for which more than one call was detected for the same locus. One can easily link the multiple calls together by ***Alleles*** > ***Sort alleles*** ( ).

- ***Alleles*** > ***Show non-summary calls*** ( ) filters out the loci that have discrepant allele calls as defined from all available data obtained by the two allele identification algorithms. As a result, no allele number is present in the wgMLST summary for that locus. Typically, this includes loci detected only from the assembly-free algorithm for which one or more alleles were found but no corresponding allele sequence was present in the allele database. In addition, in case the assembly-free algorithm found a so far unknown allele and the assembly-based algorithm found a closest match for an allele sequence that could not be submitted e.g. due to degenerate IUPAC code in the de novo assembled allele sequence, these loci are also included in the non-summary calls.

- ***Alleles*** > ***Show only calls for submission*** ( ) filters out the loci for which allele sequences, obtained by the assembly-based algorithm, can be submitted to the allele database as tentative alleles. In case the automatic submission of alleles upon import in the database was enabled (see 13), only the alleles that did not surpass the submission criteria based on minimum homology and maximum number of gaps are displayed.

On top of the views described here, an additional filtering can be applied based on the defined subschemes in the wgMLST character experiment type. All available subschemes can be used as filter by toggling the subschemes from the drop-down list in the toolbar from the *Alleles* panel. In most reference databases following views have been defined at the curator level and are synchronized upon installation: the default view **All loci**, the **Core loci**, the **MLST** view for the traditional seven housekeeping loci, and the **wgMLST loci** view containing all loci except the ones present in the **MLST** view. User-defined views - if defined - can also be selected from the list.

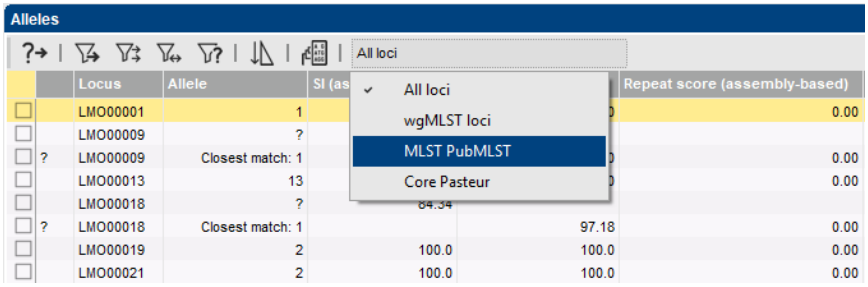

Figure 12.12: Filter based on subschemes.

12.3 The quality character type experiment

The parameters displayed in the *Quality control* dialog box (see Figure 12.2) are stored in the character experiment type **quality**.

Double-clicking on the **quality** experiment in the *Experiment types* panel opens the *Character type* window, displaying all parameters. The quality parameters are grouped based on the data sets and algorithms and the view can be restricted to each of these groups: ***Raw data statistics***, ***Raw data statistics (after trimming)***, ***De novo assembly***, ***Assembly-free calls***, ***Assembly-based calls***, and ***Summary calls***.

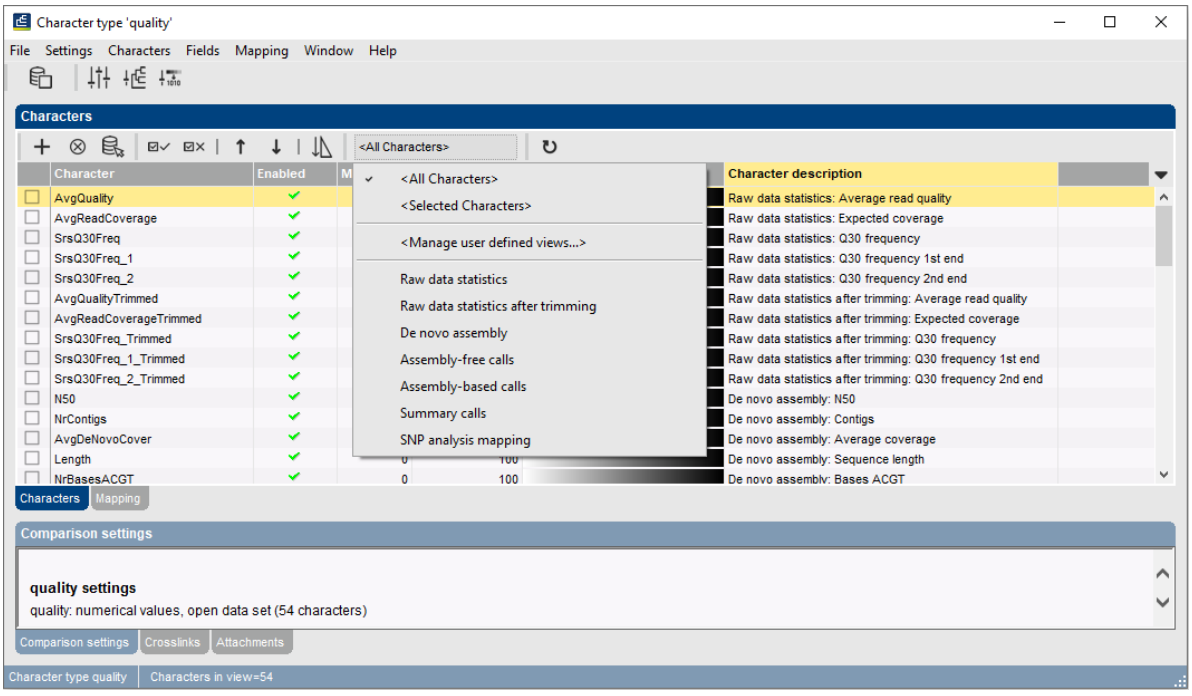

Figure 12.13: Quality character type experiment.

The quality parameters for a selection of entries can be consulted in the *Comparison* window. When clicking the icon next to the experiment name **quality** in the *Experiments* panel, the quality data is displayed in the *Experiment data* panel. Default, the ***Character names*** are displayed in the header of the *Experiment data* panel. These names correspond to the first column in the *Character type* window (see Figure 12.13). To display the ***Character descriptions***, update the display name:

A detailed explanation of each parameter can be found in 12.4.

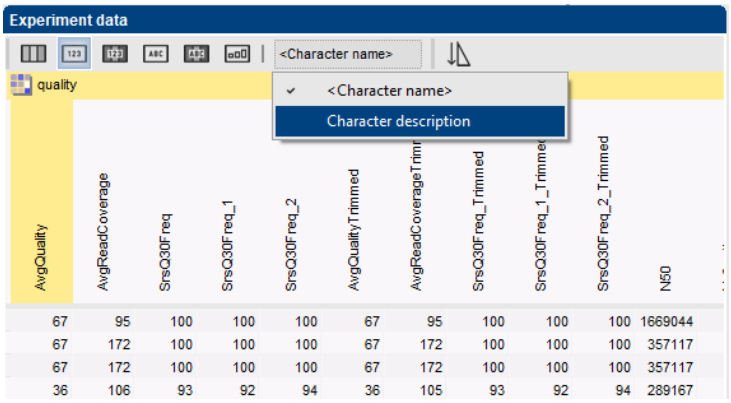

Figure 12.14: Character descriptions.

12.4 The quality parameters

An overview of the quality criteria is given below. Parameters are grouped by the type of calculation they are associated with. The character name used in the **quality** character type experiment is shown between brackets next to the name used by the *Quality control* dialog box.

Raw data statistics

- ***Average read quality (AvgQuality)***: The average quality of the sequence read set using the quality scores from the raw data.
- ***Expected coverage (AvgReadCoverage)***: The expected coverage for each base. Sum of the lengths of all reads divided by the expected sequence length.
- ***Q30 (SrsQ30)***: Total number of bases present in the (paired end) data files that have a quality score of 30 or higher.
- ***Q30 1st end (SrsQ30_1)***: Number of bases present in the first end reads that have a quality score of 30 or higher.
- ***Q30 2nd end (SrsQ30_2)***: Number of bases present in the second end reads that have a quality score of 30 or higher.
- ***Q30 frequency (SrsQ30Freq)***: Number of bases that have a quality score of 30 or higher, expressed as a percentage of the total number of bases present in the (paired end) data files.
- ***Q30 frequency 1st end (SrsQ30Freq_1)***: Number of bases present in the first end reads that have a quality score of 30 or higher, expressed as a percentage of the total number of bases present in the first end reads.
- ***Q30 frequency 2nd end (SrsQ30Freq_2)***: Number of bases present in the second end reads that have a quality score of 30 or higher, expressed as a percentage of the total number of bases present in the second end reads.

Raw data statistics (after trimming)

Same parameters as the ”Raw data statistics” parameters but with the suffix ”Trimmed”. These parameters apply to the trimmed sequence read sets. The ”Raw data statistics” are based on the raw sequence read set.

De novo assembly

- ***N50 (N50)***: Length of the median contig (in terms of sequence length).
- ***Contigs (NrContigs)***: The number of contigs in the assembled sequence.
- ***Bases ACGT (NrBasesACGT)***: Number of bases A, C, G, and T.
- ***Bases non ACGTN (NrNonACGT)***: Number of ambiguous bases (not taking N bases into account).
- ***Bases N (NrBasesN)***: Number of bases N.
- ***Sequence length (Length)***: Length of the assembled sequence. This should be close to the expected sequence length as defined by the curator.
- ***Average coverage (AvgDeNovoCover)***: Average base coverage of all bases included in the assembled sequence.

Assembly-free calls

- ***Average coverage (KeywordCov)***: The average keyword coverage. The keyword coverage is the number of keywords found by the assembly-free calling algorithm for the allele. Only the keyword coverages of the preferred alleles are included in the calculations if multiple alleles have been found for a locus.
- ***Multiple alleles (NrAFMultiple)***: Number of loci with multiple allele hits. In such cases a preferred allele hit is chosen (the one with the lowest allele number).
- ***Perfect matches (NrAFPerfect)***: Number of loci with at least one known allele hit that is 100% identical to an approved allele in the curator database.
- ***Present alleles (NrAFPresent)***: Number of loci with at least one allele hit (unknown and known).

Assembly-based calls

- ***Multiple alleles (NrBAFMultiple)***: Number of loci with multiple allele hits. Similar to the assembly-free calling algorithm, the preferred allele hit is again the one with the lowest allele number.
- ***Perfect matches (NrBAFPerfect)***: Number of loci with at least one known allele hit that is 100% identical to an approved allele in the curator database.
- ***Alleles to submit (NrToBeSubmitted)***: Number of loci with an allele hit eligible for submission to the curator database. Only allele hits with a sequence identity of at least a user-specified threshold (and less than 100%) and whose sequence contains only non-ambiguous bases can be submitted.
- ***Submitted alleles (NrAlreadySubmitted)***: Number of loci which have already been submitted to the curator database.

- ***Present alleles (NrBAFPresent)***: Number of loci with at least one allele hit (= perfect (100%) matches and non-perfect matches). Must be close to the expected number of loci for the organism as defined by the curator.
- ***Average locus coverage (AvgLocusCover)***: Average base coverage for the allele sequences of the preferred alleles. Only alleles for which coverage data is available (either forward, reverse or both directions) are included in the calculations.

Summary calls

The summary loci are obtained by combining the assembly-free and assembly-based calls. If both methods returned allele calls, the summary is defined as the alleles that are similar between both analyses. If for a specific locus, the allele call is only available from one algorithm, that allele call is also included in the summary.

- ***Unknown alleles (NrConsensusUnknown)***: Number of loci for which the assembly-free allele calling algorithm concluded that the locus is present and for which the assembly-free calls nor the assembly-based calls algorithms, if the latter was run, were unable to find an allele.
- ***Multiple alleles (NrConsensusMultiple)***: Number of loci with multiple allele hits. As is the case for both allele identification algorithms, the preferred allele hit here is the one with the lowest allele number.
- ***Discrepant alleles (NrDifferent)***: Number of loci for which there is no overlap in sets of known alleles found by the two allele identification algorithms. This parameter can therefore only be a nonzero value if both algorithms were run.
- ***Confirmed alleles (NrConsensusConfirmed)***: Number of loci that are called with both methods and for which both methods give the same allele id(s).
- ***Present alleles (NrConsensus)***: Number of loci with at least one allele hit. Must be close to the expected number of loci for the organism as set by the curator.
- ***% core present (CorePercent)***: Percentage of loci found (known and unknown) belonging to the subset of core loci. This parameter is only calculated if the curator has defined such a core subset and is not available for all organism schemes.

Chapter 13

Submitting new alleles to the allele database

There are two ways of submitting new allele sequences as tentative alleles to the reference allele database.

1. If automatic submission is defined in the wgMLST settings (see 3 for automatic submission of new alleles in the *Calculation engine settings* dialog box), the new alleles will automatically be submitted to the reference allele database upon import of the assembly-based wgMLST results to the BIONUMERICS database.
2. From the *wgMLST quality assessment* window, new alleles can also be submitted manually. Typically, one can select the view on the imperfect and new matches by selecting ***Alleles*** $>$ ***Show only calls for submission*** ( ) to get an overview of possible new alleles. The user can now go through the list to verify some of the new alleles. Once these matches are verified, the selected allele sequences can be submitted by selecting ***Alleles*** $>$ ***Submit new alleles*** ( ). This command submits new alleles from the current selection eligible for submission to the curator database, and updates the summary calls.

At curator side, the new alleles enter the allele database as *tentative* alleles. The status of these alleles remains tentative until approval of the alleles by the curator of the organism-specific allele database. As long as the status of an allele remains tentative, no matches against this allele are reported by the assembly-based algorithm. Only when new alleles are submitted, the allele sequence can be matched with the allele ID of an allele marked as 'tentative' in the allele database.

Chapter 14

Assigning sequence types

Based on a specific wgMLST subscheme, sequence types can be assigned for the entry selection. Select ***WGS tools*** > ***Assign wgMLST sequence types...***. This will open the *Assign sequence types* dialog box (see Figure 14.1).

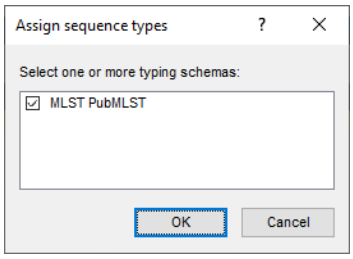

Figure 14.1: The *Assign sequence types* dialog box.

This dialog box lists all wgMLST subschemes that have sequence types available. For any subscheme that has its check-box checked, sequence types will be assigned when the <***OK***> button is pressed.

It is the curator of the wgMLST allele database who determines on which subschemes sequence types can be determined. Sequence type assignment is never possible on wgMLST subschemes that are only available in the client database and not present in the allele database.

Assigning sequence types for a subscheme containing thousands of loci (e.g. the wgMLST or even the core subscheme) generally does not make much sense: sequence types can only be determined if *all* loci are assigned an allele ID and the larger the subscheme, the lower the chance on having a complete allelic profile.

For each selected entry, one or more lists of allelic profiles (one list per typing scheme) are sent to the allele database and sequence type information is returned. For each typing scheme, the sequence type is saved to the corresponding information field for each selected entry. The information field was automatically created during the initial *WGS tools plugin* installation or during a synchronization (see 6).

A message box reports how many sequence types were found for the selected entries. In case an allelic profiles is incomplete, i.e. there was no allele called for one or more loci, a sequence type cannot be assigned. All incomplete allelic profiles are listed in an error report.

Possible values that are filled out in the database during sequence type assignment are:

- **publicSTxxx** (with xxx a number): A public sequence type, i.e. conform the nomenclature

from the external wgMLST service (either BIGSdb or Enterobase, depending on the organism).

- **N/A**: The allelic profile is complete, but no sequence type is available for this profile on the external wgMLST service. Sequence types might be available when the action is repeated later on.
- **STxxx**: The sequence type is assigned on the Calculation Engine and therefore different from the public nomenclature. In the latest BIONUMERICS version, this only occurs for subschemes that have no public sequence types (i.e. not available on BIGSdb or Enterobase).
- Field remains empty: this means that the allelic profile is incomplete, which was indicated in the error message that appears after assignment.

Chapter 15

Analyzing wgMLST profiles

15.1 Cluster analysis of wgMLST data

A cluster analysis on the wgMLST character experiment (or a subscheme thereof; see also 15.2) is created in the *Comparison* window or the *Advanced cluster analysis* window.

First, create a comparison for the selected set of entries. By default, the aspect 'All Characters' is used. One can modify the character aspect to be used in the cluster analysis by selecting the required subscheme from the aspect drop-down list (see Figure 15.1).

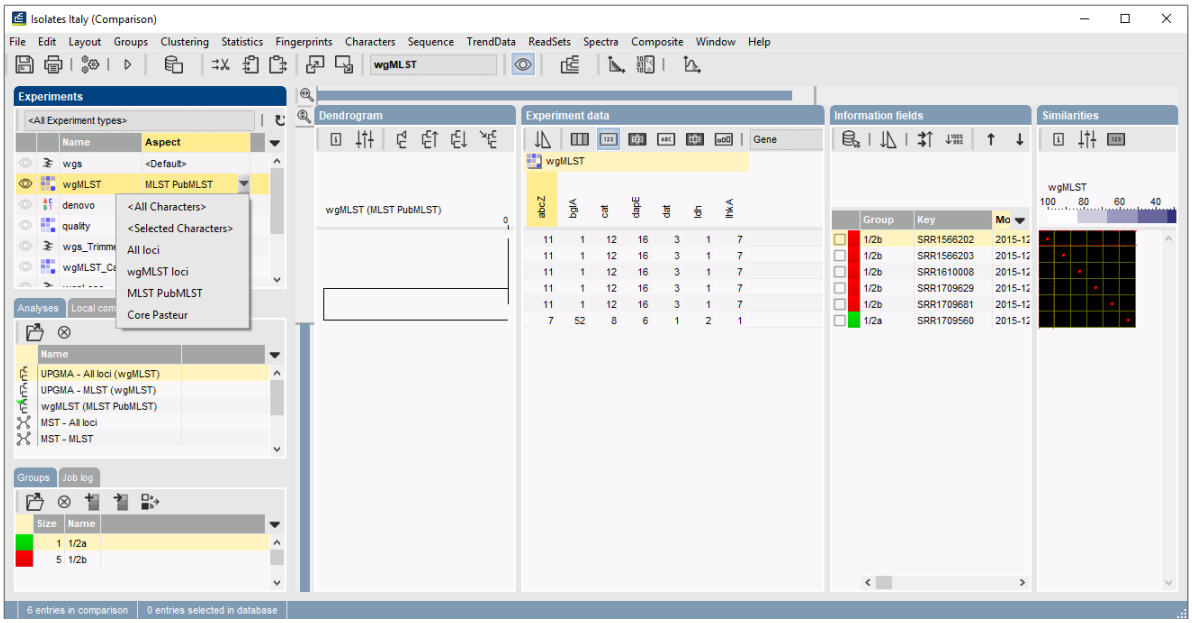

Figure 15.1: wgMLST cluster analysis of the aspect 'MLST' in the *Comparison* window.

Traditional similarity-based clustering can be executed by selecting ***Clustering*** > ***Calculate*** > ***Cluster analysis (similarity matrix)...***. One can use e.g. the ***Categorical (values)*** similarity coefficient and ***UPGMA*** clustering.

Alternatively, in the *Advanced cluster analysis* window e.g. minimum spanning trees can be calculated from all wgMLST loci (see Figure 15.2) by selecting ***Clustering*** > ***Calculate*** > ***Advanced cluster analysis...***, and using the template *MST for categorical data* .

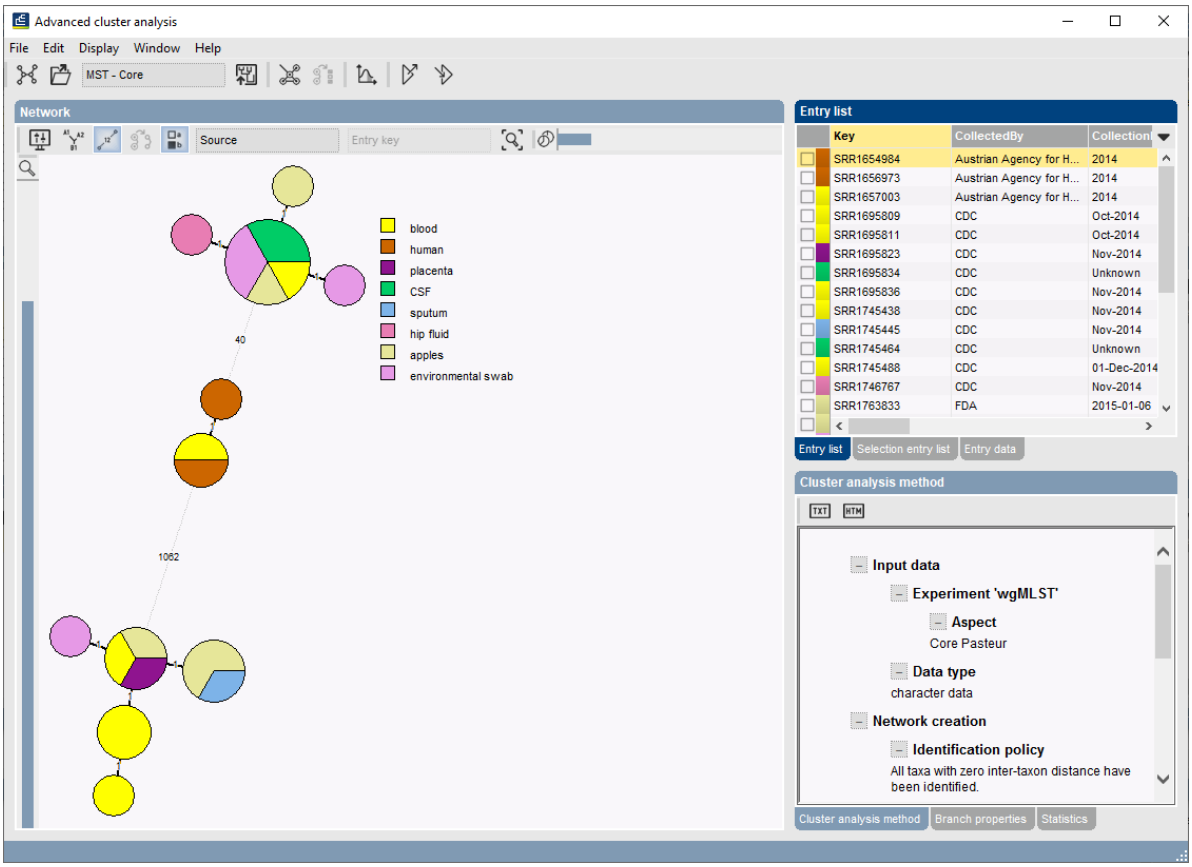

Figure 15.2: wgMLST cluster analysis in the *Advanced cluster analysis* window.

15.2 wgMLST subschemes as character views

A valuable addition in the analysis of wgMLST is the use of subschemes, i.e. subsets of wgMLST loci that are of interest for answering a specific research question. *Character views* can be created within the **wgMLST** character experiment to define these subschemes.

Within the **wgMLST** character type experiment, one or more character views can be defined by the user. Character views defined by the curator in the wgMLST allele database are synchronized upon installation. These include e.g. the core loci, or the MLST view for the traditional seven housekeeping loci (see Figure 15.3).

The user can create as many additional character views (i.e. local character views) as needed. This can be done in two ways. The first method is based on a character selection. The second method is based on a dynamic query using the character information fields.

To create a character view, open the *Character type* window by double-clicking the character experiment type. In the *Character type* window, one can switch between different character views from the drop-down list in the *Characters* panel, as indicated in Figure 15.3. After selecting a character view, the *Character type* window is updated, and the number of characters in view is displayed in the status bar at the bottom of the *Character type* window.

The list of available views can be queried from the *Manage character views* dialog box (see Figure 15.4) after selecting ***Characters*** > ***Character Views*** > ***Manage user defined views...*** (<All Characters>).

From the *Manage character views* dialog box, wgMLST views can be edited, renamed, deleted ...

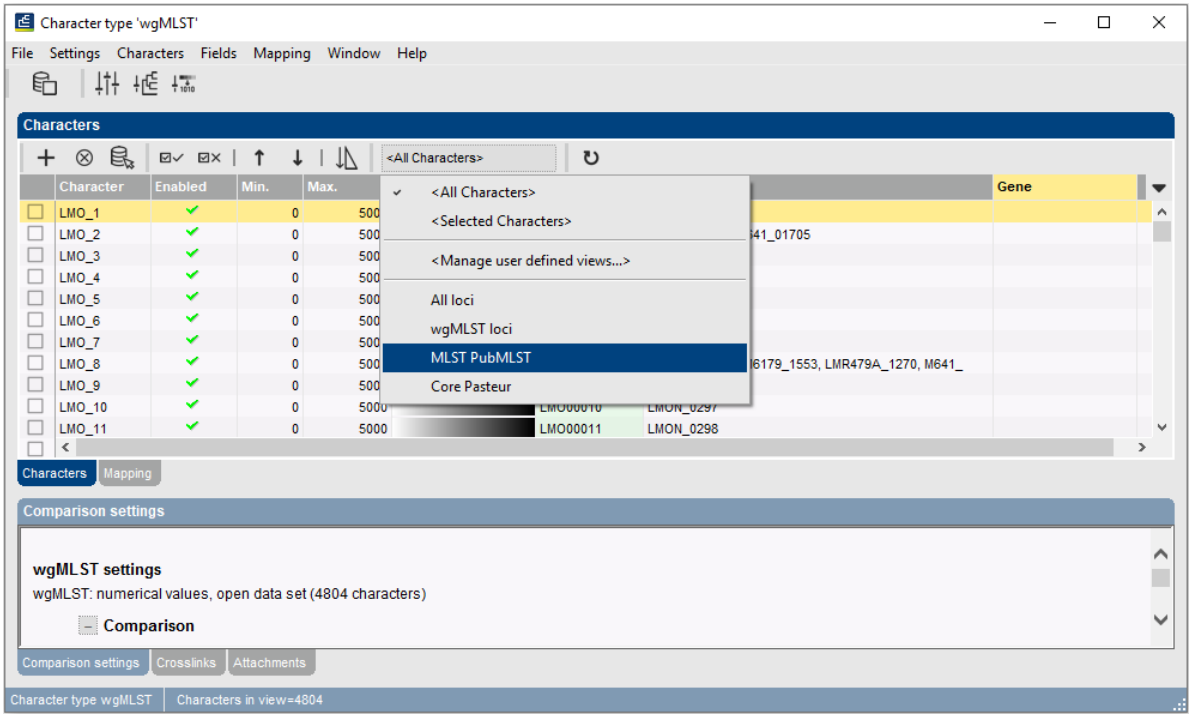

Figure 15.3: Character views in the **wgMLST** character experiment type, here for *Listeria monocytogenes*.

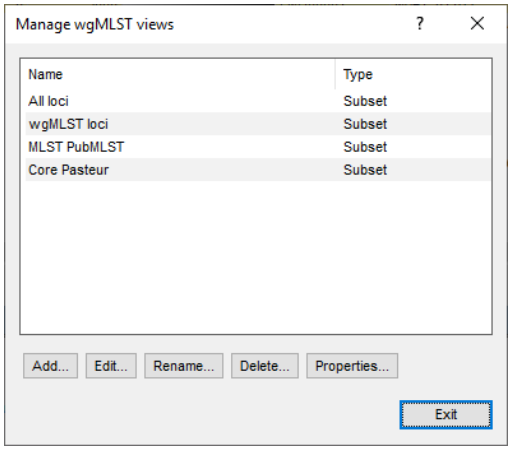

Figure 15.4: The *Manage character views* dialog box.

A view can be based on the current selection or can be based on a dynamic query using the character information fields.

To add a subset-based view, first select the characters that will be part of the subset, next create the subset from the *Manage character views* dialog box by selecting <***Add...***>. This will open the *New character view* dialog box (see Figure 15.5).

In the *New character view* dialog box, a name can be defined for the new view and the view type needs to be specified. For the subset-based view this is sufficient information.

When defining a query-based view, the *Query view editor* dialog box opens, where the query can be defined as statements on the character field values. Once the query is validated, it is added to the list in the *Manage character views* dialog box.

Existing query-based views can be edited by selecting <***Edit...***>. This will open the the *Query*

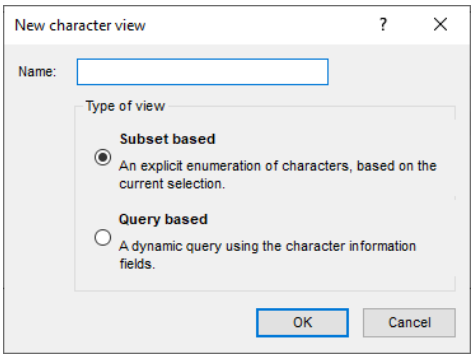

Figure 15.5: The *New character view* dialog box.

view editor dialog box again for evaluation of the character query. Subset-based views and views imported from the curator database cannot be edited. User-defined views can be renamed by selecting <***Rename...***>. Existing views can be deleted by selecting <***Delete...***>. After confirmation, the view is permanently deleted from the database. The object properties on a selected view can be accessed in the *Object access* dialog box by selecting <***Properties...***>.

Chapter 16

Import of sample-specific allele sequences to the database

Once the wgMLST allele results have been imported in the database, it is possible to obtain the actual allele sequences for a specific wgMLST locus or combination of loci, i.e. for all loci present in a defined subschemes. First, the entries need to be selected for which allele sequences should be retrieved and stored in the database. By selecting ***WGS tools*** > ***Store wgMLST locus sequences...***, the *Store sequences* dialog box opens (see Figure 16.1).

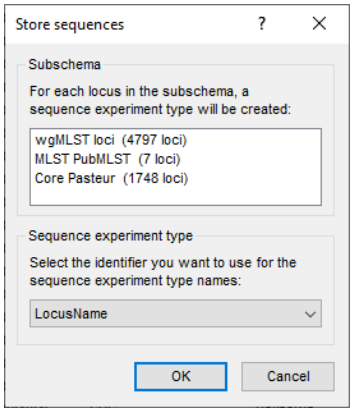

Figure 16.1: The *Store sequences* dialog box.

From the *Store sequences* dialog box, one can define for which subschemes all loci should be imported in the database. For each locus, a separate sequence experiment will be created. The name of the sequence experiments can be defined from one of the custom field in the **wgMLST** character experiment. The sequence experiments name can be picked from the drop-down list, and by default contains the Locus name, the Locus tag, the Gene name and the Character name from the **wgMLST** character experiment.

For each of the selected entries in the database, the respective allele sequence will be retrieved from the curator database and stored in the sequence experiment type. This allows to further analyze the wgMLST allele sequences in the *Sequence alignment* window or the *Comparison* window.

Chapter 17

Core and pan genome analysis

The pan-genome of a bacterial species consists of a core and an accessory gene pool. As the wgMLST locus set is defined as pan-genomics scheme over all available organism genome sequences, the analysis can be limited to the pan-genomic and/or core genomic loci for the selected sample set in the comparison.

For a selected set of samples, the core set of loci can be defined as follows. First, create a *Comparison* window for the selected database entries. Next, highlight the **wgMLST** character experiment and select ***Statistics*** > ***Core locus analysis...*** in the *Comparison* window. This open the *Core locus analysis* dialog box (see Figure 17.1).

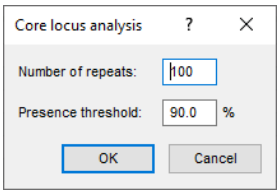

Figure 17.1: The *Core locus analysis* dialog box.

The determination of the number of core loci is based on sub-sampling the entries in the comparison. As such, the ***Number of repeats*** can be defined, i.e. the number of subsamples taken from the comparison set.

The ***Presence threshold*** indicates the minimum presence (expressed in %) for a locus to be called within the core. Entering “90”, will imply that only loci present in 90% of the entry selection will be identified as core loci. For a very strict analysis, one can put the presence threshold at “100”, limiting the core to only those loci which are present in all the entries under evaluation i.e. present in the comparison.

After analysis, the results open in the *Charts and statistics* window. After creating e.g. a profile chart (via ***Plot*** > ***Add new plot from selected properties...*** (+)) on the average, the minimum and the maximum number of loci, the number of core loci can be derived (see Figure 17.2).

In addition, all the core loci are selected in the **wgMLST** character experiment and if required, a subscheme can easily be created on this character selection.

For the same entries, the pan locus set can be defined by selecting ***Statistics*** > ***Pan locus analysis...***. Similar as for the *Core locus analysis* dialog box, the ***Number of repeats*** and ***Presence threshold*** can be defined in the *Pan locus analysis* dialog box (see Figure 17.3).

Similar to the determination of the number of core loci, the number of pan loci is also based on sub-sampling the entries in the comparison. As such, the ***Number of repeats*** can be defined, i.e.

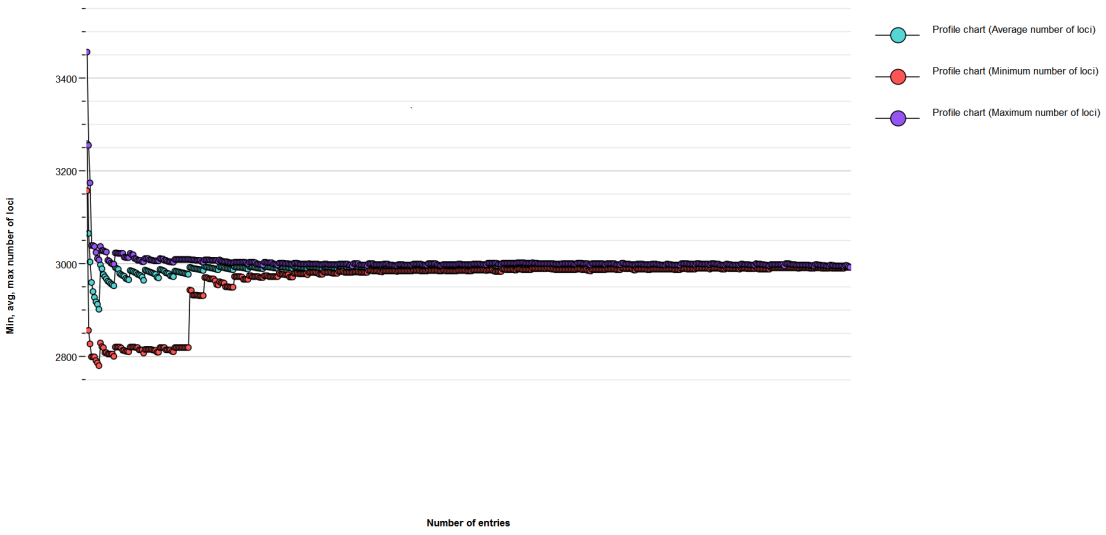

Figure 17.2: The wgMLST core locus analysis for 513 *Listeria* entries resulted in 2992 core loci (Number of repeats: 25; Presence threshold: 90%).

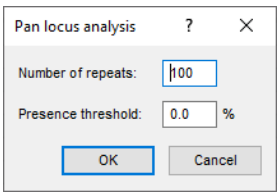

Figure 17.3: The *Pan locus analysis* dialog box.

the number of subsamples taken from the comparison set.

The ***Presence threshold*** indicates the minimum presence (expressed in %) for a locus to be called within the pan loci. Entering “5”, will imply that only loci present in at least 5% of the selected entries will be identified as pan loci. For a very non-restrictive analysis, one can put the presence threshold at “0”, defining the pan loci as all the loci which are present in at least one of the entries under evaluation.

After calculation of the pan loci, the results open in the *Charts and statistics* window, where the profile charts can be created on the average, the minimum and maximum number of loci (see Figure 17.4). After this analysis, all the pan loci are selected in the **wgMLST** character experiment and a subscheme can be created on the character selection.

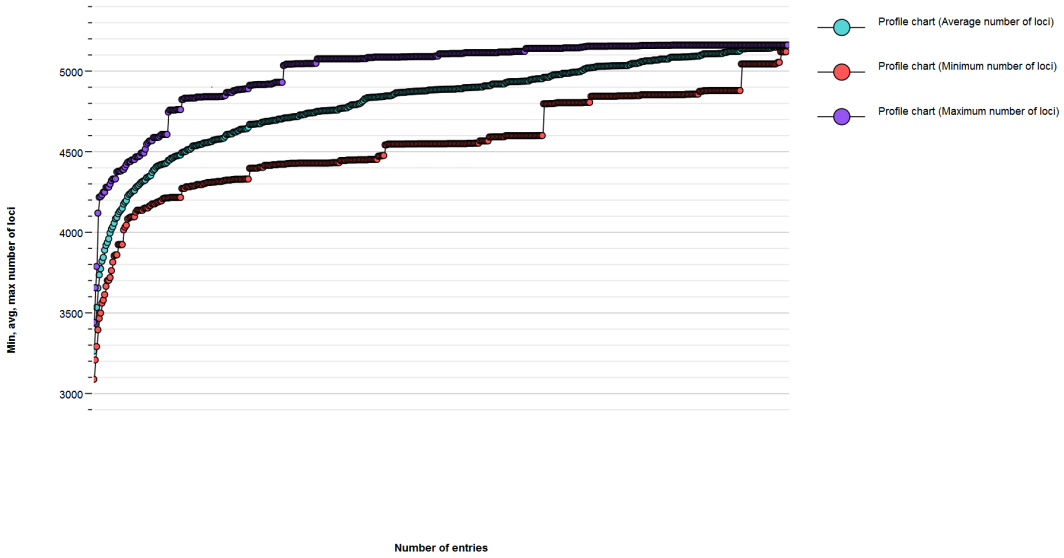

Figure 17.4: The wgMLST pan locus analysis for 513 *Listeria* entries resulted in 5161 pan loci (Number of repeats: 25; Presence threshold: 0%).

Chapter 18

wgMLST nomenclature synchronization

18.1 Introduction

With wgMLST nomenclature we refer to locus definitions, allele number assignments and optionally also sequence type assignments within a wgMLST schema and its subschemas. Unless they start from exactly the same wgMLST schema, two wgMLST services will likely use different locus definitions. However, while not all loci are the same in two schemas, often subsets of loci (subschemas) are shared among schemas, albeit with different locus identifiers. To enable comparison, the locus IDs from one service need to be ”translated” into the locus IDs of the other service. In nearly all cases, each wgMLST service uses its own allele numbering. This means that the exact same sequence will be assigned an allele number on one service and a different number on the other. The only exception to this rule is when both services are connected to the same allele database, which is the case e.g. when a Calculation Engine project is set up in a master / slave connection to another project on another instance of the Calculation Engine.

To be able to compare results obtained with different wgMLST services, BIONUMERICS contains a tool to synchronize between the nomenclature used by the wgMLST schema to which your Calculation Engine project connects and that used by an external wgMLST service. This tool is referred to as **allele mapping** in the software.

For a given organism, one or more allele mapping experiments might be present. The latter are defined by the allele database curators for all clients that connect to the allele database. Mapping experiments consist of a list of loci for which the allele numbers are ”translated” from one taxonomy to another. Clients only need to enable the mapping experiment(s) and run the mapping on a selection of entries. The allelic profiles from the external wgMLST service will be stored in the mapping experiment type.

With missing allele numbers in a profile, it can make sense to re-run the mapping at a later time if the allele ID is known then by the external wgMLST service.

18.2 Activating an allele mapping experiment

Depending on the organism, one or more *allele mapping experiments* might be available. Proceed as follows to see which allele mapping experiments are available (and hence against which public nomenclature can be synchronized):

2.1 Select ***WGS tools*** > ***Settings...***.

2.2 In the *Calculation engine settings* dialog box that appears, click on the *wgMLST* tab.

Available allele mappings are listed in the ***Allele mapping experiments*** list (see Figure 18.1).

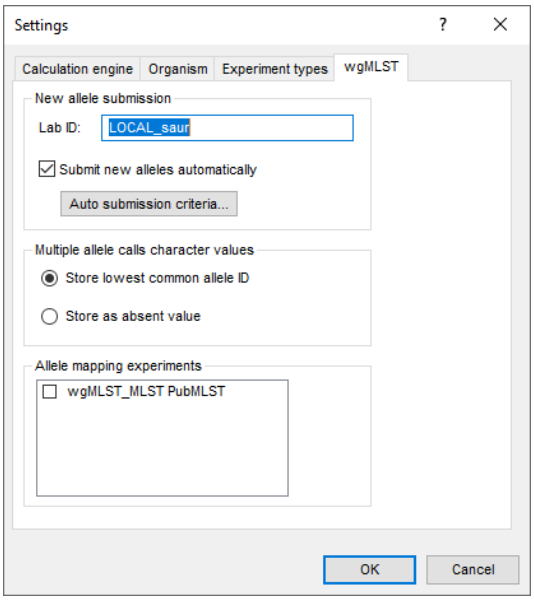

Figure 18.1: The *wgMLST* tab of the *Calculation engine settings* dialog box, showing a single allele mapping experiment for the *Staphylococcus aureus* MLST schema on PubMLST.

If you are aware of a public schema (e.g. MLST, cgMLST, eMLST) being available for your organism on BIGSdb or Enterobase, which is not listed as a mapping experiment, please contact the allele database curators with the request to add a mapping experiment for this schema.

To activate a mapping experiment, check its check box and press <***OK***> to close the *Calculation engine settings* dialog box. A character experiment type with the same name will be automatically created. Each character in this character experiment type corresponds to a locus ID from the external wgMLST service. A character information field called 'Original Locus ID' contains the internal locus identifier as used in the **wgMLST** experiment type (see Figure 18.2).

18.3 Getting allelic profiles and sequence types

As soon as one or more mapping experiments are activated, allelic profiles can be obtained for a selection of entries via ***WGS tools*** > ***Get alleles mapping***. This action "translates" the allele numbers stored in the **wgMLST** experiment type into the corresponding allele numbers used by the external wgMLST service. This action is done for all active allele mapping experiments and will overwrite any existing data in the character experiments of the selected entries.

The allelic profiles can be compared in the *Comparison* window, just as any other character set.

Sequence types can be assigned as outlined in 14. For mapping experiments, the sequence types are assigned by the external wgMLST service and hence correspond to the "public" nomenclature.

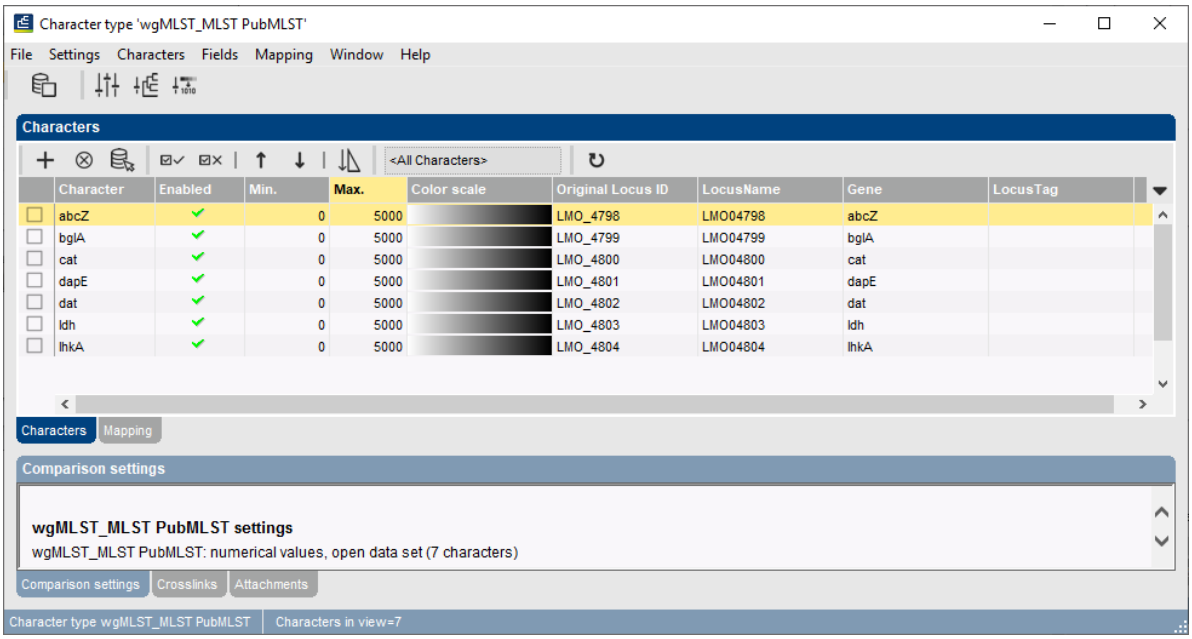

Figure 18.2: The *Character type* window, showing a mapping experiment for the *Campylobacter jejuni* MLST schema on BIGSdb.

Specifically for subschemas corresponding to public MLST (i.e. 7 housekeeping genes) schemes, the allele numbers in the **wgMLST** experiment type will be largely (but not completely!) the same as those obtained from e.g. PubMLST. This is because public MLST allele numbers were taken over in the wgMLST allele database at time of creation. From that moment on, both databases evolved independently and new alleles might were assigned a different number on PubMLST as in the wgMLST allele database on the Calculation Engine. However, the allele numbers in the mapping experiments are those assigned by the external wgMLST service (e.g. PubMLST BIGSdb) and hence correspond to the ”public” nomenclature. The same observation can be made for sequence types (see also 14).

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

Bibliography

[1] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

[2] Steve Davis, James B Pettengill, Yan Luo, Justin Payne, Al Shpuntoff, Hugh Rand, and Errol Strain. Cfsan snp pipeline: an automated method for constructing snp matrices from next-generation sequence data. *PeerJ Computer Science*, 1:e20, 2015.

[3] S Gladman, T Seemann, Victorian Bioinformatics Consortium, et al. Velvet optimiser: for automatically optimising the primary parameter options for the velvet de novo sequence assembler. *Victorian Bioinformatics Consortium, Monash University, Melbourne, Australia*, 2008.

[4] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[5] Morgan N Price, Paramvir S Dehal, and Adam P Arkin. Fasttree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3), 2010.

[6] Torsten Seemann. Prokka: rapid prokaryotic genome annotation. *Bioinformatics*, 30(14):2068–2069, 2014.

[7] Alexandre Souvorov, Richa Agarwala, and David J Lipman. Skesa: strategic k-mer extension for scrupulous assemblies. *Genome biology*, 19(1):153, 2018.

[8] Alexandros Stamatakis. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

[9] Ryan R Wick, Louise M Judd, Claire L Gorrie, and Kathryn E Holt. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS computational biology*, 13(6):e1005595, 2017.

[10] Matei Zaharia, William J Bolosky, Kristal Curtis, Armando Fox, David Patterson, Scott Shenker, Ion Stoica, Richard M Karp, and Taylor Sittler. Faster and more accurate sequence alignment with snap. *arXiv preprint arXiv:1111.5572*, 2011.

[11] D.R. Zerbino and E. Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821, 2008.

[12] Shaokang Zhang, Yanlong Yin, Marcus B Jones, Zhenzhen Zhang, Brooke L Deatherage Kaiser, Blake A Dinsmore, Collette Fitzgerald, Patricia I Fields, and Xiangyu Deng. Salmonella serotype determination utilizing high-throughput genome sequencing data. *Journal of clinical microbiology*, 53(5):1685–1692, 2015.