CLC **Microbial Genomics** Module

USER MANUAL

User manual for

CLC Microbial Genomics Module 25.1

Windows, macOS and Linux

June 20, 2025

This software is for research purposes only.

QIAGEN Aarhus AS
Kalkværksvej 5, 11.
DK - 8000 Aarhus C
Denmark

Contents

Part I

Introduction

Chapter 1

Introduction

Welcome to *CLC Microbial Genomics Module 25.1* – a software package supporting your daily bioinformatics work.

1.1 The concept of QIAGEN CLC Microbial Genomics Module

QIAGEN CLC Microbial Genomics Module includes tools for microbial community analysis as well as tools for epidemiological typing of microbial isolates.

Microbiome composition analysis based on 16S rRNA and other commonly used metagenome derived amplicon data is fully supported. The primary output of the clustering, tallying and taxonomic assignment processes is an OTU abundance table that lists the abundances of OTUs in the samples under investigation. In addition, analyses based on whole shotgun metagenomic data are also available, leading to taxonomic profiling abundance tables. CLC Microbial Genomics Module also offers the possibility to investigate biological functions associated with complex communities using Gene Ontology (GO) and Pfam databases to annotate whole shotgun metagenomic data in functional abundance tables. All abundance tables are viewable through a number of intuitive visualization options. Secondary analyses include estimations of alpha and beta diversities, in addition to various statistical tests for differential abundance.

Tools for NGS-MLST typing and identification of antimicrobial resistance genes are included in CLC Microbial Genomics Module to enable epidemiological typing of microbial isolates using NGS data. In cases when the precise identity of the isolated species is not known, the tool automatically detects the most closely related reference genome in NCBI's RefSeq bacterial genome collection and the corresponding MLST scheme from MLST.net or PubMLST.org. The powerful new CLC metadata framework allows fast and intuitive browsing, sorting, filtering and selection of samples and associated metadata, including results obtained during analysis. This metadata framework provides a dashboard-like overview for easy filtering and selection of samples for other analyses such as k-mer or SNP tree reconstruction and visualisation for outbreak analysis.

For convenience, expert-configured workflows for microbiome analysis as well as epidemiological typing allow the user to get from raw NGS reads through data processing and statistical analysis to the final graphical results in very few steps. Reference databases and MLST schemes needed to perform the analyses are automatically downloadable using dedicated tools, and can be easily customized to fit the specific needs of your research.

Most tools delivered by this plugin are located in the Microbial Genomics Module folder under the Tools menu. Exceptions to this are noted in the manual sections for those tools. Template workflows delivered by this plugin are located in the Microbial Workflows folder under the Workflows menu.

The CLC Microbial Genomics Module is frequently updated. A detailed list of new features, improvements, bug fixes, and changes is available at https://digitalinsights.qiagen.com/clc-microbial-genomics-module-latest-improvements/.

1.2 Contact information

QIAGEN CLC Microbial Genomics Module is developed by:

QIAGEN Aarhus A/S
Kalkværksvej 5, 11.
DK - 8000 Aarhus C
Denmark

https://digitalinsights.qiagen.com/

Email: ts-bioinformatics@qiagen.com

The QIAGEN Aarhus team continuously improves products with your interests in mind. We welcome feedback and suggestions for new features or improvements. How to contact us is described at: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Contact_information_citation.html

You can also make use of our online documentation resources, including:

- Core product manuals https://digitalinsights.qiagen.com/technical-support/manuals/
- Plugin manuals https://digitalinsights.qiagen.com/products-overview/plugins/
- Tutorials https://digitalinsights.qiagen.com/support/tutorials/
- Frequently Asked Questions https://qiagen.my.salesforce-sites.com/KnowledgeBase/KnowledgeNavigatorPage

1.3 System requirements

In addition to meeting the system requirements of the ***CLC Genomics Workbench*** or the ***CLC Genomics Server***, the following requirements must be met:

- An AMD/Intel CPU that supports AVX2 or an Apple M series CPU is required for the tools below:
  - Annotate with DIAMOND
  - Annotate CDS with Best DIAMOND Hits
  - Find Resistance with ShortBRED
  - Type with MLST Scheme

Special requirements for the MLST Scheme tools

The system requirement for the MLST Scheme tools depends on the size of the MLST schemes (both the number of alleles and the number of sequence types). A laptop with 16GB of memory is normally sufficient for 7-gene schemes or cgMLST schemes based on a moderate number of isolates. Downloading and constructing or typing with larger schemes may require more memory, and in general we recommend at least 64GB of memory when working with cg/wgMLST schemes based on more than 100 isolates.

Special requirements for OTU Clustering

The memory requirement of *Reference based* OTU clustering depends on the size of the reference database used; more and longer sequences require more run time and memory. Newer version of common choices (e.g. the full SILVA SSU database) are likely to be too large for a 16 GB machine. Instead we recommend using clustered databases (e.g. the SILVA SSU 99% database) and/or otherwise filtering and subsetting the database, to minimize its size.

Special requirements for Classify Long Read Amplicons

The memory requirement for **Classify Long Read Amplicons** depends on both sample size and the size of reference database. The following are examples of maximum memory usage given different sample and database sizes.

Sample size	Database size	Memory usage
100,000 reads	9 MB / 50,000 sequences	15 GB
100,000 reads	13 MB / 27,000 sequences	10 GB
1,000,000 reads	9 MB / 50,000 sequences	25 GB
1,000,000 reads	13 MB / 27,000 sequences MB	20 GB

Large reference databases like the unclustered SILVA SSU database, are expected to require more than 32GB of available memory.

Special requirements for Taxonomic Profiling

The performance of the Taxonomic Profiling tool depends on the reference database used - the more complete a database, the better the taxonomic profiling. However, running Taxonomic Profiling with a given database size will require at least the same amount of memory. For example, a 14 GB database requires at least 16 GB of RAM, and a 56 GB database requires a minimum of 64 GB RAM. When creating your reference database with the Download Custom Microbial Reference Database tool, you will get a warning about the memory requirements needed for running the Taxonomic Profiling tool with this database.

Special requirements for De Novo Assemble Small Genome

On Linux, this tool requires Linux RHEL 8 and later and supported versions of SUSE Linux Enterprise Server 15.3 and later. The tool is expected to run without problem on other recent Linux systems, but we do not guarantee this. At least 16 GB RAM is recommended for running

De Novo Assemble Small Genome.

Special requirements for De Novo Assemble Metagenome

At least 16 GB RAM is recommended for running De Novo Assemble Metagenome.

Special requirements for Create Whole Metagenome Index

The amount of free disk space needed in the temporary files directory depends on the size of the database. Approximately 1.7 bytes are required per base in the database. For example, to create an index from a 90 gigabase database, you need 90 Gb x 1.7 GB/Gb $\approx$ 150 GB of free temporary disk space.

To learn more about temporary disk space and how to choose a different location, see https://resources.qiagenbioinformatics.com/manuals/workbenchdeployment/current/index.php?manual=Temporary_data.html.

Compatibility

CLC Microbial Genomics Module 25.1 and CLC Microbial Genomics Server Extension 25.1 can be installed on *CLC Genomics Workbench* 25.0.1 and *CLC Genomics Server* 25.0.1, respectively, and on later versions in the same major release line.

1.4 Installing modules

Note: In order to install plugins and modules, the *CLC Workbench* must be run in administrator mode. On Windows, you can do this by right-clicking the program shortcut and choosing "Run as Administrator". On Linux and Mac, it means you must launch the program such that it is run by an administrative user.

Plugins and modules are installed and uninstalled using the Workbench Plugin Manager. To open the Plugin Manager, click on the **Plugins ( ) button** in the top Toolbar, or go to the menu option:

Utilities | **Manage Plugins... ( )**

The Plugin Manager has two tabs at the top:

- **Manage Plugins** An overview of your installed plugins and modules is provided under this tab.
- **Download Plugins** Plugins and modules available to download and install are listed in this tab.

To install a plugin, click on the **Download Plugins** tab (figure 1.1). Select a plugin. Information about it will be shown in the right hand panel. Click on the **Download and Install** button to install the plugin.

Accepting the license agreement

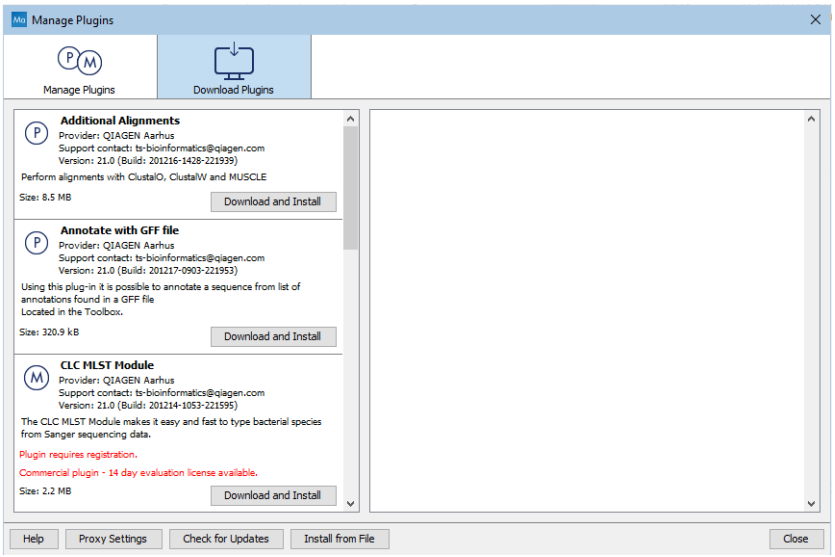


Figure 1.1: *Plugins and modules available for installation are listed in the Plugin Manager under the Download Plugins tab.*

The End User License Agreement (EULA) must be read and accepted as part of the installation process. Please read the EULA text carefully, and if you agree to it, check the box next to the text **I accept these terms**. If further information is requested from you, please fill this in before clicking on the **Finish** button.

Installing a cpa file

If you have a .cpa installer file for QIAGEN CLC Microbial Genomics Module, you can install it by clicking on the **Install from File** button at the bottom of the Plugin Manager.

If you are working on a system not connected to the internet, plugin and module .cpa files can be downloaded from https://digitalinsights.qiagen.com/products-overview/plugins/using a networked machine, and then transferred to the non-networked machine for installation.

Restart to complete the installation

Newly installed plugins and modules will be available for use after restarting the software. When you close the Plugin Manager, a dialog appears offering the opportunity to restart the *CLC Workbench*.

1.4.1 Licensing modules

When you have installed the QIAGEN CLC Microbial Genomics Module and start a tool from that module for the first time, the License Assistant will open (figure 1.2).

The License Assistant can also be launched by opening the Workbench Plugin Manager, selecting the installed module from under the Manage Plugins tab, and clicking on the button labeled *Import License*.

To install a license, the *CLC Workbench* must be run in administrator mode. On Windows, you

can do this by right-clicking the program shortcut and choosing "Run as Administrator". On Linux and Mac, it means you must launch the program such that it is run by an administrative user.

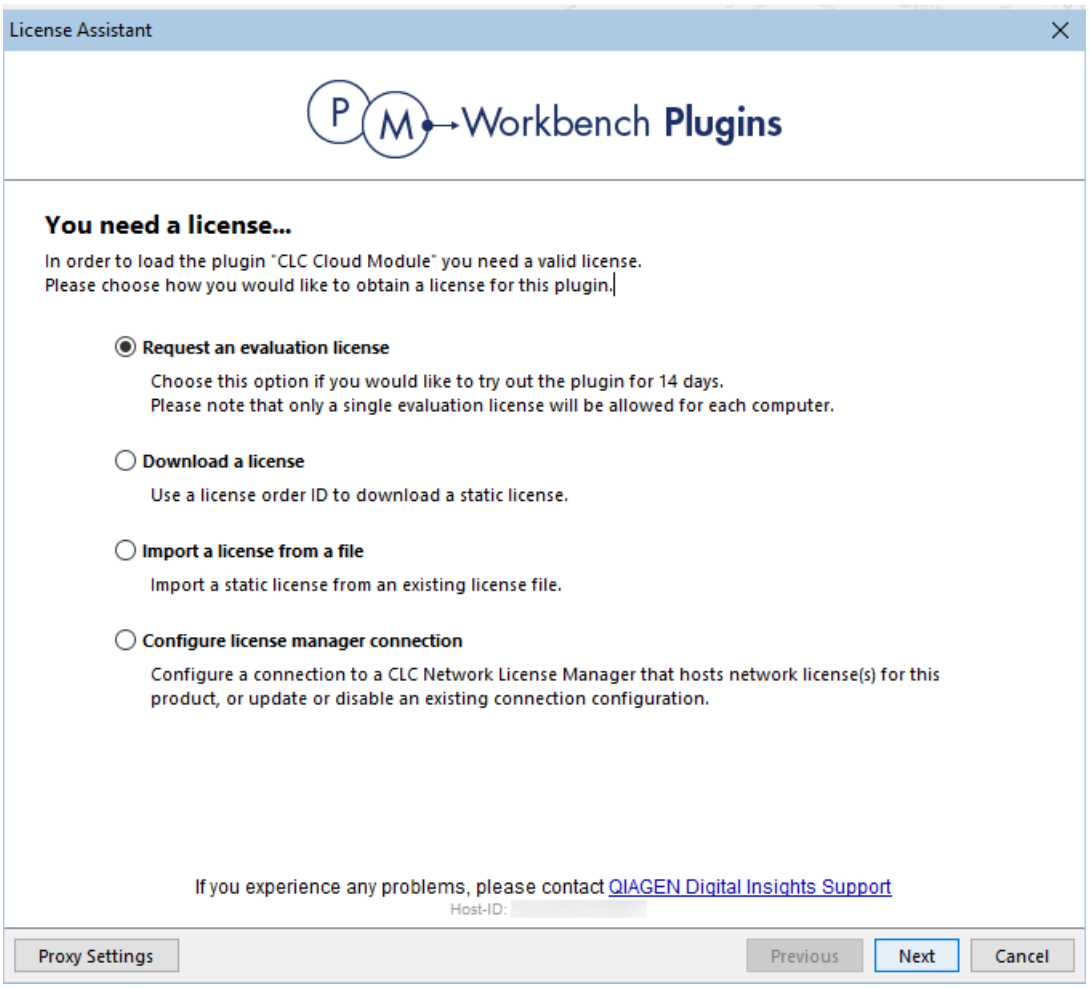


Figure 1.2: *The License Assistant provides options for licensing modules installed on the Workbench.*

The following options are available:

- **Request an evaluation license**. Request a fully functional, time-limited license.
- **Download a license**. Use the license order ID received when you purchased the software to download and install a license file.
- **Import a license from a file**. Import an existing license file, for example a file downloaded from the web-based licensing system.
- **Configure license manager connection**. If your organization has a *CLC Network License Manager*, select this option to configure the connection to it.

These options are described in detail in sections under https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Workbench_Licenses.html.

To download licenses, including evaluation licenses, your machine must have access to the external network. To install licenses on non-networked machines, please see https://resources.

qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Download_static_license_on_non_networked_machine.html.

1.4.2 Uninstalling modules

Plugins and modules are uninstalled using the Workbench Plugin Manager.To open the Plugin Manager, click on the **Plugins ( ) button** in the top Toolbar, or go to the menu option:

Utilities | Manage Plugins... ( )

This will open the Plugin Manager (figure 1.3). Installed plugins and modules are shown under the Manage Plugins tab of the Plugins Manager.

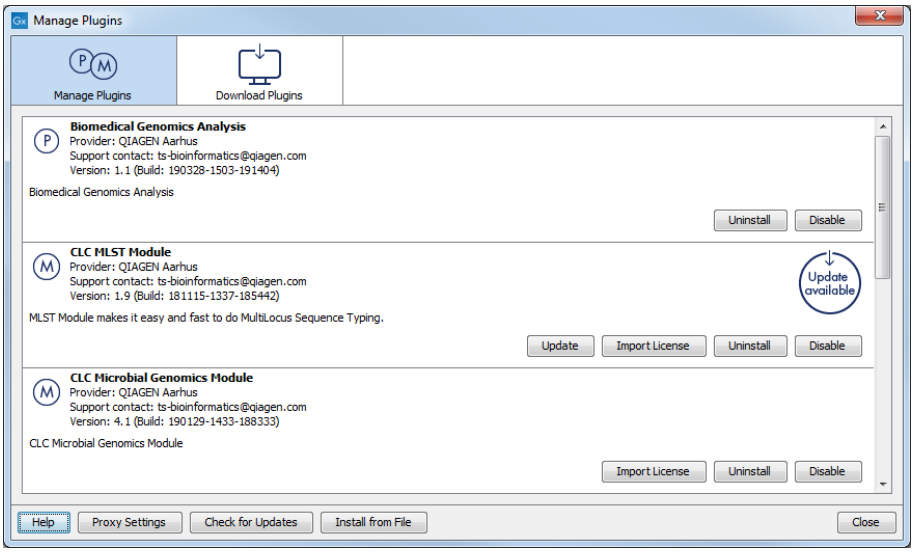


Figure 1.3: *Installed plugins and modules are listed in the Plugins Manager under the Manage Plugins tab.*

To uninstall a plugin or module, click on its entry in the list, and click on the **Uninstall** button.

Plugins and modules are not uninstalled until the Workbench is restarted. When you close the Plugin Manager, a dialog appears offering the opportunity to restart the *CLC Workbench*.

Disabling a plugin without uninstalling it

If you do not want a plugin to be loaded the next time you start the Workbench, select it in the list under the Manage Plugins tab and click on the **Disable** button.

1.5 Installing server extensions

To use the tools and functionalities of QIAGEN CLC Microbial Genomics Module on a *CLC Server*:

1. You need to purchase a license to run tools delivered by the CLC Microbial Genomics Server Extension.
2. A *CLC Server* administrator must install the license on the single server, or on the master node in a job node or grid node setup, as described in section 1.5.1.

3. A *CLC Server* administrator must install the CLC Microbial Genomics Server Extension on the *CLC Server*, as described below.

Download and install server plugins and server extensions

Plugins, including server extensions (commercial plugins), are installed by going to the **Extensions ( )** tab in the web administrative interface of the single server, or the master node of a job node or grid nod setup, and opening the **Download Plugins ( )** area (figure 1.4).

Figure 1.4: *Installing plugins and server extensions is done in the Download Plugins area under the Extensions tab.*

If the machine has access to the external network, plugins can be both downloaded and installed via the *CLC Server* administrative interface. To do this, locate the plugin in the list under the **Download Plugins** ( ) area and click on the **Download and Install...** button.

To download and install multiple plugins at once on a networked machine, check the "Select for download and install" box beside each relevant plugin, and then click on the **Download and Install All...** button.

If you are working on a machine without access to the external network, server plugin (.cpa) files can be downloaded from: `https://digitalinsights.qiagen.com/products-overview/plugins/` and installed by browsing for the downloaded file and clicking on the **Install from File...** button.

The *CLC Server* must be restarted to complete the installation or removal of plugins and server extensions. All jobs still in the queue at the time the server is shut down will be dropped and would need to be resubmitted. To minimize the impact on users, the server can be put into Maintenance Mode. In brief: running in Maintenance Mode allows current jobs to run, but no new jobs to be submitted, and users cannot log in. The *CLC Server* can then be restarted when desired. Each time you install or remove a plugin, you will be offered the opportunity to enter Maintenance Mode. You will also be offered the option to restart the *CLC Server*. If you choose not to restart when prompted, you can restart later using the option under the **Server maintenance ( )** tab.

For job node setups only:

- Once the *master CLC Server* is up and running normally, then restart each *job node CLC Server* so that the plugin is ready to run on each node. This is handled for you if you restart the server using the functionality under

  Management ( ) | Server maintenance ( )

- In the web administrative interface on the *master* CLC Server, check that the plugin is enabled for each job node.

Installation and updating of plugins on connected job nodes requires that direct data transfer from client systems has been enabled, which is done by the *CLC Server* administrator, under the "External data" tab.

Grid workers will be re-deployed when a plugin is installed on the master server. Thus, no further action is needed to enable the newly installed plugin to be used on grid nodes.

Managing installed server plugins

Installed plugins can be updated or uninstalled, from under the **Manage Plugins** ( ) area (figure 1.5), under the **Extensions ( )** tab.

The list of tools delivered with a server plugin can be seen by clicking on the **Plugin contents** link to expand that section. Workflows delivered with a server plugin are not shown in this listing.

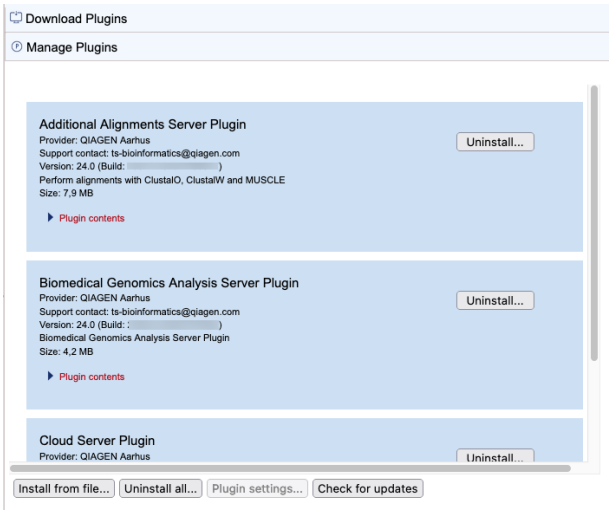


Figure 1.5: *Managing installed plugins and server extensions is done in the Manage Plugins area under the Extensions tab. Clicking on Plugin contents opens a list of the tools delivered by the plugin.*

Links to related documentation

- Logging into the *CLC Server* web administrative interface: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Logging_into_administrative_interface.html
- Maintenance Mode: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Server_maintenance.html

- Restarting the server: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Starting_stopping_server.html
- Plugins on job node setups: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Installing_Server_plugins_on_job_nodes.html
- Grid worker re-deployment: https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Overview_Model_II.html

Plugin compatibility with the server software

The version of plugins and server extensions installed must be compatible with the version of the *CLC Server* being run. A message is written under an installed plugin's name if it is not compatible with the version of the *CLC Server* software running.

When upgrading to a new major version of the *CLC Server*, all plugins will need to be updated. This means removing the old version and installing a new version.

Incompatibilities can also arise when updating to a new bug fix or minor feature release of the *CLC Server*. We recommend opening the **Manage Plugins** area after any server software upgrade to check for messages about the installed plugins.

Licensing server extensions is described in section 1.5.1.

1.5.1 Licensing server extensions

Licenses are installed on a single server or on the master node of a job node or grid node setup.

To download and install a license:

- Log into the web administrative interface of the single server or master node as an administrative user.
- Under the **Management ( )** tab, open the **Download License ( )** tab.
- Enter the Order ID supplied by QIAGEN into the Order ID field and click on the "Download and Install License..." button (figure 1.6).

Please contact ts-bioinformatics@qiagen.com if you have not received an Order ID.

The *CLC Server* must be restarted for new license files to be loaded. Details about restarting can be found at https://resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Starting_stopping_server.html.

Each time you download a license file, a new file is created in the `licenses` folder under the *CLC Server* installation area. *If you are upgrading* an existing license file, *delete the old file* from this area before restarting.

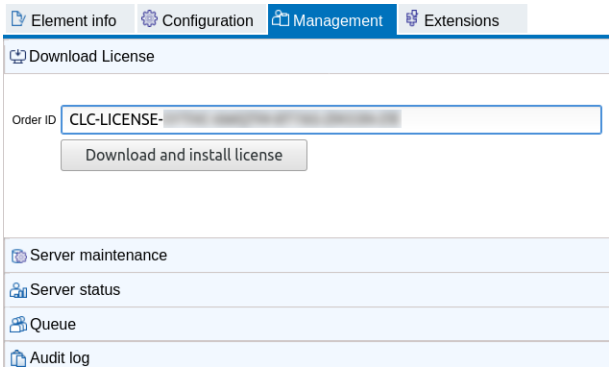


Figure 1.6: *License management is done under the Management tab.*

Part II

Core Functionalities

Chapter 2

Microbial template workflows

Template workflows are provided as example workflows. They can be launched as they are from under the Workflows menu, or copies can be easily opened, allowing you to optimize the workflow to fit your specific application.

To open a template workflow to view the design or edit it, you can:

- Right-click on the workflow name in the Toolbox in the lower, left side of the Workbench under:

  Workflows | **Template Workflows**

  and select the option **Open Copy of Workflow** from the right-click menu.

  or

- Open the Workflow Manager by clicking on the **Workflows** button () in the top toolbar, and choose **Manage Workflows**.

  Click on the **Template Workflows** tab and then select the workflow you wish to edit. Then click on the **Open Copy of Workflow** button.

For an introduction to workflows and information on how to configure workflow elements, see Workflows.

In the following sections, we describe the template workflows distributed with CLC Microbial Genomics Module.

2.1 Taxonomic Analysis template workflows

The Taxonomic Analysis template workflows are found at:

Workflows | **Template Workflows** () | **Microbial Workflows** () | **Metagenomics** () | **Taxonomic Analysis** ()

2.1.1 Data QC and Remove Background Reads

The **Data QC and Remove Background Reads** template workflow performs trimming of reads, creates a QC report and cleans the dataset from background DNA, leaving back only the reads

that match the reference genome(s).

To run the workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **Data QC and Remove Background Reads** ( )

1. Specify the sample(s) or folder(s) of samples you would like to analyze.
2. Specify a **Trim adapter list** if your sequences contain adapters (see `Adapter trimming`).
3. In the "Taxonomic Profiling" step (figure 2.1), select the "Species of interest taxpro index" you will use to map the reads, and the "Background taxpro index". Reference databases can be obtained by using the Download Curated Microbial Reference Database tool (section 16.1) or Download Custom Microbial Reference Database tool (section 16.2). For custom reference databases, indexes can be built with the Create Taxonomic Profiling Index tool (section 16.5).
4. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).

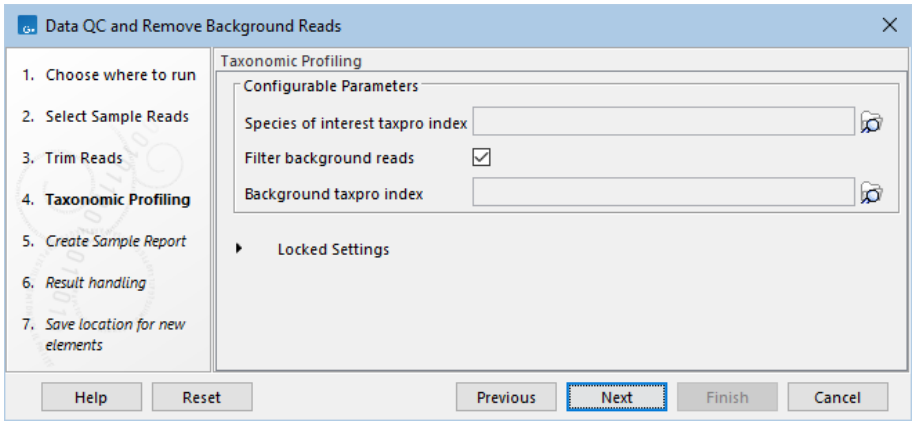


Figure 2.1: *Select the species of interest taxpro index and a background taxpro index to remove possible contamination.*

The workflow produces the following outputs:

- **Cleaned reads**. Folder containing trimmed reads mapping to the species of interest taxpro index of reference genome(s).
- **Background reads**. Folder containing reads mapping to the background taxpro index.
- **Unmapped reads**. Folder containing reads not mapping to the species of interest or the background taxpro index.
- **QC & Reports**. Folder containing the individual reports generated during the analysis.
- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

The Sample report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

2.1.2 Data QC and Taxonomic Profiling

The **Data QC and Taxonomic Profiling** template workflow combines the Taxonomic Profiling tool with a trimming step and additionally creates sequencing QC reports. The workflow outputs both a raw and a refined taxonomic profiling abundance table as well as additional reports on the trimming, QC and taxonomic analysis.

To run the workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **Data QC and Taxonomic Profiling** ( )

1. Specify the sample(s) or folder(s) of samples you would like to analyze.
2. Specify a **Trim adapter list** if your sequences contain adapters (see `Adapter trimming`).
3. In the "Taxonomic Profiling" step (figure 2.2), choose the index of references that you wish to map the reads against. You could also remove host DNA by specifying a host genome index (e.g., Homo sapiens GRCh38). Reference databases can be obtained by using the Download Curated Microbial Reference Database tool (section 16.1) or Download Custom Microbial Reference Database tool (section 16.2). For custom reference databases, indexes can be built with the Create Taxonomic Profiling Index tool (section 16.5).
4. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).

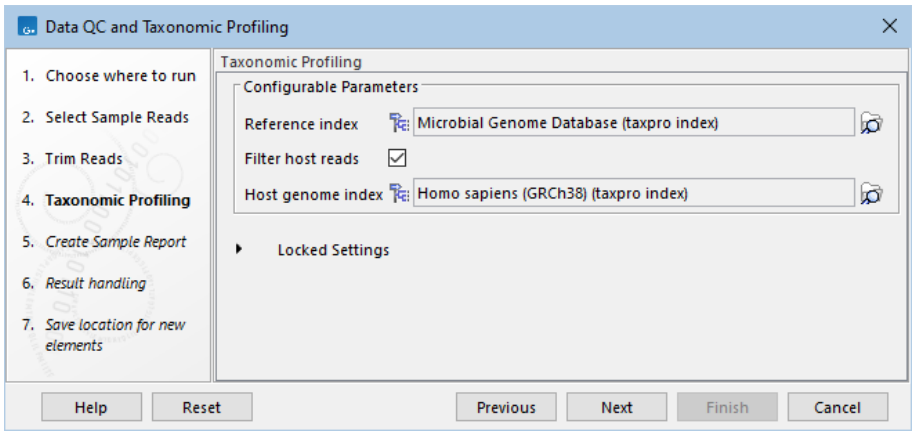


Figure 2.2: *Specify the reference database. You can also check the option "Filter host reads" and specify the host genome.*

The workflow produces the following outputs:

- **Raw abundance table**. The Taxonomic Profiling abundance table.
- **Refined abundance table**. The Taxonomic Profiling abundance table after being refined by running it through **Refine Abundance Table** (section 7.2). The Refined abundance table has

been aggregated on species level and filtered to exclude taxa with relative abundance below 1%. This is the recommended practice for taxonomic profiling in order to avoid drawing wrong conclusions from the results. See more about the other filtering options available at section 7.2.1.

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

The abundance table displays the names of the identified taxa, along with their full taxonomy, the total amount of reads associated with that taxon, and a coverage estimate. The table can be visualized using the Stacked bar charts and stacked area charts function, as well as the Sunburst charts (see section 6.4.3).

The Sample report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

2.1.3 Merge and Estimate Alpha and Beta diversities

The **Merge and Estimate Alpha and Beta diversities** template workflow requires several abundance tables as input file. The first tool of the workflow is the **Merge Abundance Tables**. The output is a single merged abundance table that will be used as input for two additional tools, the **Alpha diversity** tool and the **Beta diversity** tool. Running this workflow will therefore give three outputs: a diversity report for the alpha diversity, a PCoA for the beta diversity and a merged abundance table.

If your data is whole metagenome sequencing data use this workflow. Otherwise use **Estimate Alpha and Beta Diversities** (section 2.2.3).

To run the workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **Merge and Estimate Alpha and Beta diversities** ( )

In the first step, select several abundance tables (figure 2.3).

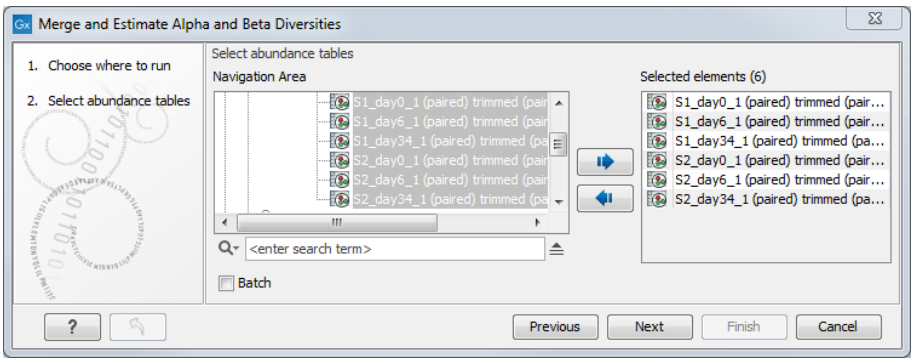


Figure 2.3: *Select abundance tables.*

In the second and third steps, you can choose parameters for the Alpha Diversity and for the Beta Diversity analyses. The parameters are described in section 7.4 and section 7.5.

The Merge and Estimate Alpha and Beta Diversities workflow generates the results seen in figure 2.4.

Figure 2.4: *Results from the Merge and Estimate Alpha and Beta Diversities workflow.*

Please refer to section 7.4 and section 7.5 to learn more about interpreting these results.

2.1.4 QC, Assemble and Bin Pangenomes

The **QC, Assemble and Bin Pangenomes** template workflow guides you through the key steps to analyze whole-genome shotgun metagenomic reads and assign them to clusters of sequences (bins) using the tools **Bin Pangenomes by Taxonomy** and **Bin Pangenomes by Sequence**. The inputs to the workflow are short reads belonging to a single metagenome sample (can be split in multiple sequence lists).

To run the workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **QC, Assemble and Bin Pangenomes** ( )

1. Specify the sample you would like to analyze.
2. Specify a **Trim adapter list** if your sequences contain adapters (see `Adapter trimming`).
3. Specify the minimum contig length, the type of de novo assembly you wish to perform (fast, or optimized for longer contigs), and whether you wish to perform scaffolding (figure 2.5).
4. For taxonomic binning of the assembled contigs, a Taxonomic Profiling Index must be provided (figure 2.6). Reference databases can be obtained by using the Download Curated Microbial Reference Database tool (section 16.1) or Download Custom Microbial Reference Database tool (section 16.2). For custom reference databases, indexes can be built with the Create Taxonomic Profiling Index tool (section 16.5).
5. In the next dialog (figure 2.7), configure the parameters for the Bin Pangenomes by Sequence tool. You can set the minimum contig length to exclude shorter contigs, as binning by sequence requires longer sequences for good results. You can also choose the maximum number of iterations that should be performed, and how to label singletons (bins with a single contig).
6. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).

The workflow produces the following outputs:

- **Reads binned by taxonomy**. Folder containing sequence list(s) of reads binned by **Bin Pangenomes by Taxonomy**.
- **Reads binned by sequence**. Folder containing sequence list(s) of reads binned by **Bin Pangenomes by Sequence**.

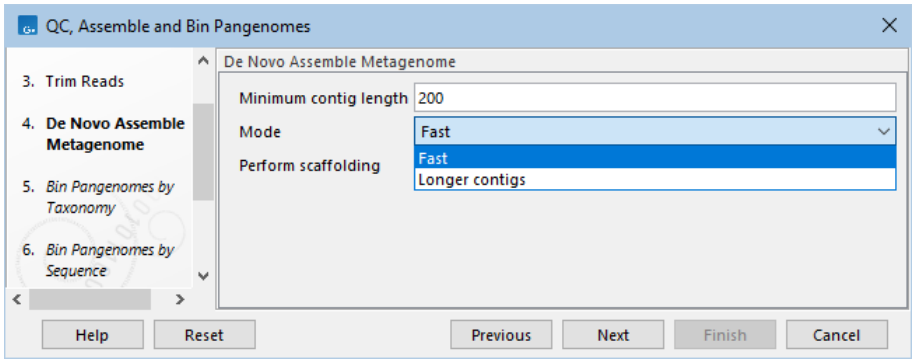


Figure 2.5: *Parameters for the De Novo Assemble Metagenome tool.*

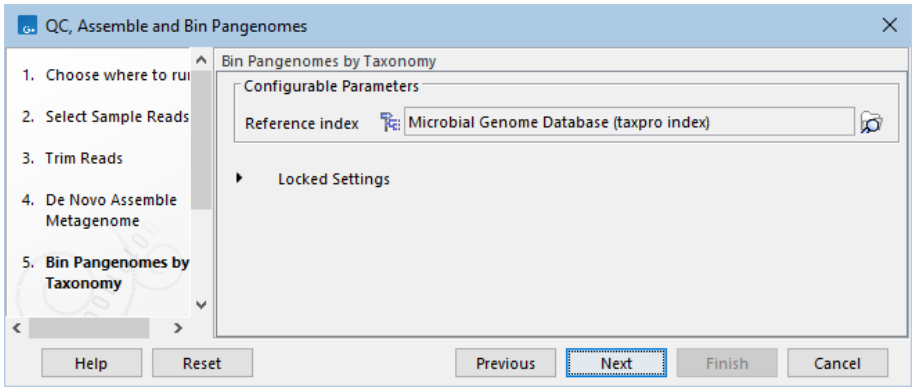


Figure 2.6: *Select the reference index for Bin Pangenomes by Taxonomy.*

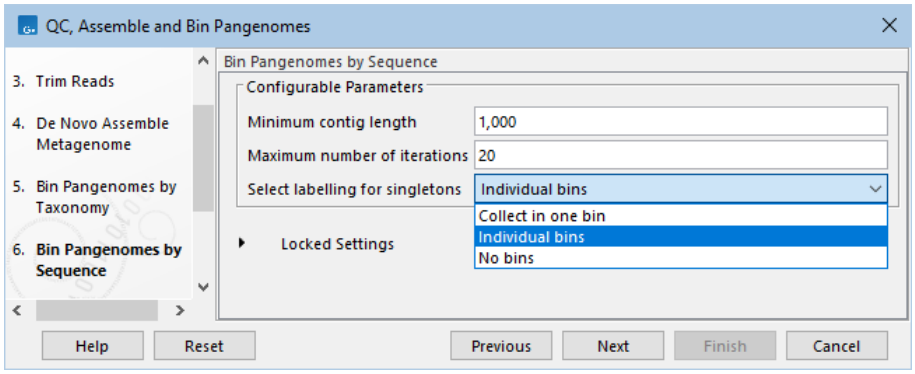


Figure 2.7: *Configure the Bin Pangenomes by Sequence.*

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
- **Binned contigs**. A combined sequence list of all the contigs binned by either taxonomy or sequence.
- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

The Sample report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

Additionally, you will find the "De novo assemble metagenome report" in the "QC & Reports" subfolder. For a detailed description, see (section 4.2).

Individual bins can be extracted from the sequence and contig lists by filtering by the bin label in the "Assembly_ID" column, either manually in the table view of the sequence list or by using `Filter on Custom Criteria` or `Split Sequence List`. Contigs can be used for downstream analysis such as reference-based assembly (or re-assembly), functional analysis, typing etc.

2.2 Amplicon-Based Analysis template workflows

2.2.1 Data QC and OTU Clustering

The **Data QC and OTU Clustering** workflow is meant for amplicon sequencing data. It trims reads and performs either reference-based or de novo OTU clustering. The resulting abundance table can optionally be filtered. The workflow additionally runs **QC for Sequencing Reads**, which can be used to assess the quality of the raw reads.

Filter Samples Based on Number of Reads filters samples with fewer than 100 reads. If multiple samples are used for the input, samples that have fewer than half of the median number of reads will be excluded.

This template workflow is available from:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Data QC and OTU Clustering** ( )

1. Specify the sample(s) or folder(s) of samples you would like to analyze.
2. Specify a **Trim adapter list** if your sequences contain adapters (see `Adapter trimming`).
3. Choose whether to run the de novo or reference-based OTU clustering and set the available similarity parameters. If selecting *Reference based OTU clustering*, choose whether to allow creation of new OTUs and provide an OTU database. Reference databases can be downloaded using **Download Amplicon-Based Reference Database** (see section 15.1).
4. Various options and filters can be set for refining the abundance table after clustering (see section 7.2). Note that if *De novo OTU clustering* was chosen in the previous step, then the Aggregation level must be set to "Do not aggregate".
5. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).

The workflow produces the following outputs:

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
- **OTU abundance table**. An abundance table containing all the samples input into the workflow, and refined according to the parameters set in the **Refine Abundance Table** step. Rows in the table will have taxonomies if reference-based OTU clustering was chosen.

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

The Sample report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

2.2.2 Detect Amplicon Sequence Variants and Assign Taxonomies

The Detect Amplicon Sequence Variants and Assign Taxonomies workflow processes reads from amplicon sequencing to yield a merged multi-sample (if applicable) ASV (Amplicon Sequence Variant) abundance table, and subsequently assigns taxonomies to the ASVs (amplicon sequence variants).

We recommend making preliminary evaluations of the read lengths and qualities, to decide on parameter settings like read length. This can be done by running a single sample through the workflow, and taking a look at the resulting trim report section *Read length before / after trimming*.

Launching the workflow

The **Detect Amplicon Sequence Variants and Assign Taxonomies** template workflow is available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Detect Amplicon Sequence Variants and Assign Taxonomies** ( )

Launch the workflow and step through the wizard.

1. Select the sequence list(s) containing the reads to process and click on **Next**.
2. Select a *Taxonomic Profiling Index* and click on **Next**.
3. The 'Configure batching' and 'Batch overview' steps can be left as is, or configured as described in Launching workflows individually and in batches.
4. Select a *Trim Adapter List* if relevant for your application. The *Trim Adapter List* should correspond to the adapters used for sequencing. If no input is provided, the tool will skip the adapter trimming step. Click on **Next**.
5. Choose the trim length to use for **Detect Amplicon Sequence Variants** and decide whether to remove chimeras by toggling the *Remove chimeras* box (figure 2.8). Click on **Next**.

   The optimal read length setting will depend on the length of your reads after trimming. We recommend that you have a look at the trim report section *Read length before / after trimming* if you are unsure about what value to set.
6. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see Create Sample Report).

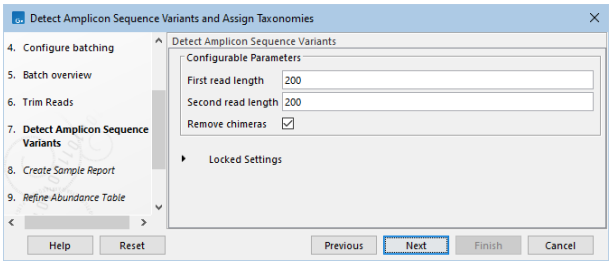


Figure 2.8: *Wizard step for selecting read trim length and whether to remove chimeras in the Detect Amplicon Sequence Variants tool.*

7. Various options and filters can be set for refining the merged abundance table (see section 7.2).

8. Finally, select a location to save outputs to and click on **Finish**.

Workflow outputs

The batch-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.
- **Analysis results**. Folder containing results output during analysis.
  - **ASV table**. ASV abundance table with abundances for each detected ASV.
  - **ASV sequence list**. Sequence list containing the detected ASVs.
- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Assign taxonomies report**. Report output by **Assign Taxonomies to Sequences in Abundance Table** (section 7.3). Is also in the Combined report.
- **Refine abundance table report**. Report output by **Refine Abundance Table** (section 7.2.2). Is also in the Combined report.
- **Combined report**. Combined report of all sample reports and the Assign taxonomies and Refine abundance table reports. The combined report contains all quality control information and analysis results. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.

- **Merged and refined abundance table with taxonomies**. Abundance table containing abundance results with taxonomy from all samples input into the workflow. If parameters were set in the **Refine Abundance Table** step, the abundance table will have been refined accordingly.

The Combined report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

2.2.3 Estimate Alpha and Beta Diversities

The **Estimate Alpha and Beta Diversities** template workflow takes an abundance table with sequences as input. If your data is amplicon sequencing data use this workflow. Otherwise use **Merge and Estimate Alpha and Beta Diversities** (section 2.1.3).

Remember to add metadata to the abundance table before starting the workflow. Adding metadata can be done very early on, by importing metadata and associating reads to it before generating an abundance table. The metadata will propagate to the abundance table automatically. When working with reads that were not associated with metadata in the first place, it is always possible to add metadata to an already existing abundance table with the tool **Add Metadata to Abundance Table** (section 7.9).

The workflow is available at:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Metagenomics** () | **Amplicon-Based Analysis** () | **Estimate Alpha and Beta Diversities** ()

The first tool of the workflow is **Refine Abundance Table** which filters features with less than 10 combined abundance. The output is a reduced abundance table that will be used as input for downstream analysis:

- **Align OTUs using MUSCLE**, a tool that will produce an alignment used as input for `Maximum Likelihood Phylogeny`, which will in turn output a phylogenetic tree also used as input in the following two tools.
  - **Alpha diversity**.
  - **Beta diversity**.

Running this workflow will therefore give the following outputs: a phylogenetic tree of the sequences, a diversity plot for the alpha diversity and a PCoA plot for the beta diversity.

2.3 Typing and Epidemiology template workflows

The Typing and Epidemiology template workflows are found at:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Typing and Epidemiology** ()

2.3.1 Compare Variants Across Samples

Compare Variants Across Samples can be used to compare samples originating from strains or species sharing a common reference. Input should be sequence lists of trimmed reads for which

host reads have been removed, e.g. using **Taxonomic Profiling**, see section 6.4.

As the workflow removes duplicate mapped reads, amplicon data is not recommended as input. However, the workflow can be modified to work on amplicon data by opening a copy of the workflow, removing the **Remove Duplicate Mapped Reads** tool and saving the modified workflow.

To run the Compare Variants Across Samples workflow, go to

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Typing and Epidemiology** () | **Compare Variants Across Samples** ()

- Select two or more samples as input.
- Select the reference track to use. The reference should match all the samples selected.
- Select a CDS track associated with the reference.
- Define batch units. For details, see `Running part of a workflow multiple times`.
- Check that batching is as intended.
- In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).
- In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Choose to save the results (we recommend to create a new folder for it) and click **Finish**.

The output will be saved in the location you chose.

The batch-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.
- **Track list**. Collection of all the tracks in the "Tracks" folder, and the input Reference and CDS tracks.
- **Tracks**. Folder containing all tracks output during analysis.
  - **Annotated variant track**. Output from the **Low Frequency Variant Detection** tool after coverage and quality filtering. Note: Multiple variant track files from monoploid data that are based on the same reference genome can be exported to a single VCF file using the Multi-VCF exporter.
  - **Amino acid track**. Amino acid track including amino acid changes resulting from the called variants.
  - **Read mapping**. Mapping of the reads to the specified reference. For increased sensitivity, duplicate mapped reads are removed before local realignment.

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Combined report**. Combined report of all sample reports. The combined report contains all quality control information and analysis results. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.
- **Variant track list for all samples**. The track combines the variant tracks for all analyzed samples.
- **SNP tree report**. Summarizes the applied filtering settings in the **Create SNP tree** tool, as well as a summary of ignored positions attributed to the different read mappings.
- **SNP Matrix**. A matrix containing the pairwise number of SNP differences between all pairs of samples included in the analysis.
- **SNP tree**. The output tree built from the SNPs called in all samples. A number of different visualizations are available, see section 9.1.2.

The Combined report should be inspected in order to determine whether the quality of the sequencing reads and the analysis results are acceptable.

For more information on the Create SNP Tree tool, see section 9.1.

2.3.2 Create MLST Scheme with Sequence Types

The **Create MLST Scheme with Sequence Types** workflow creates a MLST scheme from references and adds sequence types by typing references and adding the results to the scheme.

To run the **Create MLST Scheme with Sequence Types** template workflow, go to

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Typing and Epidemiology** () | **Create MLST Scheme with Sequence Types** ()

You can select one or more assemblies as input (figure 2.9). At least one of the assemblies must be annotated with CDS regions.

In "Create MLST scheme" dialog (figure 2.10), the settings for the scheme creation can be viewed and changed.

The parameters that can be set are:

- **MLST Type:** specifies the fraction of assemblies a locus must be present in to be included in the scheme. Options are: Core genome (corresponding to a fraction of 0.9), Whole genome (corresponding to a fraction of 0.1) or custom fraction.
- **Genetic code:** specifies the genetic code matching the input assemblies for a codon check.

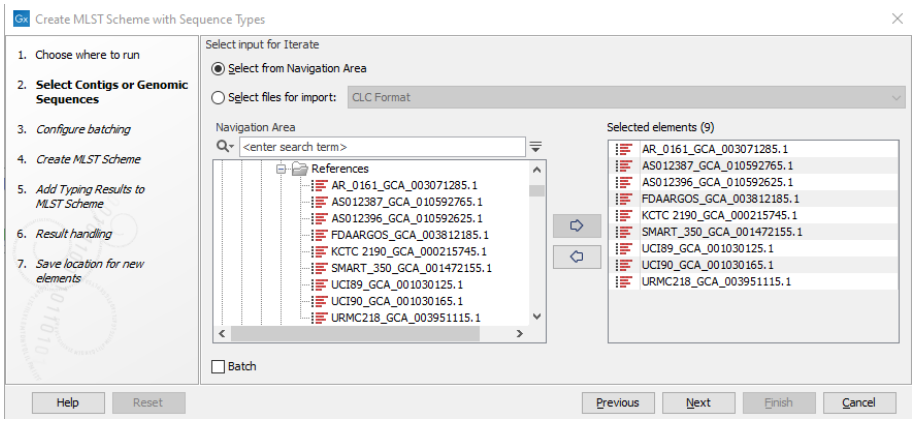


Figure 2.9: *Select the high-quality references serving as the basis for the scheme*

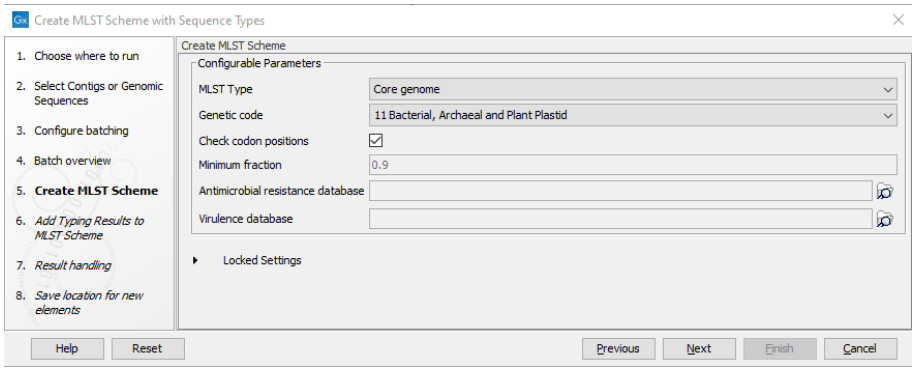


Figure 2.10: *Parameters for creating the initial scheme*

- **Check codon positions:** if enabled, loci failing the specified codon check will not appear in the scheme. This should be disabled when working with organisms containing spliced genes.
- **Minimum fraction:** specifies the required fraction if custom fraction was selected in **MLST Type**.
- **Antimicrobial resistance database:** optional setting for specifying an antimicrobial resistance database to use for annotating loci in the scheme.
- **Virulence database:** optional setting for specifying a virulence database to use for annotating loci in the scheme.

In "Add Typing Results to MLST scheme" dialog (figure 2.11), sequence types will be added to the scheme. In addition, the following parameters can be specified:

- **Allow incomplete novel alleles:** whether only complete novel alleles (containing both start and stop codon) should be allowed. If incomplete novel alleles are not allowed, a sequence type with incomplete alleles for a locus will be added with missing alleles for that locus. If **Check codon positions** has been disabled (see figure 2.10), all alleles will be incomplete and consequently it will be necessary to allow adding incomplete alleles.
- **Comparing a known to a missing allele:** how to treat missing alleles when comparing a locus for a pair of sequence types.

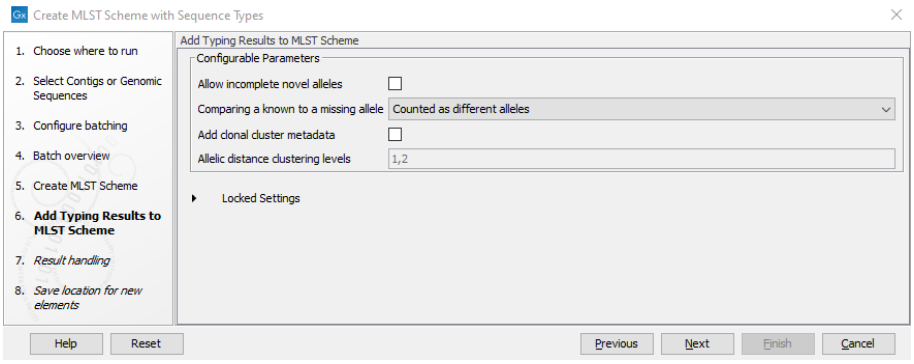


Figure 2.11: *Add Typing Results settings*

- **Add clonal cluster metadata:** if selected, clonal cluster data will be added as metadata.
- **Allele distance clustering levels:** if clonal cluster data is added, specifies the allelic distance thresholds for adding clustering information.

In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Saving the output will generate the files shown in (figure 2.12) and optionally, a workflow result metadata table.

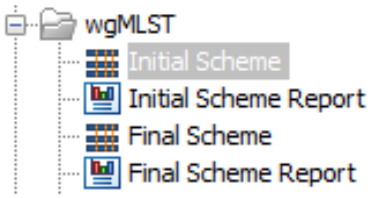


Figure 2.12: *The output from Create MLST Scheme with Sequence Types*

- **Initial Scheme Report:** the report from **Create MLST Scheme** tool.
- **Initial Scheme:** an empty scheme containing only loci.
- **Final Scheme Report:** the report from **Add Typing Results to MLST Scheme** tool.
- **Final Scheme:** the complete MLST Scheme containing loci and sequence types.

For more information on the tools and MLST schemes, see section 10.

2.3.3 Map to Specified Reference

Once analysis has been performed using the **Type Among Multiple Species** workflow, the best matching reference is listed in the Result Metadata table (figure 2.13, see column Best match).

If all your samples share the same common reference, you can proceed to additional analyses without delay.

However there are cases where your samples have different Best match reference for a particular MLST scheme. And because creating a SNP Tree require a single common reference, you will

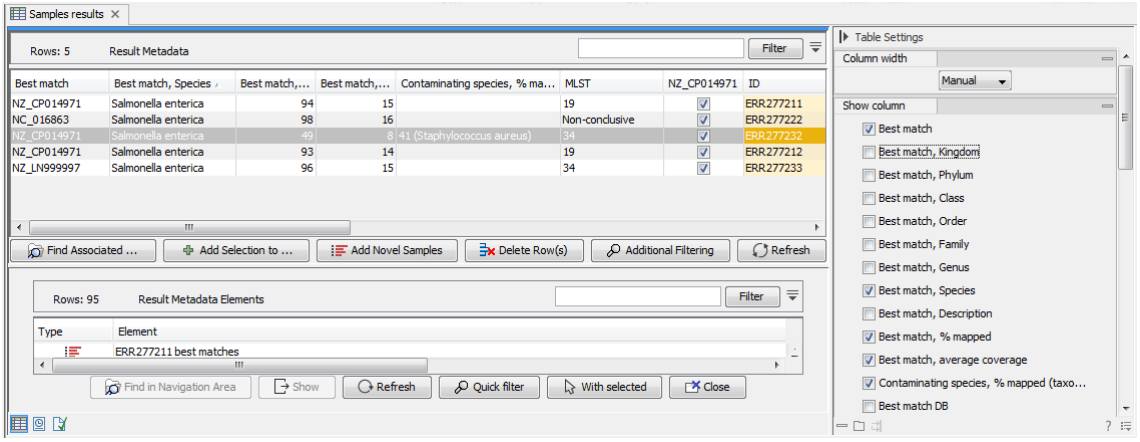


Figure 2.13: *Best match references are listed for each row in the Result Metadata Table.*

need to identify the best matching common reference for all your samples using a K-mer Tree, as well as subsequently re-map your samples to this common reference.

If you already know the common reference for the sample you want to use to create a SNP tree, you can directly specify that reference in the re-map workflow. Otherwise, finding a common reference is described in more details in section 8.1.1.

In short, **to identify a common reference across multiple clades within the Result Metadata Table:**

- **Select** samples to which a common best matching references should be identified.
- **Click** on the **Find Associated Data** ( ) button to find their associated Metadata Elements.
- **Click** on the **Quick Filtering** ( ) button and select the option **Filter for K-mer Tree** to find Metadata Elements with the Role = Trimmed Reads.
- **Select** the relevant Metadata Element files.
- **Click** on the **With selected** ( ) button.
- **Select** the **Create K-mer Tree** action and follow the wizard as described in section 9.2.

The common reference, chosen as sharing the closest common ancestor with the clade of isolates under study in the k-mer tree, is subsequently used as a reference for the **Map to Specified Reference** workflow (figure 2.14) that will perform a re-mapping of the reads followed by variant calling.

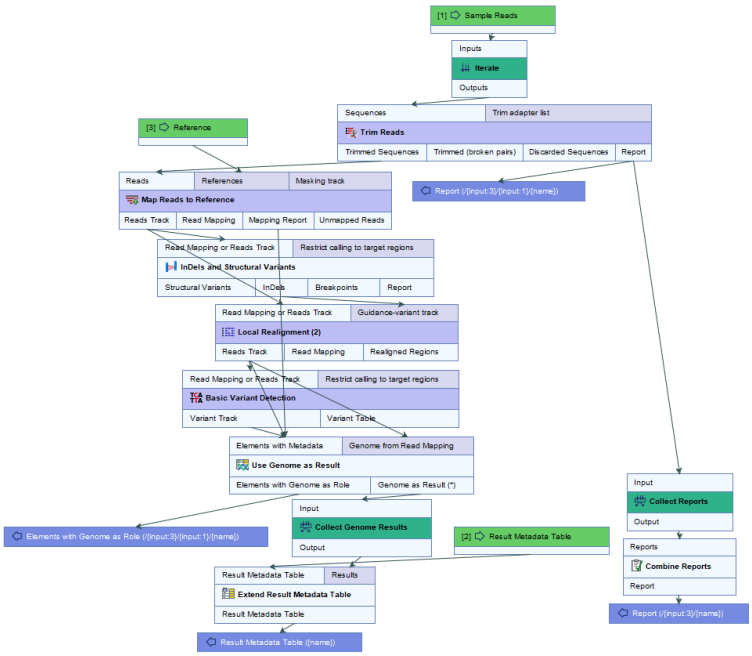


Figure 2.14: *Overview of the template Map to Specified Reference workflow.*

How to run the Map to Specified Reference workflow

The **Map to Specified Reference** template workflow is intended for read mapping and variant calling of the samples against a common reference. To run this workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Typing and Epidemiology** ( ) | **Map to Specified Reference** ( )

1. Specify the sample(s) or folder(s) of samples you would like to type (figure 2.15) and click **Next**. Remember that if you select several items, they will be run as batch units.

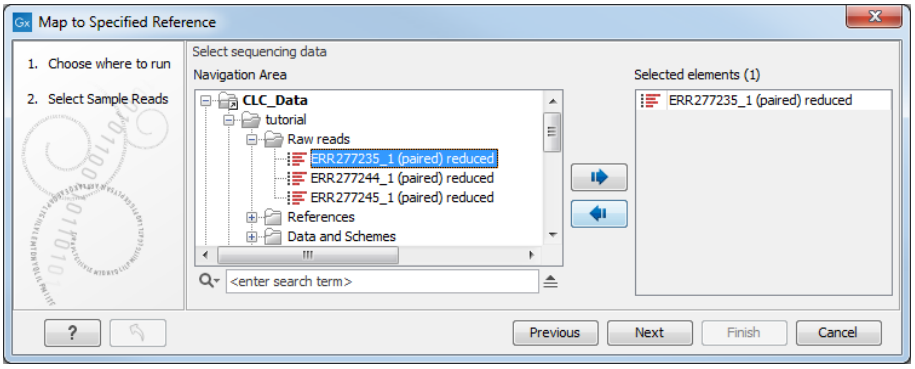


Figure 2.15: *Select the reads from the sample(s) you would like to type.*

2. Specify the **Result Metadata Table** you want to use (figure 2.16) and click **Next**.

3. Select the reference you obtained from the previous workflows - provided that it was the same reference for all the samples you want to re-map - or determined earlier from your K-mer tree if the samples you want to re-map had different best match references. Click **Next**.

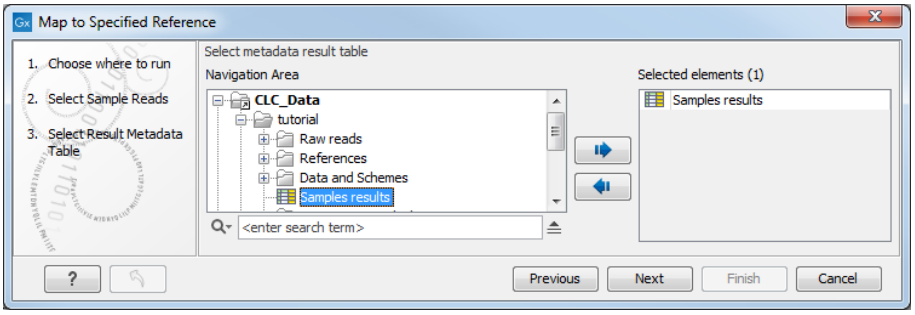


Figure 2.16: *Select the metadata table you would like to use.*

4. Define batch units using organisation of input data to create one run per input or use a metadata table to define batch units. Click **Next**.

5. The next wizard window gives you an overview of the samples present in the selected folder(s). Choose which of these samples you want to analyze in case you are not interested in analyzing all the samples from a particular folder (figure 2.17).

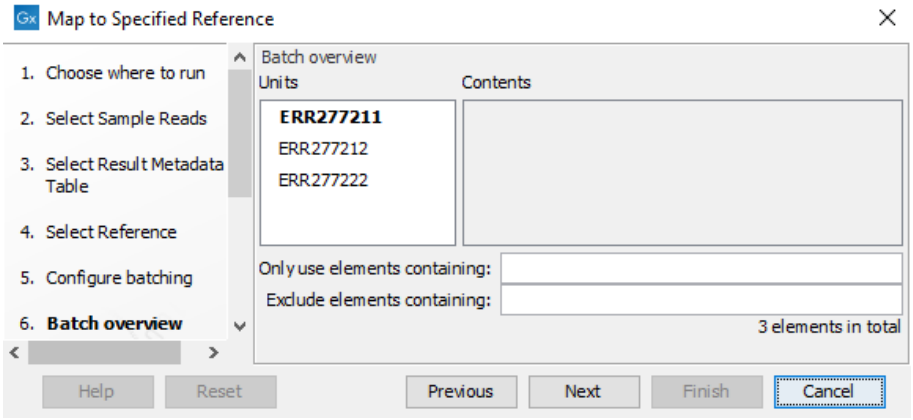


Figure 2.17: *Choose which of the samples present in the selected folder(s) you want to analyze.*

6. You can specify a **trim adapter list** and set up parameters if you would like to trim your sequences from adapters. Specifying a trim adapter list is optional but recommended to ensure the highest quality data for your typing analysis (figure 2.18). For more information, see `Trim adapter list`.

   The parameters that can be set are:

   - **Ambiguous trim**: if checked, this option trims the sequence ends based on the presence of ambiguous nucleotides (typically N).
   - **Ambiguous limit**: defines the maximal number of ambiguous nucleotides allowed in the sequence after trimming.
   - **Quality trim**: if checked, and if the sequence files contain quality scores from a base-caller algorithm, this information can be used for trimming sequence ends.
   - **Quality limit**: defines the minimal value of the Phred score for which bases will not be trimmed.

   Click **Next**.

7. Specify the parameters for the **Maps Reads to Reference** tool (figure 2.19).

   The parameters that can be set are:

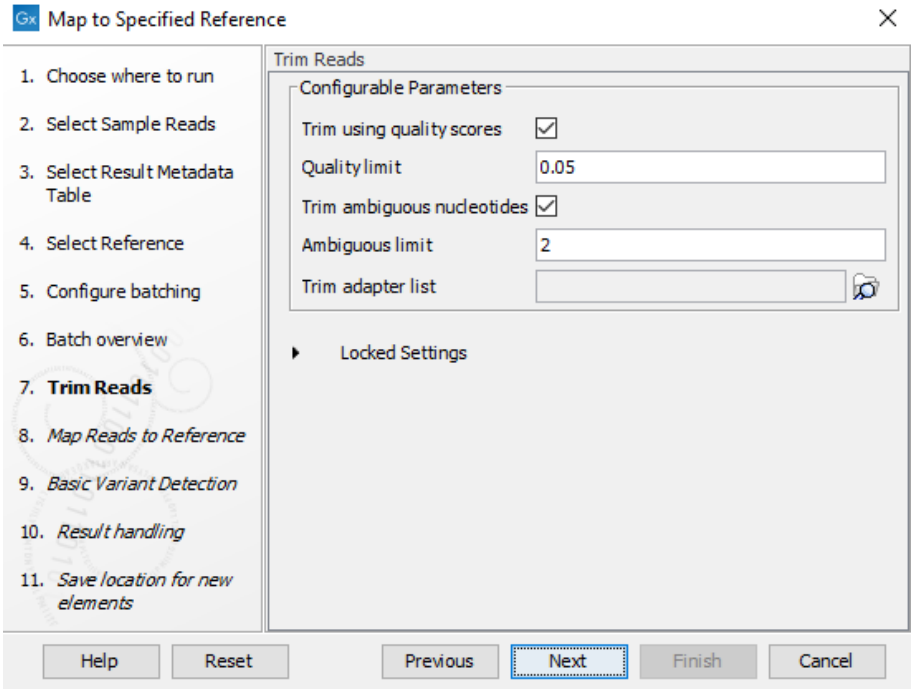


Figure 2.18: *You can choose to trim adapter sequences from your sequencing reads.*

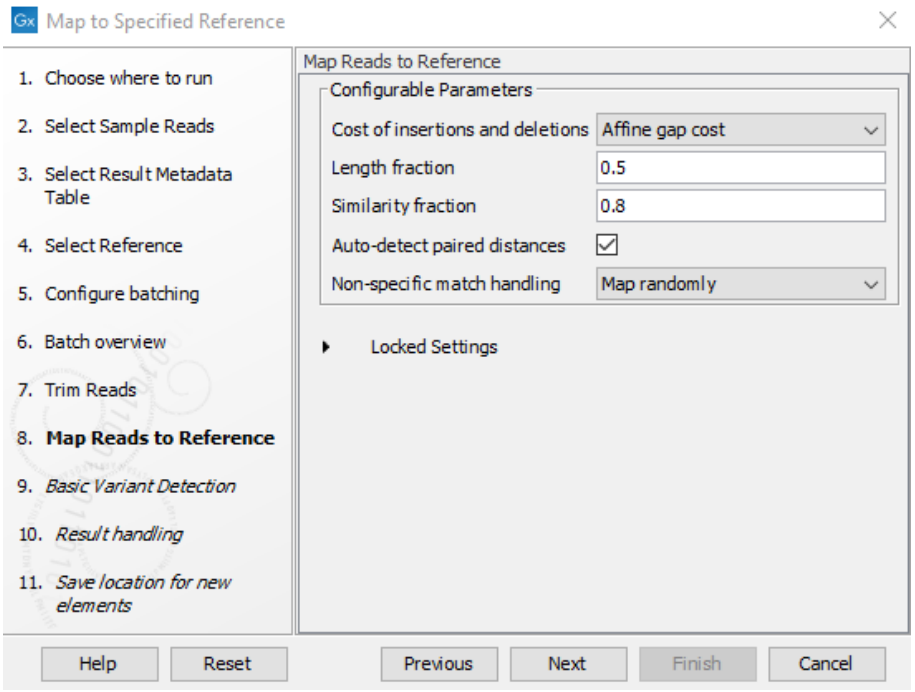


Figure 2.19: *Specify the parameters for the Maps Reads to Reference tool.*

- **Cost of insertion and deletions**: You can choose affine or linear gap cost.
- **Length fraction**: The minimum percentage of the total alignment length that must match the reference sequence at the selected similarity fraction. A fraction of 0.5 means that at least half of the alignment must match the reference sequence before the read is included in the mapping (if the similarity fraction is set to 1). **Note** that the minimal seed (word) size for read mapping is 15 bp, so reads shorter than this will

not be mapped.

- **Similarity fraction**: The minimum percentage identity between the aligned region of the read and the reference sequence. For example, if the identity should be at least 80% for the read to be included in the mapping, set this value to 0.8. **Note** that the similarity fraction relates to the length fraction, i.e., when the length fraction is set to 50% then at least 50% of the alignment must have at least 80% identity
- **Auto-detect paired sequences**: This will determine the paired distance (insert size) of paired data sets. If several paired sequence lists are used as input, a separate calculation is done for each one to allow for different libraries in the same run.
- **Non-specific match handling**: You can choose from the drop down menu whether you would like to ignore or map randomly the non specific matches.

Click **Next**.

8. Specify the parameters for the **Basic Variant Detection** tool (figure 2.20) before clicking **Next**.

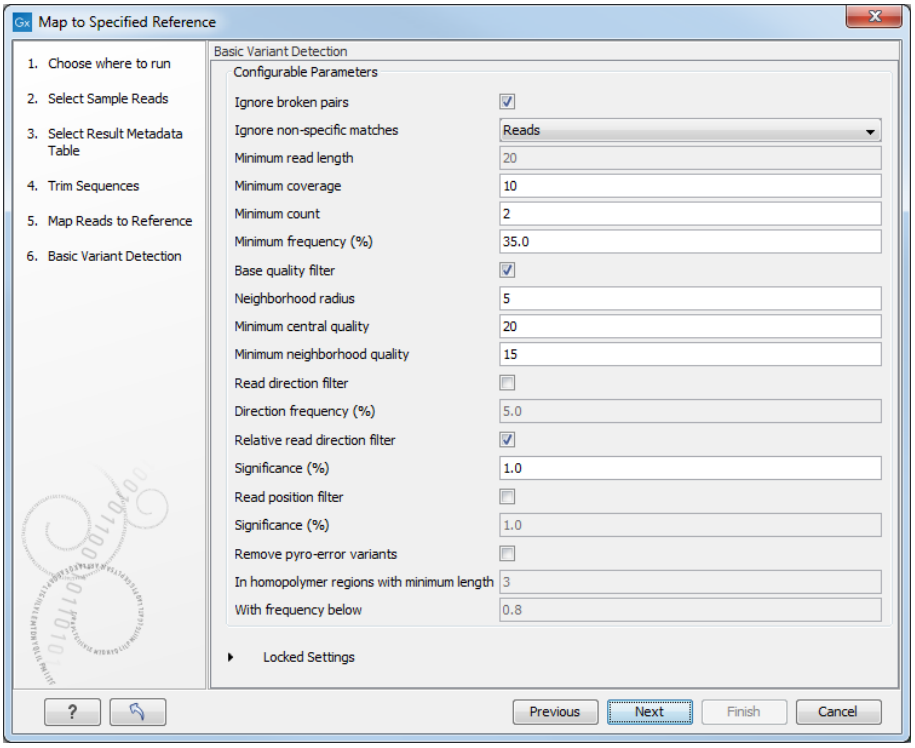


Figure 2.20: *Specify the parameters to be used for the Basic Variant Detection tool.*

The parameters that can be set are:

- **Ignore broken pairs**: You can choose to ignore broken pairs by clicking this option.
- **Ignore non-specific matches**: You can choose to ignore non-specific matches between reads, regions or to not ignore them at all.
- **Minimum read length**: Only variants in reads longer than this size are called.
- **Minimum coverage**: Only variants in regions covered by at least this many reads are called.
- **Minimum count**: Only variants that are present in at least this many reads are called.

- **Minimum frequency %**: Only variants that are present at least at the specified frequency (calculated as count/coverage) are called.
- **Base quality filter**: The base quality filter can be used to ignore the reads whose nucleotide at the potential variant position is of dubious quality.
- **Neighborhood radius**: Determine how far away from the current variant the quality assessment should extend.
- **Minimum central quality**: Reads whose central base has a quality below the specified value will be ignored. This parameter does not apply to deletions since there is no "central base" in these cases.
- **Minimum neighborhood quality**: Reads for which the minimum quality of the bases is below the specified value will be ignored.
- **Read direction filters**: The read direction filter removes variants that are almost exclusively present in either forward or reverse reads.
- **Direction frequency %**: Variants that are not supported by at least this frequency of reads from each direction are removed.
- **Relative read direction filter**: The relative read direction filter attempts to do the same thing as the Read direction filter, but does this in a statistical, rather than absolute, sense: it tests whether the distribution among forward and reverse reads of the variant carrying reads is different from that of the total set of reads covering the site. The statistical, rather than absolute, approach makes the filter less stringent.
- **Significance %**: Variants whose read direction distribution is significantly different from the expected with a test at this level, are removed. The lower you set the significance cut-off, the fewer variants will be filtered out.
- **Read position filter**: It removes variants that are located differently in the reads carrying it than would be expected given the general location of the reads covering the variant site.
- **Significance %**: Variants whose read position distribution is significantly different from the expected with a test at this level, are removed. The lower you set the significance cut-off, the fewer variants will be filtered out.
- **Remove pyro-error variants**: This filter can be used to remove insertions and deletions in the reads that are likely to be due to pyro-like errors in homopolymer regions. There are two parameters that must be specified for this filter:
- **In homopolymer regions with minimum length**: Only insertion or deletion variants in homopolymer regions of at least this length will be removed.
- **With frequency below**: Only insertion or deletion variants whose frequency (ignoring all non-reference and non-homopolymer variant reads) is lower than this threshold will be removed.

9. In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Choose to save the results and click on the button labeled **Finish**.

Four outputs are generated per input sample (figure 2.21):

- **Mapping Summary report**: Summary report about the mapping process, see `Summary mapping report`.

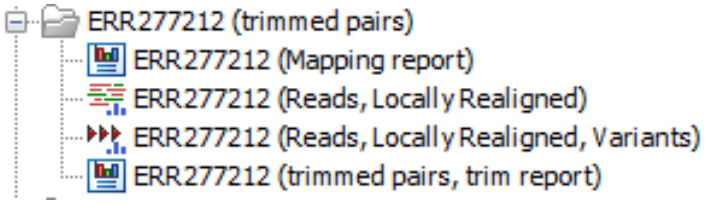


Figure 2.21: *Output files from the Map to Specified Reference workflow.*

- **Trim report**: Summary report for the trimming, see `Trim output`.
- **Reads Track**: Output from the **Local Realignment** tool
- **Variant Track**: Output from the **Basic Variant Detection** tool. Note: Multiple variant track files from monoploid data that are based on the same reference genome can be exported to a single VCF file using the Multi-VCF exporter.

You now have the data necessary to create a SNP tree for your samples as explained in section 9.1.

The tool will output, among other files, variant tracks. Note: Multiple variant track files from monoploid data that are based on the same reference genome can be exported to a single VCF file using the Multi-VCF exporter.

2.3.4 Type Among Multiple Species

The **Type Among Multiple Species** workflow is designed for typing a sample among multiple predefined species.

It allows for identification of the closest matching reference species among the specified reference list(s) that may represent multiple species. The workflow identifies the associated MLST scheme and type, determines variants found when mapping the sample data against the identified best matching reference, and finds occurring resistance genes if they match genes within the specified resistance database.

The workflow also automatically associates the analysis results to the specified Result Metadata Table. For details about searching and quick filtering among the sample metadata and generated analysis result data (see section 20.2.2).

Preliminary steps to run the Type Among Multiple Species workflow

Before starting the workflow,

- Download microbial genomes using either **Download Custom Microbial Reference Database** (section 16.2), the prokaryotic databases from **Download Curated Microbial Reference Database** (section 16.1), or using **Download Pathogen Reference Database** (section 16.3).
- Download the MLST schemes using the **Download MLST Scheme** tool (see section 14.2).
- Download the database for the **Find Resistance with Nucleotide Database** tool using the **Download Resistance Database** tool (see section 18.1).
- Create a New Result Metadata table using the **Create Result Metadata Table** tool (see section 20.2.1).

How to run the Type Among Multiple Species workflow

To run the workflow, go to:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Typing and Epidemiology** ( ) | **Type Among Multiple Species** ( )

1. Specify the sample(s) or folder(s) of samples you would like to type and click **Next**. Remember that if you select several items, they will be run as batch units.
2. Specify the *Result Metadata Table* you want to add your results to and click **Next**.
3. Define batch units. For details, see `Running part of a workflow multiple times`.
4. Check that batching is as intended.
5. If your reads contain adapters, add an appropriate Trim adapter list. Click **Next**.
6. Choose the species-specific *References* to be used by the **Find Best Matches using K-mer Spectra** tool (figure 2.22). Click **Next**.

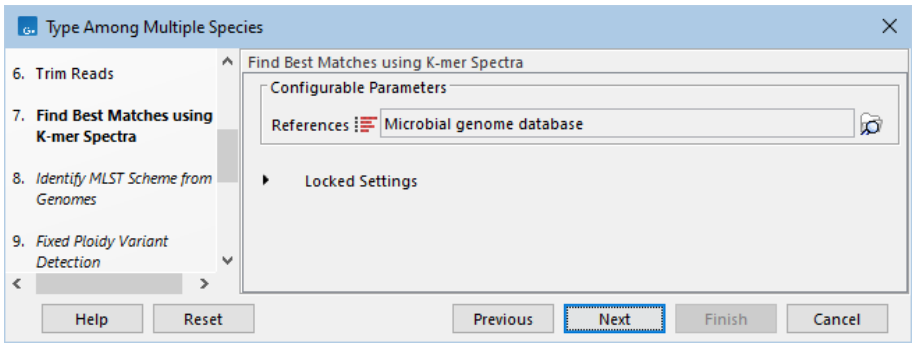


Figure 2.22: *Specify the references for the Find Best Matches using K-mer Spectra tool.*

7. Specify the *MLST Schemes* to be used for the **Identify MLST Scheme from Genomes** tool so they correspond to the chosen reference list(s) (figure 2.23).

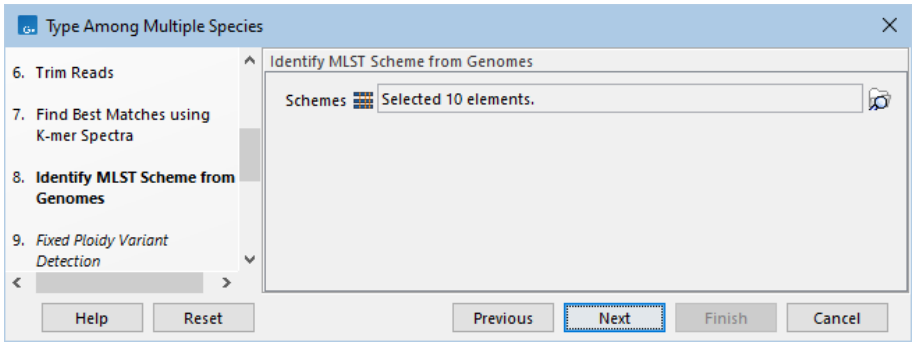


Figure 2.23: *Specify the schemes that best describe your sample(s).*

8. Specify the parameters for the **Fixed Ploidy Variant Detection** tool (figure 2.24) before clicking **Next**. For detailed information about all the filters, see `Fixed Ploidy Variant Detection` and `Variant Detection – filters`.

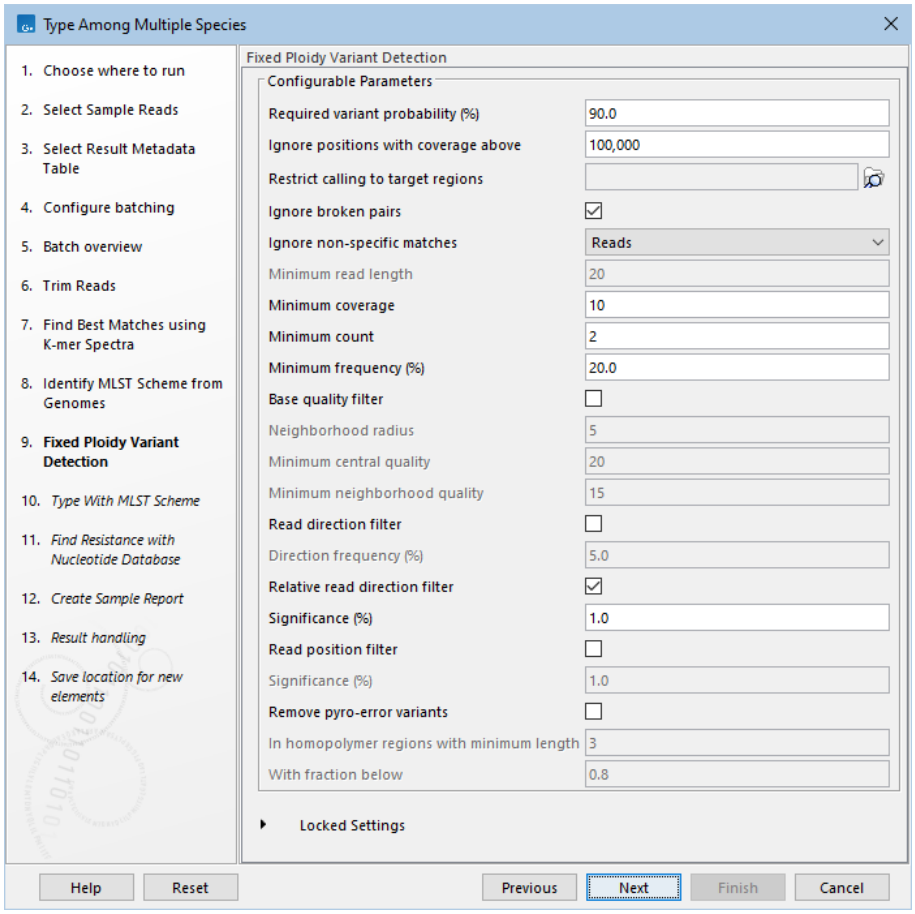


Figure 2.24: *Specify the parameters to be used for the Fixed Ploidy Variant Detection tool.*

9. Specify the parameters for the **Type with MLST Scheme** tool (figure 2.25).

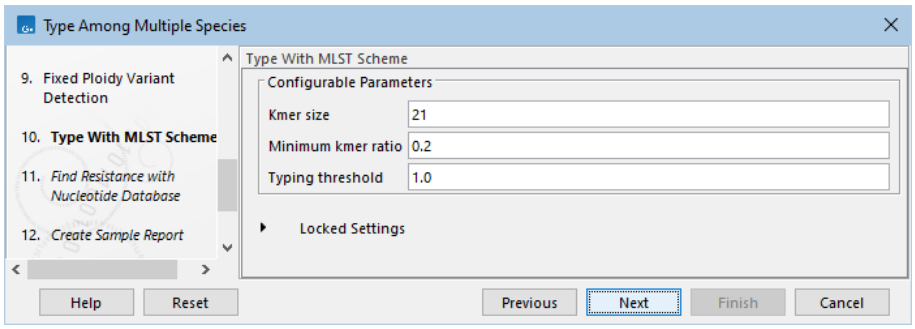


Figure 2.25: *Specify the parameters for MLST typing.*

The parameters that can be set are:

- **Kmer size**. Determines the number of nucleotides in the kmer - raising this setting might increase specificity at the cost of some sensitivity.
- **Minimum kmer ratio**. The minimum kmer ratio of the least occurring kmer and the average kmer hit count. If an allele scores higher than this threshold it is classified as a high-confidence call.

- **Typing threshold**. The typing threshold determines how many of the kmers in a sequences type need be identified before a typing is considered conclusive. The default setting of 1.0 means that all kmers in all alleles must be matched. Lowering the setting to 0.99 would mean that on avergae 99% of all kmers in all the alleles of a given sequence type must be detected before the sequence type is considered conclusive.

Click **Next**.

10. Specify the *Resistance Database* (figure 2.26) and set the parameters for the **Find Resistance with Nucleotide Database** tool.

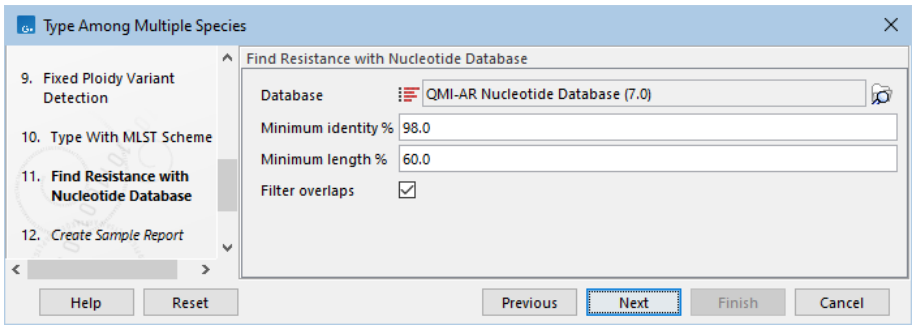


Figure 2.26: *Specify the resistance database to be used for the Find Resistance with Nucleotide Database tool.*

The parameters that can be set are:

- **Minimum identity %**. The threshold for the minimum percentage of nucleotides that are identical between the best matching resistance gene in the database and the corresponding sequence in the genome.
- **Minimum length %**. The percentage of the resistance gene length that a sequence must overlap to count as a hit for that gene.
- **Filter overlaps**. Extra filtering of results per contig, where one hit is contained by the other with a preference for the hit with the higher number of aligned nucleotides (length * identity).

Click **Next**.

11. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).

12. In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Choose to save the results (we recommend to create a new folder for it) and click **Finish**.

The output will be saved in the location you chose, and eligible results will also be added automatically to the Metadata Result table.

The batch-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be

found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

- **Resistance Results**. Folder containing results from de novo assembly and resistance calling.
  - **Assembled contigs list**. Contig list from the **De novo assembly** tool.
  - **Resistance table**. The result table from the **Find Resistance with Nucleotide Database** tool, reporting the found resistance.
- **Typing Results**. Folder containing results from typing and variant calling.
  - **Best match**. Reference sequence that matches best the data according to the **Find Best Matches using K-mer Spectra** tool.
  - **Best matches table**. Table containing the best matching sequence(s), a list of all (maximum 100) significantly matching references and the various statistical values applied.
  - **Sequence lists**. List(s) of the sequences that were successfully trimmed and mapped to the best reference.
  - **Best match read mapping**. Mapping of the reads using the Best Match as reference.
  - **Typing result**. Output from the **Type with MLST Scheme** tool, including information on kmer fractions, kmer hit counts, and alleles identified and called.
  - **Variant track**. Output from the **Fixed Ploidy Variant Detection** tool. Note: Multiple variant track files from monoploid data that are based on the same reference genome can be exported to a single VCF file using the Multi-VCF exporter.
- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Combined report**. Combined report of all sample reports. The combined report contains all quality control information and analysis results. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.
- **Results Metadata Table**. A table containing summary information of results for each sample analyzed and a quick way to find the associated files. In particular, the column "Best match, average coverage" can help when deciding whether a best match is significant, well covered and of good quality. This is especially useful when a sample has low quality but is not contaminated.

Through the Result Metadata Table, it is possible to filter among sample metadata and analysis results. By clicking **Find Associated Data** () and optionally performing additional filtering, it is possible to perform additional analyses on a selected subset directly from this Table, such as:

- Generation of SNP trees based on the same reference used for read mapping and variant detection (section 9.1).
- Generation of *K*-mer Trees for identification of the closest common reference across samples (section 9.2).
- Run validated workflows (workflows that are associated with a Result Metadata Table and saved in your Navigation Area).

2.3.5 Type a Known Species

The **Type a Known Species** workflow is designed for typing of samples representing a single known species. It identifies the associated MLST, determines variants found when mapping the sample data against the specified reference, and finds occurring resistance genes if they match genes within the specified resistance database.

Preliminary steps to run the Type a Known Species workflow

Before starting the workflow,

- Download microbial genomes using either **Download Custom Microbial Reference Database** (section 16.2), the prokaryotic databases from **Download Curated Microbial Reference Database** (section 16.1), or using **Download Pathogen Reference Database** (section 16.3).
- Download the MLST schemes using the **Download MLST Scheme** tool (see section 14.2).
- Download the database for the **Find Resistance with Nucleotide Database** tool using the **Download Resistance Database** tool (see section 18.1).
- Create a New Result Metadata table using the **Create Result Metadata Table** tool (see section 20.2.1).

How to run the Type a Known Species workflow

To run the workflow, go to:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Typing and Epidemiology** () | **Type a Known Species** ()

1. Specify the sample(s) or folder(s) of samples you would like to type and click **Next**. Remember that if you select several items, they will be run as batch units.
2. Specify the *Result Metadata Table* you want to add your results to and click **Next**.
3. Define batch units. For details, see `Running part of a workflow multiple times`.
4. Check that batching is as intended.
5. If your reads contain adapters, add an appropriate Trim adapter list. Click **Next**.
6. Choose the *Reference* for **Map Reads to Reference** (figure 2.27). Click **Next**.

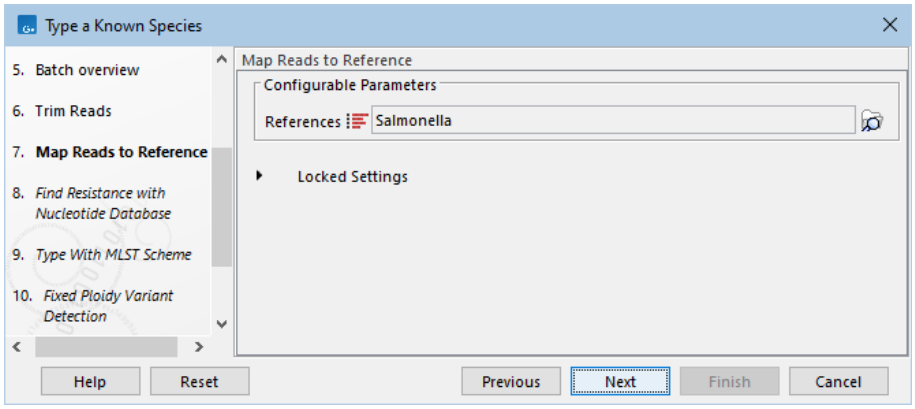


Figure 2.27: *Specify the reference for the Map Reads to Reference tool.*

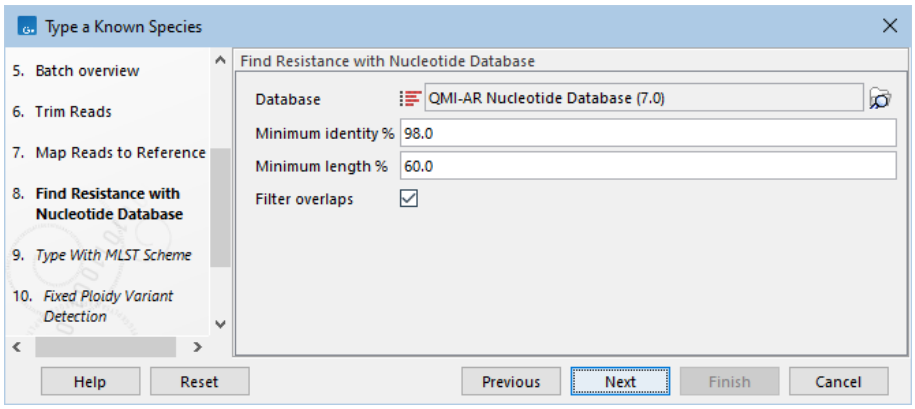


Figure 2.28: *Specify the resistance database to be used for the Find Resistance with Nucleotide Database tool.*

7. Specify the *Resistance Database* (figure 2.28) and set the parameters for the **Find Resistance with Nucleotide Database** tool.

   The parameters that can be set are:

   - **Minimum identity %**. The threshold for the minimum percentage of nucleotides that are identical between the best matching resistance gene in the database and the corresponding sequence in the genome.
   - **Minimum length %**. The percentage of the resistance gene length that a sequence must overlap to count as a hit for that gene.
   - **Filter overlaps**. Extra filtering of results per contig, where one hit is contained by the other with a preference for the hit with the higher number of aligned nucleotides (length * identity).

   Click **Next**.

8. Specify the *MLST Scheme* and set the parameters for the **Type with MLST Scheme** tool (figure 2.29).

   The parameters that can be set are:

   - **Kmer size**. Determines the number of nucleotides in the kmer - raising this setting might increase specificity at the cost of some sensitivity.

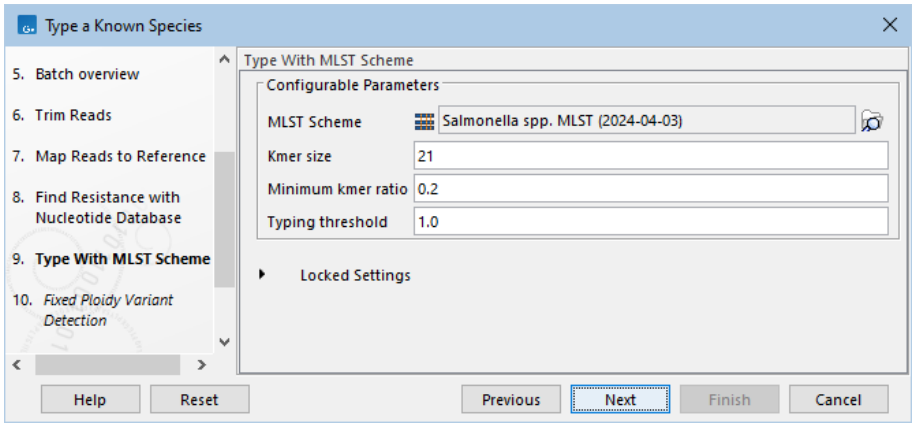


Figure 2.29: *Specify the parameters for MLST typing.*

- **Minimum kmer ratio**. The minimum kmer ratio of the least occurring kmer and the average kmer hit count. If an allele scores higher than this threshold it is classified as a high-confidence call.
- **Typing threshold**. The typing threshold determines how many of the kmers in a sequences type need be identified before a typing is considered conclusive. The default setting of 1.0 means that all kmers in all alleles must be matched. Lowering the setting to 0.99 would mean that on avergae 99% of all kmers in all the alleles of a given sequence type must be detected before the sequence type is considered conclusive.

Click **Next**.

9. Specify the parameters for the **Fixed Ploidy Variant Detection** tool (figure 2.30) before clicking **Next**. For detailed information about all the filters, see `Fixed Ploidy Variant Detection` and `Variant Detection - filters`.
10. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`).
11. In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Choose to save the results (we recommend to create a new folder for it) and click **Finish**.

The output will be saved in the location you chose, and eligible results will also be added automatically to the Metadata Result table.

The batch-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.
- **Resistance Results**. Folder containing results from de novo assembly and resistance calling.

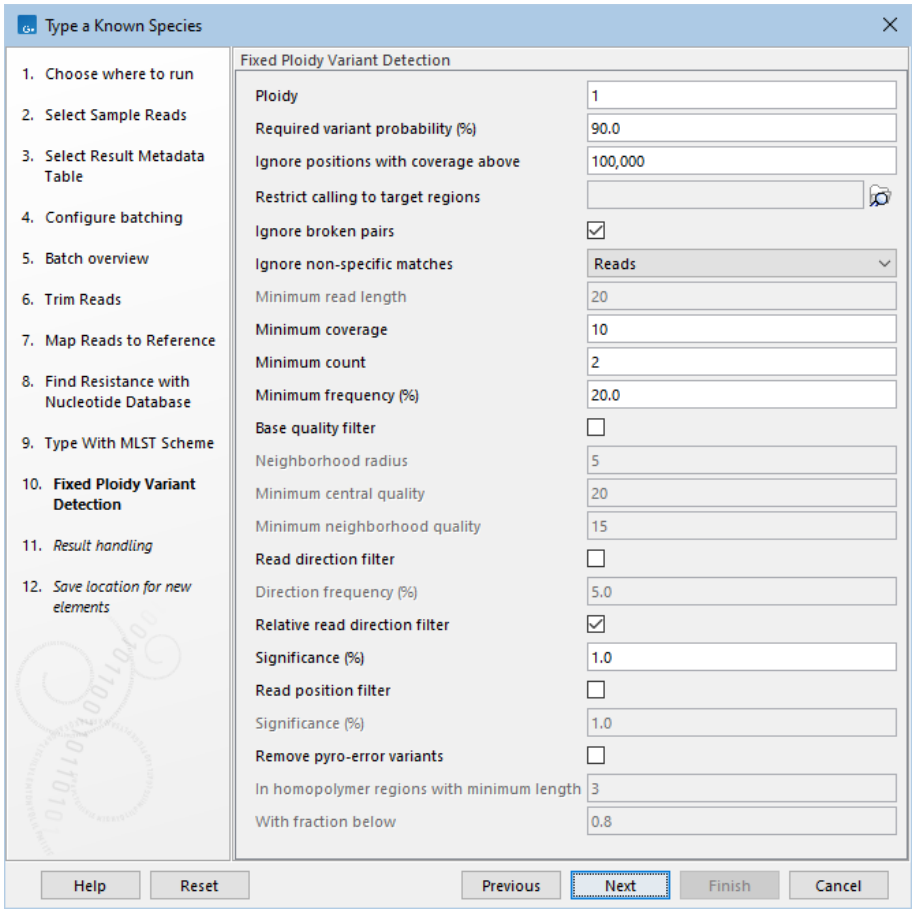


Figure 2.30: *Specify the parameters to be used for the Fixed Ploidy Variant Detection tool.*

  - **Assembled contigs list**. Contig list from the **De novo assembly** tool.
  - **Resistance table**. The result table from the **Find Resistance with Nucleotide Database** tool, reporting the found resistance.

- **Typing Results**. Folder containing results from typing and variant calling.
  - **Sequence lists**. List(s) of the sequences that were successfully trimmed and mapped to the best reference.
  - **Read mapping**. Mapping of the reads to the specified reference.
  - **Typing result**. Output from the **Type with MLST Scheme** tool, including information on kmer fractions, kmer hit counts, and alleles identified and called.
  - **Variant track**. Output from the **Fixed Ploidy Variant Detection** tool. Note: Multiple variant track files from monoploid data that are based on the same reference genome can be exported to a single VCF file using the Multi-VCF exporter.

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Combined report**. Combined report of all sample reports. The combined report contains all quality control information and analysis results. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.
- **Results Metadata Table**. A table containing summary information of results for each sample analyzed and a quick way to find the associated files.

Through the Result Metadata Table, it is possible to filter among sample metadata and analysis results. By clicking **Find Associated Data** () and optionally performing additional filtering, it is possible to perform additional analyses on a selected subset directly from this Table, such as:

- Generation of SNP trees based on the same reference used for read mapping and variant detection (section 9.1).
- Generation of K-mer Trees for identification of the closest common reference across samples (section 9.2).
- Run validated workflows (workflows that are associated with a Result Metadata Table and saved in your Navigation Area).

2.3.6 Type Influenza Strain

The **Type Influenza Strain** template workflow is designed for typing targeted sequencing Influenza virus samples. It combines the **Type with Consensus Refinement** (section 8.3) tool with a trimming step to determine the type and subtype of the Influenza virus. The workflow outputs a consensus sequence list, a list of the closest matching reference sequences, and annotation tracks (Genes and CDS) that are combined into a track list for easier visualization. Additionally, the workflow includes a `Low Frequency Variant Detection` step to detect potential mixed-infection variants, which can also be inspected in the same track list, both in nucleotide and amino acid formats.

Preliminary steps to run the Type a Known Species workflow

Before starting the workflow, download the Influenza Segment References data set using the `Reference Data Manager`.

How to run the Type Influenza Strain workflow

To run the workflow, go to:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Typing and Epidemiology** () | **Type Influenza Strain** ()

1. Specify the sample(s) or folder(s) of samples you would like to type and click **Next**.
2. Click **Next** to use the automatically selected **Influenza Segment References** data set.
3. Define batch units. For details, see `Running workflows in batch mode`.
4. Check that batching is as intended.
5. If your reads contain adapters, add an appropriate Trim adapter list. Click **Next**.

6. Choose values for the parameters used for mixed infection variant detection (see Low Frequency Variant Detection).
7. In the Create Sample Report step, various summary items have been set. These are guidelines to help evaluate the quality of the results (see Create Sample Report).
8. In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters. Choose to save the results (we recommend creating a new folder for it) and click **Finish**.

Outputs from the Type Influenza Strain workflow

The workflow produces the following outputs:

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.
- **Sequences**. Folder containing consensus sequence list and list of best-matching reference sequences.
- **Tracks**. Folder containing various tracks.
  - **Consensus sequence**, **Gene** and **CDS** tracks based on the best-matching references.
  - **Read mapping**. Reads track of the sample reads mapping to the consensus sequence.
  - **Candidate mixed infection variant**. Variant track containing potential mixed infection variants.
  - **Candidate amino acid changes**. Track to see amino acids and changes in coding sequences due to potential mixed infections.
- **Consensus Genome Browser**. A track list containing the consensus sequences, the annotation tracks, variant tracks and the read mappings. Individual tracks are located in the **Tracks** folder.
- **Influenza Typing Report**. Sample report containing results of the analysis. The sample report is curated to contain the most important information for analysis interpretation, but all full reports can be found in the **QC & Reports** folder.

2.4 QIAseq Analysis template workflows

The QIAseq Analysis template workflows are found at:

Workflows | **Template Workflows** () | **Microbial Workflows** () | **QIAseq Analysis** ()

The QIAseq template workflows are configured for selection of Reference Data Sets, making them simple to launch while helping to ensure that the same reference data is used consistently. QIAGEN Reference Sets are available for download using the Reference Data Manager in the *CLC Genomics Workbench*. Reference data for a specific workflow can also be downloaded via the workflow launch wizard.

For further information about how to work with reference data when launching a workflow, and how to configure workflows to support use of Reference Data Sets, see Reference data and workflows.
For further information, see QIAGEN Sets.

Launching the QIAseq Analysis template workflows

The following sections describe how to launch the QIAseq Analysis template workflows from the **Workflows** menu. They can also be launched from the **QIAseq Panel Analysis Assistant**, where they can be found in the *xHYB Viral and Bacterial* category (see section 2.5).

2.4.1 Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host)

The **Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host)** template workflow performs spoligotyping for lineage detection and identifies high-frequency antimicrobial drug resistance variants. It is suitable for analysis of samples from human hosts generated with the QIAseq xHYB Mycobacterium tuberculosis Panel. Optionally, the workflow also detects and types Mycobacteriaceae, if the *QIAseq xHYB NTM-ID Panel* was used in conjunction with the *QIAseq xHYB Mycobacterium tuberculosis Panel*.

To analyze samples not from human hosts, you can create a copy of the workflow and edit it to fit your specific application, see Template workflows. Since the workflow element **Map Reads to Human Control Genes** is relevant for human data only, you should delete this. In addition, if a host genome is not relevant for you application, open the **Taxonomic Profiling** workflow element, and uncheck *Filter host reads*.
Once the workflow copy is customized, you can install it to make it available from under the **Workflows** menu (see Workflow installation).

To run the workflow using a variant database other than the default one, you need to modify the workflow elements where the database name appears as a column header, such as **Filter for WHO variants** and **WHO variant associated with resistance**.

QIAGEN Reference Data Set

The *QIAseq xHYB Mycobacterium tuberculosis Panel* Reference Data Set contains reference data relevant for this template workflow. It includes the Mycobacterium tuberculosis reference genome H37Rv and the WHO Mycobacterium tuberculosis variant database based on the WHO Mycobacterium tuberculosis mutation catalogue (see section 18.2). Like the template workflow, the reference data set is designed for human samples. It contains both a human host taxonomic profiling index and a sequence list with human control genes for use in the workflow step **Map Reads to Human Control Genes**.

For performing Mycobacteriaceae typing analysis a version of the *QIAseq xHYB Mycobacterium tuberculosis Panel* Reference Data Set, which contains the hsp65 reference database needed, is also available (for more, see section 2.4.1).

Data in the *QIAseq xHYB Mycobacterium tuberculosis Panel* set not already downloaded can be downloaded during the launch of the workflow. It can also be downloaded, as well as managed, using the Reference Data Manager, which can be opened by clicking on the **Manage Reference Data (** **)** button in the Toolbar. Click on the **QIAGEN Sets Reference Data Library** tab in the Reference Data Manager and search for the set by entering terms from its name in the search field.

For analysis of samples not from human hosts: If a non-human host is relevant for your application, you can create a host taxonomic profiling index from your host reference genome using **Create Taxonomic Profiling Index**, see section 16.5.

The workflow analysis

The raw Mycobacterium tuberculosis whole genome sequencing reads are trimmed for low quality, read-through adapter sequences, and G homopolymers. Trimmed reads are used as input for the separate spoligotyping analysis.

In the Taxonomic Profiling step, reads that map to the human host index are filtered. As a quality control step, these reads are subsequently mapped to the human control genes defined for the panel. In addition to human reads, reads identified as belonging to taxonomies other than Mycobacterium tuberculosis are excluded from downstream analysis.

The remaining reads are mapped to the Mycobacterium tuberculosis reference genome, and variants are called from this read mapping. The reference genome may differ from the lineage reported by the spoligotyping step. Using the same reference genome for mapping and variant calling across samples ensures comparability of variants and facilitates alignment with variant databases, such as the WHO Mycobacterium tuberculosis mutation catalogue, which are based on a specific genome. Variant calling is optimized for calling resistance in the dominant strain of an infection: variants with frequency beneath 50% will typically not be reported.

Detected variants are compared to the WHO drug resistance variant database and annotated with drug resistance information. Larger InDels that cannot be matched to the variant database exactly (e.g. whole-gene deletions), but that overlap with possible resistance InDels, are reported as candidate InDels and annotated with information from all resistance InDels that they overlap (for more, see section 2.4.1).

The analysis can also detect and type Mycobacteriaceae (for more, see section 2.4.1).

Launching the workflow

Before launching the workflow, make sure to download the *QIAseq xHYB Mycobacterium tuberculosis Panel* reference data set.

The **Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host) workflow** is available at:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **QIAseq Analysis** () | **Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host)** ()

Launch the workflow and step through the wizard.

1. Select whether to perform Mycobacteriaceae typing analysis. If the *QIAseq xHYB NTM-ID Panel* was used in conjunction with the *QIAseq xHYB Mycobacterium tuberculosis Panel*, select "Yes" (for more, see section 2.4.1).
2. Select the sequence list(s) containing the sample reads. If selecting multiple inputs from different samples, check the **Batch** option, see Running workflows in batch mode.

3. Select a reference data set or select "Use specified data elements". The latter runs the workflow using default elements, which can be viewed by clicking the "workflow roles" text just above the option.

4. If **Batch** was checked in step 1, choose whether batch units should be defined based on organization of the input data, or by provided metadata. In the next step, review the batch units resulting from your selections above.

5. Specify the spoligotyping settings (figure 2.31). Using the default values is usually sufficient, but we recommend taking a look at the spoligotyping report afterwards to make sure the results are as expected.

6. If you selected "Yes" for performing Mycobacteriaceae typing analysis, the parameters for filtering references can be changed. This might be necessary if the expected Mycobacteriaceae species is present in the sample at a very low abundance. The default settings are expected to work in most cases. For more information about the filters, see **Find Best References using Read Mapping** (section 8.2).

7. Finally, select a location to save outputs to.

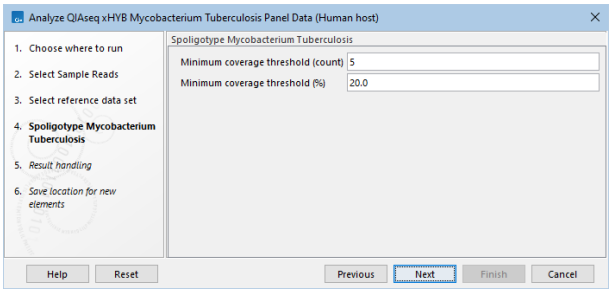


Figure 2.31: *Select the minimum threshold settings for spoligotyping.*

Workflow outputs and how to interpret

The outputs provided by the workflow are:

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.
- **Tracks**. Folder containing various tracks.
  - **Genome**, **Gene** and **CDS** tracks based on the Mycobacterium tuberculosis reference used.
  - **Human_control_genes_read_mapping**. Track to see mapping of the human host reads to the control genes.
  - **Read_mapping**. Reads track of the sample reads mapping to the reference genome.
  - **Amino_acid_track**. Track to see amino acids and potential changes in coding sequences of the reference genome.

  - **WHO_mycobacterium_tuberculosis_variant_database_v1.0 (filtered)**. The WHO resistance database, filtered to only contain Insertions and Deletions overlapping with candidate InDels (for more, see section 2.4.1).

- **Variants**. Folder containing all the variant tracks generated during the analysis.
  - **Raw_variants**. Variant track containing all raw variants detected by `Fixed Ploidy Variant Detection` i.e., before adjusting with **Join Nearby Variants** (section 13.4) and annotating.
  - **Filtered_InDels**. InDels detected by `InDels and Structural Variants` that were not already present in the "Raw_variants" track. Only InDels with a variant ratio over 0.5 are reported. These InDels are later merged with the other variants and included in the "Annotated variants" track.
  - **WHO_variants_detected**. Variant track containing only variants from the WHO resistance database.
  - **Novel_variants_detected**. Variant track containing only variants that are *not* graded by the WHO.
  - **WHO_candidate_InDels**. Annotation track containing insertions, deletions and "complexes" that may correspond to a WHO-graded variant, but which it was not possible to match to the resistance database exactly (for more, see section 2.4.1).

- **Genome Browser**. A track list containing the reference genome, gene, CDS, read mapping, variant, candidate InDels, and amino acid changes tracks.

- **QIAseq xHYB Mycobacterium Tuberculosis Analysis Report**. Sample report containing results of the analysis. The sample report is curated to contain the most important information for analysis interpretation, but all full reports can be found in the **QC & Reports** folder.

- **Annotated variants**. Variant track containing all detected variants and non-candidate InDels after readjustment and annotated with WHO resistance, amino acid changes and gene information.

If you selected "Yes" for performing Mycobacteriaceae typing analysis, some additional outputs are provided:

- **NTM-ID Panel Analysis**. (Only if a positive result was detected). Folder containing results from the analysis.
  - **Mycobacteriaceae reads**. Sequence lists (single and paired) containing reads from the input that mapped to the hsp65 references before refinement of references.
  - **Mycobacteriaceae read mapping**. Reads track of the reads mapped to the final hsp65 references.

- **QIAseq xHYB NTM-ID Analysis Report**. The report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout this report or can be found in the **QC & Reports** folder.
  The report icon will be colored based on whether Mycobacteriaceae was detected (for more, see section 2.4.1):

- A green dot on the report icon indicates detection of at least one Mycobacteriaceae species.
- A red dot on the report icon indicates no detection of Mycobacteriaceae species.

The sample report "QIAseq xHYB Mycobacterium Tuberculosis Analysis Report" is the main output of the workflow. This allows for easy overview of the analysis results, both in terms of quality control and detected drug resistance for the sample. An example of the report can be seen in figure 2.32.

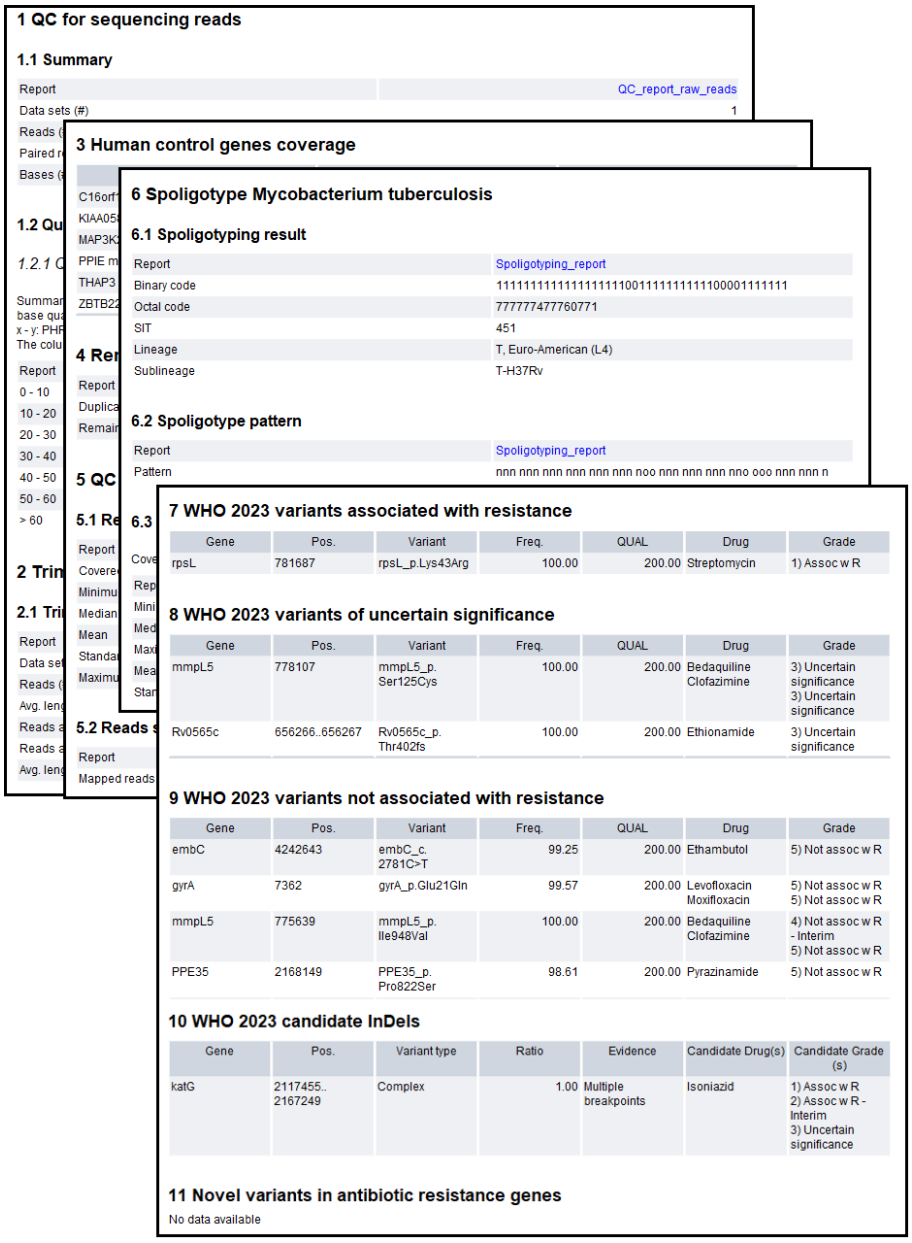


Figure 2.32: *An example report from the Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host) workflow.*

The report contains the following sections:

- Sections 1-5 contain quality metrics for the analysis:
  - **QC for sequencing reads**. A summary of the number of raw reads and their quality. If the reads are of too low quality, the results may be unreliable.
  - **Trim reads**. A summary of the read trimming. If the percentage of reads after trim is low or the average read length after trimming is considerably lower than before trimming, it may be a sign that something is wrong with the sample reads.
  - **Human control genes coverage**. A summary of the host reads mapping to the human control genes. The coverage can be low, but there should be some reads mapping to the genes. If not, something may have gone wrong during the sample prep, or the sample was not made with the QIAseq xHYB Mycobacterium tuberculosis Panel.
  - **Remove duplicate mapped reads**. A high percentage of duplicates may indicate that the sample contains little gDNA.
  - **QC for read mapping**. For the QIAseq xHYB Mycobacterium tuberculosis panel, the coverage percentage should be close to 100%. Also, most of the reads after trimming (see Reads after trim in the **Trim reads** section) should be mapped. If this is not the case, there may have been an issue with the sample prep.
- Sections 6-11 contain lineage and variant results from the analysis:
  - **Spoligotype Mycobacterium tuberculosis**. Results of spoligotyping. This reports on the detected SIT, lineage, sublineage, and spoligotype pattern. It can be a good idea to take a look at the coverage plot in the full spoligotyping report (/QC & Reports/Spoligotyping_report), to ascertain whether the minimum threshold has been correctly set. For additional information about the spoligotype report content, see section 11.1.2.
  - **WHO 2023 variants associated with resistance**. Variants detected in the sample that have been graded "1)" or "2)" for at least one drug by the WHO. As variants can be graded for multiple drugs with different grades, this section may contain grades of "3)" and higher as well. For information about WHO grading, see section 18.2.
  - **WHO 2023 variants of uncertain significance**. Variants detected in the sample that have been graded "3)" for at least one drug by the WHO, but not "1)" or "2)". As variants can be graded for multiple drugs with different grades, this section may contain grades of "4)" and higher as well.
  - **WHO 2023 variants not associated with resistance**. Variants detected in the sample that have only been graded "4)" or "5)" by the WHO.
  - **WHO 2023 candidate InDels**. InDels detected in the sample that overlap one or more WHO-graded InDels (for more, see section 2.4.1).
  - **Novel variants in antibiotic resistance genes**. Variants detected in the sample, but that are not graded by the WHO. The report only contains variants in known resistance genes, and excludes variants in protein-coding regions that result in synonymous mutations. To view all detected novel variants, look at the "/Variants/Novel_variants_detected" variant track.

The variant table reports contain the following columns:

- **Gene**. For WHO variants, this is the gene with which the variant is associated. For Novel variants, it is the gene in which the variant is located.

- **Pos.**. The genomic position of the variant within the reference genome.
- **Variant**. (Only WHO variants). The name(s) of the variant as given by WHO. The name consists of the gene in which the variant is located, along with the corresponding position and change, either as a nucleotide or amino acid change.
- **AA change**. (Only Novel variants). This describes the change on the protein level. For example, single amino-acid changes caused by SNVs are listed as p.Gly261Cys, denoting that in the protein sequence (hence the "p.") the Glycine at position 261 is changed into Cysteine. Frame-shifts caused by nucleotide insertions and deletions are listed with the extension *fs*, for example p.Pro244fs denoting a frameshift at position 244 coding for Proline. For further details about HGVS nomenclature as relates to proteins, see http://varnomen.hgvs.org/recommendations/protein/.
- **Freq.**. The number of reads supporting the allele divided by the number of reads covering the position of the variant. Note that variants with frequency beneath 50% will typically not be reported.
- **QUAL**. Measure of the significance of a variant, i.e., a quantification of the evidence (read count) supporting the variant, relative to the coverage and what could be expected to be seen by chance, given the error rates in the data. For additional information, see Variant tracks.
- **Drug**. (Only WHO variants). The antimicrobial resistance drug(s) for which the variant is graded.
- **Grade**. (Only WHO variants). The grade of drug resistance determined for the variant.

The candidate InDels table report contains the following unique columns (for more, see section 2.4.1):

- **Variant type**. The type of variant detected, either "Deletion", "Insertion" or "Complex". A "Complex" variant indicates that more than two breakpoints give rise to the structural variant.
- **Ratio**. Ratio of reads calculated as the sum of the 'Non perfect mapped' reads for the breakpoints used to infer the InDel, divided by the sum of the 'Non perfect mapped' and 'Perfect mapped' reads for the breakpoints used to infer the InDel. Note that variants with ratio beneath 50% will not be reported.
- **Evidence**. The mapping evidence on which the call of the InDel was based (see Theoretically expected structural variant signatures).
- **Candidate Drug(s)**. The antimicrobial resistance drug(s) for which variants overlapping with the candidate InDel are graded.
- **Candidate Grade(s)**. The grade(s) of drug resistance determined for variants overlapping with the candidate InDel.

If no variants are detected in a section of the report, it will say "No data available".

For more info on the WHO variant database, including the resistance grades, see section 18.2.

WHO 2023 candidate InDels

Candidate InDels are structural variants that overlap, but do not exactly match, a WHO-graded variant. These include large deletions that may cause loss of function of a resistance-associated gene. Only deletions that overlap with a WHO deletion, and insertions that overlap with a WHO insertion are included. Complexes are included if they overlap with either.

Candidate InDels are called by **InDels and Structural Variants** as Deletions, Insertions or Complexes. A complex is usually called in regions with more than 2 signature breakpoints (see `Structural Variants and InDels output`).

As candidate InDels may overlap with many resistance-associated variants, these are not listed individually. Instead the "Candidate Drug(s)" column includes all possible drugs to which the variants may confer resistance. Similarly, the "Candidate Grade(s)" column includes all possible grades of resistance associated with those variants. To avoid redundancy, each drug and grade will only be reported once in the column, even if multiple variants are associated with that drug and grade.

A candidate InDel is not a guarantee of resistance or susceptibility, but an indicator that one should take a closer look at that location in the read mapping, to evaluate whether the variant is of interest.

A good way to investigate a candidate InDel further is to open up the "Genome Browser" track list output from the analysis and zoom into the candidate InDel's location. In figure 2.33 it is clear from the read mapping that a large deletion is present where the "Complex" is called.

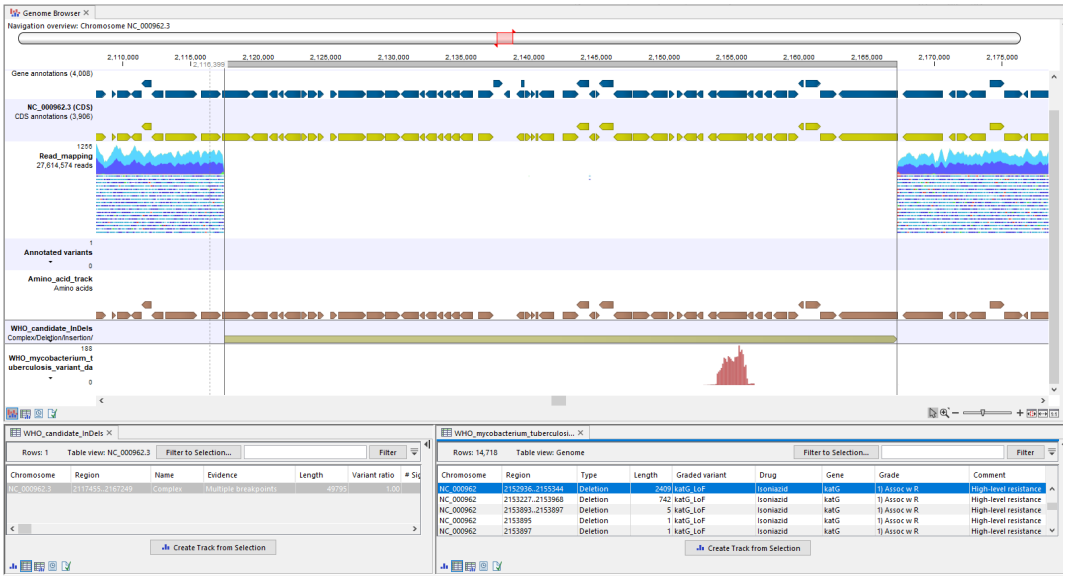


Figure 2.33: *A candidate complex called in a region of the genome where the read mapping clearly lacks coverage, indicating that the complex is a deletion. In the filtered WHO resistance database track (bottom), it can be seen that the candidate complex, now confirmed to be a deletion, overlaps with multiple large WHO LoF deletions.*

Candidate InDels are annotated with both WHO insertions and deletions, so it is necessary to take a closer look at the variants to determine whether candidate drug resistance from the report is supported. The "WHO_mycobacterium_tuberculosis_variant_database_v1.0 (filtered)" track in the Genome Browser can help to investigate whether the InDel overlaps with a meaningful

WHO variant. In figure 2.33 the candidate deletion overlaps with multiple WHO loss of function deletions, which confer resistance to the drug Isoniazid. It can be inferred that a large deletion will confer similar resistance (see also pages 88 and 102 about "feature_ablation" in [WHO, 2023]).

Mycobacteriaceae typing analysis

The Mycobacteriaceae typing analysis is intended for samples where the *QIAseq xHYB NTM-ID Panel* was used in conjunction with the *QIAseq xHYB Mycobacterium tuberculosis Panel*. It performs the analysis in the same way as the **Analyze QIAseq xHYB NTM-ID Panel Data (Human host)** (section 2.4.2) template workflow. A description of the analysis is available under **The workflow analysis** (section 2.4.2).

The reads used as input for the Mycobacteriaceae analysis in this workflow, are extracted from the hsp65 gene region of the H37Rv read mapping. Due to the high level of similarity between hsp65 genes from different Mycobacteriaceae species, reads are expected to map to this region, even if they don't come from H37Rv.

The "QIAseq xHYB NTM-ID Analysis Report" report contains the following sections:

- **Summary**. A summary of the QC summary item "Percentage reads mapped to reference" and whether it passed (green) or failed (red).
- **Find best references using read mapping**. Contains a summary of how many of the input reads mapped to the hsp65 references and how many were unmapped. These are the mapping statistics before refinement of the references, and may not match the number of reads mapped to the reference(s) in the following sections. To see more details of the results prior to refinement, see the "Find_best_reference_report" in the "QC & Reports" folder.
- **QC for Mycobacteriaceae mapping**. (Only if a positive result was detected). Contains the name of the reference(s) detected after refinement of references and mapping statistics for the reads mapped to them. The bottom of **Workflow outputs and how to interpret** (section 2.4.2) contains details about the columns.

2.4.2 Analyze QIAseq xHYB NTM-ID Panel Data (Human host)

The **Analyze QIAseq xHYB NTM-ID Panel Data (Human host)** template workflow detects and types species from the Mycobacteriaceae family. It is suitable for analysis of samples from human hosts generated with the *QIAseq xHYB NTM-ID Panel* and can detect both the presence of Mycobacterium tuberculosis and Non-Tuberculosis Mycobacteria (NTM) by targeting the 65-kDa heat shock protein (hsp65) gene.

If the *QIAseq xHYB NTM-ID Panel* was used in conjunction with the *QIAseq xHYB Mycobacterium tuberculosis Panel*, use the template workflow **Analyze QIAseq xHYB Mycobacterium Tuberculosis Panel Data (Human host)** (section 2.4.1) instead.

To analyze samples not from human hosts, you can create a copy of the workflow and edit it to fit your specific application, see `Template workflows`. Since the workflow element **QC for Targeted Sequencing** is relevant for human data only, you should delete this. In addition, if a host genome is not relevant for you application, you can remove the "Host reference" input from

the Find Best References using Read Mapping step.
Once the workflow copy is customized, you can install it to make it available from under the **Workflows** menu (see `Workflow installation`).

QIAGEN Reference Data Set

The *QIAseq xHYB NTM-ID Panel* Reference Data Set contains reference data relevant for this template workflow. It includes a non-redundant reference database of the hsp65 gene, used for detection and typing of Mycobacteriaceae. Like the template workflow, the reference data set is designed for human samples, and additionally contains a human host reference and an annotation track of human control gene regions.

Data in the *QIAseq xHYB NTM-ID Panel* set not already downloaded can be downloaded during the launch of the workflow. It can also be downloaded, as well as managed, using the Reference Data Manager, which can be opened by clicking on the **Manage Reference Data ( )** button in the Toolbar. Click on the **QIAGEN Sets Reference Data Library** tab in the Reference Data Manager and search for the set by entering terms from its name in the search field.

For analysis of samples not from human hosts: If a non-human host is relevant for your application, you can download a host genome using Download Custom Microbial Reference Database (section 16.2).

The workflow analysis

The raw reads are trimmed for low quality, read-through adapter sequences, and G homopolymers. If a `Trim adapter list` is supplied, these adapters will also be trimmed.

Trimmed reads are mapped to the references of Mycobacteriaceae hsp65 genes and the human host reference simultaneously using **Find Best References using Read Mapping** (section 8.2). Due to the high level of similarity between hsp65 genes from different Mycobacteriaceae species, the reads are mapped with stringent mapping parameters.

This results in an intial set of hsp65 reads and possible references. If more than one possible reference is detected for the sample reads, the analysis will try to refine the references by only looking at non-ambiguous reads mapping to this subset of the references. This helps to resolve false positive species calls as a result of the high level of similarity within the target gene.

While the detected species may contain a "variant" name (e.g. "Mycobacterium tuberculosis variant bovis"), be advised that the hsp65 gene is usually not specific enough for strain level typing - only species level typing. For mixed infections involving more than one Mycobacteriaceae species, the lower detection limit is 3% abundance relative to the most abundant species.

After reference refinement, all of the hsp65 reads will be re-mapped to the final refined list of references, and the detected species and read mapping statistics are output in the report.

The human control gene regions are used for QC for Targeted Sequencing. The *QIAseq xHYB NTM-ID Panel* contains probes for these regions as an indicator of succesful hybrid capture.

Launching the workflow

The **Analyze QIAseq xHYB NTM-ID Panel Data (Human host) workflow** is available at:

Workflows | Template Workflows ( ) | Microbial Workflows ( ) | QIAseq Analysis ( ) | Analyze QIAseq xHYB NTM-ID Panel Data (Human host) ( )

Launch the workflow and step through the wizard.

1. Select the sequence list(s) containing the sample reads. If selecting multiple inputs from different samples, check the **Batch** option, see `Running workflows in batch mode`.
2. Select a reference data set or select "Use specified data elements". The latter runs the workflow using default elements, which can be viewed by clicking the "workflow roles" text just above the option.
3. If **Batch** was checked in step 1, choose whether batch units should be defined based on organization of the input data, or by provided metadata. In the next step, review the batch units resulting from your selections above.
4. If your reads contain adapters, add an appropriate Trim adapter list. Click **Next**.
5. The parameters for filtering references can be changed (figure 2.34). This might be necessary if the expected Mycobacteriaceae species is present in the sample at a very low abundance. The default settings are expected to work in most cases. For more information about the filters, see **Find Best References using Read Mapping** (section 8.2).
6. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results (see `Create Sample Report`). Thresholds can be changed, if the defaults are too stringent for the input samples.
7. Finally, select a location to save outputs to.

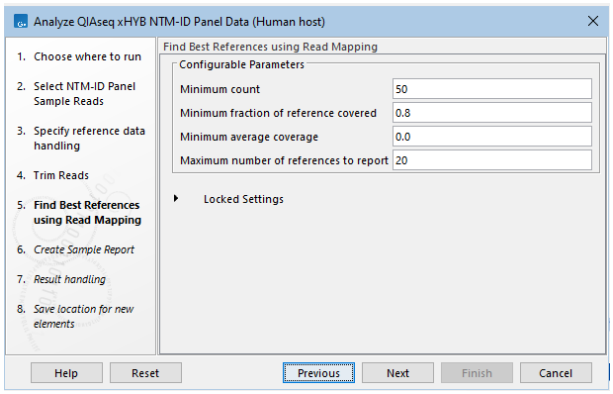


Figure 2.34: *Parameters for filtering references can be changed.*

Workflow outputs and how to interpret

The outputs provided by the workflow are:

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

  - Mycobacteriaceae full mapping statistics table, which the *Mycobacteriaceae mapping report* is based on.

- **Outputs**. Folder containing results from the analysis.
  - **Human host read mapping**. Track to see mapping of the human host reads and from which the *QC for human control genes* report section was derived.
  - **Mycobacteriaceae reads**. (Only if a positive result was detected). Sequence lists (single and paired) containing reads from the input that mapped to the hsp65 references before refinement of references.
  - **Mycobacteriaceae read mapping**. (Only if a positive result was detected). Reads track of the reads mapped to the final hsp65 references.

- **Typing Report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met (see the "Quality control" section in the sample report for specifics):
  - A green dot on the report icon indicates detection of at least one Mycobacteriaceae species, and all quality control thresholds passed.
  - A yellow dot on the report icon indicates detection of at least one Mycobacteriaceae species, but not all quality control thresholds passed.
  - A red dot on the report icon indicates no detection of Mycobacteriaceae species. The report must be opened to determine whether quality control thresholds passed.

The **Typing Report** is the main output of the workflow. This allows for easy overview of the analysis results, both in terms of quality control and detected Mycobacteriaceae for the sample. An example of the report can be seen in figure 2.35.

The report contains the following sections:

- Sections 1-4 contain quality metrics for the analysis:
  - **Summary**. A summary of the QC summary items and whether they passed (green) or failed (yellow). The "Percentage reads mapped to reference" will be red if no Mycobacteriaceae species was detected.
  - **QC for sequencing reads**. A summary of the number of raw reads and their quality. If the reads are of too low quality, the results may be unreliable.
  - **Trim reads**. A summary of the read trimming. If the percentage of reads after trim is low or the average read length after trimming is considerably lower than before trimming, it may be a sign that something is wrong with the sample reads.
  - **QC for human control genes**. A summary of the host reads mapping to the human control genes. The fraction of the regions covered is expected to be more than half, and with relatively high median coverage due to the hybrid capture method used. If not, something may have gone wrong during the sample prep, or the sample was not made with the *QIAseq xHYB NTM-ID Panel*.

- Sections 5 and 6 contain detection results for the analysis:

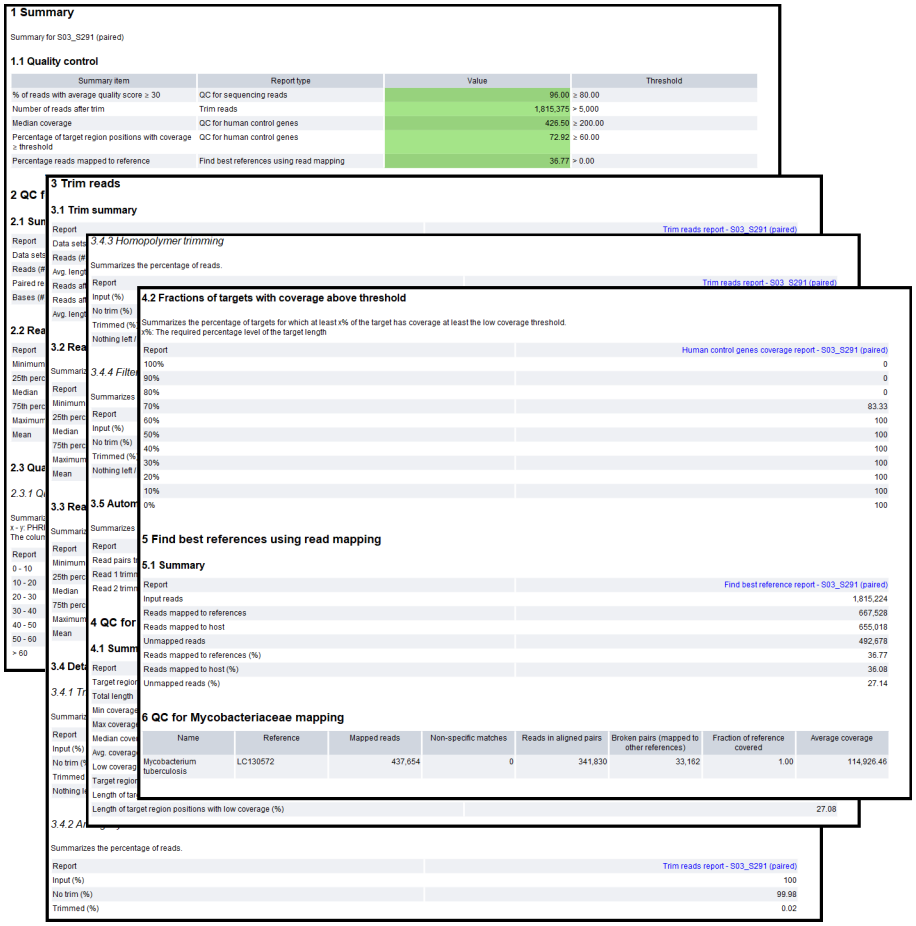


Figure 2.35: *An example report from the Analyze QIAseq xHYB NTM-ID Panel Data (Human host) workflow.*

- **Find best references using read mapping**. Contains a summary of how many of the input reads mapped to the hsp65 references vs. the host and how many were unmapped. These are the mapping statistics before refinement of the references, and may not match the number of reads mapped to the reference(s) in the following sections. To see more details of the results prior to refinement, see the "Find best reference report" in the "QC & Reports" folder.
- **QC for Mycobacteriaceae mapping**. (Only if a positive result was detected). Contains the name of the reference(s) detected after refinement of references and mapping statistics for the reads mapped to them. See below for details about the columns.

The "QC for Mycobacteriaceae mapping" table report contains the following columns:

- **Name**. The name of the Mycobacteriaceae species reference detected.
- **Reference**. The accession number of the reference.
- **Mapped reads**. The number of reads mapped to the reference.
- **Non-specific matches**. The number of reads that mapped equally well to multiple positions in the set of detected references.

- **Reads in aligned pairs**. The number of reads mapped in pairs to the reference.
- **Fraction of reference covered**. The fraction of the reference covered by at least one read.
- **Average coverage**. The number of nucleotides mapped to the reference divided by the reference length.

2.4.3 Analyze QIAseq xHYB Viral Panel Data (Human host)

The **Analyze QIAseq xHYB Viral Panel Data (Human host)** template workflow trims reads, identifies the best match reference, and calls viral variants. It is suitable for analysis of samples from human hosts generated with the QIAseq xHYB viral panels:

- QIAseq xHYB Respiratory Panel
- QIAseq xHYB Viral STI Panel
- QIAseq xHYB Adventitious Agent Panel
- QIAseq xHYB MPXV Panel
- QIAseq xHYB HepC Panel

QIAGEN reference data set

The *QIAseq xHYB Viral Panels* and *QIAseq xHYB HepC Panel* Reference Data Sets contain reference data relevant to this template workflow. The data can be downloaded and managed using the Reference Data Manager. To download the data, open the Reference Data Manager by clicking on the **Manage Reference Data ( )** button in the top Toolbar, go to the **QIAGEN Sets Reference Data Library** tab, and locate the relevant set. Data from the Reference Data Sets that have not already been downloaded can be downloaded when the workflow is run.

Like the template workflow, the Reference Data Sets are designed for human samples. They include both a human host taxonomic profiling index and a sequence list with human control genes for use in the workflow step **Map Reads to Human Control Genes**. For analysis of samples not from human hosts: If a non-human host is relevant for your application, you can create a host taxonomic profiling index from your host reference genome using **Create Taxonomic Profiling Index**, see section 16.5.

Workflow customization

You can create a copy of the template workflow and edit it to fit your specific application, see `Template workflows`. This could be useful e.g., in the following cases:

- **Your sample is from a non-human source**. Since the workflow element **Map Reads to Human Control Genes** is relevant for human data only, it can be deleted. In addition, if a host genome is not relevant for you application, open the **Taxonomic Profiling** workflow element, and uncheck *Filter host reads*.
- **You expect mixed or co-infections in your samples**. The workflow analysis and reporting is made in such a way that only one viral species is expected per sample. You can configure the *Filter references* parameters in the **Find Best References using Read Mapping** workflow element to allow for detection of multiple species according to your preferences.

Launching the workflow

The **Analyze QIAseq xHYB Viral Panel Data (Human host)** template workflow is available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **QIAseq Analysis** ( ) | **Analyze QIAseq xHYB Viral Panel Data (Human host)** ( )

Launch the workflow and step through the wizard.

1. Select the sequence list(s) containing the reads to analyze.
2. Select a reference data set or select "Use specified data elements". The latter runs the workflow using default elements, which can be viewed by clicking the "workflow roles" text just above the option.
3. Define batch units. For details, see `Running part of a workflow multiple times`.
4. Check that batching is as intended.
5. Verify or select the viral taxonomic profiling index (figure 2.36).
6. Verify or select the host taxonomic profiling index.
7. Select the viral reference database(s). If in the first step you selected e.g., the QIAseq xHYB Viral Panels reference set, you can now select which of the available viral reference databases from that set to apply (2.37). If you chose to use the specified data elements, select a reference database.
8. Verify or select the control genes.
9. If your reads contain adapters, add an appropriate Trim adapter list (see `Adapter trimming`). This is optional and not needed if the QIAseq xHYB Microbial Hyb Kit was used.
10. Specify Low Frequency Variant Detection settings, see figure 2.38.
11. Specify Extract Consensus sequence settings (figure 2.39).
12. In the "Create Full Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results.
13. Finally, select a location to save outputs to.

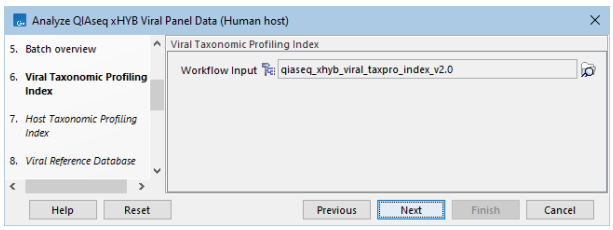


Figure 2.36: *Select viral taxonomic profiling index.*

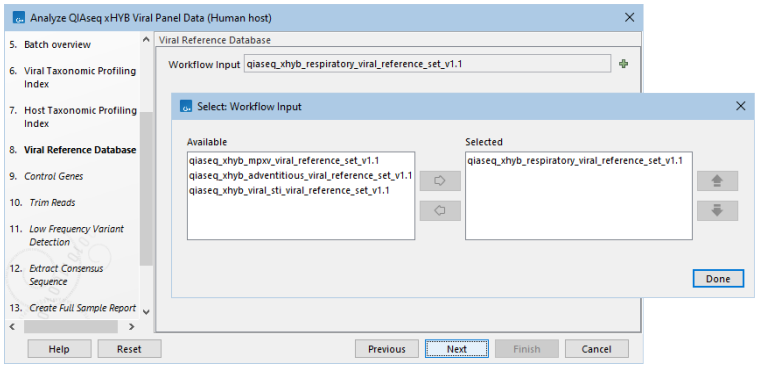


Figure 2.37: *Select one or more viral reference databases.*

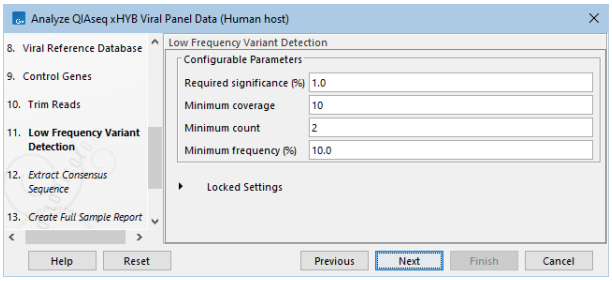


Figure 2.38: *Low frequency variant detection parameters.*

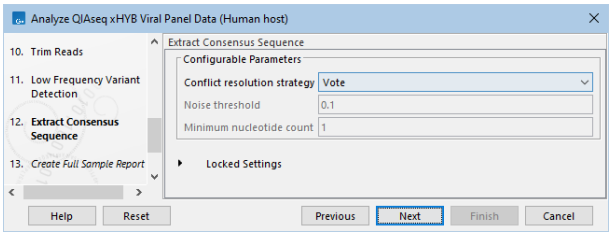


Figure 2.39: *Extract Consensus Sequence parameters.*

Workflow tools and outputs

The Analyze QIAseq xHYB Viral Panel Data (Human host) template workflow consists of the following tools.

- **QC for Sequencing Reads**. Performs basic quality control of the sequencing reads. The output, which is included in a combined report, can be used to evaluate the quality of the sequencing reads. See QC for Sequencing Reads.
- **Trim Reads**. Removes adapter sequences and low quality nucleotides. The appropriate settings for the Trim Reads tool depends on the protocol used to generate the reads. See Trim Reads.
- **Taxonomic Profiling**. Used to filter viral reads from the sample reads. See section 6.4. Host reads i.e., reads that map to the host taxonomic profiling index, do not count toward the taxonomic profiling result, but are used as input for **Map Reads to Human Control Genes**. Viral reads - reads that map to the viral taxonomic profiling index - are subsampled and later used as input for **Find Best References using Read Mapping**.

- **Map Reads to Human Control Genes**. Maps the host reads output from **Taxonomic Profiling** to the host taxonomic profiling index, to a reference of human control genes. See `Map Reads to Reference`. This serves as a QC step to verify mapping to the human control genes. For human samples, you expect to see mapping of reads to all human control genes.
- **Find Best References using Read Mapping**. Maps the viral reads output from **Taxonomic Profiling** to the selected viral reference database to identify which reference sequence is the "Best match". See section 8.2.
- **Remove Duplicate Mapped Reads**. Removes duplicate reads derived from PCR amplification (or other enrichment) during sample preparation from the mapping. See `Remove Duplicate Mapped Reads`. The output reads track is used as input for **Local Realignment**.
- **Local Realignment**. Improves the alignment of the reads in the reads track. See `Local Realignment`.
- **Low Frequency Variant Detection**. Calls variants in the read mapping that are present at low frequencies. See `Low Frequency Variant Detection`.
- **Filter on Custom Criteria** and **Filter against Known Variants**. Remove variants that fall below a set of thresholds. For this workflow, coverage >30 and frequency >20% is required. See `Variant filtering`.
- **Amino Acid Changes**. Uses the called variants to generate a track of amino acid changes. See `Amino Acid Changes`.
- **Create Mapping Graph** and **Identify Graph Threshold Areas**. Creates a track with regions with coverage below a threshold. For this workflow, the threshold is set to 30. See `Create Mapping Graph` and `Idnetify Graph Threshold Areas`.
- **Extract Consensus Sequence**. Makes a consensus sequence from the read tracks from **Local Realignment**. See `Extract Consensus Sequence`.
- **QC for Read Mapping**. Performs quality control of the read mapping. See `QC for Read Mapping`.
- **Refine Abundance Table**. Aggregates the abundance table from **Taxonomic Profiling** on species-level. See section 7.2.
- **Merge Abundance Tables**. Merges the sample-specific abundance tables to one combined abundance table. See section 7.1.
- **Create Sample Report**. Creates a single report per sample that contains all tool reports. Also allows for setting of Summary items to highlight possible issue during analysis. See `Create Sample Report`.

The sample-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be

found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.

- **Refined abundance table**. The abundance of viral species, along with their full taxonomy. See section 6.4.3 and section 7.2.
- **Consensus sequence**. Viral consensus sequence(s), extracted from the *Best match read mapping* track.
- **Track list**. Collection of all the tracks in the "Tracks" folder, except for the *Read mapping human control genes* track.
- **Viral Reads**. Folder containing sequence list(s) of reads that mapped to the viral taxonomic profiling index.
- **Tracks**. Folder containing all tracks output during analysis.
  - **Read mapping human control genes**. The host reads mapped against the control gene reference.
  - **Best match sequence**. The "Best match" reference sequence as identified by the Find Best References using Read Mapping tool.
  - **Best match CDS track**. The CDS track extracted from the "Best match" reference sequence as identified by the Find Best References using Read Mapping tool.
  - **Best match read mapping**. Reads mapped to the "Best match" viral reference. Output from **Local Realignment**.
  - **Low coverage areas**. List of low coverage regions in the *Best match read mapping* output.
  - **Annotated variant track**. List of detected variants left after filtering, annotated with amino acid changes.
  - **Amino acid track**. List of amino acid changes.
- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Combined report**. Combined report of all sample reports. The combined report contains all quality control information and analysis results, including the result best matching reference as detected by **Find Best Reference using Read Mapping**. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.
- **Merged refined abundance table**. The abundance of viral species for all samples in the workflow run. See section 6.4.3 and section 7.2. In this analysis, Taxonomic Profiling is only used for filtering viral reads. However, if the best matching reference species does not align with the most abundant species in the abundance table, or if another species has a high relative abundance compared to the best match, it may be a sign of a mixed or co-infection. As mentioned previously, this workflow does not support multiple species detection, but alterations can be made to investigate such a case further.

2.4.4 Find QIAseq xHYB AMR Markers (Human host)

The **Find QIAseq xHYB AMR Markers (Human host)** template workflow trims reads and detects antimicrobial resistance (AMR) markers.

It is suitable for analysis of samples from human hosts generated with the QIAseq xHYB AMR Panel.

To analyze non-human samples, you can create a copy of the workflow and edit it to fit your specific application, see Template workflows. Since the workflow element **Map Reads to Human Control Genes** is relevant for human data only, you should delete this.
Once the workflow copy is customized, you can install it to make it available from under the **Workflows** menu (see Workflow installation).

QIAGEN reference data set

The *QIAseq xHYB AMR Panel* Reference Data Set contains reference data relevant for this template workflow. Data in this set that is not already downloaded can be downloaded during the launch of the workflow. It can also be downloaded, as well as managed, using the Reference Data Manager, which can be opened by clicking on the **Manage Reference Data ( )** button in the Toolbar. Click on the **QIAGEN Sets Reference Data Library** tab in the Reference Data Manager and search for the set by entering terms from its name in the search field.

Launching the workflow

The **Find QIAseq xHYB AMR Markers (Human host)** template workflow is available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **QIAseq Analysis** ( ) | **Find QIAseq xHYB AMR Markers (Human host)** ( )

Launch the workflow and step through the wizard.

1. Select the sequence list(s) containing the reads to analyze.
2. Select a reference data set or select "Use specified data elements". The latter runs the workflow using default elements, which can be viewed by clicking the "workflow roles" text just above the option.
3. Define batch units. For details, see Running part of a workflow multiple times.
4. Check that batching is as intended.
5. Verify or select the reference marker database (figure 2.40).
6. Verify or select control genes.
7. If your reads contain adapters, add an appropriate Trim adapter list. This is optional and not needed if the QIAseq xHYB Microbial Hyb Kit was used.
8. In the "Create Sample Report" step various summary items have been set. These are guidelines to help evaluate the quality of the results.
9. Finally, select a location to save outputs to.

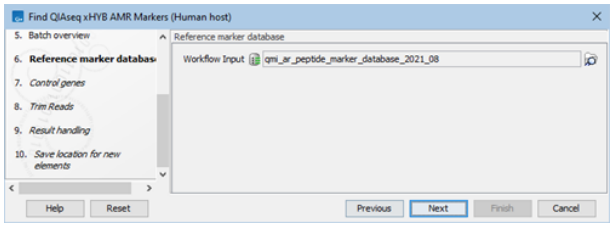


Figure 2.40: *Select the reference marker database*

Workflow tools and outputs

The Find QIAseq xHYB AMR Markers (Human host) template workflow consists of the following tools.

- **QC for Sequencing Reads**. Performs basic quality control of the sequencing reads. The output, which is included in a combined report, can be used to evaluate the quality of the sequencing reads. See `QC for Sequencing Reads`.
- **Trim Reads**. Removes adapter sequences and low quality nucleotides. The appropriate settings for the Trim Reads tool depends on the protocol used to generate the reads. See `Trim Reads`.
- **Map Reads to Human Control Genes**. Maps the host reads output from **Taxonomic Profiling** to the host taxonomic profiling index, to a reference of human control genes. See `Map Reads to Reference`. This serves as a QC step to verify mapping to the human control genes. For human samples, you expect to see mapping of reads to all human control genes.
- **QC for Read Mapping**. Performs quality control of the read mapping. See `QC for Read Mapping`.
- **Find Resistance with ShortBRED**. Detects and quantifies the presence of antimicrobial resistance marker genes of interest. See section 6.4.
- **Merge Abundance Tables**. Merges the sample-specific abundance tables to one combined abundance table. See section 7.1.
- **Create Sample Report**. Creates a single report per sample that contains all tool reports. Also allows for setting of Summary items to highlight possible issue during analysis. See `Create Sample Report`.

The sample-specific outputs provided by this workflow are:

- **Sample report**. The sample report is curated to contain the most important information for analysis interpretation. All full reports are linked throughout the Sample report or can be found in the **QC & Reports** folder. The Sample report icon will be colored based on whether Summary item thresholds were met. See the "Quality control" section in the sample report for specifics.
- **Analysis Results**. Folder containing results output during analysis.

  - **Resistance table**. The result table from **Find Resistance with ShortBRED**. The output provides the abundance of each detected AMR marker. See section 13.3.1.
  - **Read mapping human control genes**. Reads track containing the reads that mapped to the human control genes.

- **QC & Reports**. Folder containing the individual reports generated during the analysis.
  - All reports from the sample report are found here in their full length.

The combined outputs provided by this workflow are:

- **Combined report**. Combined report of all sample reports. The combined report contains all quality control information and analysis results. The combined report icon will be colored based on whether Summary item thresholds were met in each sample. See the "Quality control" section in the combined report for specifics.
- **Merged resistance table**. Provides the abundances of the detected AMR markers across samples. See section 13.3.1.

2.5 QIAseq Panel Analysis Assistant

The QIAseq Panel Analysis Assistant provides an easy entrance point for working with data generated with QIAseq panels and kits. Using the QIAseq Panel Analysis Assistant, information about the panels and kits can be accessed and available analyses can be viewed and run.

Most analyses offered via the QIAseq Panel Analysis Assistant are based on template workflows, which are available under the Workflows menu. Analyses launched using the QIAseq Panel Analysis Assistant have the appropriate reference data preselected. Additionally, some parameters are different to the template workflow, to account for the panel/kit design.

Validation of results should be performed.

To start the QIAseq Panel Analysis Assistant, go to:

Workflows | **Template Workflows** | **QIAseq Panel Analysis Assistant (** **)**

This opens a wizard listing different categories on the left, and analyses in the selected category on the right (figure 2.41).

An analysis can be:

- A pre-configured template workflow, available from the Workflows menu.
- A pre-configured analysis tool, available from the Tools menu.
- A tool only available from within the QIAseq Panel Analysis Assistant.

Once an analysis has been selected, it can be started using **Run**. Additional actions for the selected analysis are available under **More**.

For detailed information on the QIAseq Panel Analysis Assistant, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QIAseq_Panel_Analysis_Assistant.html.

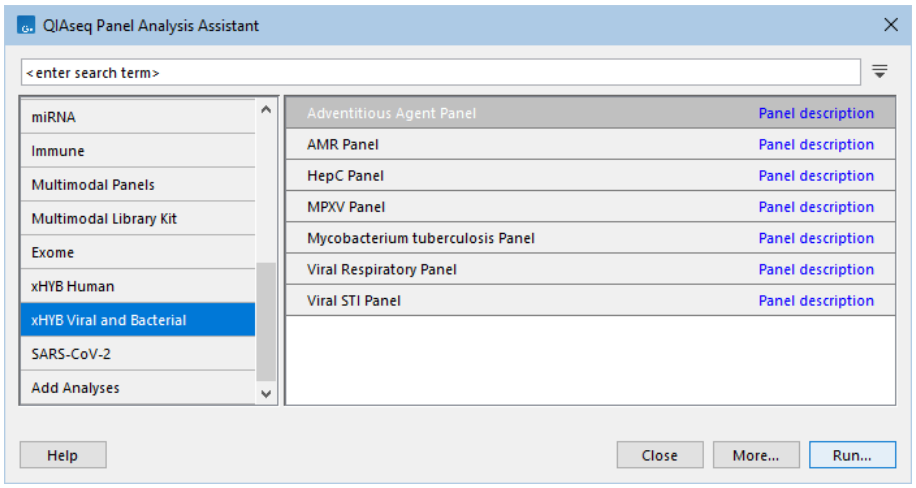


Figure 2.41: *The QIAseq Panel Analysis Assistant. Multiple analyses are available for the xHYB Viral and Bacterial category. The "Panel description" links to more information about the panel.*

Part III

De Novo Sequencing

Chapter 3

De Novo Assemble Small Genome

De Novo Assemble Small Genome facilitates assembly of a microbial genome from short next-generation sequencing reads with high and uniform coverage. It is based on the open source tool SPAdes [Prjibelski et al., 2020].

The tool is equivalent to running SPAdes v4.0.0 with the option `--isolate`. Paired reads with forward-reverse orientation are supplied as paired-end libraries; paired reads with reverse-forward orientation are supplied as high-quality mate-pair libraries; all other types of reads are supplied as one single read library. For more details on the options used by SPAdes, please see https://ablab.github.io/spades/.

The tool requires high and uniform coverage across the genome. High coverage means >50x, though lower values may work satisfactorily. You can estimate the coverage of your data as follows:

- Obtain an estimated size of the genome you intend to assemble.
- Obtain the total number of nucleotides in your input data by running **QC for Sequencing Reads** on your reads, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html.
- Divide the estimated size of the genome by the number of nucleotides in your input data.

For small genomes, **De Novo Assemble Small Genome** will typically produce a higher quality assembly than the **De Novo Assembly** tool from *CLC Genomics Workbench* (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=De_Novo_Assembly.html) but at the cost of taking more time and using more memory. The *CLC Genomics Workbench* tool should be preferred when coverage is not high, or not uniform, or the sample does not consist solely of whole genome sequencing reads from a bacterial or virus isolate.

Before you begin the Assembly

Input Data Quality Good quality data is key to a successful assembly. We strongly recommend using the Trim Reads tool:

- Trimming based on quality can reduce the number of sequencing errors that make their way to the assembler. This reduces the number of spurious words generated

during an initial assembly phase. This then reduces the number of words that will need to be discarded in the graph building stage.

- Trimming adapters from sequences is crucial for generating correct results. Adapter sequences remaining on sequences can lead to the assembler spending considerable time trying to join regions that are not biologically relevant. In other words this can lead to the assembly taking a long time and yielding misleading results.

For requirements for the De Novo Assemble Metagenome tool, see (section 1.3).

3.1 De Novo Assemble Small Genome parameters

To run the De Novo Assemble Small Genome tool, go to:

Tools | **De Novo Sequencing ( )** | **De Novo Assemble Small Genome ( )**

Select one or more sequence lists.

Click **Next** to set the assembly parameters (figure 3.1):

- **Minimum contig length**. The minimum length of contigs included in the output. Shorter contigs will be filtered.
- **Keep circular contigs**. When enabled, the minimum contig length filtering is not applied to circular contigs. This means that all circular contigs will be output regardless of length.

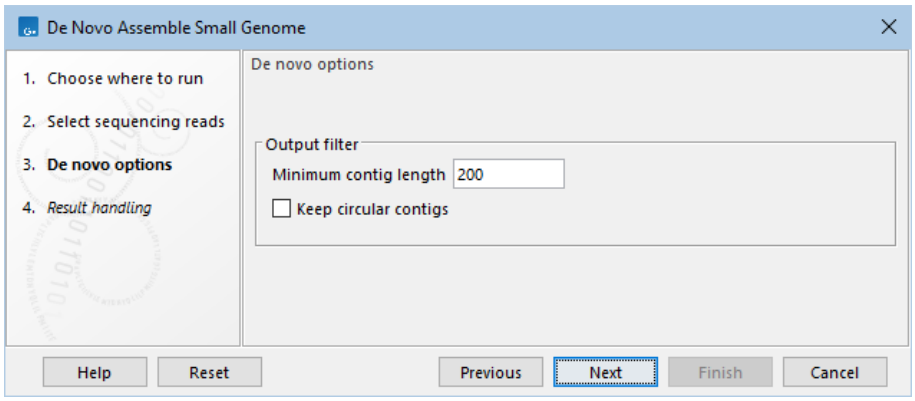


Figure 3.1: *The De Novo Assemble Small Genome options.*

3.2 De Novo Assemble Small Genomes output

The tool outputs a list of contigs and an optional summary report:

Contigs

The main assembly output is a sequence list of contigs . This can also be opened in table view.

De Novo Assemble Small Genome report

The assembly report contains information on the base and length distributions of the contigs. An example of the first sections of the report is shown in figure 3.2.

- **Nucleotide distribution**.
- **Contig measurements**. Statistics about the number and lengths of contigs.
  - **Contigs**. The number of contigs.
  - **Minimum, Maximum, Average**. Minimum, maximum and average contig length.
  - **N50**. The length of the shortest contig in sets of contigs of equal length or longer, where the summed length of contigs is at least 50% of the total contig length. As such, N50 is the shortest contig length that must be included to cover 50% of the assembly.
  - **N90**. The length of the shortest contig in a set of contigs of equal length or longer, where the summed length of contigs is at least 90% of the total contig length. As such, N90 is the shortest contig length that must be included to cover 90% of the assembly. N90 will be equal to or smaller than N50.
  - **Total**. The number of bases in the contigs. This can be used for comparison with the estimated genome size to evaluate how much of the genome sequence is included in the assembly.
- **Contig length distribution**. The number of contigs found at a specific length.
- **Accumulated contig length**. The y-axis shows the summed contig length, while the x-axis represents the number of contigs, arranged with the largest contigs first. This provides insight into the number of contigs required to cover, for instance, half of the genome.

Evaluating and Refining the Assembly

Three key points to look for in assessing assembly quality are contiguity, completeness, and correctness.

Contiguity: How many contigs are there?

A high N50 and low number of contigs relative to your expected number of chromosomes are ideal. If you aren't sure what type of N50 and contig number might be reasonable to expect, you could try to get an idea by looking at existing assemblies of a similar genome, should these exist. For an even better sense of what would be reasonable for your data, you could make comparisons to an assembly of a similar genome, assembled using a similar amount and type of data. If your assembly results include a large number of very small contigs, it may be that you set the minimum contig length filter too low. Very small contigs, particularly those of low coverage, can generally be ignored.

Completeness: How much of the genome is captured in the assembly?

If a total genome length of 5MB is expected based on existing literature or similar genomes that have already been assembled, but the sum of all contig lengths is

only 3.5MB, you may wish to try the De Novo Assembly tool, which has tuneable parameters https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=De_Novo_Assembly.html

Depending on the resources available for the organism you are working on, you might also assess assembly completeness by aligning the assembled contig sequences to a known reference. You can then check for regions of the reference genome that have not been covered by the assembled contigs. Whether this is sensible depends on the sample and reference organisms and what is known about their expected differences.

Correctness: Do the contigs that have been assembled accurately represent the genome?

One key question in assessing correctness is whether the assembly is contaminated with any foreign organism sequence data. To check this, you could run a BLAST search using your assembled contigs as query sequences against a database containing possible contaminant species data.

In addition to BLAST, checking the coverage can help to identify contaminant sequence data. The coverage of a contaminant contig is often different from the desired organism so you can compare the potential contaminant contigs to the rest of the assembled contigs. To check for these types of coverage differences between contigs you may:

- Map your reads used as input for the de novo assembly to your contigs;
- Create a Detailed Mapping Report;
- In the Result handling step of the wizard, check the option to Create separate table with statistics for each mapping;
- Review the average coverage for each contig in this resulting table.

If there are contigs that have good matches to a very different organism and there are discernible coverage differences, you could either consider removing those contigs from the assembly, or run a new assembly after removing the contaminant reads. One way to remove the contaminant reads would be to run a read mapping against the foreign organism's genome and to check the option to Collect unmapped reads. The unmapped reads Sequence List should now be clean of the contamination. You can then use this set of reads in a new de novo assembly.

Assessing the correctness of an assembly also involves making sure the assembler did not join segments of sequences that should not have been joined - or checking for mis-assemblies. This is more difficult. One option for identifying mis-assemblies is to try running the InDels and Structural Variants tool. If this tool identifies structural variation within the assembly, that could indicate an issue that should be investigated.

Post assembly improvements

The **CLC Genome Finishing Module** has been developed to reduce the extensive workload associated with genome finishing and to facilitate as many steps in the procedure as possible. The module can be downloaded from the Workbench Plugin Manager, or from our website at https://digitalinsights.qiagen.com/plugins/clc-genome-finishing-module/. A free trial license is available, as described at https://resources.qiagenbioinformatics.com/manuals/clcgenomefinishing/current/index.php?manual=Licensing_modules.html.

1 Nucleotide distribution

Nucleotide	Count	Frequency (%)
Adenine (A)	918,532	33.26
Cytosine (C)	457,082	16.55
Guanine (G)	446,497	16.17
Thymine (T)	939,571	34.02
Any nucleotide (N)	200	0.01

2 Contig measurements

Contigs	26
Minimum	78
Maximum	606,520
Average	106,226
N50	328,603
N90	61,520
Total	2,761,882

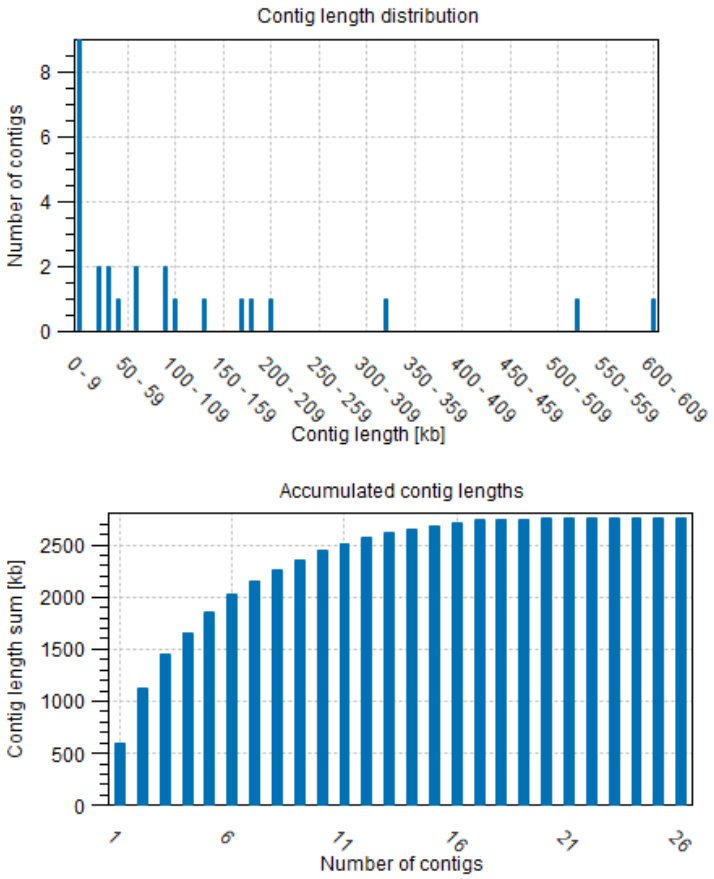


Figure 3.2: *De Novo Assemble Small Genome report*

Chapter 4

De Novo Assemble Metagenome

The De Novo Assemble Metagenome tool is designed for de novo assembly of short sequencing reads from mixed-community samples, such as soil, ocean water, or human gut.
For assembly of single-organism isolates, use the De Novo Assemble Small Genome tool (section 3), or a de novo assembly tool from *CLC Genomics Workbench* (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=De_Novo_sequencing.html).

The De Novo Assemble Metagenome tool produces a list of contigs that can be used for downstream analysis.

Before assembly, adapters should be removed from the sequencing reads using the Trim Reads tool (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_Reads.html). The presence of adapters can result in the assembler trying to join regions that are not biologically relevant, leading to an assembly taking a long time and yielding misleading results.

Quality trimming before assembly is not generally necessary as the assembler can weed out or correct bad quality regions. However, trimming of low quality regions may decrease the amount of memory needed for the de novo assembly, which can be an advantage when working with large datasets.

4.1 De Novo Assemble Metagenome parameters

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Metagenomics** () | **De Novo Assemble Metagenome** ()

Select the sequence lists or single sequences to assemble.

Set assembly parameters (figure 4.1).

- **Minimum contig length**. Contigs below this length will not be reported. For very complex datasets containing reads from many closely related species, the assembler will often produce shorter contigs. For such cases, it is recommended to set a lower threshold in order to cover a larger proportion of the metagenome with contigs. Reversely, for

metagenomes of low complexity, it is often wise to set a higher threshold in order to avoid duplication.

- **Execution mode**
  - **Fast**. The assembler is iterated once with a predifined wordsize ($k = 21$).
  - **Longer contigs**. the assembler is iterated three times with increasing wordsize ($k = 21, 41, 61$), using the contigs from the previous iteration as input in the next iteration together with the input reads.

  Fast mode produces contigs of very high quality very fast, while the Longer contigs mode produces significantly longer contigs, possibly with slightly more misassemblies. Longer contigs mode requires up to three times more computation time.

- **Perform scaffolding**. If selected, as the last step of the assembly process the assembler attempts to join contigs using paired-end information. Since paired-end information is needed to perform scaffolding, this option is disabled for single-end sequences.

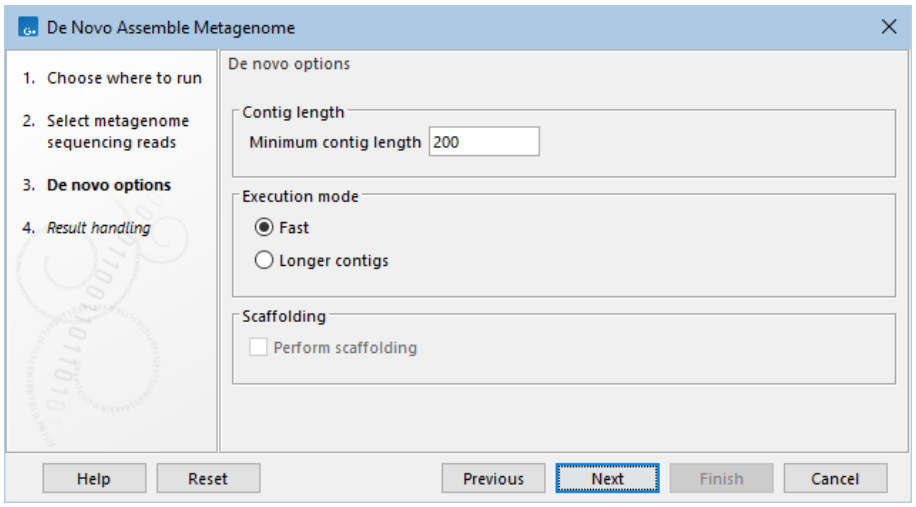


Figure 4.1: *Setting parameters for the assembly.*

4.2 De Novo Assemble Metagenome output

The tool outputs a list of contigs and an optional assembly report.

Contigs

The main assembly output is a sequence list of contigs. This can also be opened in table view.

If **Perform scaffolding** was selected, scaffolds will appear at the bottom of the contig list as scaffold_1, scaffold_2, etc.

De Novo Assemble Metagenome assembly report

The assembly summary report contains statistics on reads and contigs (figure 4.2).

- **Basic statistics on input reads**.
- **Distribution of nucleotide reads**.
- **Distribution of read lengths**.
- **Basic statistics on contigs**. This section includes statistics about the number and lengths of contigs.
  - **Number of contigs**.
  - **Number of contigs > 1kb**.
  - **Total length of contigs**.
  - **Total length of contigs > 1kb**.
  - **Minimum, maximum, mean, and median contig length**.
  - **N10, N25, N50, N75 and N90**. The N25 contig set is calculated by adding up the lengths of the biggest contigs until you reach 25 % of the total contig length. The minimum contig length in this set is the number that is usually used to report the N25 value of a de novo assembly. Likewise for the remaining N values.
  - **Number of Ns per 100kb**.
- **Distribution of nucleotides in contigs**.
- **Contig length distribution**. The number of contigs found at a specific length.
- **Accumulated contig reads**. This shows the summed up contig length on the y axis and the number of contigs on the x axis, with the biggest contigs ranked first. This answers the question: how many contigs are needed to cover e.g., half of the genome.

1.5 Basic statistics on contigs

Measurement	Length or count
Number of contigs	23
Number of contigs > 1kb	19
Total length of contigs	676,346
Total length of contigs > 1kb	673,743
Minimum contig length	228
Maximum contig length	165,261
Mean contig length	29,406
Median contig length	7,993
N10	165,261
N25	105,633
N50	88,924
N75	46,911
N90	29,619
Number of Ns per 100kb	5.91

1.6 Distribution of nucleotides in contigs

Nucleotide	Count	Frequency
Adenine (A)	144,829	21.4%
Cytosine (C)	198,674	29.4%
Guanine (G)	189,567	28.0%
Thymine (T)	143,236	21.2%
Any nucleotide (N)	40	0.0%

Figure 4.2: *The De Novo Assemble Metagenome report, selected sections.*

Part IV

Metagenomics

Chapter 5

Amplicon-Based Analysis

In the Amplicon-Based Analysis folder you will find tools for analyzing amplicon data.

With **OTU Clustering** and accompanying OTU tools, you can cluster reads at e.g. 97% similarity, into so-called Operational Taxonomic Units (OTUs).

Detect Amplicon Sequence Variants offers a higher resolution alternative. It uses error profiling to distinguish biological nucleotide differences from sequencing errors.

Template workflows for amplicon-based analysis are available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis ( )**

For more information on the template workflows, see section 2.2.

5.1 Normalize OTU Table by Copy Number

One way to correct OTU abundance tables is to take the rRNA copy number of the detected species into account and divide the detected read number for each OTU by the rRNA copy number. This can be done with the Normalize OTU Table by Copy Number (beta) tool. Note that this algorithm corrects the species distribution for each sample individually to get a more realistic picture of the species distribution in a sample. In order to normalize OTU abundance tables across samples one can use the Create Normalized Abundance Subtable button (see section 5.3.2), but many tools use an internal cross-sample normalization strategy.

In order to run this tool, an Amplicon Multiplication Table is required. Such tables can be imported with Import PICRUSt2 Multiplication Table (beta) 17.6.

To run the tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Normalize OTU Table by Copy Number (beta)** ( )

Select the OTU table you want to normalize first and click "Next".

In the next wizard step (see figure 5.1) you can choose the Amplicon Multiplication Table to use.

The normalization works in the same way as the functional inference, except for that the functional

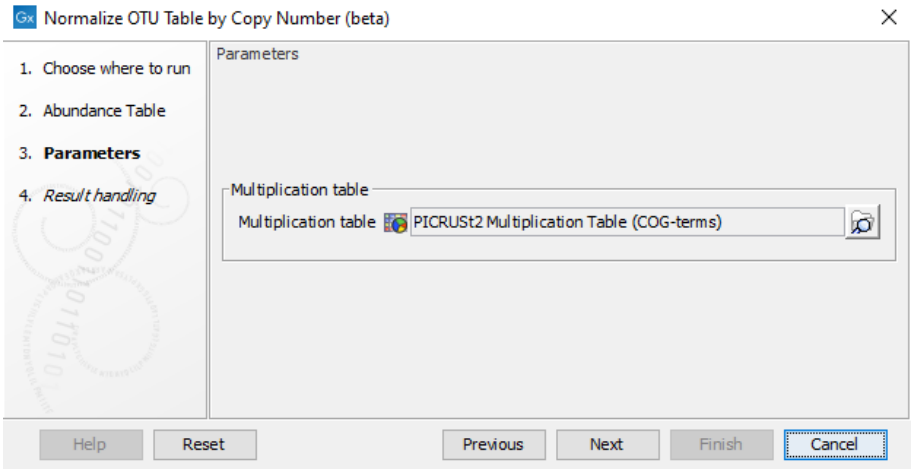


Figure 5.1: *Selecting an Amplicon Multiplication Table for normalizing OTU tables.*

inference step is left out, see section 12.8.

5.2 Filter Samples Based on Number of Reads

In order to cluster accurately samples, they should have comparable coverage. Sometimes, however, DNA extraction, PCR amplification, library construction or sequencing has not been entirely successful, and a fraction of the resulting sequencing data will be represented by too few reads. These samples should be excluded from further analysis using the **Filter Samples Based on Number of Reads** tool.

To run the tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Filter Samples Based on Number of Reads** ( )

The tool requires that the input reads from each sample must be either all paired or all single. This check ensures that the samples are comparable, as the number of reads before merging paired reads is twice as great as the number of merged reads.

The threshold for determining whether a sample has sufficient coverage is specified by the parameters **minimum number of reads** and **minimum percent from the median**. The algorithm filters out all samples whose number of reads is less than the **minimum number of reads** or less than the **minimum percent from the median** times the median number of reads across all samples.

The primary output is a table describing how many reads are in a particular sample and if they passed or failed the quality control (see figure 5.2).

1 Number of reads

Sample	Number of reads	Notes
GT-A-A_L001_R1_001 (paired) merged trimmed fixedLength	855	Number of reads too low
GT-A-B_L001_R1_001 (paired) merged trimmed fixedLength	6304	Passed
GT-A-C_L001_R1_001 (paired) merged trimmed fixedLength	10432	Passed
GT-B-A_L001_R1_001 (paired) merged trimmed fixedLength	7283	Passed

Figure 5.2: *Output table from the Filter Samples Based on Number of Reads tool.*

In the next wizard window you can decide to **Copy samples with sufficient coverage** as well as to **Copy the discarded samples**. Copying the samples with sufficient coverage will give you a new list of sequences that you can use in your following analyses.

5.3 OTU clustering

The OTU clustering tool clusters a collection of reads to operational taxonomic units.

To run the tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **OTU clustering** ( )

The tool aligns the reads to reference OTU sequences (e.g. the reference database) to create an "alignment score" for each OTU. If the input sequence is shorter, the unaligned ends of the reference are ignored. For example, if a shorter sequence has 100% identity to a fragment of a longer reference sequence, the tool will assign 100% identity and assign the read to the OTU.

In the opposite case (longer read mapping to short database reference), the unaligned ends will count as indels, and the percentage identity will be lower.

When the input consists of paired reads, the OTU clustering tool will initially group them into pairs, and align both reads of a pair to the same OTUs. Both reads of a pair will be assigned to the one OTU where they BOTH align with the highest identity possible. Finally, the tool merges both reads of the pair using a stretch of N to the fragments so that the paired read looks as much as possible like the OTU they have been assigned to. For example, the forward-reverse pair (ACGACGACG, GTAGTAGTA) will be turned into ACGACGACGnnnnnnnnnnnnnnnnnnnnTACTACTAC. Reads that cannot be merged will be independently aligned to reference OTUs.

If a read due to insufficient similarity cannot be included in an already existing OTU, the algorithm attempts to optimize the alignment score by allowing "crossover" from one database reference to another at a cost (the chimera crossover cost). To speed up the chimera crossover detection algorithm, the read is not aligned to all OTUs but only to the most promising candidates found via a k-mer search. If the best match has at least one crossover and the "constructed alignment" meets the similarity percentage threshold, the read is considered chimeric.

By default, the similarity percentage parameter is set to 97% in the OTU Clustering tool. Therefore without the chimera crossover cost, the constructed alignments difference score can only be 3% at most. The smaller the chimeric cost, the more likely it is that a read is deemed chimeric; setting it too high decreases the chimeric detection.

To add samples to existing OTU clustering results, we recommend to run OTU clustering on the new samples separately and use the tool Merge Abundance Tables described in section section 7.1 to merge the OTU tables. If re-running analysis is necessary and you wish to compare with previous results, you should keep the original sample input order. Due to the iterative nature of the clustering algorithm, changing the order of input files can lead to slightly different results. In most conceivable cases that difference does not matter, specifically when using taxonomy informed and abundance weighted distance metrics like weighted UniFrac.

5.3.1 OTU clustering parameters

After having selected the sequences you would like to cluster, the wizard offers to set some general parameters (see figure 5.3).

You can choose to perform a **De novo OTU clustering**, or you can perform a **Reference based OTU clustering**.

The following parameters can be set:

- **OTU database** Specify the reference database to be used for Reference based OTU clustering. Reference databases can be created by the **Download Amplicon-Based Reference Database** tool or the **Update Sequence Attributes in Lists** tool (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Update_Sequence_Attributes_in_Lists.html).
- **Similarity percent specified by OTU database** Will apply the same similarity percentage (see below) as what was used when creating the reference database. This parameter is available only when performing a reference based OTU clustering. Selecting this parameter will disable the similarity percent parameter.

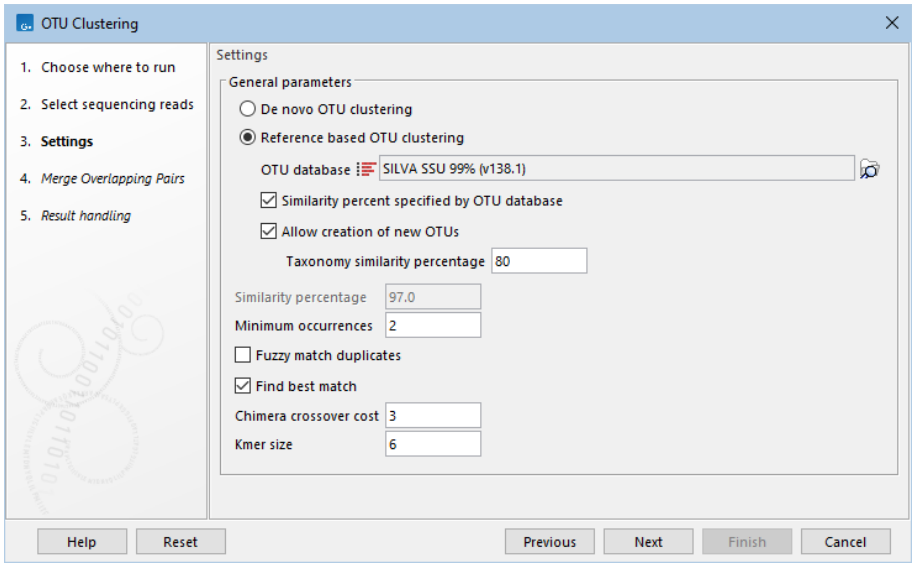


Figure 5.3: *Settings for the OTU clustering tool.*

- **Allow creation of new OTUs** Allows sequences which are not already represented at the given similarity distance in the database to form a new cluster, and a new centroid is chosen. This parameter can be set only when performing a "Reference-based OTU clustering". Disallowing the creation of new OTUs is also known as closed reference OTU picking. Note that for input data where reads do not have the same orientation, the direction of the new OTUs cannot be inferred consistently. This may cause problems in downstream analyses (e.g. for estimating phylogenetic diversity).

- **Taxonomy similarity percentage** Specifies the similarity percentage to be used when annotating new OTUs. This parameter is available only when **Allow creation of new OTUs** is selected.

- **Similarity percentage** Specifies the required percentage of identity between a read and the centroid of an OTU for the read to join the OTU cluster.

- **Minimum occurrences** Specifies the minimum number of times a sequence must be represented in the read set for it to be included in the analysis. A value of 2 means that at least two reads representing a given sequence (i.e. duplicates) must be present for that sequence to be represented in further analysis. This option can be useful for filtering out singletons.

- **Fuzzy match duplicates** Specifies how to define duplicate reads. When not selected, reads that are 100% identical are considered duplicates. When selected, reads with 2% or fewer single nucleotide differences between them, and no other differences, are considered duplicates. The reads are sorted lexicographically (dictionary order) and then processed from most abundant to the least. Using this option, two or more singleton reads that are very similar may be marked as duplicates, allowing them to be included in further processing if together, their number exceeds the "Minimum occurrences" value.

- **Find best match** If not selected, a read becomes a member of the first OTU-database entry found within the specified threshold. If the option is selected all database entries are tested and the read becomes a member of the best matching result. Note that "first" and

"all" are relative terms in this case as kmer-searches are used to speed up the process. "All" only includes the database entries that the kmer search deems close enough, i.e., database entries that cannot be within the specified threshold will be filtered out at this step. "First" is the first matching entry as returned by the kmer-search which will sort by the number of kmer-matches.

- **Chimera crossover cost** The cost of doing a chimeric crossover, i.e. the higher the cost the less likely it is that a read is marked as chimeric.
- **Kmer size**: The size of the kmer to use in regards to the kmer usage in finding the best match.

Chimera detection is performed as follows: The read being processed is split into fragments. Each fragment is then queried for matches against the database with a k-mer search. Database references that match at least one query fragment are then selected and the read is then aligned to each selected reference while allowing "crossovers". Chimera detection is performed in order to identify any chimeric sequences, i.e., amplicons formed by joining two sequences during PCR. These are artifacts that will be excluded from the regular OTU clustering, and presented in a different abundance table labeled as being chimera-specific.

In order to use the highest quality sequences for clustering, it is recommended to merge paired read data. If the read length is smaller than the amplicon size, forward and reverse reads are expected to overlap in most of their 3' regions. Therefore, one can merge the forward and reverse reads to yield one high quality representative according to some pre-selected merge parameters: the overlap region and the quality of the sequences. For example, for a designed 150 bp overlap, a maximum score of 150 is achievable, but as the real length of the overlap is unknown, a lower minimum score should be chosen. Also, some mismatches and indels should be allowed, especially if the sequence quality is not perfect. You can also set penalties for mismatch, gap and unaligned ends.

In the Merge Overlapping Pairs dialog, you can set the parameters as seen in figure 5.4.

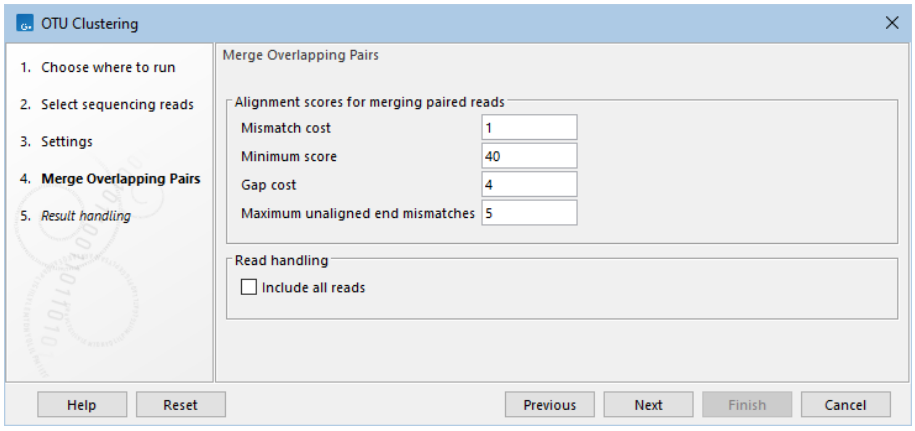


Figure 5.4: *OTU Clustering parameters for merging of overlapping pairs.*

In order to understand how these parameters should be set, an explanation of the merging algorithm is needed: Because the fragment size is not an exact number of base pairs and is different from fragment to fragment, an alignment of the two reads has to be performed. If the alignment is *good and long enough*, the reads will be merged. *Good enough* in this context means

that the alignment has to satisfy some user-specified score criteria (details below). Because of sequencing errors that typically are more abundant towards the end of the read, the alignment is not expected always to be perfect, and the user can decide how many errors are acceptable. *Long enough* in this context means that the overlap between the reads has to be non-coincidental. Merging two reads that do not really overlap leads to errors in the downstream analysis, thus it is very important to make sure that the overlap is big enough. If only a few bases overlap was required, some read pairs will match by chance, so this has to be avoided.

The following parameters are used to define what is *good enough* and *long enough*.

- **Mismatch cost** The alignment awards one point for a match, and the mismatch cost is set by this parameter. The default value is 1.
- **Minimum score** This is the minimum score of an alignment to be accepted for merging. The default value is 40. As an example: with default settings, this means that an overlap of 43 bases with one mismatch will be accepted (42 matches minus 1 for a mismatch).
- **Gap cost** This is the cost for introducing an insertion or deletion in the alignment. The default value is 4.
- **Maximum unaligned end mismatches**: The alignment is local, which means that a number of bases can be left unaligned. If the quality of the reads is dropping to be very poor towards the end of the read, and the expected overlap is long enough, it makes sense to allow some unaligned bases at the end (the default value is 5). However, this should be used with great care: a wrong decision to merge the reads leads to errors in the downstream analysis, so it is better to be conservative and accept fewer merged reads in the result. Please note that even with the alignment scores above the minimum score specified in the tool setup, the paired reads also need to have the number of end mismatches below the "Maximum unaligned end mismatches" value specified in the tool setup to be qualified for merging.

The tool accepts both paired and unpaired reads but will only merge paired reads in forward-reverse orientation. After merging, the merged reads will always be in the forward orientation.

- **Include all reads** Select this option to include the non-merged reads in the OTU clustering analysis. If some or most of your paired reads are not expected to overlap, you should check this option to include all reads in the analysis. An example of an application resulting in non-overlapping paired reads would be fungal ITS sequencing, where you often sequence a larger amplicon than what can be covered by your read pairs.

5.3.2 OTU clustering outputs

Click **Next** to select outputs (figure 5.5).

In addition to the OTU abundance table, the following outputs are available:

- A sequence list of the OTUs
- A chimera abundance table with abundances for chimeras in each sample.
- A report that summarizes the results of the OTU clustering. For paired-end data, the report will include a section about the merging of overlapping paired reads.

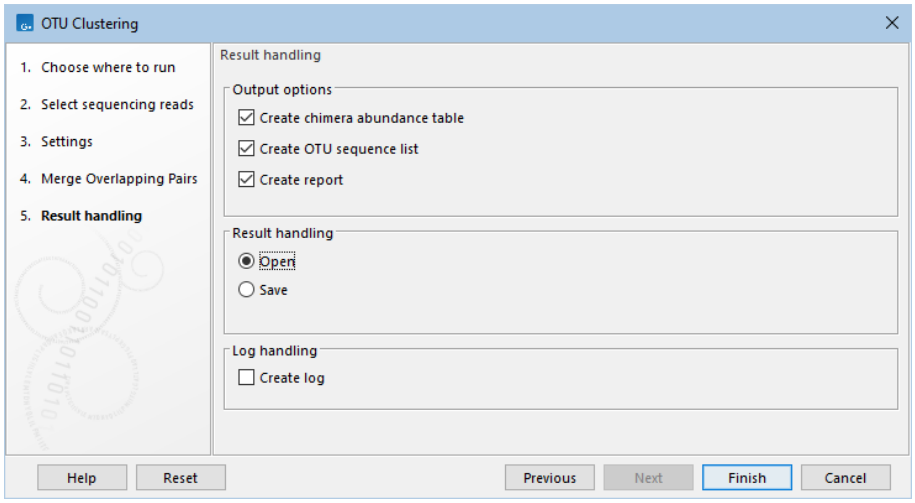


Figure 5.5: *OTU Clustering output options*

The OTU report

An example of an OTU report is shown in figure 5.6. The report contains the following sections:

- **OTU clustering**
  - **Input database size** The number of sequences in the input OTU database.
  - **Filtered database size** The number of sequences in the input OTU database having input reads mapped to it.
  - **OTUs based on database** The number of OTUs based on a sequence from the database.
  - **De novo OTUs** The number of OTUs not based on a sequence from the database.
  - **Total predicted OTUs** The total number of OTUs found.
- **Reads**
  - **Number of reads** The number of input reads
  - **Filtered reads** The number of reads filtered due to the minimum occurrences parameter. When reads are not at a specified similarity distance with the database, and the option to create new OTUs is not selected, these reads will be filtered as well.
  - **Unique reads after filtering** The number of unique reads after filtering. This is the number of candidates for OTUs before clustering.
  - **Chimeric reads** The number of reads detected as chimeric during clustering.
  - **Unique chimeric reads** The number of unique reads detected as chimeric.
  - **Reads in OTUs** The number of reads that contribute to the output OTUs.
- **Sample details**
  - **Sample** The name of the sample for which the following details are shown.
  - **Total number of reads** The number of input reads from the given sample.

  - **Filtered or chimeric reads** The number of reads from the given sample that were filtered due to the minimum occurrences parameter or detected as chimeric during clustering.
  - **Reads in OTUs** The number of reads from the given sample that contribute to the output OTUs.

- **Merging of paired reads** the following is reported for each input sample (generated if the input reads were paired)
  - **Summary** The number of merged, not merged and total paired reads.
  - **Merged pairs length distribution** Distribution of the lengths of the read pairs with the length of a read in base pairs on the x-axis and on the y-axis in the number of times a read of a given lengths has been observed.

The OTU abundance table

The OTU abundance table contains a list of OTUs, per-sample abundance values, and total abundance counts. Note that if the input contains paired-end sequences, each pair is counted as one read. There are a number of ways to visualize the contents of an OTU abundance table:

- **Table view** () (figure 5.7)

  The table displays the following columns:
  - **Name** The name of the OTU, specified by either the reference database or by the OTU representative (see below for more details).
  - **Taxonomy** The taxonomy of the OTU, as specified by the reference database when a database entry was used as Reference.
  - **Combined Abundance** The total number of reads belonging to the OTU across all samples.
  - **Min** Minimum abundance across all samples
  - **Max** Maximum abundance across all samples
  - **Mean** Mean abundance of all samples
  - **Median** Median abundance of all samples
  - **Std** Standard deviation of all samples
  - **Abundance for each sample** The number of reads belonging to the OTU in a specific sample.
  - **Sequence** The sequence of the centroid of the OTU.

  Note on OTU Names: The name is either
  - The OTU name in the reference database (e.g. 978664)
  - The name of the read used as centroid, which for sequencing data may look like random numbers and letters. If the same name is present more than once, then the OTUs will have a trailing number "-00123" like readName-12345.
  - If there is no name (for new clusters where reads have no name), something like OTU-12345 is assigned.

This will occur when one chooses the option "De novo OTU clustering" in the General parameters section of the OTU Clustering wizard, or the option "Allow creation of new OTUs". When either of these options are selected, it will be possible for the OTU clustering tool to create representative OTU sequences that are not in an existing reference database.

In the right side panel, under the tab Data, you can switch between absolute counts and relative abundances (relative abundances are computed as the ratio between the number of reads belonging to the OTU in a specific sample and the total number of reads in the sample). You can also combine absolute counts and relative abundances by taxonomic levels by selecting the appropriate phylum in the **Aggregate feature** drop-down menu. Use the option below to Hide samples for which the taxonomy at the aggregated taxonomic level is incomplete. Finally, if you have previously annotated your table with Metadata (see section 7.9), you can **Aggregate sample** by the groups previously defined in your metadata table. This is useful when analyzing replicates from the same sample origin.

Under the table, the following actions are available:

- **Create Abundance Subtable** will create a table containing only the selected rows.
- **Create Sequence Sublist** will create a sequence list containing only the selected rows.
- **Create Normalized Abundance Subtable** will create a table with all rows normalized on the values of a single selected row. The row used for normalization will disappear from the new abundance table. The normalization scales the abundance table linearly, where the scaling factor is calculated by determining the average abundance across all samples and for each sample scale it to the average for the reference. Note that to be enabled, the selected row for normalization can only have non null abundance values. If you have zero values in some samples for the control, you will need to generate a new abundance table where these samples are not present. If the abundance table is obtained from merging single-sample abundance tables, then the merge should be redone excluding the samples with zero control read counts.

- **Stacked visualization view** ( )

  In the Stacked Bar (figure 5.8) and Stacked Area Charts (figure 5.9), the metadata can be used to aggregate groups of columns (samples) by selecting the relevant metadata category in the right hand side panel. Also, the data can be aggregated at any taxonomy level selected. The relevant data points will automatically be summed accordingly.

  Holding the pointer over a colored area in any of the plots will result in the display of the corresponding taxonomy label and counts. With **Filter level** you can modify the number of features shown in the plot. For example, setting the value to 10 means that the 10 most abundant features of each sample will be shown in all columns. The remaining features are grouped into "Other", and will be shown if the option is selected in the right hand side panel. One can select which taxonomy level to color, and change the default colors manually. Colors can be be specified at the same taxonomy level as the one used to aggregate the data or at a lower level. When lower taxonomy levels are chosen in the data aggregation field, the color will be inherited in alternating shadings. It is also possible to sort samples by metadata attributes, and to show groups of samples without collapsing their stacks, as well as change the label of each stack or group of stacks. Features can be sorted by "abundance" or "name" using the drop down menu in the right hand side panel. Using the bottom right-most button (**Save/restore settings** ( )), the settings can be saved and applied in other plots, allowing visual comparisons across analyses.

- **The sunburst view** ( )

  The zoomable sunburst view lets the user select how many taxonomy level counts to display, and which level to color. Lower levels will inherit the color in alternating shadings. Taxonomy and relative abundances (the ratio between the number of reads belonging to the OTU in a specific sample and the total number of reads in the sample) are displayed in a legend to the left of the plot when hovering over the sunburst viewer with the mouse. The metadata can be used to select which sample or group of samples to show in the sunburst (figure 5.10).

  Clicking on a lower level field will render that field the center of the plot and display lower level counts in a radial view. Clicking on the center field will render the level above the current view the center of the view (figure 5.11).

5.3.3 Importing and exporting OTU abundance tables

It is possible to import a biom, a csv or an excel file as an OTU abundance table, by going to **File | Import ( ) | Standard Import... ( )** and force the input as type "OTU abundance table (.xls, .xlsx, .csv)" or "Biom (.biom)". Currently supported versions for BIOM file format are versions 1.0 and 2.1.

This importer allows users to perform statistical analyses on abundance tables that were not generated by OTU clustering tool. Note that abundance tables that are imported will not contain metadata or grouping information, and thus metadata has to be re-applied using the Add Metadata to Abundance Table tool after import.

For example, Terminal Restriction Fragment Length Polymorphism (TRFLP) data can be imported and treated similarly as OTU abundance tables. However, all sequence-based actions cannot be applied to this data (i.e., multiple sequence alignment, tree reconstruction and phylogenetic tree measure estimation).

The importer recognizes the following column headers:

- **Name** The name of the OTU, specified by either the reference database or by the OTU representative.
- **Taxonomy** The taxonomy of the OTU, as specified by the reference database when a database entry was used as reference, e.g "Bacteria; Bacillota; Bacilli; Lactobacillales; Lactobacillaceae; Lactobacillus; Lactobacillus gasseri".
- **Sequence** The sequence of the centroid of the OTU.
- Any other header of a column with integer values: The header is interpreted as sample name and the values as abundance values. Values must be absolute counts and not relative abundances.

It is furthermore possible to export abundance tables to different formats, but it is recommended to use the Biological Observation Matrix (biom) file format (http://biom-format.org) as a standardized format. Currently, the only supported version for export is 2.1.

Sunbursts graphs can be exported in the following formats: *.jpg, *.tif, *.png, *.ps, *.eps, *.svg.

1 OTU clustering

Input database size	99,322
Filtered database size	694
OTUs based on database	479
De novo OTUs	137
Total predicted OTUs	616

2 Reads

Number of reads	63,823
Filtered reads	57,176
Unique reads after filtering	2,605
Chimeric reads	122
Unique chimeric reads	58
Reads in OTUs	6,525

3 Sample details

Sample	Total number of reads	Filtered or chimeric reads	Reads in OTUs
CrimeSite1-replicateA (paired)	9,387	8,237	1,150
CrimeSite1-replicateB (paired)	10,209	9,037	1,172
Site2-replicateA (paired)	9,543	8,588	955
Site2-replicateB (paired)	12,086	10,712	1,374
Site3-replicateA (paired)	13,591	12,531	1,060
Site3-replicateB (paired)	9,007	8,193	814

4 Merging of paired reads

4.1 Sample CrimeSite1-replicateA (paired)

4.1.1 Summary

	Number of reads	Percentage
Merged	15,336	81.69%
Not merged	3,438	18.31%
Total	18,774	100%

4.1.2 Merged pairs length distribution

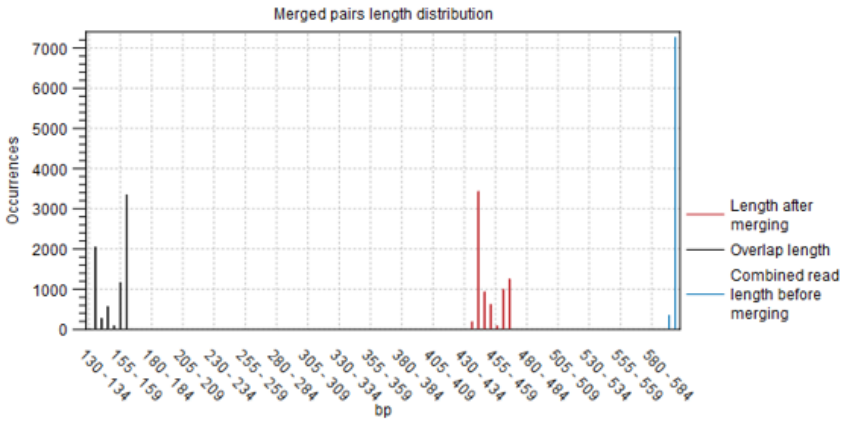


Figure 5.6: *Example of report produced by the OTU clustering tool.*

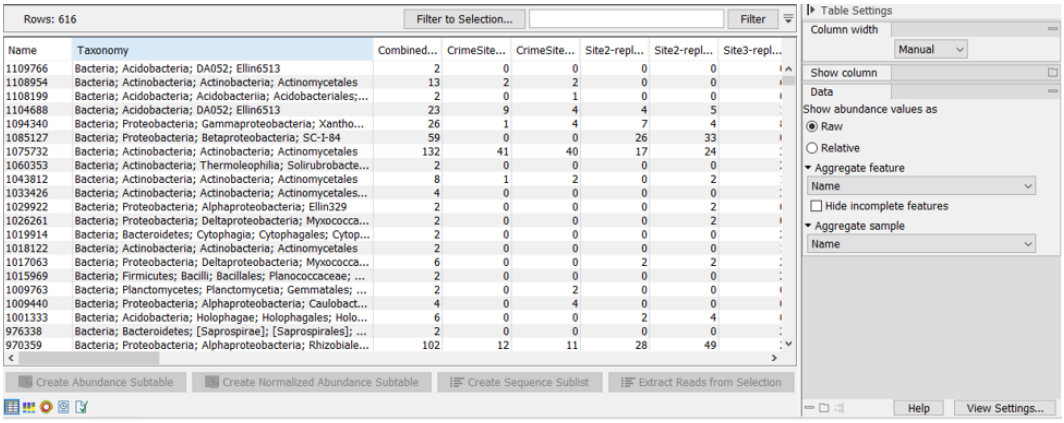


Figure 5.7: *OTU abundance table.*

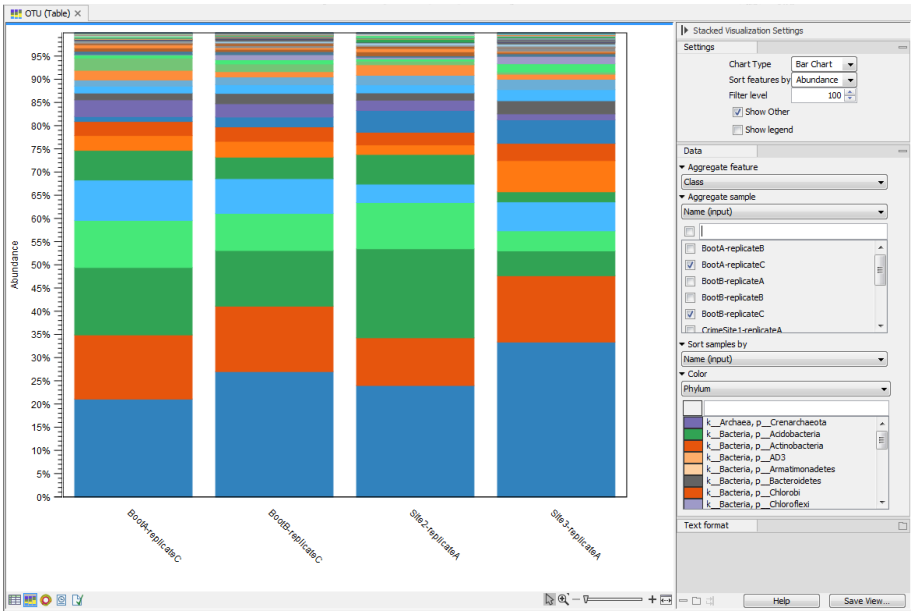


Figure 5.8: *Stacked bar of the microbial community at the class level for 4 different samples.*

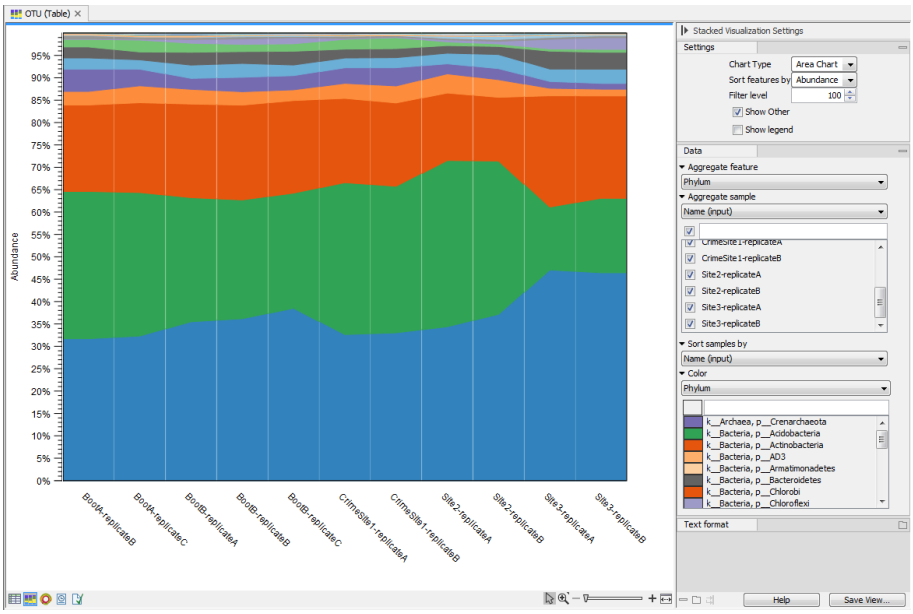


Figure 5.9: *Stacked area of the microbial community at the phylum level for 11 different sites.*

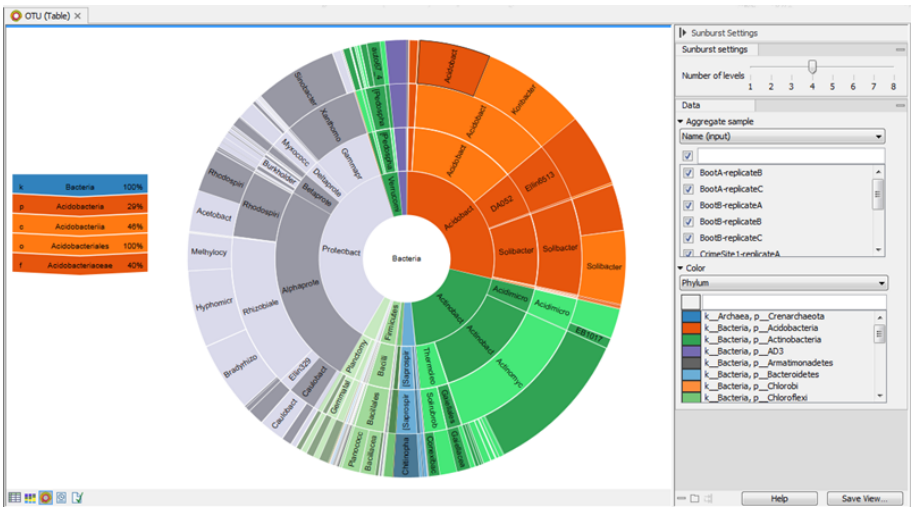


Figure 5.10: *Sunburst view of the microbial community showing all taxa belonging to the kingdom bacteria.*

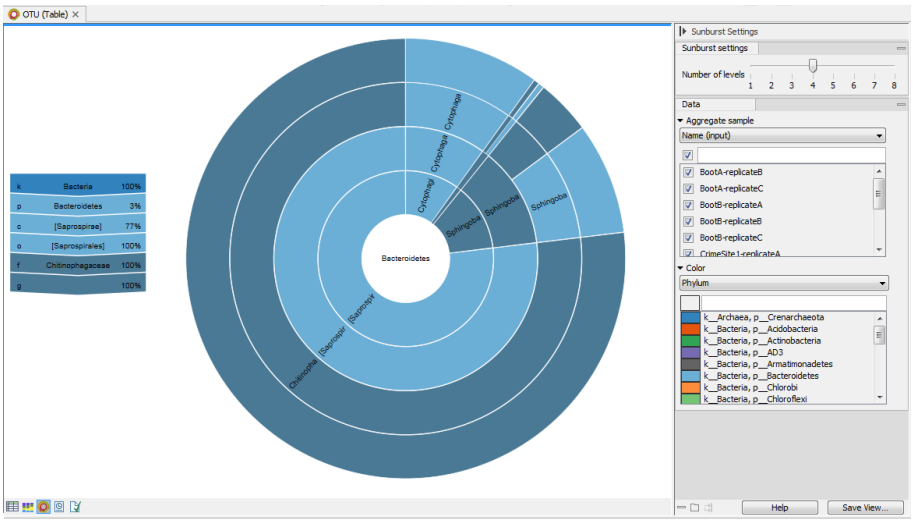


Figure 5.11: *Sunburst view of the microbial community zoomed to show all taxa belonging to the phylum Bacteroidetes.*

5.4 Align OTUs using MUSCLE

To estimate Alpha and Beta diversity, OTUs are initially aligned with MUSCLE using the **Align OTUs using MUSCLE** tool. To launch this tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Align OTUs using MUSCLE** ( )

Choose an OTU abundance table as input. The next wizard window allows you to set up the alignment parameters with MUSCLE (figure 5.12).

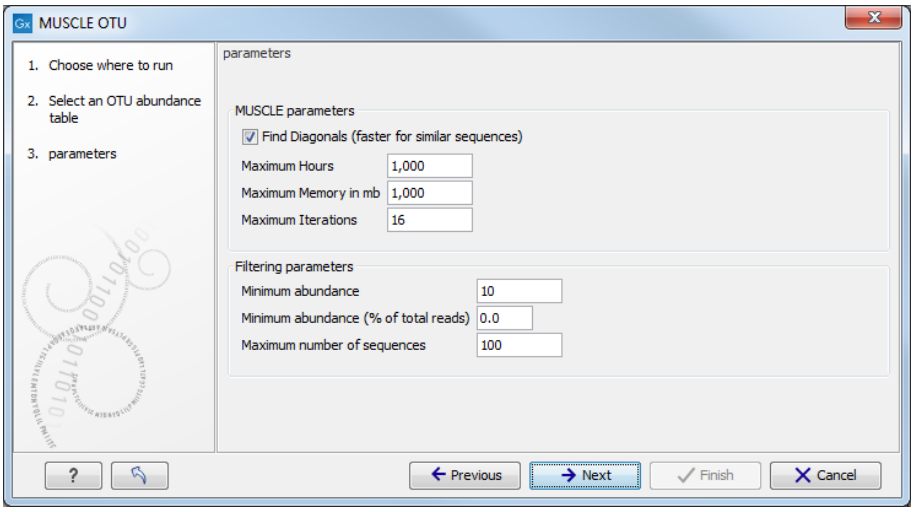


Figure 5.12: *Set up parameters for aligning sequences with MUSCLE.*

- **Find Diagonals**: you can decide on some restrictive parameters for your analysis: the **Maximum Hours** the analysis should last, the **Maximum Memory in mb** that should be used for the analysis, or the **Maximum Iterations** the analysis should make. The latter is set to 16 by default.

- **Filtering Parameters**: The algorithm filters out all OTUs whose combined abundance across all samples is less than the **minimum combined abundance** or whose combined abundance is less than the **minimum combined abundance (% of all the reads)** across all samples. The default value for the Minimum combined abundance is set at 10. Moreover, you can specify the **Maximum number of sequences to be aligned**, so that only the sequences with the highest combined abundances will be used. **Note** that reducing the number of sequences will speed up the alignment and the construction of phylogeny trees.

Note that by default only the top 100 most abundant OTUs are aligned using MUSCLE and used to reconstruct the phylogeny tree in the next step. This phylogenetic tree is used for calculating the phylogenetic diversity and the UniFrac distances, so these measures disregard the low abundance OTUs by default. If more OTUs are to be included, the default settings for the MUSCLE alignment need to be changed accordingly.

For further analysis with the Alpha and Beta diversity tools, save the alignment and construct a phylogenetic tree using the **Maximum Likelihood Phylogeny** tool, available at:

> **Tools** | **Classical Sequence Analysis** ( ) | **Alignments and Trees** ( )| **Maximum Likelihood Phylogeny** ( )

For more information, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Maximum_Likelihood_Phylogeny.html.

5.5 Detect Amplicon Sequence Variants

The Detect Amplicon Sequence Variants tool infers sequence variants from amplicon data. The tool uses error profiling to distinguish biological nucleotide differences from sequencing errors, making it possible to resolve amplicon sequence variants (ASVs) down to the level of single nucleotide differences. The algorithm is inspired by DADA2, [Callahan et al., 2016].

The Detect Amplicon Sequence Variants analysis includes the following steps:

- Initial filtering and length trimming ensures that reads are of the same length and optimized for the subsequent analysis:
  - **Length trimming** Reads are trimmed from the 3' end to the user-defined length. Reads shorter than this are removed.
  - **Ambiguity filter** Reads containing ambiguous bases are discarded.
  - **Expected Errors filter** Reads with more expected errors than the user-defined threshold are discarded.
- **Dereplication** Produces an intermediate list of unique sequences.
- **Denoising** This iterative process estimates a sample-specific error model. This error model is then used to distinguish biological nucleotide differences from likely sequencing errors and generate the list of candidate amplicon sequence variants.
- **Remove chimeras** Sequences that are assessed as being chimeras are discarded.
- **Merging unique read pairs** For paired read dataset, unique read pairs are merged. Pairs with insufficient overlap (<12 bases), are discarded.

A template workflow with a proposed analysis pipeline - trimming reads, detecting amplicon sequence variants, merging ASV tables, and assigning taxonomies - is available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Detect Amplicon Sequence Variants and Assign Taxonomies workflow** ( )

For more information, see section 2.2.2.

5.5.1 Detect Amplicon Sequence Variants parameters

To run the **Detect Amplicon Sequence Variants** tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Detect Amplicon Sequence Variants** ( )

Select the single- or paired-end sequence lists to be analyzed. For paired reads, pairs should have a minimum overlap of 12 bases. Data should be trimmed beforehand to remove adapters and poor quality nucleotides. This can be done using **Trim Reads** (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_Reads.html).

Set the Trim and filter parameters (figure 5.13):

- **First/Second read length**: Reads are trimmed to the given length from the 3' end. Reads shorter than this will be discarded. For paired reads you may set different values for the first and second read in a pair.

  What values to set will depend on your sequencing protocol and how reads are trimmed prior to being used as input for the Detect Amplicon Sequence Variants tool. We recommend that you have a look at the Trim report section *Read length before / after trimming* if you are unsure about what value to set.

  You must use the same length setting for all samples that will be compared in downstream analysis.

- **Maximum expected errors per read**: The maximum number of expected errors allowed for a read. Reads with more expected errors will be discarded.

- **Remove chimeras**: If selected, reads identified as chimeras will be discarded.

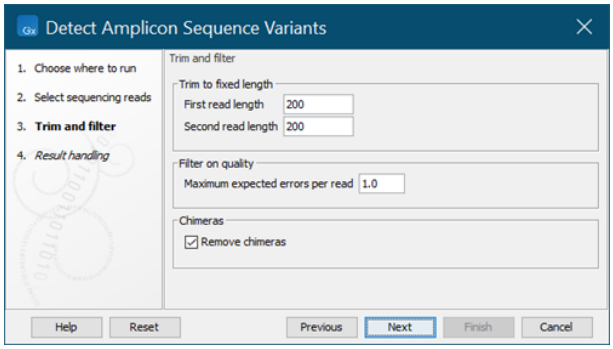


Figure 5.13: *Trim and filter parameter settings*

5.5.2 Detect Amplicon Sequence Variants output

Click **Next** to select the output (figure 5.14).

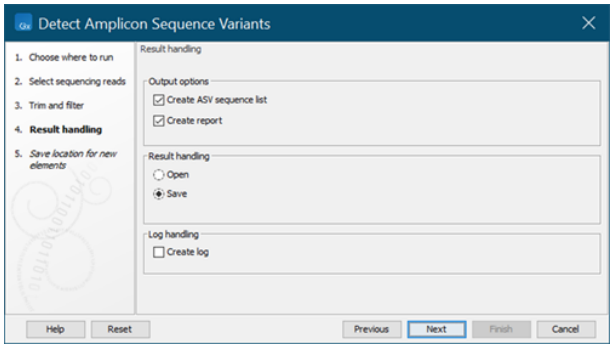


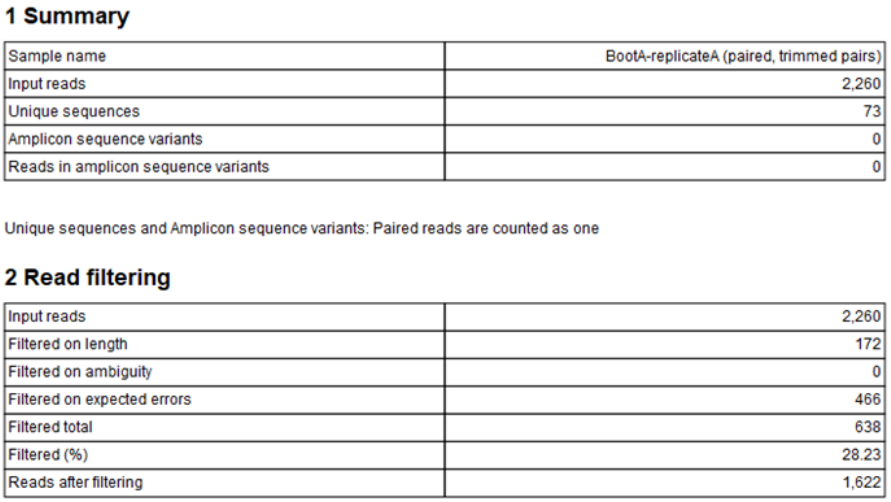
Figure 5.14: *Detect Amplicon Sequence Variants output options*

In addition to an ASV abundance table, the following outputs are available:

- Click **Create ASV sequence list** to generate a sequence list with the detected amplicon sequence variants.

- Click **Create report** to generate a summary report.

The ASV report



1 Summary

Sample name	BootA-replicateA (paired, trimmed pairs)
Input reads	2,260
Unique sequences	73
Amplicon sequence variants	0
Reads in amplicon sequence variants	0

Unique sequences and Amplicon sequence variants: Paired reads are counted as one

2 Read filtering

Input reads	2,260
Filtered on length	172
Filtered on ambiguity	0
Filtered on expected errors	466
Filtered total	638
Filtered (%)	28.23
Reads after filtering	1,622

Figure 5.15: *First sections of the Detect Amplicon Sequence Variants report*

- **Summary** (figure 5.15)
  - **Sample name** The name of the sample.
  - **Input reads** The number of input reads.
  - **Unique sequences** The number of unique sequences detected in the input reads. Read pairs are counted as one.
  - **Amplicon sequences variants** The number of amplicon sequences variant detected in the input reads. Read pairs are counted as one.
  - **Reads in amplicon sequence variants** The number of reads grouped into ASVs.
- **Read filtering**
  - **Input reads** The number of input reads.
  - **Filtered on length** The number of reads that were removed because they were shorter than the defined length threshold.
  - **Filtered on ambiguity** The number of reads that were removed because they contained ambiguous bases.
  - **Filtered on expected errors** The number of reads that were removed because they exceeded the *Maximum expected errors* threshold.
  - **Filtered total** The total number of filtered reads.
  - **Filtered (%)** The percentage of reads filtered.
  - **Reads after filtering** The number of reads left after filtering.
- **Distribution of expected errors**
- **Read lengths** Plot of read lengths before and after length trimming and filtering.
- **Unique sequences**
- **Merging of unique read pairs**

  - **Number of unique pairs** The number of read pairs.
  - **Unique pair with insufficient overlap** The number of pairs that had insufficient overlap and were discarded.
  - **Merged unique pairs** The number of read pairs that were successfully merged.

- **Error model estimation**
  - **R1** Error model computed based on forward reads.
  - **R2** Error model computed based on reversed reads.

The ASV abundance table

The ASV abundance table contains the detected amplicon sequence variants (ASVs) and the abundance of each ASV. The Detect Amplicon Sequence Variants tool produces one ASV abundance table per sample. To go beyond single sample ASVs, you can combine tables and enrich them with metadata using the following tools:

- **Merge Abundance Tables** Creates a merged, multi-sample ASV abundance table that allows you to compare abundances across samples, see section 7.1.
- **Add Metadata to Abundance Table** Adds sample metadata to your table. This allows you to aggregate samples based on attributes. This is useful for instance when analyzing replicates from the same sample origin. See section 7.9 for information on how to add metadata.
- **Assign Taxonomies to Sequences in Abundance Table** Assigns taxonomy annotations to the ASVs. You can aggregate ASVs by taxonomy level, (see section 7.3).

In the following, we focus on the single sample ASV abundance table, but include a few hints about additional features and options that could be of use for **merged ASV abundance tables**, **ASV abundance tables with sample metadata**, and **ASV abundance tables with assigned taxonomies**.

There are a number of ways to visualize the ASV abundance table:

- **Table view** () (figure 5.16) The table displays the following columns, some of which are of use mostly for merged, multi-sample ASV tables:
  - **ID** The ID of the ASV.
  - **Name** The name of the ASV. The name is generated as an MD5 hash ID, why identical ASVs will have the same name across ASV tables.
  - **Combined Abundance** The total number of reads belonging to the ASV across samples.
  - **Min** Minimum abundance across all samples
  - **Max** Maximum abundance across all samples
  - **Mean** Mean abundance of all samples
  - **Median** Median abundance of all samples
  - **Std** Standard deviation of all samples

- **Abundance for each sample** The number of reads belonging to the ASV in a specific sample.
- **Sequence** The sequence of the detected ASV.

In the **Data** section of the **Side panel**, switch between *Raw* and *Relative* abundance. Relative abundance is computed as the ratio between the number of reads belonging to an ASV and the total number of reads in the sample.

For **merged ASV abundance tables with sample metadata**, use the setting **Aggregate sample** to aggregate samples based on metadata attributes, e.g. replicates from the same sample origin.

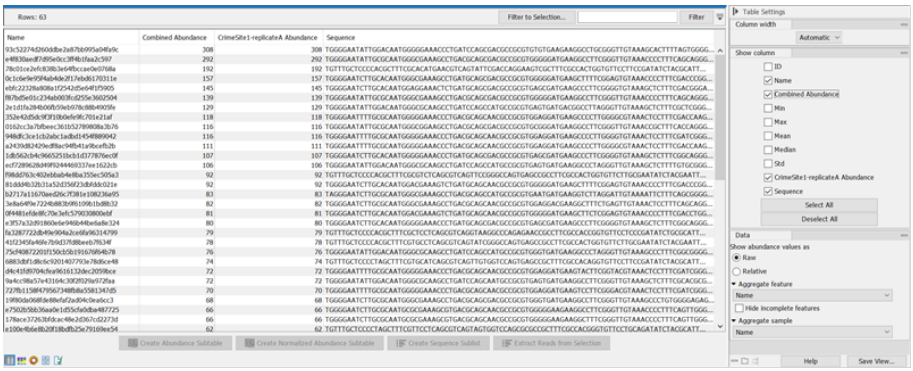


Figure 5.16: *The ASV abundance table for a single sample*

Below the table, the following actions are available:

- **Create Abundance Subtable** will create a table containing only the selected rows.
- **Create Sequence Sublist** will create a sequence list containing only the selected rows.

For **merged ASV abundance tables**, an additional action is available:

- **Create Normalized Abundance Subtable** will create a table with all rows normalized on the values of a single selected row. The row used for normalization will disappear from the new abundance table. The normalization scales the abundance table linearly. The scaling factor is calculated by determining the average abundance across all samples and for each sample scale it to the average for the reference. Note that to be enabled, the selected row must have abundance values for all samples. If you have empty values for some samples for the ASVs you wish to use as control, you will need to generate a new abundance table where those samples are not included. If the abundance table is obtained from merging single-sample abundance tables, then the merge should be redone excluding the samples with zero control read counts.

- **Stacked Visualization view** ( )

Adjust the **Side panel** setting **Aggregate feature** to *Name* (figure 5.17) for a visual representation of the relative abundance of ASVs in your sample. Hover over a color to see the name and count of the corresponding ASV.

Use **Filter level** to adjust the number of features shown in the plot. Setting the value to 10 gives you the 10 most abundant ASVs with remaining ASVs grouped as "Other".

For **merged ASV abundance tables**, additional **Side panel** settings may be of use:

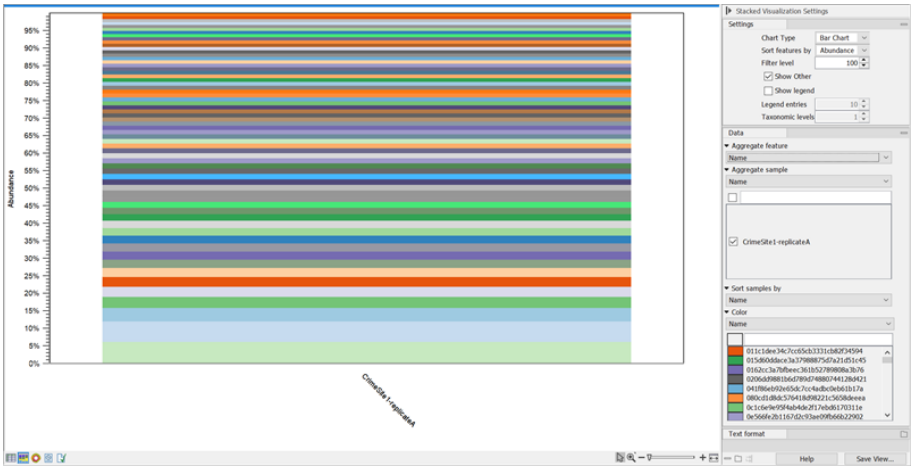


Figure 5.17: *The ASV abundance table Stacked Bar Chart with ASVs aggregated on Name shows the relative abundance of ASVs.*

- With **Chart type** you can switch between *Bar Chart* (figure 5.19) and *Area Chart* (figure 5.18).
- When **sample metadata** is applied, use **Aggregate sample** to aggregate based on metadata attributes e.g. replicates from the same sample origin.

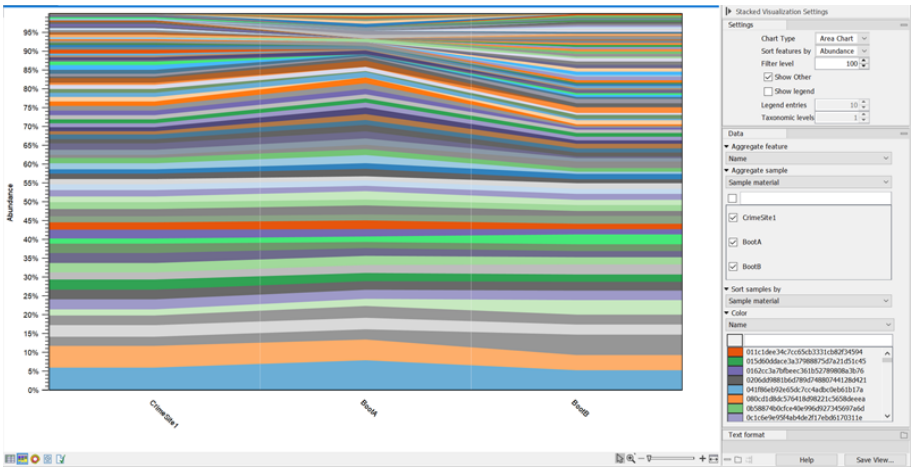


Figure 5.18: *Stacked Area Chart of a merged ASV abundance table Area chart, ASVs aggregated on Name, samples aggregated on Sample material. This shows the ASV abundance at 3 different sample sites for a total of 6 samples.*

For **ASV Abundance tables with assigned taxonomies**, you can aggregate ASVs by taxonomy level (figure 5.19).

- **Sunburst view** ( )

  The Sunburst view is only available for **ASV abundance tables with assigned taxonomies**.

  The plot is zoomable. Use **Side panel** settings to select how many taxonomy levels to display, and how these should be colored. Lower taxonomy levels will inherit the color from higher levels with different shades. Hover over the plot to view a legend with taxonomy and relative abundances for the highlighted section (figure 5.20).

  Click on a lower level field to render that field the center of the plot and display lower level counts in a radial view. Click on the center field to render the level above the current view

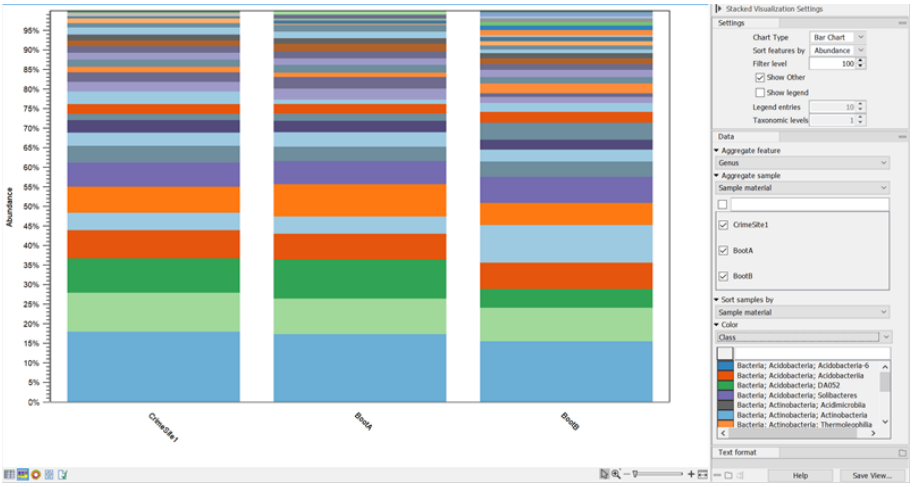


Figure 5.19: *Stacked Bar Chart of a merged ASV abundance tables with assigned taxonomies aggregated on Genus level.*

the center of the view.

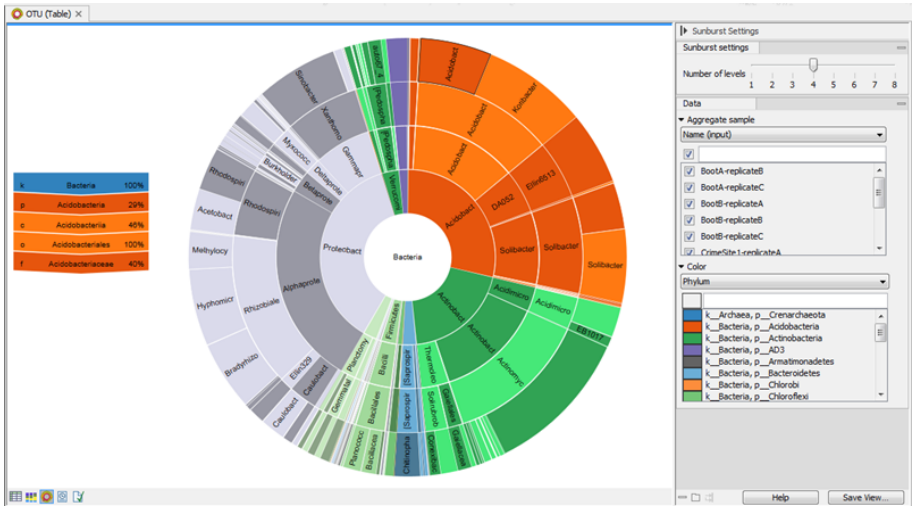


Figure 5.20: *Sunburst view of the microbial community showing all taxa belonging to the kingdom bacteria.*

5.5.3 Importing ASV abundance tables

You can import files containing amplicon sequence variant counts in tabular format (.xls/.xlsx/.csv) as abundance tables using Standard import:

1. Click on the **Import** () icon in the Toolbar and choose **Standard Import**, or go to **File** | **Import** () | **Standard Import** ().
2. Select files to import by clicking on the **Add files** button. Alternatively, specify folders containing files to import by clicking on the **Add folders** button.
3. Choose **Force import as type** and select *ASV abundance table (.xls/.xlsx/.csv)*.
4. Click on **Finish**.

5.6 Classify Long Read Amplicons

Classify Long Read Amplicons is meant for classification of long-read single-end amplicon sequencing data and was inspired by [Curry et al., 2022]. Reads are mapped to an amplicon reference database of choice and subsequently assigned the most likely taxonomy based on the probability calculated from a series of expectation maximization rounds. The tool can be used for both error-prone and high-accuracy reads.

Note: the tool has not been tested with error-prone PacBio (PacBio CLR) reads, but there is no reason to suspect these reads to be incompatible with the tool. Should you experience issues, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

For tool memory requirements, see (section 1.3). Comprehensive reference databases are needed for reliable species-level resolution, but large databases also require more memory. For example, the RefSeq Prokaryotic 16S database (see section 15.1) should not require more than 16GB RAM, but due to its limited number of sequences, only genus-level resolution is to be expected. The tool does not deduplicate the provided reference database, so make sure that databases are non-redundant in order to save memory and runtime.

The underlying algorithm works in five overall steps:

1. **Mapping reads to references**. Reads are mapped to the provided reference database. At this stage each read gets assigned one primary alignment based on mapping parameters, but up to 50 secondary alignments are retained for the following expectation maximization step.

2. **Calculate error model**. An error model of the probability of each alignment type is calculated from all primary alignments. The alignment types are mismatch, insertion, deletion, and softclip.

3. **Initialize alignment probabilities**. For each read-to-taxonomy pair the probability of the alignment is calculated based on the error model. Depending on the composition of the reference database given, a read may map to multiple references with the same taxonomy i.e., multiple identical read-to-taxonomy pairs exist. In that case, the highest alignment probability between them is used.
   In this step the initial probability of the taxonomies is also set with the assumption that all taxonomies in the reference are equally likely.

4. **Expectation maximization**. The algorithm loops through multiple rounds of expectation maximization. In each round, the probability that each read came from that taxonomy for each read-to-taxonomy pair is calculated using the alignment and taxonomy probabilities. The taxonomy probabilities are then updated and the log-likelihood of the estimate is calculated. This loop continues until the increase in log-likelihood falls below the threshold (>0.01) compared to the previous iteration.

5. **Abundance threshold cut-off and reassignment**. When the log-likelihood increase falls below the threshold, a final round of expectation maximization is entered. Here, the taxonomies falling below the set minimum abundance threshold are removed and their assigned abundance is reassigned to the most likely taxonomies among the retained taxonomies.

5.6.1 Classify Long Read Amplicons parameters

To run the **Classify Long Read Amplicons** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Amplicon-Based Analysis** ( ) | **Classify Long Read Amplicons** ( )

Classify Long Read Amplicons takes single-end long-read amplicon reads as input. The data should be trimmed for adapters, barcodes, and preferably also primers. Quality scores are not needed for the tool to work and you do not need to quality trim the reads. Since the algorithm infers an error model based on the read alignments, samples from different runs should not be analyzed simultaneously. Instead, you can run samples in "Batch" mode.

In the wizard, three categories of parameter settings need to be set (figure 5.21):

- **Select reference**. Specify the reference database to be used. Reference databases can be downloaded with the Download Amplicon-Based Reference Database tool (section 15.1) or created by adding taxonomies to sequence lists using the Update Sequence Attributes in Lists tool (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Update_Sequence_Attributes_in_Lists.html).
  Reads receive the taxonomy available on the reference sequence assigned i.e., if the reference has any missing taxonomy levels, these will also be missing from the final abundance table.
- **Read alignment**. Select whether to let the mapping algorithm set mapping parameters automatically based on the read platform used, or whether to manually override them. If choosing "Manual", the parameters can be set in the boxes below.
- **Abundance estimation**. Set the "Minimum relative abundance". Any taxa that have relative abundance below this threshold after the expectation maximization loop converges, will be removed and the abundance from these taxa will be reassigned to other probable taxa among the retained taxa.
  Note: If you are going to perform differential abundance analysis (section 7.7) following the classification of amplicons it may be an idea to set "Minimum abundance threshold" to 0. Differential abundance analysis on few samples with few features (taxa) can cause poor dispersion estimates. Should you later want to remove low abundance taxa, you can create an abundance subtable after filtering the table manually using the advanced filters in the top right of the table view.

5.6.2 Classify Long Read Amplicons output

The tool outputs an abundance table. In addition, the following outputs are available in the final wizard step (figure 5.22):

- **Collect unmapped reads**. Outputs a sequence list with all the reads that could not be mapped to the reference database.
- **Create report**. Outputs a summary report.

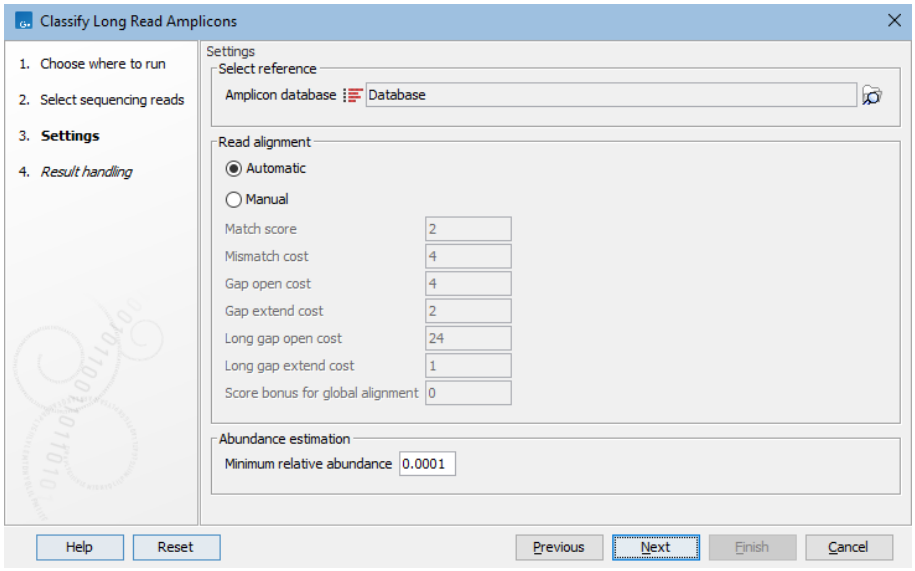


Figure 5.21: *Classify Long Read Amplicons parameter settings.*

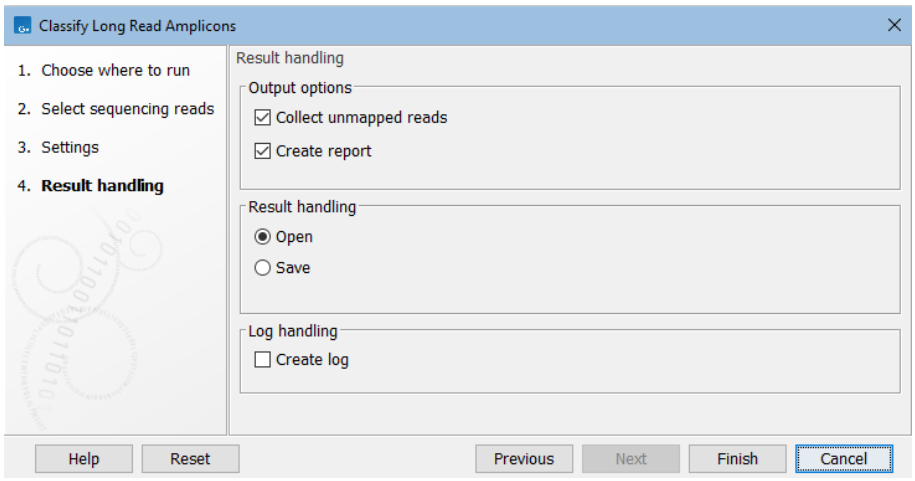


Figure 5.22: *Classify Long Read Amplicons output options.*

The Classify Long Read Amplicons abundance table

The abundance table output by **Classify Long Read Amplicons** contains a list of the identified taxonomies that passed the minimum coverage threshold, and the abundance assigned to each taxonomy. Given the probabilistic nature of the algorithm, the reported abundance is not equivalent to a read count, but rather it represents an estimated abundance. The estimate can contain fractions of reads, but the final reported abundance is rounded to the nearest integer.

In contrast to OTU or ASV abundance tables, Classify Long Read Amplicons abundance tables do not contain a sequence for the taxonomies. This is because the reads are not assigned to a sequence but to a taxonomy. Multiple reference sequences may have the same taxonomy (depending on the provided reference database), so all reads assigned to any of the sequences will count towards the abundance for the taxonomy.

Otherwise, the abundance table contains the same columns, views, and options as the OTU abundance table. See section 5.3.2 for a detailed description of the abundance table options.

The Classify Long Read Amplicons report

An example report can be seen in figure 5.23.

The report contains summary statistics of the results, which can be used for quality checking and verification. It is divided into four sections:

- **Classification summary**. Number of unique taxonomic level reported for each taxonomic level.
- **Classification of reads**. Statistics of the classification given in number of reads and in percentage of input reads.
  - **Input reads**. The number of reads in the input sequence list(s).
  - **Assigned reads**. The number of reads that could be mapped to the reference database.
  - **Unclassified reads**. The number of reads that could *not* be mapped to the reference database. If a large percentage of the reads are unclassified, it could mean that the sample is contaminated, or that the reference database is not comprehensive enough.
- **Minimum abundance filter**.
  - **Filtered taxa**. The number of taxa removed from the final result due to having relative abundance below the minimum abundance threshold.
  - **Reassigned abundance**. The sum of the abundance of the removed taxa which has been reassigned to the most likely taxa among the retained taxa.
- **Abundance distribution**. A scatter plot of the relative abundance of reported features before and after reassigning abundance of features below the minimum abundance threshold. Points will fall on the line or above it, but if one or a few point(s) lie significantly higher than the line, it could mean that that feature has been artificially inflated in the abundance table.

1 Classification summary

Taxonomic level	Number of classifications
Kingdom	2
Phylum	10
Class	13
Order	17
Family	21
Genus	24
Species	26

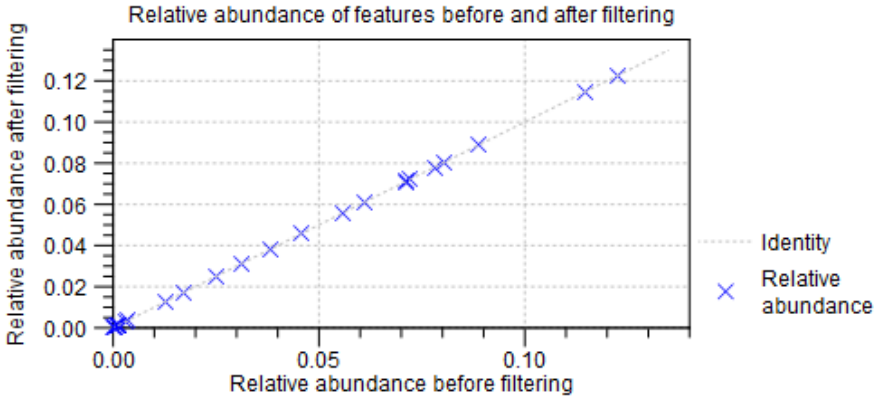
2 Classification of reads

	Reads	% of reads
Input reads	47,933	100.00
Assigned reads	47,933	100.00
Unclassified reads	0	0.00

3 Minimum abundance filter

Filtered taxa	8
Reassigned abundance	8

4 Abundance distribution



Reads that mapped to features removed by the Minimum Relative Abundance filter were subsequently reallocated, resulting in disparities between the relative abundance values of the remaining features before and after filtering. This can be observed in the form of data points departing from the diagonal line of the plot. The plot exclusively displays features that survived the Minimum Relative Abundance filter.

Figure 5.23: *Classify Long Read Amplicons report.*

Chapter 6

Taxonomic Analysis

Template workflows for taxonomic analysis are available at:

> **Workflows** | **Template Workflows** () | **Microbial Workflows** () | **Metagenomics** () | **Taxonomic Analysis** ()

For more information, see section 2.1.

6.1 Contig Binning

In order to characterize microbial communities, it is key to resolve their composition, diversity and function. With recent advancements in sequencing techniques, whole metagenome shotgun sequencing is becoming standard in metagenomics. Because the output of this technique is a mixture of short DNA fragments belonging to various genomes, computational algorithms for clustering of related sequences are necessary. This approach is globally referred to as sequence binning, and it facilitates downstream analysis steps including: retrieval of metabolic and marker genes; core genome and housekeeping genes analysis; MLST, MLSA and phylogenetic analysis; rRNA and probe design; metagenome re-assembly.

There are two types of binning methods: a) taxonomy dependent and b) taxonomy independent. The first is implemented here through the Bin Pangenomes by Taxonomy tool and the second via the Bin Pangenomes by Sequence tool [Sedlar et al., 2017]. The performance of approach a) is limited to the completeness of an existing database, whereas approach b) usually suffers from a lack of precision. In order to leverage the full strength of the two approaches a combined analysis is encouraged. The template workflow **QC, Assemble and Bin Pangenomes** (section 2.1.4) facilitates this as it employs both methodologies to generate lists of contigs of assembled, binned contigs.

6.1.1 Bin Pangenomes by Taxonomy

This tool assigns contigs and the reads they are composed of into bins with other contigs presumably of closely related taxonomy. For this we use a microbial reference (genome) sequence database, which comprises sequences with taxonomic information. Furthermore, in order to separate contigs that originate from plasmids from those of genomic origin, the Bin Pangenomes by Taxonomy tool additionally takes a plasmid database as input.

Binning occurs in 4 consecutive steps:

1. Obtain taxonomic information for reads and plasmids using the Taxonomic Profiling tool (see section 6.4).
2. Map reads to contigs using the Map Reads to Contigs tool (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Map_Reads_Contigs.html).
3. Assign taxonomic and plasmid labels to contigs. The contigs are assigned the most specific taxonomy that is assigned to at least a certain fraction of the reads that map to that contig.
4. Group contigs with the same taxonomy into bins, and reject those for which no taxonomy was assigned, or for which the assigned taxonomy was not specific enough.

To start the tool, go to:

Tools | Microbial Genomics Module ( ) | Metagenomics ( ) | Taxonomic Analysis ( ) | Bin Pangenomes by Taxonomy ( )

The Bin Pangenomes by Taxonomy tool takes one or several single or paired-end read files as input (figure 6.1).

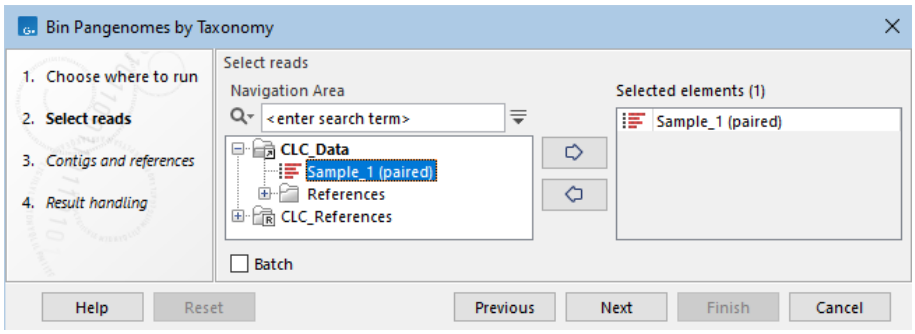


Figure 6.1: *Select the reads.*

The tool is designed to work on contigs assembled from the same set of reads used as input, previously assembled using the De Novo Assemble Metagenome tool (as in the workflow, see section 2.1.4). You can also specify here the minimum contig length desired (figure 6.2).

As reference databases, one or two Taxonomic Profiling index files can be provided:

- The file provided as "Reference index" is used to find taxonomic information for the reads.
- The file provided as "Plasmid reference index" (if the "Find plasmid information" option is checked) is used to distinguish genomic reads from plasmid reads.

Both references can be obtained by using the Download Curated Microbial Reference Database tool (section 16.1) or Download Custom Microbial Reference Database tool (section 16.2). If using the latter, the indexes can be built with the Create Taxonomic Profiling Index tool (section 16.5).

Depending on the dataset, it may be necessary to adapt the contig purity settings, where "Maximum level" refers to a maximum level in the taxonomic tree and where a specific "Minimum

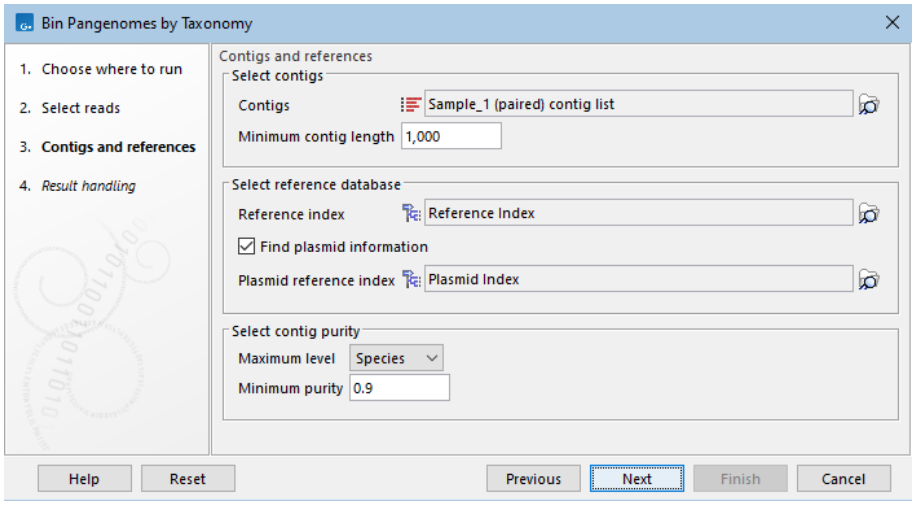


Figure 6.2: *Select the references and specify the parameters needed for running the tool.*

purity" per contig needs to be reached in order for it to be considered a part of a bin. For example, if Maximum level = Genus and Minimum purity = 0.8 and 512 reads map to a given contig, at least 0.8 * 512 = 410 reads need to have the same Genus level taxonomy in order for the contig to become part of the respective bin. If more precise taxonomic information is available (e.g., on Species level) with the requested minimum purity, this information will be used instead.

The "Result handling" dialog allows you to specify outputs (figure 6.3):

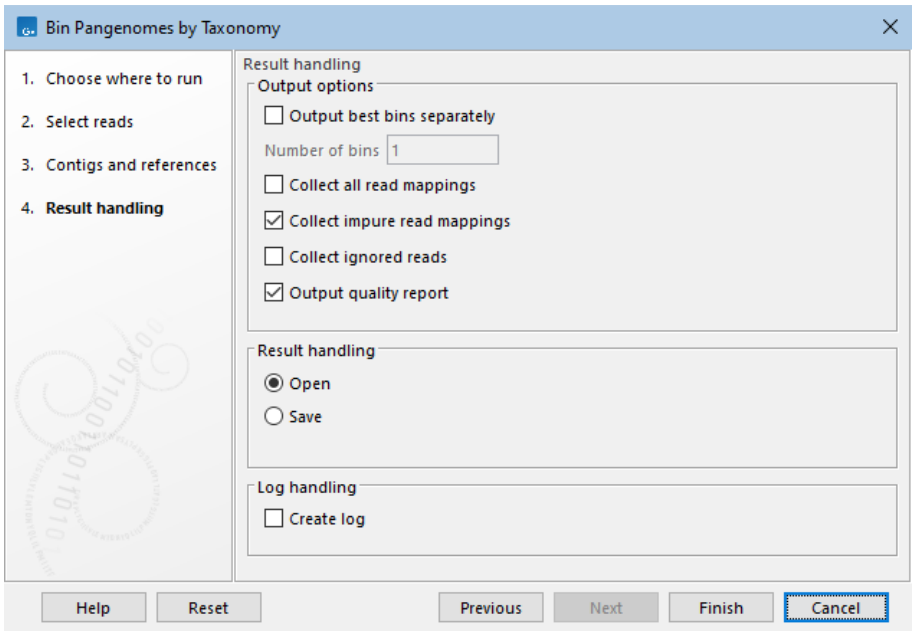


Figure 6.3: *Specify the outputs needed.*

- Choose to output a certain number of the best bins separately, which means that a chosen number of bins will be written to separate outputs. In this context, "best" is defined by completeness, estimated as the number of contig nucleotides in the bin divided by the number of nucleotides in the assigned reference genome.

- Specify whether to collect the read mappings and which kind (all, only impure contigs)
- Collect ignored reads (i.e., reads not mapping to contigs)
- Output a quality report for the bins (see section 6.1.2).

The standard output of the Bin Pangenomes by Taxonomy tool consists of a list of (binned) contigs and one sequence list per input reads file (or two for paired reads) where each of the sequences is labeled according to its most probable origin and bin it ended up in (the bin annotation is stored as "Assembly ID" annotation in order for it to work seamlessly with other tools). Also, a column called "isPlasmid" provides a true/false label whether the contig/read was mapped respectively to a plasmid or a genome. The tool can also output a Taxonomy binning report.

6.1.2 The Taxonomy Binning Report

The taxonomy binning report has the following sections:

- **Contigs**
  - **Accepted.** The number of contigs that have the required minimum contig length, and can be assigned a taxonomy with the specified "Maximum level" and "Minimum purity".
  - **Rejected.** The remaining contigs.
- **Reads.** The number of reads that are unmapped, or that map to the accepted and rejected contigs. All reads mapping to contigs are counted regardless of whether or not these support the assigned taxonomy.
- **Bins.** A bin is created for each assigned taxonomy, and for each contig for which no taxonomy can be assigned.
  - **Accepted.** Bins that contain accepted contigs.
  - **Rejected.** The remaining bins.
- **Accepted contig bins / Rejected contig bins.** These two tables are sorted by "Approximate completeness". They contain the same columns:
  - **Bin.** An identifier for the bin. This takes the form "TaxBinX" where X is a number starting from 0. There is no significance to the number of a tax bin. The identifier matches the "Assembly ID" on the binned contigs.
  - **Taxonomy.** The taxonomy of the bin. For "Accepted contig bins" this is always at least as specific as the "Maximum level". For "Rejected contig bins" it may be less specific or Unknown. Note that Unknown only means that no taxonomy was found at the required "Minimum purity". For example, if Minimum purity is 0.9, then a bin will be labeled as Unknown even if 89% of reads are assigned the same species.
  - **Taxonomic level (plasmid).** The level at which the taxonomy is assigned, e.g. Species. If the taxonomy is assigned via the provided "Plasmid reference index", then the word "(plasmid)" will be added e.g. Species (plasmid).
  - **Contigs.** The number of contigs in the bin.

- **Nucleotides contigs.** The number of nucleotides in the contigs.
- **Reads.** The number of reads mapping to the contigs.
- **Nucleotides reads.** The number of nucleotides in the mapped reads, regardless of how or if they mapped to the contig. This number therefore includes unaligned ends, and counts overlapping nucleotides of read pairs twice.
- **Approximate completeness.** To calculate the approximate completeness, **Nucleotides contigs** is divided by the average length of the sequences in the reference indexes that both 1) have the same taxonomy, and 2) have reads mapping to them. This number can exceed 1, either because multiple contigs may be assembled for one sequence in the reference index, or because some reads may map to a short reference index with the same taxonomy, but for which no contig is assembled.
- **Taxonomic purity (read level).** The purity as a percentage out of 100. This will either be 0.00 or greater than or equal to the "Minimum purity" setting. This is because taxonomies are not assigned when the purity is less than the specified Minimum purity.
- **Average contig coverage.** This is calculated as **Nucleotides reads** / **Nucleotides contigs**. This is a rough estimate of coverage because Nucleotides reads does not take account of how the nucleotides in the reads mapped to the contig.

6.1.3 Bin Pangenomes by Sequence

Binning by sequence is done irrespective of a database, only depending on content and coverage. To have both sources of information available, the Bin Pangenomes by Sequence tool takes read mappings to contigs as input, where there should be one read mapping per technical replicate (each mapping to the same contigs) in order to make most use of coverage information across all samples. However, if read mappings are not available, the Bin Pangenomes by Sequence tool also takes plain sequence lists of contigs as input.

The Bin Pangenomes by Sequence algorithm is based on the MetaBAT [Kang et al., 2015] and SCIMM [Kelley and Salzberg, 2010] algorithms with several modifications:

- A logistic regression model (simliar to MetaBAT) may use a variable number of parameters. The number of trusted parameters is adjusted to the number of contigs in a bin and the parameters are adjusted during the algorithm. Only Kmer features are used.
- The interpolated Markov Models of SCIMM are replaced by variable order markov models.
- Random Projections are used to speed up the search for the centers in the proximity of a contig in combined Kmer-Coverage space.
- Poisson-Mixture models are used to fit the coverage distribution on contigs.

The tool was designed for sample sizes on the order of 100 000 contigs. It does not support substantially larger data sets.

To start the tool, go to:

Tools | **Microbial Genomics Module** () | **Metagenomics** () | **Taxonomic Analysis** () | **Bin Pangenomes by Sequence** ()

The Bin Pangenomes by Sequence takes one sequence list of contigs or one read mapping per sample as input (figure 6.4).

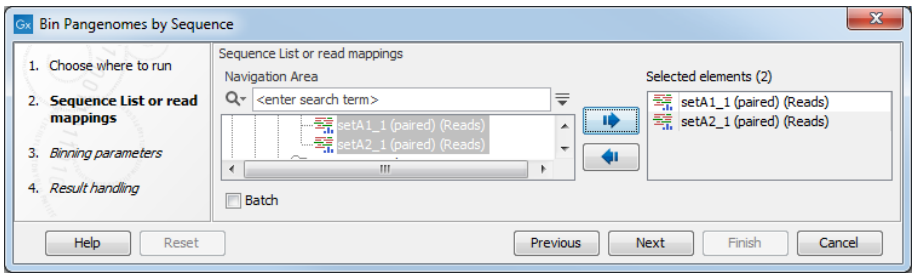


Figure 6.4: *Select the contigs or read mappings.*

In the next dialog (figure 6.5), the several parameters can be specified:

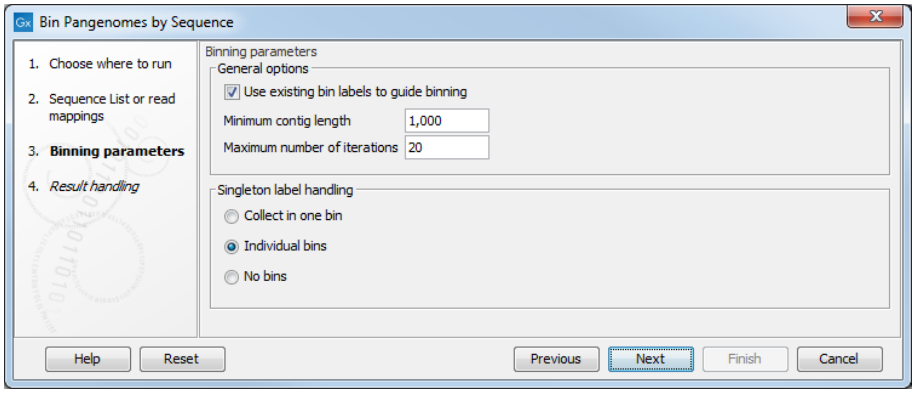


Figure 6.5: *Configuration of the Bin Pangenomes by Sequence.*

- **Use existing labels to guide binning** may be used to improve binning quality and speed. For read mapping inputs, labels assigned to the reads by the Bin Pangenomes by Taxonomy are used, while for sequence list inputs the Assembly ID (see section 22) labels assigned to the contigs are used.
- **Minimum contig length** specifies the minimal length for contigs to be considered (should be at least 1000 to obtain decent bin qualities
- **Maximum number of iterations** specifies how many purification steps at most should be made.
- **Singleton label handling** decides whether singletons should be collected in one bin, kept in individuals bin, or not included in any bins.

Finally, in the "Result handling" dialog, it may be specified whether the reads of the binned contigs should be labelled and collected.

The standard output of the Bin Pangenomes by Sequence tool consists of a Sequence binning report, a contig list with their assigned bin listed in the Assembly_ID column, and as many read lists as read mappings were used as input in the tool, where reads have been assigned the bin of the contig they belong to.

6.2 Identify Viral Integration Sites

The **Identify Viral Integration Sites** tool searches for likely viral/host integration events. The tool works by searching for regions with reads with unaligned ends and/or discordant paired reads, where one read in the pair maps to the host, and the other read maps to a virus.

Notice: this tool can only be used for protocols such as hybrid capture, which specifically enriches for viral genomes while capturing at least some chimeric reads that map to both host and virus genomes.

The approach is the following:

- First, the input reads are mapped simultaneously against the host genome (e.g. human) and a viral database. Internally, the reads are mapped using the 'Find Best References using Read Mapping' tool. Any ambiguous reads are randomly assigned, corresponding to the standard "Non-specific match handling = Map randomly" read mapper option. This produces a host read mapping, and read mappings for all identified viruses. These read mappings are then scanned for potential breakpoints ends, which are the positions showing a pattern of unaligned ends.
- The potential breakpoint ends are filtered based on the following criteria:
  - The number of reads with unaligned ends must be higher than the user-specified criteria
  - The number of reads with unaligned ends must be more than 5% of the maximum for the position with the highest number of unaligned ends for the chromosome/virus.
- For the host, we collect and map all the unaligned ends for a given position against the viral genomes. Then we look at the position where the majority of the reads map on the viral genome, and check if there is a potential breakpoint within 50 bp of that position. Notice: we choose the closest viral breakpoint, and we always choose the read mapping position where the majority of reads map.
- Finally, we look at the broken read pairs on the host genome, where one read was within 500 bp of the host breakpoint (and on the same side as the aligned part of the reads found during the scan for unaligned ends), while the other read in the pair mapped to the virus. If this number of broken reads is larger than a user-specified threshold, the host/virus breakpoint ends are considered a sound match, and we add the host/virus breakpoint to our list of identified breakpoints.

To launch the **Identify Viral Integration Sites** tool, go to:

Tools | **Microbial Genomics Module** () | **Metagenomics** () | **Taxonomic Analysis** () | **Identify Viral Integration Sites** ()

One or more single or paired-end read files can be provided as input.

After selecting the input reads, it is possible to specify the host and virus references, and adjust the detection parameters, see (figure 6.6).

The following parameters are available:

- **Viral references** The viral sequences. The breakpoints identified from the read mappings against the human reference will be tested against these sequences.

- **Viral annotations** Annotations, such as a Gene or CDS track for the viral sequences. Notice, that these annotations can also be present on the viral sequence input if this is a sequence list. In this case, specifying the annotations here will be used instead of any annotations present on the viral sequence list.
- **Host references** The host sequences.
- **Host annotations** Annotations, such as a Gene or CDS track for the host sequences. Notice, that these annotations can also be present on the host sequence input if this is a sequence list. In that case, specifying the annotations here will overwrite any annotations present on a host sequence list.
- **Minimum number of reads on a virus** At least this many reads must map to a virus before it is included in the analysis.
- **Minimum relative virus abundance to most abundant virus** A reference must have at least this fraction of the reads of the most abundant virus.
- **Minimum virus coverage** The minimum number of nucleotides mapped to the virus reference divided by the reference length before it is included in the analysis.
- **Minimum reads with unaligned ends supporting site** The minimum number of reads required with an unaligned end starting at the same position.
- **Minimum host/virus broken pairs supporting site** The minimum number of paired reads spanning the breakpoint site, where one read maps to the virus, and the other to the host.
- **Minimum ratio between unaligned and aligned** The minimum ratio between reads supporting a breakpoint, and reads with no unaligned ends. This is only checked for the host genome.
- **Minimum unaligned end length** Minimum length of unaligned ends to be considered as supporting a breakpoint.
- **Nearby genes distance** If host genes are located within this distance of an integration event (in basepairs) they are reported in the table view, and in the report.

The final step is to specify the output objects, see (figure 6.7). The following options are available:

- **Create breakpoint visualization** Creates a graphical visualization and a table with breakpoints. This element is explained in more detail in the next section.
- **Create report** Creates a summary report.
- **Create host breakpoint tracks** Creates a feature track with detected breakpoints.
- **Create viral breakpoint tracks** Creates a feature track with detected breakpoints for the identified viruses.
- **Create host mappings** Creates a read mapping for the host references.
- **Create viral mapping tracks** Creates a read mapping for the (detected) viral references.

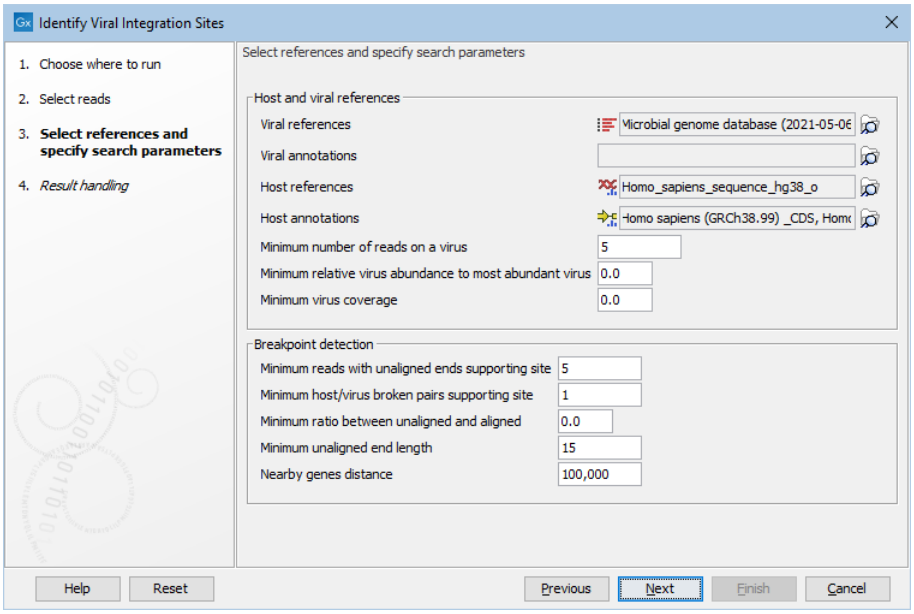


Figure 6.6: *Select references and adjust detection options.*

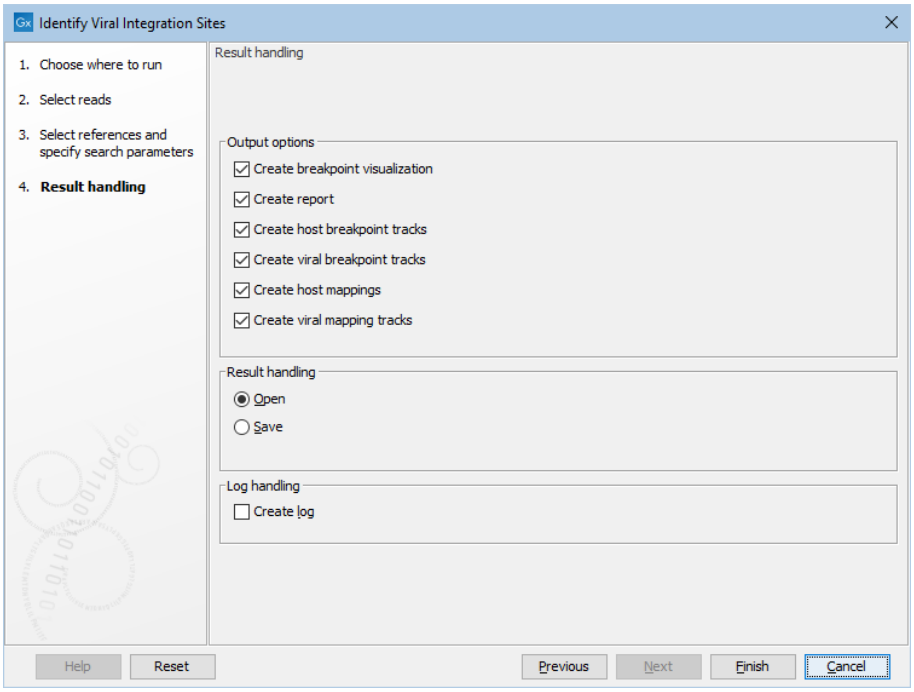


Figure 6.7: *Select output options.*

6.2.1 The Viral Integration Viewer

The viral integration viewer presents a graphical view of a virus together with the host genome, shown in a circular plot, see (figure 6.8).

The viral integration viewer is synchronized to the table view: selecting a breakpoint in the table view, will choose the corresponding breakpoint in the graphical view, and vice versa.

The upper left quadrant is a view of the virus. Notice, that if several viruses are detected in a sample, it is possible to choose between them using the **Virus** drop-down in the sidepanel view.

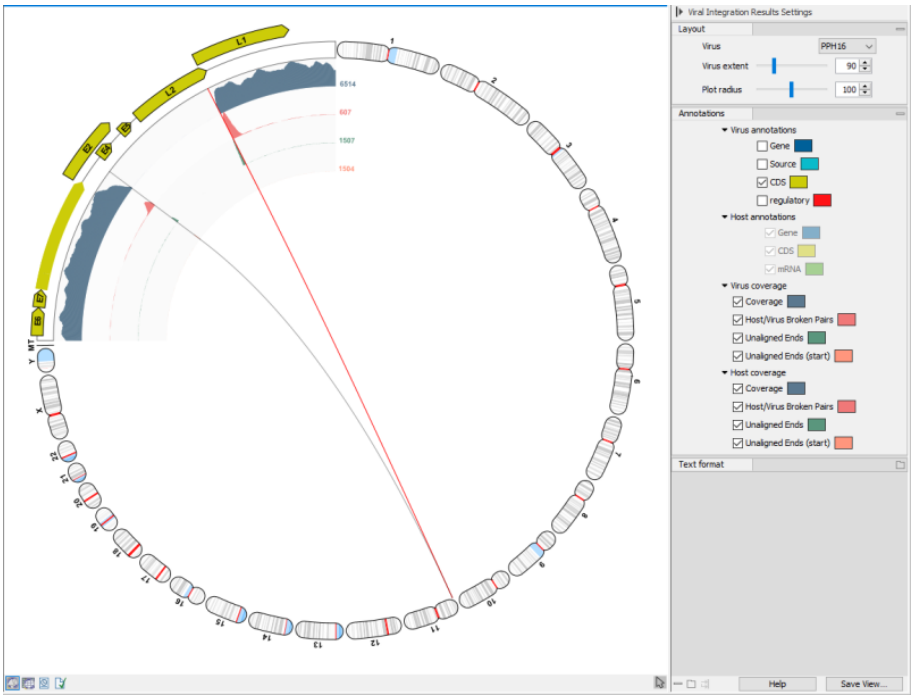


Figure 6.8: *The viral integration viewer.*

It is possible to zoom in on the host genome, by using the mouse-wheel on a section of the host genome, see (figure 6.9). It is also possible to double-click a breakpoint to zoom in and center on the breakpoint on the host genome.

Most elements (coverages, annotations, genome positions, breakpoints) show tooltips with additional information when hovering them with the mouse.

The table view, (figure 6.10), shows summary information for the identified breakpoints, including the host and virus regions, the number of unaligned reads supporting the host and virus end of the breakpoint, and the number of broken host/virus pairs supporting the breakpoint.

The table view also lists any genes that overlap with the breakpoint position. This information is only available if Gene, CDS, or mRNA tracks are provided for the host genome. If CDS and/or mRNA tracks are provided, an additional qualifier ("exon" or "intron") will be added to the output.

6.2.2 The Viral Integration Report

The viral integration report contains an overall summary for the sample. Notice, that reports from multiple samples may be combined using the 'Combine Report' tool.

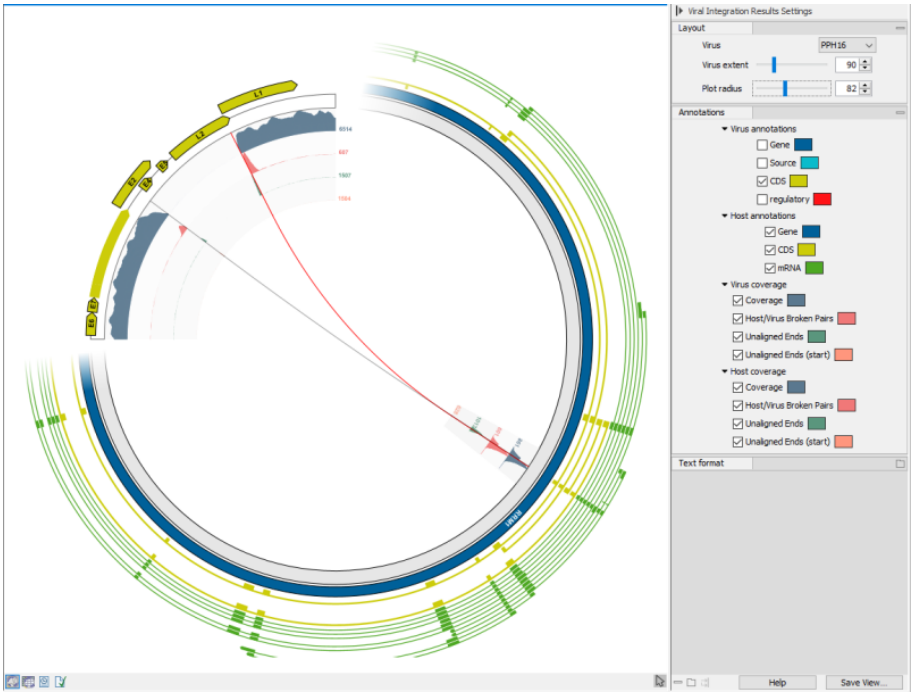


Figure 6.9: *The viral integration viewer when zoomed in.*

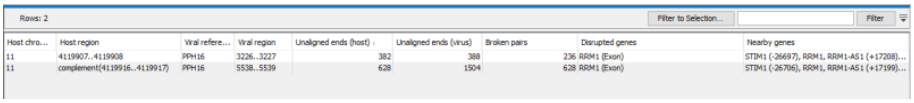


Figure 6.10: *The table view for the detected viral breakpoints.*

1 Identify Viral Integration Sites summary

Input reads	290,140
Host reads	84,432
Virus reads	193,379
Unmapped reads	12,329
Host reads (%)	29.10
Virus reads (%)	66.65
Unmapped reads (%)	4.25
Breakpoints identified	2
Viruses identified	4

2 Identify Viral Integration Sites virus content

Virus	Reads mapped
AY057438	321
X05015	80
PPH16	96,891
X74476	210

3 Identify Viral Integration Sites breakpoint summary

Host chromosome	Host region	Virus	Virus region	Unaligned ends host	Unaligned ends virus	Broken reads host	Disrupted genes	Nearby genes
11	complement(4119916..4119917)	PPH16	5538..5539	628	1,504	628	RRM1 (Exon)	STIM1 (-26706), RRM1, RRM1-AS1 (+17199), OR55B1P (+26195), AC015689.1 (+57223), AC015689.2 (+90437)
11	4119907..4119908	PPH16	3226..3227	382	388	236	RRM1 (Exon)	STIM1 (-26697), RRM1, RRM1-AS1 (+17208), OR55B1P (+26204), AC015689.1 (+67232), AC015689.2 (+90446)

Figure 6.11: *The report for the Identify Viral Integration Sites tool.*

6.3 Classify Whole Metagenome Data

Classify Whole Metagenome Data performs taxonomic classification of whole metagenome sequencing data using the same principles as the open-source tools Kraken2 [Wood et al., 2019] and Bracken [Lu et al., 2017]. This tool uses a fast and memory-efficient algorithm that facilitates analysis with large, comprehensive reference indexes, enabling accurate species-level resolution on laptops.

Each read is classified to the most specific taxonomic level. If a read match equally well to

multiple references, it may be assigned to a higher taxonomic level. Reads that do not match the database are marked as unclassified.

The number of reads assigned to each taxonomic level is used to create an abundance table, excluding any taxonomy with fewer than 10 reads. Some reads may have matched higher taxonomic levels. These reads are probabilistically reassigned to species level using Bayes' theorem, considering the likelihood that a read from a certain species may be redundant and therefore assigned to a higher taxonomic level.

6.3.1 Classify Whole Metagenome Data parameters

To run the Classify Whole Metagenome Data tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **Classify Whole Metagenome Data** ( )

In the first dialog, select the sequence list to analyze (figure 6.12).

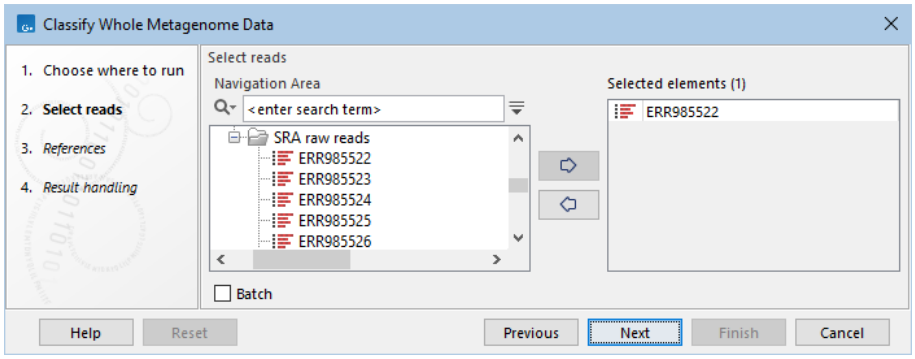


Figure 6.12: *Select a sequence list as input.*

Click **Next** to select reference databases (figure 6.13):

- **Reference index**. Select the metagenome index. If your sample contains host reads, the index should include the host genome.
  Curated reference databases and metagenome indexes are available from **Download Curated Microbial Reference Database** (section 16.1). Alternatively, create your own index using **Create Whole Metagenome Index** (section 16.4).
- **Host genome**. Define the taxa of the host or background genome(s) that may be present in your sample, but that is not the target of your analysis. Reads assigned to the host genome are omitted from the abundance table output. As an example, to analyze the microbiome from human gut samples and exclude human reads from the result, specify the taxon "Homo sapiens" as host genome.

6.3.2 Classify Whole Metagenome Data output

Click **Next** to select the outputs.

The tool outputs an abundance table. In addition, the following options are available:

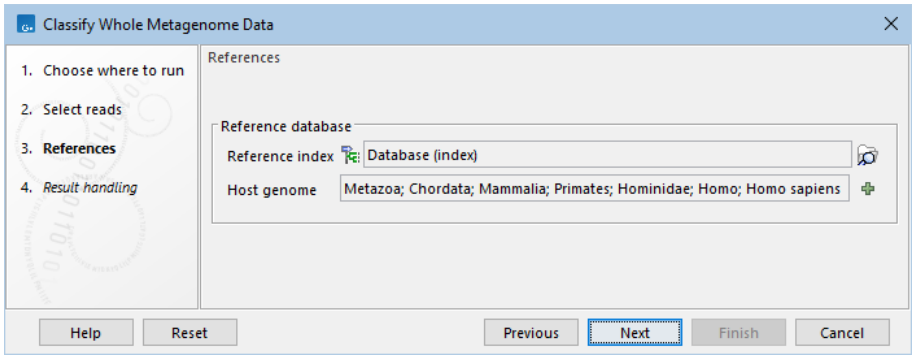


Figure 6.13: *Select the reference index and, if relevant, define the host genome taxa.*

- **Collect reference database reads**. Creates a sequence list for each input with reads that were assigned to the reference database.
- **Collect host genome reads**. Creates a sequence list for each input with reads that were assigned to the host genome.
- **Collect unmapped reads**. Creates a sequence list for each input with unassigned reads.
- **Create report**. Creates a summary report.

Sequence list output

The sequence list of reference database reads and host genome reads contain the **Taxonomy** attribute with the full taxonomy of the taxon to which the read was assigned.

Classify Whole Metagenome Data report

The Classify Whole Metagenome Data report (figure 6.14) contains information about the number of classified taxa per taxonomic level, and classification of reads.

- **Taxonomic summary**. The number of detected taxa for each taxonomic level.
- **Classification of reads**.
  - **Reference database matches**. The number of reads assigned to the reference index, excluding host genome taxa.
  - **Host matches**. The number of reads assigned to host genome taxa.
  - **Unclassified reads**. The number of reads that could not be assigned to the reference index. If a large percentage of the reads are unclassified, it could mean that the sample is contaminated, or that the reference index is not comprehensive enough.

6.3.3 Classify Whole Metagenome Data abundance table

The abundance table contains the taxa identified in your sample.
To create a multi-sample abundance table, use the Merge Abundance Tables tool (section 7.1). Some of the options mentioned in the following are only relevant for multi-sample abundance tables.

1 Taxonomic summary

Taxonomic level	Number of classifications
Kingdom	3
Phylum	7
Class	8
Order	11
Family	16
Genus	49
Species	290

2 Classification of reads

	Reads
Reference database matches	1,432,169
Host matches	228
Unclassified reads	232
Total	1,432,629

Figure 6.14: *The Classify Whole Metagenome Data report.*

The abundance table can be visualized in Table view (), Stacked Visualization view (), and Sunburst view ().

Table view ()

The table displays a number of columns, some of which are available only when the table is not aggregated by taxonomy (figure 6.15):

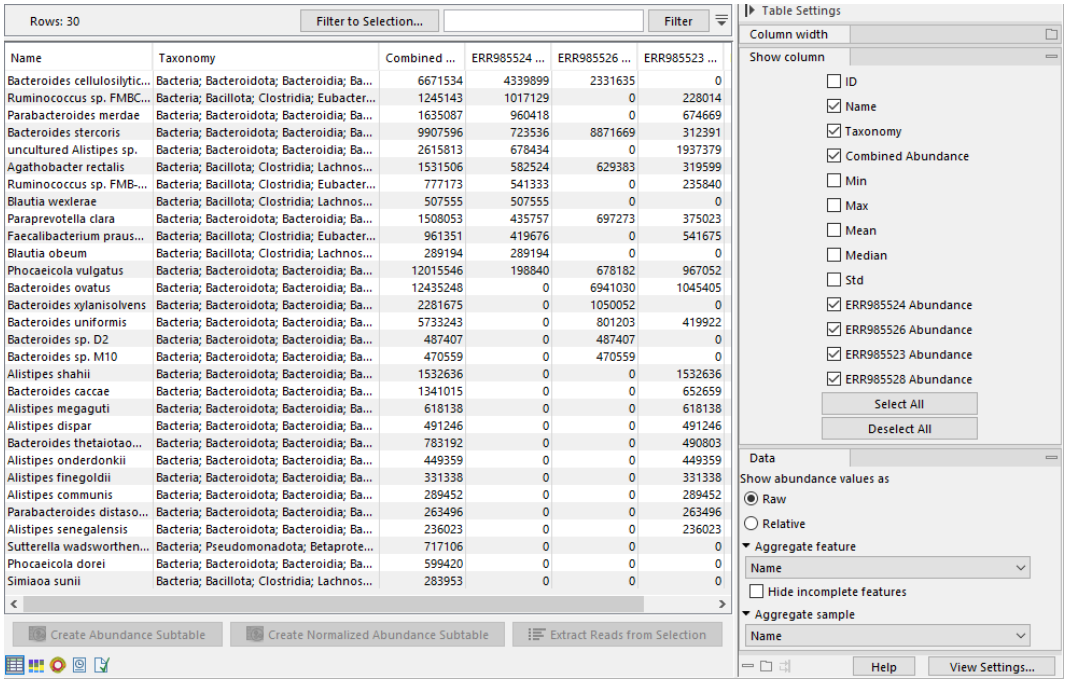


Figure 6.15: *The table view of the abundance table lists name, taxonomy, abundance etc. of the identified taxa. The example shows a multi-sample table.*

- **ID**. The ID of the taxon.
- **Taxonomy**: The taxonomy of the taxon as specified by the reference database.
- **Combined Abundance**: Total abundance across samples.
- **Min**, **Max**, **Mean**, **Median** and **Std**. Minimum, maximum, mean, median and standard deviation of abundance values across samples.
- **Abundance**. Number of reads assigned to the taxa.

The **Side panel** contains the following settings of relevance to the Classify Whole Metagenome Data abundance table:

- **Show abundance values as**. Switch between Raw and Relative abundance. Relative abundance is calculated as: Relative abundance = (Abundance) / (Sum of abundances).
- **Aggregate feature**. Select a taxonomic level to aggregate abundance values by. For example, select *Family* to display abundance values per Family as opposed to per genome.
- **Aggregate sample**. Select a metadata attribute to aggregate samples with same metadata value into one column with combined abundance values.

Below the table, the following actions are available:

- **Create Abundance Subtable**. Creates a table containing only the selected rows.
- **Extract Reads from Selection**. Extracts reads that were uniquely associated with the selected rows. This option is available if you selected the output Reference database matches in the Classify Whole Metagenome Data tool dialog.

Stacked Visualization view ()

The Stacked Visualization view displays the relative abundance of each taxon.

Use the **Side panel** setting **Bar type** to switch between Bar Chart (figure 6.16) and Area Chart (figure 6.17). The charts can be scaled by percentage, where all bars have the same height of 100%, or counts, where the bar heights are proportional to the number of counts. Different colored bars or areas represent different taxa. A column represents a sample or - if aggregated by sample level - a group of samples.

Hold your pointer over an area to have the full taxonomy and abundance value displayed in a tooltip.

You can adjusted the view further via the **Side panel** settings. Options include:

- **Aggregate feature**. Select a taxonomic level to aggregate abundance values by. For example, select *Family* to have each section in the plot represent a Family instead of a genome.
- **Aggregate sample**. Select a metadata attribute to aggregate samples with same metadata value into one column with combined abundance values. (Relvant for multi-sample abundance tables only). Use the checkboxes below to specify which samples or groups of samples to include in the plot.

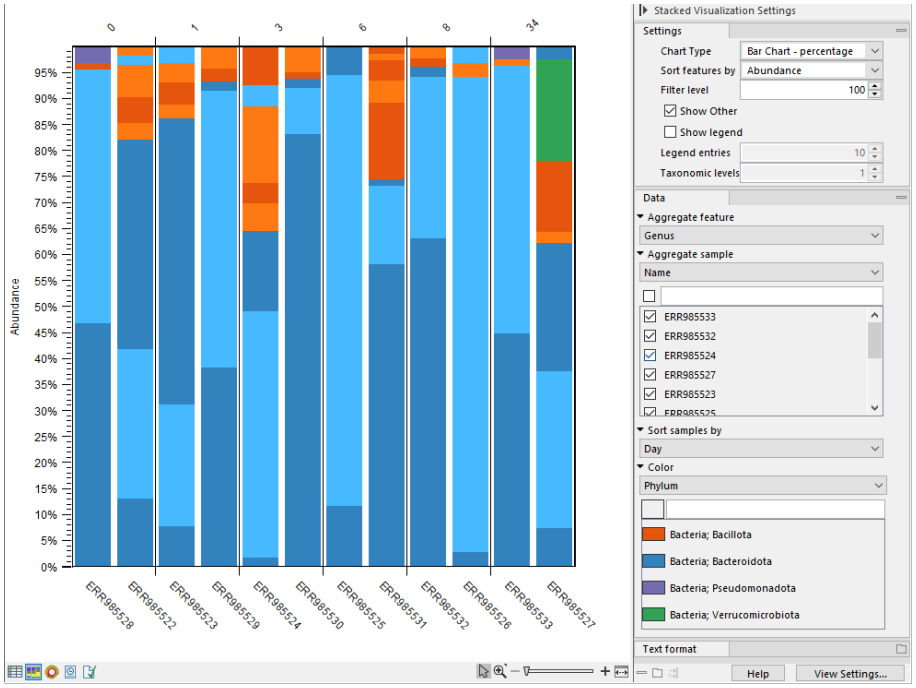


Figure 6.16: *Stacked bar chart.*

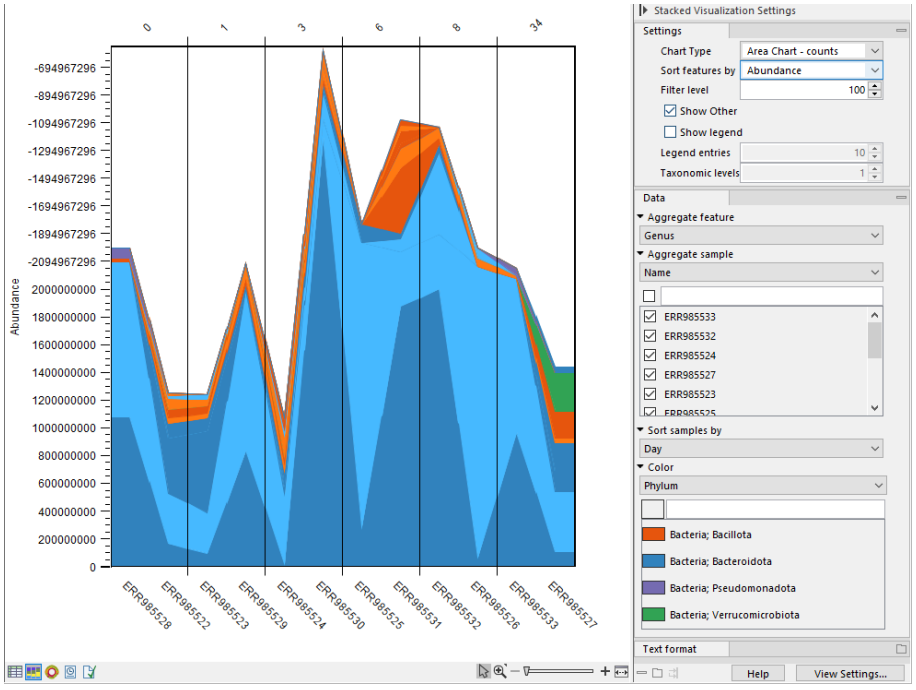


Figure 6.17: *Stacked area chart.*

- **Sort samples by**. Select a metadata attribute to sort samples by the corresponding attribute values. The values are listed above the plot (figure 6.16).

- **Color**. Select a taxonomic level to color the plot by. As an example, if you select *Phylum*, all taxa belonging to the same Phylum will get different shades of the same base color.

Sunburst view ( )

The plot is zoomable. Click on a section to zoom in and render the plot with this section at the center. Click on the center of the plot to zoom out one level at a time.

Hold your pointer over the plot to have the legend reflect the highlighted section (figure 6.18).

Use the **Side panel** settings to adjust the plot:

- **Number of levels**. Select the maximum number of taxonomic levels to display.
- **Aggregate sample**. Select a metadata attribute to group aggregate samples by, and use the checkboxes below to specify which samples or groups of samples to include in the plot.
- **Color**. Select a taxonomic level to color the plot by. Lower levels will inherent the color and get different shades of the same color.

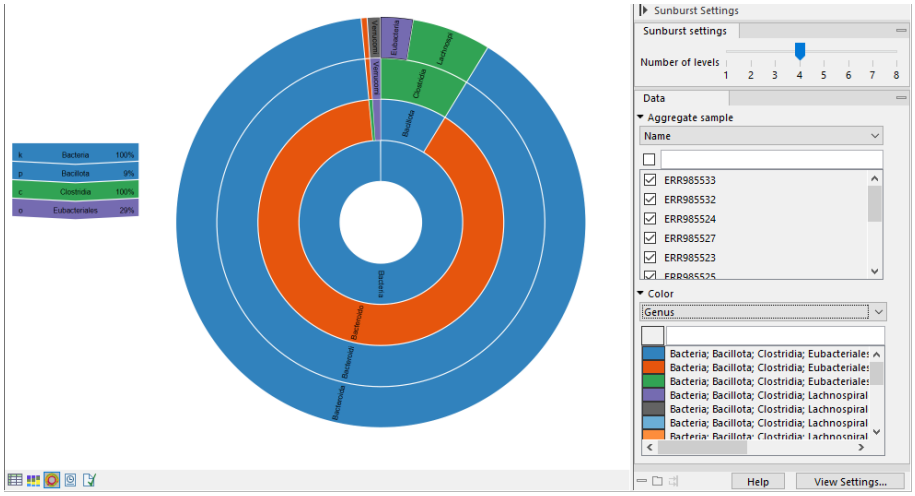


Figure 6.18: *Sunburst view.*

6.4 Taxonomic Profiling

Taxonomic profiling provides insight into the taxonomic composition of whole metagenome samples and estimates the relative abundance of the detected taxa.

Reads are mapped to a reference genome database and are assigned to a reference genome or higher taxonomy level based on their mapping quality score, i.e. the confidence that the read is correctly mapped:

- Reads that map to only one reference location are assigned to that genome.
- Reads that map best to one reference location, but where other almost as good alternatives are found, are assigned to the taxonomy one level up from the best-match genome.
- Reads that map equally well to more than one reference location are assigned to the lowest common ancestor.

If a host genome is provided, reads that map better to this are filtered. Reads are mapped individually to the reference genome database and the host genome. Reads that map to both are assigned to the match with higher mapping score.

For paired reads, when a read pair is broken, either because only one read in the pair matches, or the distance or relative orientation is wrong, both reads are discarded.

Following mapping of reads, qualification and quantification steps refine the results:

- **Qualification**. Determines whether a particular taxon is represented in a sample. This calculation is based on a confidence scores; of whether a reference sequence was assigned reads by pure chance. Any taxon with a confidence score < 0.995 will be ignored and reads will be reassigned to its closest qualified ancestor. By construction, the confidence score is very close to 1.0 except on the Kingdom level of the taxonomy, thus it is not reported.
- **Quantification**. Calculates the abundance of qualified taxa based on the number of assigned reads.

  For data sets with varying read length, the abundance values may optionally be adjusted to correct for a skewed read distribution between taxa, see *Adjust for read length variation* in section 6.4.1.

6.4.1 Taxonomic Profiling parameters

To run the Taxonomic Profiling tool, go to

Tools | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Taxonomic Analysis** ( ) | **Taxonomic Profiling** ( )

In the first dialog, select the sequence list to analyze (figure 6.19).

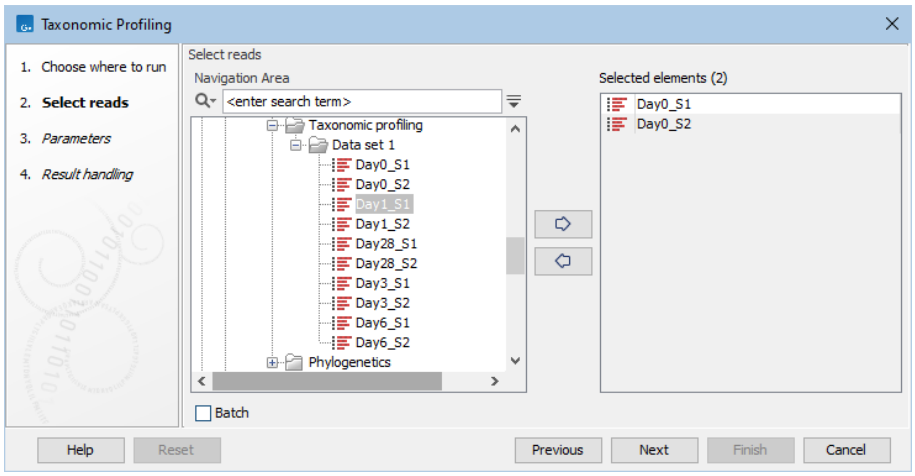


Figure 6.19: *Select a sequence list as input.*

When sequences are selected, click **Next**, and you will see the dialog in (figure 6.20).

Select reference databases:

- **Reference index**. The database of reference genomes that you are analyzing for.

- **Filter host reads**. If this is checked, reads that map better to the specified host genome are filtered and will not count toward taxonomic results.
- **Host genome index**. The host genome, or background genome, represents the reference sequences that may be present in your sample, but that are not the target of your analysis. As an example, to analyze the microbiome from human gut samples, you would specify the human reference index to filter reads that map against the human genome.

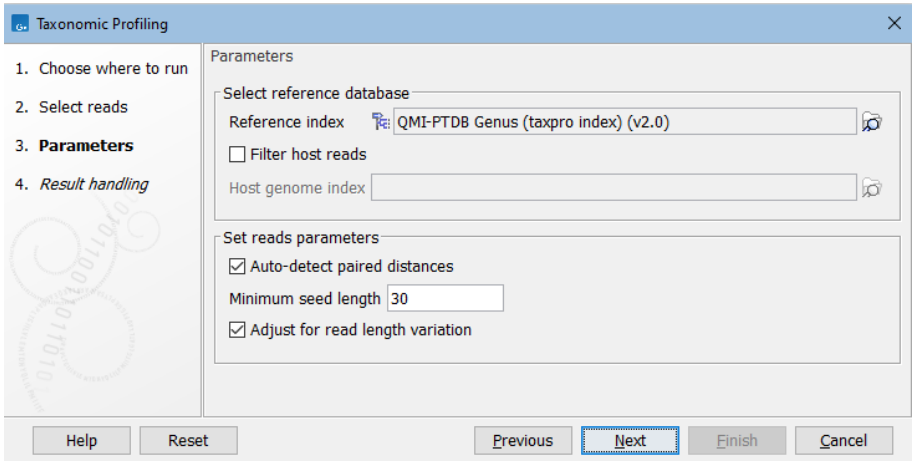


Figure 6.20: *Set the parameters for taxonomic profiling.*

Curated reference databases and taxonomic profiling indexes are available with the Download Curated Microbial Reference Database tool (section 16.1). Alternatively, you can use the Download Custom Microbial Reference Database tool (section 16.2) to create your own custom reference database. To create indexes from reference databases and host genomes, use the Create Taxonomic Profiling Index tool (section 16.5).
Metagenome indexes, available from Download Curated Microbial Reference Database tool (section 16.1) or created with Create Whole Metagenome Index (section 16.4), are not supported.

Under *Set reads parameters*, the following options are available:

- **Auto-detect paired distances**. For paired data, this default choice will automatically calculate the distance between reads in pairs as follows:

  1. A sample of 100,000 reads is extracted randomly from the full data set and mapped against the reference index using a very wide distance interval.
  2. The distribution of distances between the paired reads is analyzed using a method that investigates the shape of the distribution and finds the 0.5% boundaries of the peak. These values make up the distance interval. If fewer than 10,000 reads are mapped as pairs, the range is calculated using the standard deviation.
  3. The full sample is mapped using the calculated distance interval.
  4. The history of the result records the distance interval used.

  If the automatic detection of paired distances is not checked, the tool will use the information about minimum and maximum distance recorded on the input sequence lists.

- **Minimum seed length**. The minimum number of nucleotides with which a read must map to a reference sequence for it to be considered a valid match. Increasing this value will give higher precision of called taxa (true positives). Lowering it will result in more taxa being called, but at the cost of precision (more false positives).
  Apart from the Minimum seed length parameters, reads are mapped with standard read mapping parameters (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html).
- **Adjust for read length variation**. This option is recommended for data sets with varying read lengths to adjust for skewed read distribution between taxa. If checked, abundance and coverage values are adjusted by weighting the reads assigned to a taxon by the number of nucleotides mapped to the taxon. Calculation of abundance and coverage with and without this option checked is explained in section 6.4.3.

6.4.2 Taxonomic Profiling output

Clicking **Next** will allow you to specify the output as shown in figure 6.21.

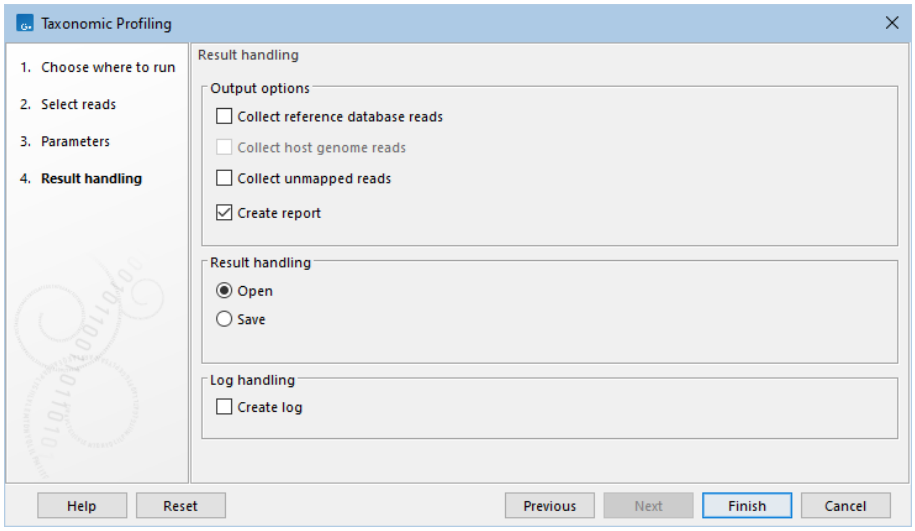


Figure 6.21: *Specify the output.*

The following outputs are available:

- **Collect reference database reads**. Creates a sequence list for each input with reads that were assigned to the reference database.
- **Collect host genome reads**. Creates a sequence list for each input with reads that were assigned to the host genome.
- **Collect unmapped reads**. Creates a sequence list for each input with unassigned reads.
- **Create report**. Creates a summary report.

Sequence list output

The sequence lists with reads that were assigned to the reference database and host genome contain the following annotations:

- **Mapping Quality Score**. Reads with quality score <10 will have been assigned to a higher taxonomy level.
- **Mapping Score**. The score for the read alignment (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html).

In addition, the list of reference database reads contains the **Taxonomy** annotation with the full taxonomy of the taxon to which the read was assigned.

Taxonomic Profiling report

The Taxonomic Profiling report (figure 6.22) contains information about the taxonomic profiling run and databases used.

- **Taxonomic summary**. The number of detected taxa for each taxonomic level.
- **Classification of reads**. Information about the number of reads that were assigned to reference and host databases, or left unclassified. For database matches, both the total number of reads and the number of uniquely matched reads are provided.
- **Reference database summary**. Information about the reference database. Reads that map across Kingdoms will count as a reference database match, but will not contribute to values in the abundance table as no meaningful taxon can be assigned.
- **Host database summary**. Information about the host genome.
- **Auto-detect paired distances**. The calculated paired distance range (provided when the corresponding option was applied).

6.4.3 Taxonomic Profiling abundance table

The abundance table contains the taxa identified in your sample.
To create a multi-sample abundance table, use the Merge Abundance Tables tool (section 7.1). Some of the options mentioned in the following are relevant for multi-sample abundance tables only.

The abundance table can be visualized in Table view (), Stacked Visualization view (), and Sunburst view ().

Table view ()

The table displays a number of columns, some of which are available only when the table is not aggregated by taxonomy (figure 6.23):

- **ID**. The ID of the reference genome or taxon.
- **Name**. The name of the taxon as specified by the reference database.
  If the name contains the text '(Unknown)', this indicates that the taxon corresponds to a higher-level node in the taxonomy, and that this node had a significant amount of reads associated with ancestor taxa that are present in the database but were disqualified. This suggests that there was some organism in the sample for which there is no exact match in the reference database but that is most likely closely related to this taxon.

1 Taxonomic summary

Taxonomic level	Number of classifications
Kingdom	1
Phylum	5
Class	6
Order	10
Family	12
Genus	14
Species	15

2 Classification of reads

	Reads	Uniquely matching reads
Reference database matches	3,306	3,224
Host matches	0	N/A
Unclassified reads	12,466	N/A
Total	15,772	3,224
Reassigned reads	3	N/A

3 Reference database summary

Number of sequences	62,647
Number of basepairs	183,813,930

4 Host database summary

Number of sequences	25
Number of basepairs	3,088,286,401

5 Auto-detected paired distances

Sequence list	Paired distance estimate
Sample A (paired)	398 - 502

Figure 6.22: *The Taxonomic Profiling report.*

- **Taxonomy**: The taxonomy of the taxon as specified by the reference database.
- **Assembly ID**: The ID of the assembly as specified by the reference database. Typically a GenBank assembly accession number.
- **Combined Abundance**: Total abundance across samples.
- **Min**, **Max**, **Mean**, **Median** and **Std**. Minimum, maximum, mean, median and standard deviation of abundance values across samples.
- **Abundance**. Number of reads assigned to the taxon.

  If the option *Adjust for read length variation* is checked, the abundance value will be adjusted

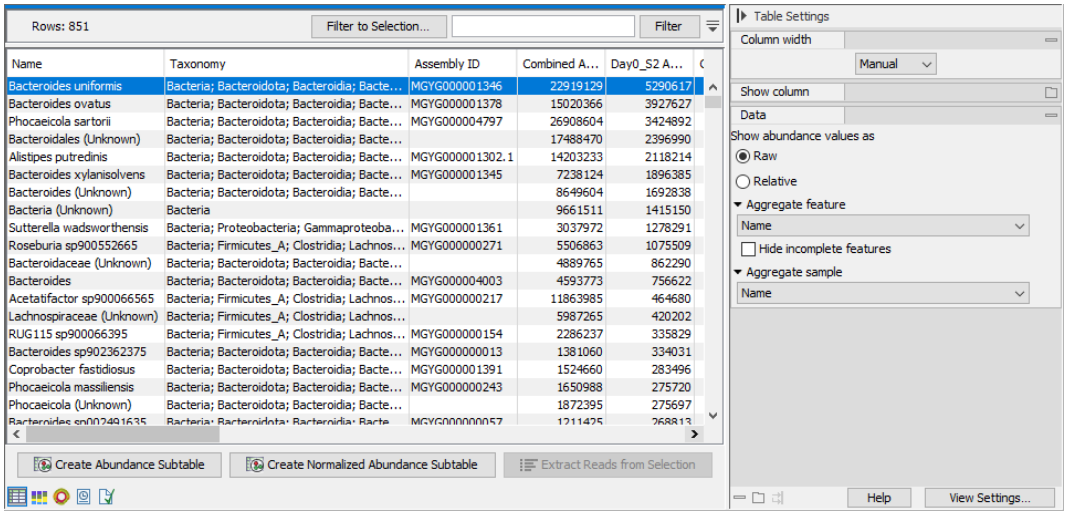


Figure 6.23: *The table view of the abundance table lists name, taxonomy, abundance etc. of the identified taxa.*

by weighing the reads assigned to a taxon by the total number of nucleotides mapped to the taxon:

- Adjusted abundance = (abundance in nucleotides) / (average mapped read length)

For data sets where all reads have similar length, the adjusted abundance will be very similar to the raw read count. Occasionally, weighting may lead to zero reads assigned to a qualified taxon if there are only few, shorter than average reads assigned to this taxon.

- **Coverage**: The coverage estimate of the sample. Coverage calculation depends on the representation of the taxon:
  - The taxon is represented by a single-sequence genome:
    - Coverage = (Weighted nucleotides matching the genome sequence) / (genome sequence length).
      The weight is adjusted based on the number of ambiguous matches for the individual reads. A unique match equals a maximum weight of 1.
  - The taxon is represented by a multi-sequence genome:
    - *Adjust for read length variation* is checked: Coverage = (total number of nucleotides assigned to the genome sequences) / (total genome sequence length).
    - *Adjust for read length variation* is unchecked: Coverage = ((total number of reads assigned to the genome sequences) x (average sample read length)) / (total genome sequence length).
  - The taxon is a parent of filtered, unqualified genomes. The reads from the filtered genomes were reassigned to the parent taxon:
    - *Adjust for read length variation* is checked: Coverage = (total number of nucleotides initially assigned to the filtered genomes) / (average sequence length of filtered genomes).
    - *Adjust for read length variation* is unchecked: Coverage = ((total number of reads initially assigned to the filtered genomes) x (average sample read length)) / (average sequence length of filtered genomes).

The **Side panel** contains the following settings:

- **Show abundance values as**. Switch between Raw and Relative abundance. Relative abundance is calculated as: Relative abundance = (Abundance) / (Sum of abundances).
- **Aggregate feature**. Select a taxonomic level to aggregate abundance values by. For example, select *Family* to display abundance values per Family as opposed to per genome.
- **Hide incomplete features**. Hides features that are not resolved to the taxonomic level selected with the option above.
- **Aggregate sample**. Select a metadata attribute to aggregate samples with same metadata value into one column with combined abundance values.

Below the table, the following actions are available:

- **Create Abundance Subtable**. Creates a table containing only the selected rows.
- **Create Normalized Abundance Subtable**. Creates a table with all rows normalized by the values of a single selected row. The row used for normalization will disappear from the new abundance table. The normalization scales the abundance table linearly, where the scaling factor is calculated by determining the average abundance across all samples and for each sample scale it to the average for the reference. Note that to be enabled, all abundance values in the selected rows must be larger than zero.
- **Extract Reads from Selection**. Extracts reads that were uniquely associated with the selected rows. This option is available if you selected the output *Reads matching the reference database* in the Taxonomic Profiling tool dialog.

Stacked Visualization view ()

The Stacked Visualization view displays the relative abundance of each feature. Use the **Side panel** setting **Bar type** to switch between Bar Chart (figure 6.24) and Area Chart (figure 6.25). The charts can be scaled by percentage, where all bars have the same height of 100%, or counts, where the bar heights are proportional to the number of counts. Different colored bars or areas represent different features. A column represents a sample or - if aggregated by sample level - a group of samples.

Hold your pointer over an area to have the full taxonomy and abundance value displayed in a tooltip.

You can adjusted the view further via the **Side panel** settings. Selected options are:

- **Aggregate feature**. Select a taxonomic level to aggregate abundance values by. For example, select *Family* to have each section in the plot represent a Family instead of a genome.
- **Aggregate sample**. Select a metadata attribute to aggregate samples with same metadata value into one column with combined abundance values. (Relvant for multi-sample abundance tables only). Use the checkboxes below to specify which samples or groups of samples to include in the plot.

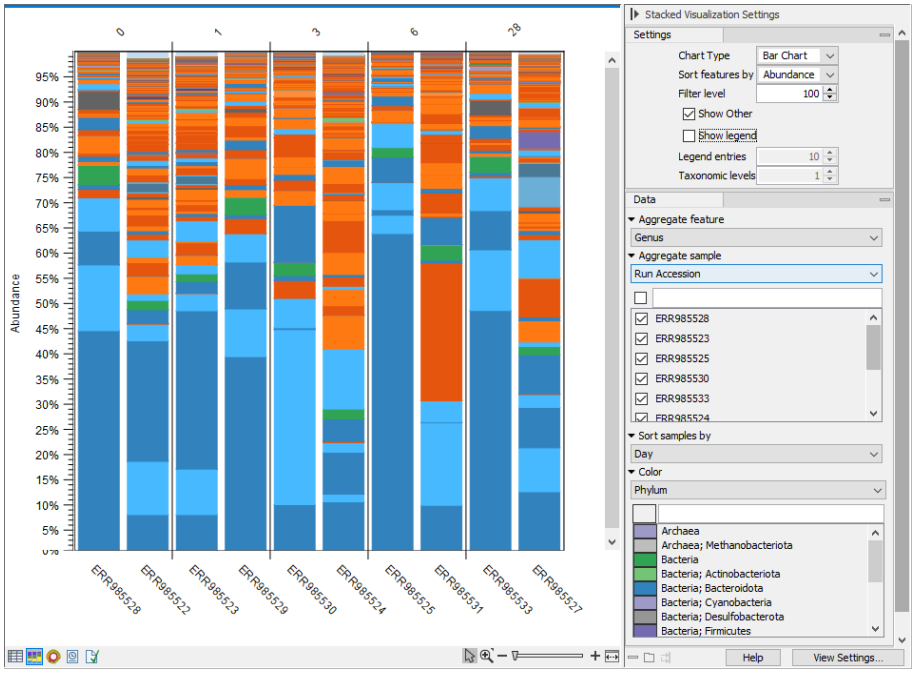


Figure 6.24: *Stacked bar chart.*

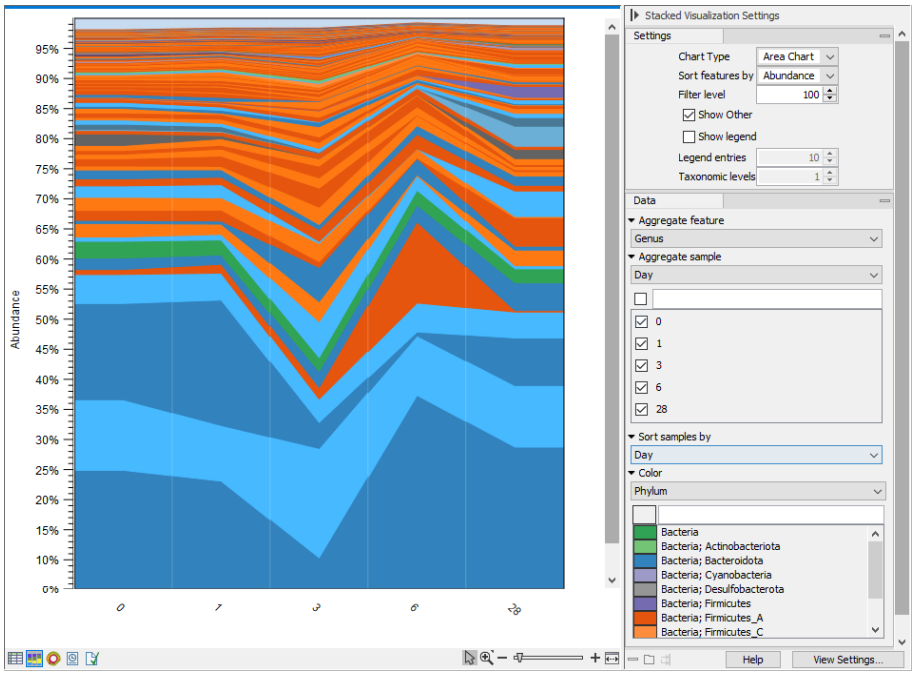


Figure 6.25: *Stacked area chart.*

- **Sort samples by**. Select a metadata attribute to sort samples by the corresponding attribute values. The values are listed above the plot (figure 6.24).
- **Color**. Select a taxonomic level to color the plot by. As an example, if you select *Phylum*, all features belonging to the same Phylum will get different shades of the same base color.

Sunburst view ()

The plot is zoomable. Click on a section to zoom in and render the plot with this section at the center. Click on the center of the plot to zoom out one level at a time.

Hold your pointer over the plot to have the legend reflect the highlighted section (figure 6.26).

Use the **Side panel** settings to adjust the plot:

- **Number of levels**. Select the maximum number of taxonomic levels to display.
- **Aggregate sample**. Select a metadata attribute to group aggregate samples by, and use the checkboxes below to specify which samples or groups of samples to include in the plot.
- **Color**. Select a taxonomic level to color the plot by. Lower levels will inherent the color and get different shades of the same color.

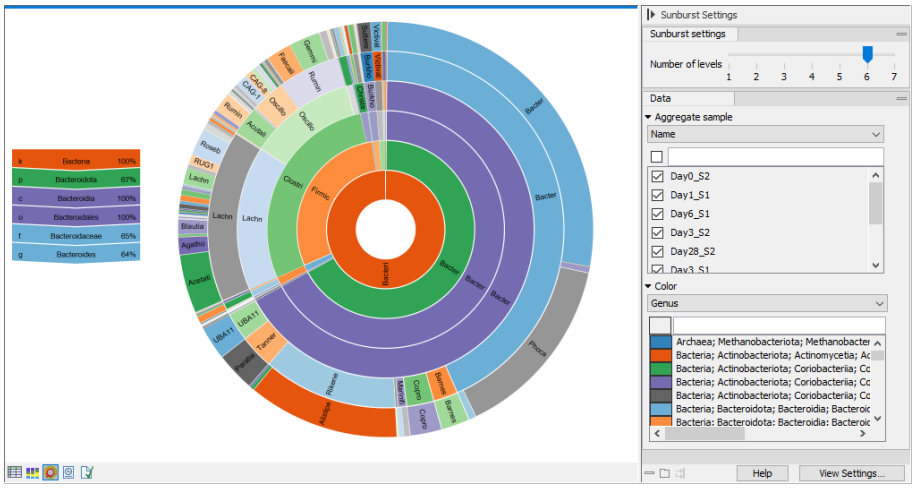


Figure 6.26: *Sunburst view.*

Chapter 7

Abundance Analysis

7.1 Merge Abundance Tables

Merge Abundance Tables merges abundance table from different samples. The abundance tables must be of the same type.

For the following types of abundance tables, the tool will merge based on ID or, if no ID is found, on Name:

- OTU tables
- Whole metagenome taxonomic profiling abundance tables
- Functional profile abundance tables
- Resistance abundance tables

Metadata from input tables is transferred to the merged table.

For ASV abundance tables, merging is based on the ASV sequences. To avoid conflicts, taxonomy annotations will be cleared. Use **Assign Taxonomies to Sequences in Abundance Table** to add taxonomies to the merged table, see section 7.3.

To run the **Merge Abundance Tables** tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Metagenomics** () | **Abundance Analysis** () | **Merge Abundance Tables** ()

Select the tables to merge. These must be of the same type (see above).

7.1.1 Merge Abundance Tables output

The tool outputs a merged abundance table and an optional report.

Merge Abundance Tables report

For each input and the output abundance table, the report contains the following information (figure 7.1):

- **Number of samples**. The number of samples in the abundance table.
- **Number of features**. The number of features in the non-aggregated abundance table (rows).
- **Total abundance**. The total abundance across features and samples. This corresponds to the sum of the "Combined Abundance" column.

1 Input

Name	Number of samples	Number of features	Total abundance
S1_day0_1 (paired, trimmed pairs) (taxonomic profile)	1	66	1,803,646
S1_day34_1 (paired, trimmed pairs) (taxonomic profile)	1	45	1,803,096
S1_day6_1 (paired, trimmed pairs) (taxonomic profile)	1	26	1,895,134
S2_day0_1 (paired, trimmed pairs) (taxonomic profile)	1	27	1,920,924
S2_day34_1 (paired, trimmed pairs) (taxonomic profile)	1	29	1,922,029
S2_day6_1 (paired, trimmed pairs) (taxonomic profile)	1	33	1,913,986

2 Output

Name	Number of samples	Number of features	Total abundance
S1_day0_1 (paired, trimmed pairs) (taxonomic profile) (merged abundance table)	6	90	11,258,815

Figure 7.1: *The Merge Abundance Tables report.*

7.2 Refine Abundance Table

Refine Abundance Table reduces the number of rows in an abundance table by aggregating rows at a higher level in a taxonomy or by filtering rows that do not meet certain criteria. It is possible to aggregate and filter at the same time; in this case, the table is first aggregated, and then filtering is applied. The optional report output includes, among others, a list of the top 50 features as determined by combined abundance. For example, if the table is aggregated at Genus level, the report will list the 50 most abundant genera.

7.2.1 Refine Abundance Table parameters

To run the **Refine Abundance Table** tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Metagenomics** () | **Abundance Analysis** () | **Refine Abundance Table** ()

Select the abundance table to be aggregated or filtered (figure 7.2). Both single and multi-sample abundance tables are supported.

Click on **Next** to set aggregation level.

This tool detects the taxonomy from the input and populates the **Aggregation level** dropdown accordingly (figure 7.3). To perform no aggregation, choose "Do not aggregate" in the dropdown.

When running in a workflow, a taxonomy must be selected that matches the taxonomy of the input abundance table. If the desired taxonomy is not provided in the dropdown, it is possible to choose a **Custom Taxonomy** and enter the level as free text.

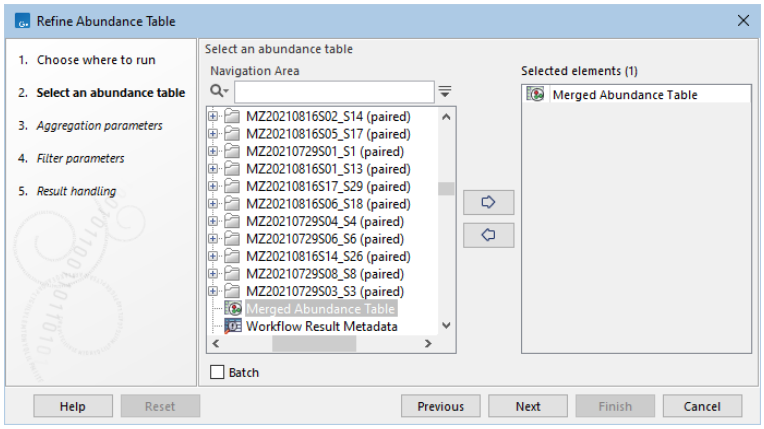


Figure 7.2: *Select an abundance table as input.*

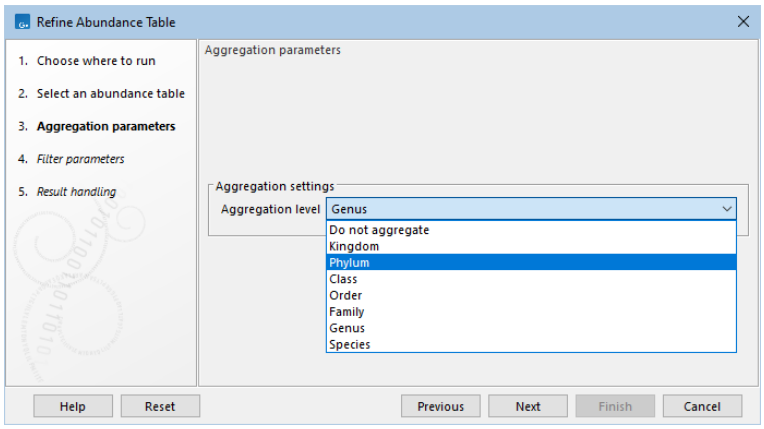


Figure 7.3: *The Aggregation level dropdown is populated based on input taxonomy.*

Three types of filters can be applied after aggregation (figure 7.4):

- Taxonomy filters
  - **Remove features with no taxonomy.** Remove features with no taxonomic information. Examples might include de novo OTUs, ASVs, or results from the Build Functional Profile tool.
  - **Remove features with incomplete taxonomy.** Remove features where the most specific term in the taxonomy after aggregation is undefined. For example, if a table is aggregated at Phylum level, features that lack a Phylum indication will be removed by this filter.
- Prevalence filters

  These filters are designed to remove rows from tables that contain a large number of samples, where the presence of the row would otherwise tend to give a false positive call of differential abundance. The filter is independent of the test statistic [Nearing et al., 2022].
  - **Filter on minimum % of samples.** Remove features with non-zero abundance in less than the given percentage of samples.

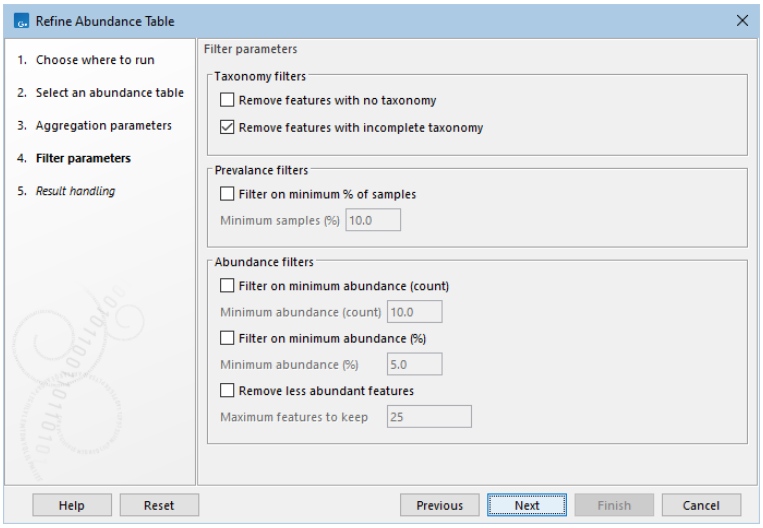


Figure 7.4: *Filter features based on taxonomy, prevalence and abundance.*

  - **Minimum samples (%).** Only keep rows that have non-zero abundance in greater than or equal to this percentage of samples (after rounding up to the nearest whole number). For example, if there are up to 10 samples, and this is set to 10%, then at least 1 sample must have non-zero abundance. For 11-20 samples and 10%, then at least 2 samples must have non-zero abundance. Be careful to set this value such that:
    1. It corresponds to at most the size of the smallest group of samples (a group being samples with similar abundance profile) as otherwise rows that are specifically present in that group will be removed. For example, if you set the percentage so that it corresponds to 4 samples, and your setup has groups A, B, and C of sizes groupA: 3, groupB: 4, groupC: 4, the filter will remove rows that have non-zero abundance in only groupA.
    2. It is sufficiently large to have an effect. For example, with 6 samples, the default value of 10% will not have an effect. To remove rows that have non-zero abundance in at least two samples, one might set the value to 20% (20% of 6 is 1.2, which is rounded up to 2 samples).

- Abundance filters
  - **Filter on minimum abundance (count).** Remove features whose combined abundance is less than a given value.
  - **Minimum abundance (count).** The minimum combined abundance to keep.
  - **Filter on minimum abundance (%).** Remove features whose combined abundance is less than a given percentage of the total abundance of all the features in the table.
  - **Minimum abundance (%).** The minimum percentage to keep. Note that e.g. if the percentage is set to 1.0%, then at most 100 rows can be returned.
  - **Remove less abundant features.** Features are sorted by combined abundance. Only a fixed number of rows of the sorted table will be kept by this filter.
  - **Maximum features to keep.** The fixed number of rows that may pass the filter. Note that, because all filters are applied independently of each other, the output abundance table may have fewer rows than this, because another filter has removed rows.

When filtering, note that the Differential Abundance Analysis tool (see section 7.7), assumes that rows with similar abundance will have similar parameters, and uses this to improve parameter estimates by sharing information across rows. Therefore differential abundance results for aggressively filtered tables may not be as sensitive or precise as results for tables that contain more rows.

7.2.2 Refine Abundance Table output

The tool outputs an abundance table and an optional report.

Refine Abundance Table report

1 Taxonomy summary

Taxonomic level	Number of classifications
Kingdom	1
Phylum	10
Class	12
Order	25
Family	46
Genus	142
Species	0

2 Filtered features

Filtered due to	Filtered features
Missing feature	0
Incomplete feature	44
Samples filter	0
Minimum abundance filter	0
Top N features filter	0
Total filtered	44

3 Top features

Feature	Abundance	Abundance (% of unfiltered)
Bacteroides	3,014,515	38.21
Phocaeicola	1,041,859	13.20
Alistipes	705,653	8.94
Barnesiella	338,580	4.29
Parabacteroides	251,130	3.18
Akkermansia	180,186	2.28
Faecalibacterium	149,043	1.89
Sutterella	135,421	1.72
Coprobacter	107,772	1.37
Agathobacter	91,124	1.15
Roseburia	54,441	0.69
Lachnospira	51,957	0.66
Ruminiclostridium_E	43,061	0.55

Figure 7.5: *The Refine Abundance Table report.*

The Refine Abundance Table report (figure 7.5) contains a summary of the number of features at the given taxonomic level in the output table. If aggregation has been performed, the number of features at a taxonomic level more specific than the aggregated level will be zero.

The report also contains a summary of why different features were filtered. When a feature is filtered by multiple filters, it is listed as being filtered by the first of the filters in the table that it failed.

Finally, the report lists the top 50 features in the output table when sorted by abundance. The corresponding abundances are shown both as a raw abundance, and as a percentage of the total

abundance of the table after any aggregation and before any filtering.

7.3 Assign Taxonomies to Sequences in Abundance Table

The Assign Taxonomies to Sequences in Abundance Table tool lets you add taxonomies to abundance table features that have sequences associated. This is useful for annotating amplicon sequence variant (ASV) tables, and OTU tables with de novo OTUs where sequences are not annotated by the initial analysis tools.

The tool requires a reference index and works by mapping each sequence from the abundance table to this reference index. The underlying analysis is the same as for **Taxonomic Profiling**, see section 6.4.

Creating the required reference index

You create a reference index using **Create Taxonomic Profiling Index**, see section 16.5. As input, you will need a reference database, i.e. a sequence list containing reference sequences with taxonomy annotations.

Reference database can be obtained using one of the Download Database tools. The choice of reference database depends on your data.

For amplicon data, consider the reference databases available with **Download Amplicon-Based Reference Database**, see section 15.1.

For whole genome data, you may use the databases from **Download Curated Microbial Reference Database**, see section 16.1. Alternatively, create your own reference database with **Download Custom Microbial Reference Database**, see section 16.2.

Running the tool

To run the **Assign Taxonomies to Sequences in Abundance Table** tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Metagenomics** () | **Abundance Analysis** () | **Assign Taxonomies to Sequences in Abundance Table** ()

Select the abundance table with sequences to be annotated.

Select the reference index to map the abundance table sequences to (figure 7.6).

Choose settings for taxonomic assignment:

- **Minimum similarity percentage** Sequences in the abundance table must be at least this similar to sequence in the reference index to be matched and get a new taxonomy assigned.
- **Clear existing taxonomy** All existing abundance table taxonomy annotations are removed. Only abundance table sequences with a reference index match will get a taxonomy assignment.
- **Overwrite existing taxonomy** Abundance table sequences with a reference index match will get a new taxonomy assigned. Sequences with no match will retain the existing taxonomy annotation.

- **Use existing taxonomy when present** Only abundance table sequences that do not already have a taxonomic annotation will get a new taxonomy assigned.

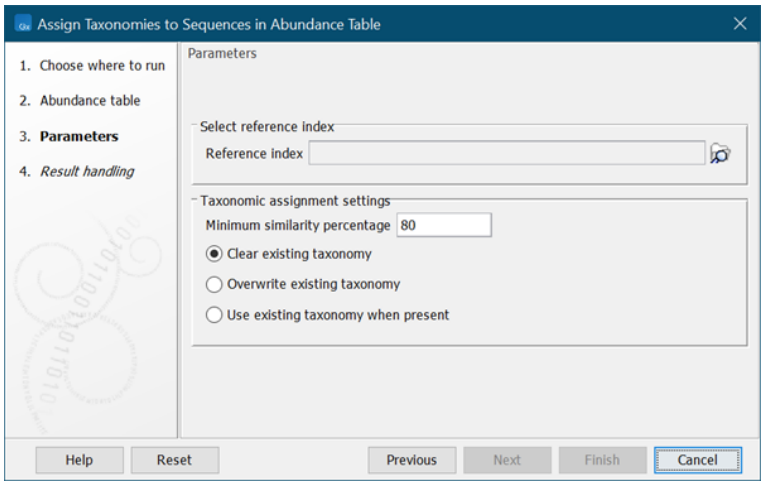


Figure 7.6: *Select reference index and set parameters for taxonomic assignment.*

Select **Create Report** to generate a report with summary information on taxonomic assignment.

Assign Taxonomies to Sequences in Abundance Table output

- **Abundance table with taxonomy annotations** The new taxonomy assignments are listed in the **Taxonomy** column.
- **Assign taxonomies report** The report contains the following sections:
  - **Summary** Information on the sequences in the abundance table.
  - **Reference index summary** Information on the reference index.
  - **Taxonomy assignment**
    - **Sequences with reference index match**: Sequences that met the Minimum similarity treshold.
      - **Taxonomy assigned (was blank)**: Taxonomy was blank, new taxonomy has been assigned.
      - **Taxonomy updated**: The existing taxonomy has been replaced.
      - **Existing taxonomy retained**: The existing taxonomy is retained. This can happen when the taxonomy of the matched reference index sequence is identical to the existing taxonomy, or when *Taxonomy assignment* was set to *Use existing taxonomy when present*.
    - **Sequences with no reference index match (insufficient similarity)**: Sequences that did not meet the Minimum similarity threshold.
      - **Existing taxonomy retained**: Sequences for which an existing taxonomy remains.
      - **No taxonomy**: Sequences with no taxonomy.

7.4 Alpha Diversity

Alpha diversity is the diversity within a particular area or ecosystem, usually expressed by the number of species (i.e., species richness) in that ecosystem. Alpha diversity estimates are calculated from a series of rarefaction analyses and hence dependent on sampling depth.

The Alpha Diversity tool takes abundance tables as input. Abundance tables can be generated in the workbench by various tools, for example: OTU clustering, Build Functional Profile and Taxonomic Profiling. With the first two tools, the abundance tables generated are count-based, and Alpha diversity measures calculated from such tables give an absolute number of species. However, when using an abundance table generated by e.g. the Taxonomic Profiling tool, Alpha diversity results will not give an absolute number of species, but rather estimates that are useful for comparative studies, i.e., to assess the depth of sequencing, or to compare different communities.

To run the **Alpha Diversity** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Abundance Analysis** ( ) | **Alpha Diversity** ( )

Choose an abundance table to use as input.

The next wizard window offers you to set up different analysis parameters (figure 7.7).

For example, you can calculate metrics at a specific taxonomic level: the tool will then aggregate the features by taxonomy (so that OTUs from the same phylum will be grouped together) before computing the metric. The default value is to not aggregate by taxonomy. You then select which diversity measures to calculate (see section 7.4.1).

If you are working with OTU abundance tables, you can also specify an appropriate phylogenetic tree for computing phylogenetic diversity. In that case, you must have aligned the OTUs and constructed a phylogeny before running the Alpha Diversity tool. Note that the "Evaluate at taxonomic level" option described above does not apply to the "Phylogenetic Diversity" metric, since that metric is not using taxonomic information, but is making use of a phylogenetic tree based on OTU sequences.

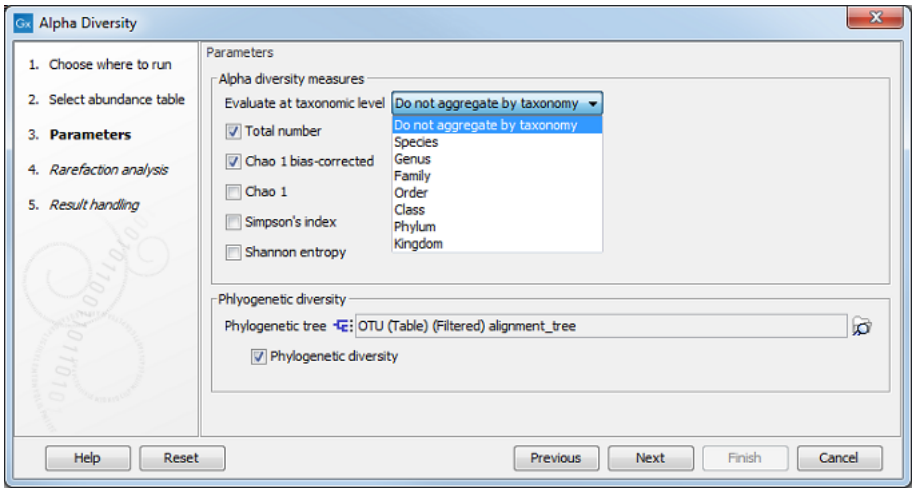


Figure 7.7: *Set up parameters for the Alpha Diversity tool.*

In the following dialog (figure 7.8), set up the rarefaction analysis parameters.

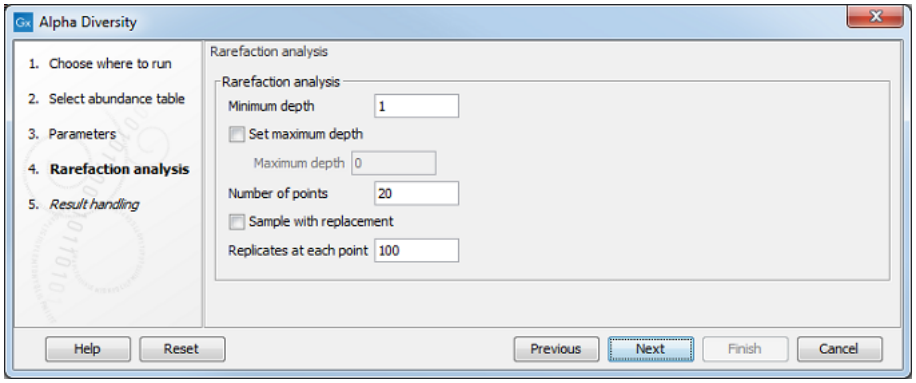


Figure 7.8: *Set up parameters for the Rarefaction analysis.*

The rarefaction analyses are done differently depending on the type of abundance table used as input. For abundance tables where abundances are counts, such as OTU and functional abundance tables, rarefaction is calculated by sub-sampling the abundances in the different samples to the same depths. For taxonomic profiling abundance tables, where abundances are coverage estimates, sub-sampling is not possible. Instead, diversity is estimated using a probabilistic model corresponding to our qualification criteria (see section 6.4).

The rarefaction analysis parameters will define the granularity of the alpha diversity curve.

- **Minimum depth to sample** is set to 1 by default.
- **Maximum depth to sample** If this option is not checked, the maximum depth is set it to the total number of reads (in the case of one sample) or the total number of reads of the sample with most reads.
- **Numbers of points** Number of different depths to be sampled. For example, if you choose to sample 5 depths between 1000 and 5000, the algorithm will sub-sample each sample at 1000, 2000, 3000, 4000, and 5000 reads.
- **Sample with replacement** Whether the sampling should be performed with or without replacement.
- **Replicates at each depth** (for counts-based abundance tables only). How many times the algorithm sub-samples the data at each depth.

The tool will generate a graph for each selected Alpha diversity measure (figure 7.9). Using the Lines and dots editor on the right hand side panel, it is possible to color samples according to groups defined by associated metadata. Note that you can filter metadata by typing the appropriate text in the field above each list of metadata elements. This is an easy way to change the visualization of a group of data at once.

Note that the option "Show derived legend info" is enabled by default (figure 7.10). According to this setting, the legend(s) for which metadata categories happen to be "shared" for all items in the legend will display the dependencies between the different categories. In this example, the "Location" category determines Dot Type, and the "Antibiotic" category determines Line Color. For this particular data set, all samples with a specific location have the same antibiotic resistance. The "Show derived legend info" option enables the legends to show such implicit

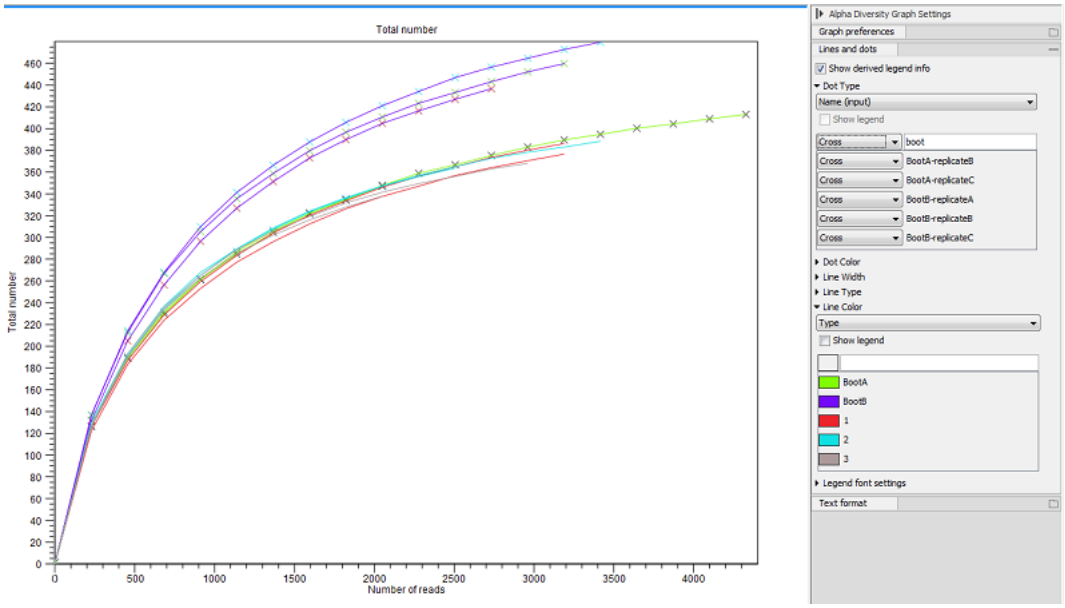


Figure 7.9: *An example of Alpha Diversity graph based on phylogenetic diversity.*

dependencies in the data. If such a visualization is not wished for, the option can be disabled, and the legend will show only the metadata category values that were explicitly selected in the right hand side panel.

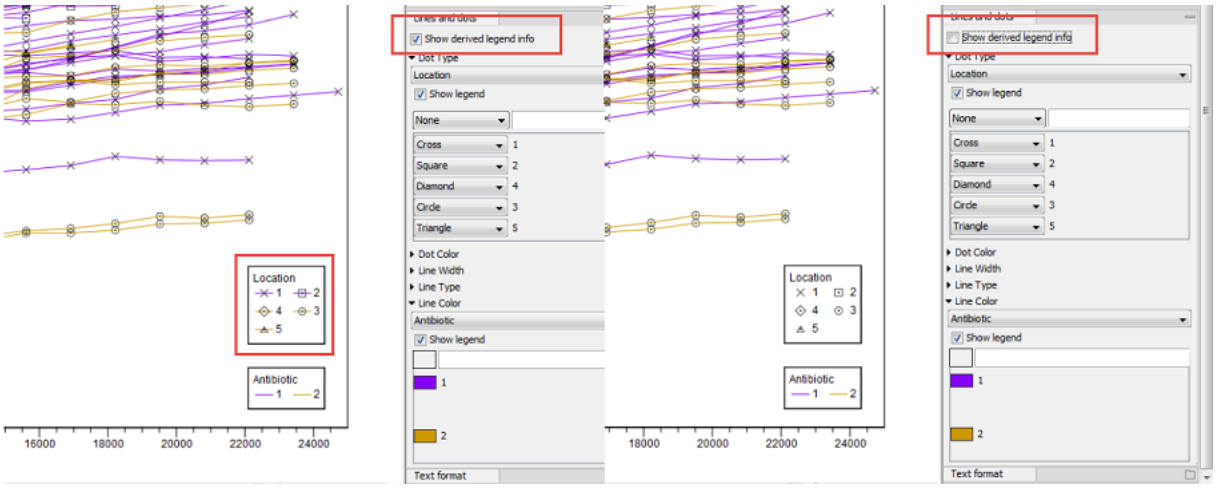


Figure 7.10: *Example of the difference between having the "Show derived legend info" enabled or disabled. When enabled, the legend helps visualize that "location" and "antibiotic" are dependent for this particular data set.*

It is also possible to view the Alpha diversity measures as Box plot to see if samples of a certain group are significantly different than those of other groups (figure 7.11). For example, one can check if soils of a certain type contain more bacterial species than other samples.

The box plot view can also display the following statistics:

- **Rarefaction level** This drop down menu allows to choose which value of the rarefaction curve should be used. The values of "Rarefaction level" are the same as the horizontal axis

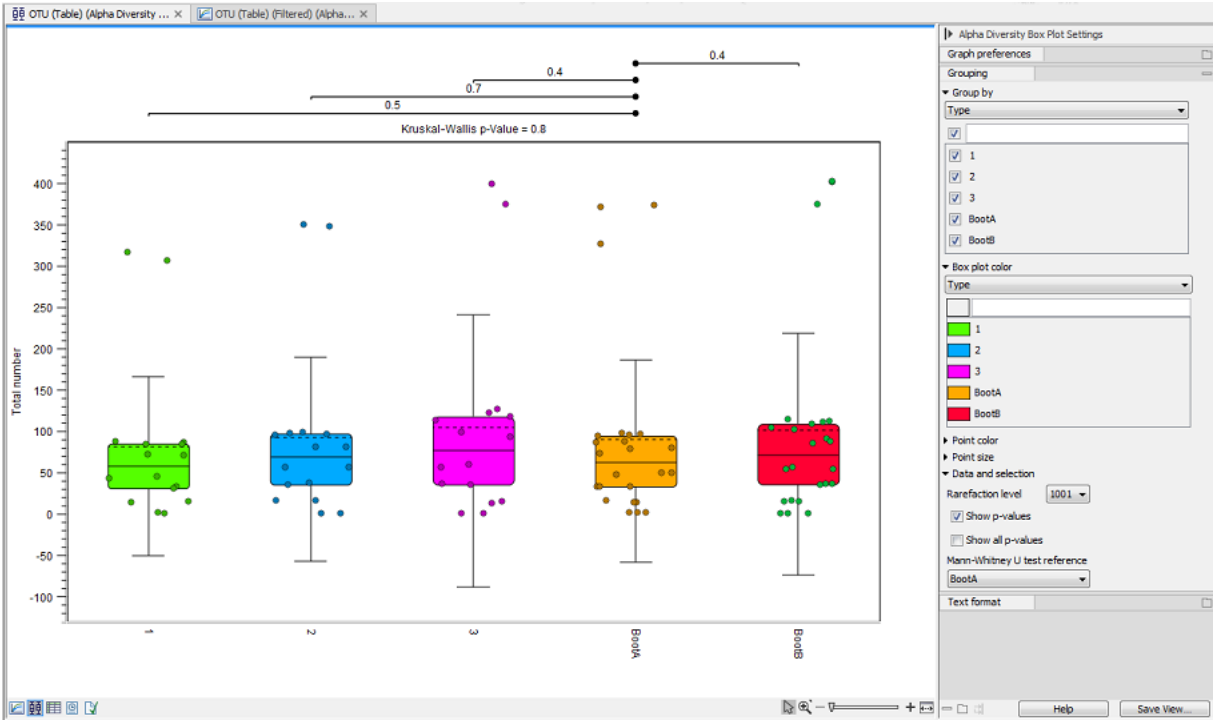


Figure 7.11: *Alpha diversities shown in a box plot.*

in the Alpha Diversity Graph view and correspond to the depths at which the input data is sub-sampled before calculating the diversity metric in the Alpha Diversity tool.

- **Kruskal-Wallis H test** This test is used to assess whether the values originate from the same distribution or whether their distribution is different depending on the group they belong to. This test is a nonparametric alternative to ANOVA (i.e., it does not depend on the data following a given distribution - the normal distribution in case of ANOVA). A significant p-value for the Kruskal-Wallis test means that at least one group follows a different distribution, but does not specify which pairs have different distributions.

- **Mann-Whitney U test** The tool therefore performs a pair-wise **Mann-Whitney U test** to specifically determine which pairs of groups follow different distributions. These statistical tests are performed when the "Show p-values" option is checked. If the "Show all p-values" is checked, all pairwise Mann-Whitney U tests are performed, while when it is not checked, only the pairs that contain the reference group specified in the "Mann-Whitney U test reference" option are considered.

7.4.1 Alpha diversity measures

The available diversity measures are:

- Total number: The number of features (e.g. OTUs when doing OTU clustering, GO terms when building functional profiles or organisms when performing taxonomic profiling) observed in the sample.

- Chao 1 bias-corrected: Chao1-bc $= D + \frac{f_1(f_1-1)}{2(f_2+1)}$.

- Chao 1: Chao1 $= D + \frac{f_1^2}{2f_2}$.
- Simpson's index: SI $= 1 - \sum_{i=1}^{n} p_i^2$.
- Shannon entropy: H $= \sum_{i=1}^{n} p_i \log_2 p_i$.

where n is the number of features; D is the number of distinct features observed in the sample; f_1 is the number of features for which only one read has been found in the sample; f_2 is the number of features for which two reads have been found in the sample; and p_i is the fraction of reads that belong to feature i.

Note that Chao-based methods deal with singletons and doubletons, i.e., rows with exactly one or two reads (counts) associated. These measures are thus not available for whole metagenome taxonomic profiles that are characterized by coverage estimate.

The following distances are also available:

- Phylogenetic diversity: $PD = \sum_{i=1}^{n} b_i I(p_i > 0)$

where n is the number of branches in the phylogenetic tree, b_i is the length of branch i; p_i is the proportion of taxa descending from branch i; and the indicator function $I(p_i > 0)$ and $I(p_i^B > 0)$ assumes the value of 1 if any taxa descending from branch i is present in the sample or 0 otherwise.

7.5 Beta Diversity

Beta diversity examines the change in species diversity between ecosystems. The analysis is done in two steps. First, the tool estimates a distance between each pair of samples (see section 7.5.1). Once the distance matrix is calculated, the beta diversity analysis tool performs Principal Coordinate Analysis (PCoA) on the distance matrices. These can be visualized by selecting the PCoA icon ( ) in the bottom of the Beta Diversity results ( ).

The Beta Diversity tool takes abundance tables as input. Abundance tables can be generated in the workbench by various tools, for example: OTU clustering, Build Functional Profile and Taxonomic Profiling.

If you are working with an OTU table, you can specify an appropriate phylogenetic tree for computing phylogenetic diversity. In that case, you must have aligned the OTUs and constructed a phylogeny before running the Beta Diversity tool.

To run the **Beta Diversity** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Abundance Analysis** ( ) | **Beta Diversity** ( )

Select an abundance table with more than one sample as input (e.g., a multi-sample OTU or merged abundance table) and set the parameters for the beta diversity analysis as shown in figure 7.12.

The output of the tool is a 3D PCoA plot (figure 7.13) that can also be seen as a table or a 2D Principal Coordinate Plot.

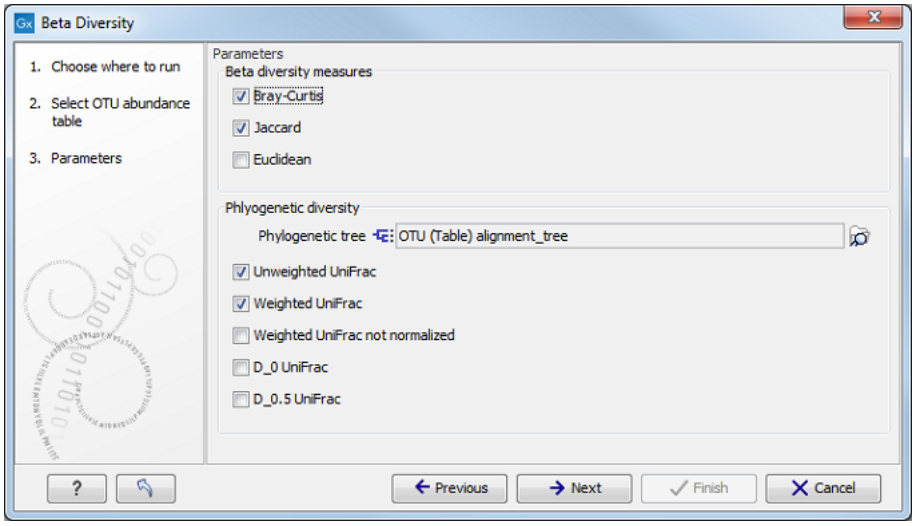


Figure 7.12: *Set up parameters for the Beta diversity tool.*

If you have problems viewing the 3D plot, please check your system matches the requirements for 3D viewers. See https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html.

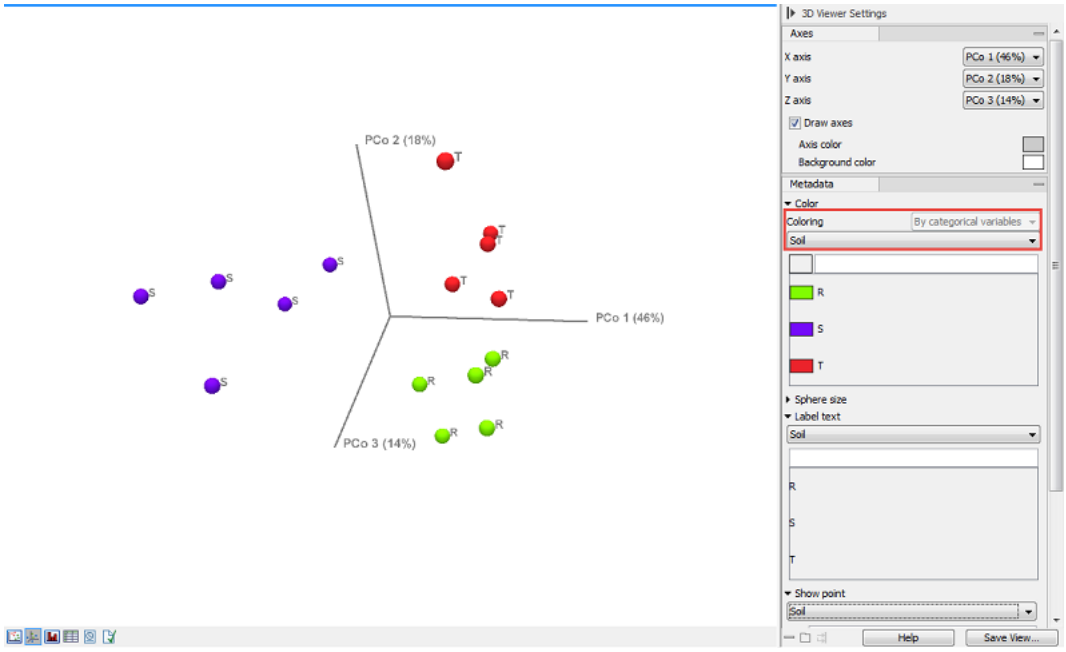


Figure 7.13: *Beta diversity results seen as a 3D PCoA, with coloring done according to the categories dedined by the metadata.*

Use the settings in the right hand Side Panel of the PCoA (2D or 3D) to modify the plot visualization.

In the section Metadata, the **Color** menu (1) allows you to choose whether you want your data to be colored according to categorical variables (the ones defined by the metadata, as seen in figure 7.13) or by abundance values (figure 7.14).

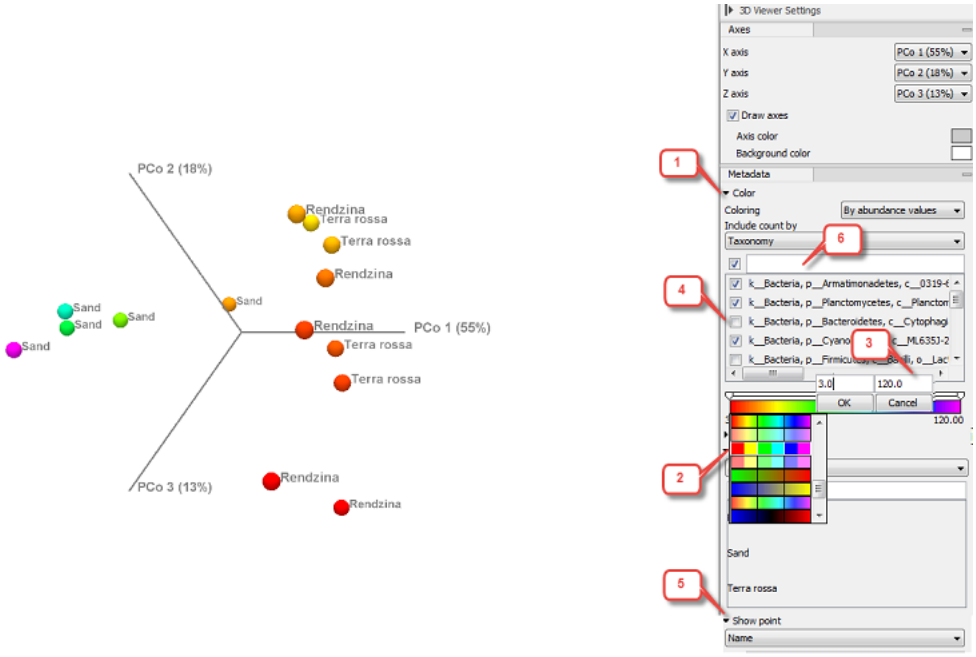


Figure 7.14: *Beta diversity results seen as a 3D PCoA, with coloring done according to taxonomic abundance values.*

Coloring per abundance values is done with a gradient scheme. Click on the gradient bar to choose from several color schemes (2). Double-click on the slider to set specific values for the gradient (3).

When coloring "By abundance values", it is possible to control the abundance value calculation through existing metadata fields such as Name, EC numbers, or Taxonomy (depending on the type of abundance table input). The drop down menu will then display the items that can be deselected (4) if you want to remove them from the abundance value calculation: this will not remove any of the data point from the PCoA view, but just change the abundance values of the affected point and therefore its coloring. To remove a data point from the plot, use the **Show point** menu (5) below in the Side Panel.

As in other metadata Side Panel features, use the field above each menu (6) to filter that menu. This is particularly helpful when menus are very long, as is the case with taxonomies for example.

7.5.1 Beta diversity measures

The following beta diversity measures are available:

- Bray-Curtis: $B = \frac{\sum_{i=1}^{n} |x_i^A - x_i^B|}{\sum_{i=1}^{n} (x_i^A + x_i^B)}$
- Jaccard: $J = 1 - \frac{\sum_{i=1}^{n} \min(x_i^A, x_i^B)}{\sum_{i=1}^{n} \max(x_i^A, x_i^B)}$
- Euclidean: $E = \sum_{i=1}^{n} \sqrt{\left(x_i^A - x_i^B\right)^2}$

where n is the number of OTUs and x_i^A and x_i^B are the abundances of OTU i in samples A and B, respectively.

The following distances are also available:

- Unweighted UniFrac: $d^{(U)} = \frac{\sum_{i=1}^{n} b_i \left| I(p_i^A>0) - I(p_i^B>0) \right|}{\sum_{i=1}^{n} b_i}$
- Weighted UniFrac: $d^{(W)} = \frac{\sum_{i=1}^{n} b_i \left| p_i^A - p_i^B \right|}{\sum_{i=1}^{n} b_i (p_i^A + p_i^B)}$
- Weighted UniFrac not normalized: $d^{(w)} = \sum_{i=1}^{n} b_i \left| p_i^A - p_i^B \right|$
- D_0 UniFrac: The generalized UniFrac distance $d^{(0)} = \frac{\sum_{i=1}^{n} b_i \left| \frac{p_i^A - p_i^B}{p_i^A + p_i^B} \right|}{\sum_{i=1}^{n} b_i}$
- D_0.5 UniFrac: The generalized UniFrac distance $d^{(0.5)} = \frac{\sum_{i=1}^{n} b_i \sqrt{p_i^A + p_i^B} \left| \frac{p_i^A - p_i^B}{p_i^A + p_i^B} \right|}{\sum_{i=1}^{n} b_i \sqrt{p_i^A + p_i^B}}$

where n is the number of branches in the phylogenetic tree, b_i is the length of branch i; p_i^A and p_i^B are the proportion of taxa descending from branch i for samples A and B, respectively; and the indicator functions $I(p_i^A > 0)$ and $I(p_i^B > 0)$ assume the value of 1 if any taxa descending from branch i is present in samples A and B, respectively, or 0 otherwise.

The unweighted UniFrac distance gives comparatively more importance to rare lineages, while the weighted UniFrac distance gives more important to abundant lineages. The generalized UniFrac distance $d^{(0.5)}$ offers a robust tradeoff [Chen et al., 2012].

7.6 PERMANOVA Analysis

PERMANOVA Analysis (PERmutational Multivariate ANalysis Of VAriance, also known as non-parameteric MANOVA [Anderson, 2001]), can be used to measure effect size and significance on beta diversity for a grouping variable. For example, it can be used to show whether OTU abundance profiles of replicate samples taken from different locations vary significantly according to the location or not. The significance is obtained by a permutation test.

To perform a PERMANOVA analysis, go to:

> **Tools** | **Microbial Genomics Module** () | **Metagenomics** () | **Abundance Analysis** () | **PERMANOVA Analysis** ()

Choose an abundance table with more than one sample as input (e.g., a multi-sample OTU or merged abundance table) and specify the metadata group you would like to test. You will need more than one replicate in the metadata group you select.

In the "Parameters" dialog (figure 7.15), you can choose which Beta diversity measure to use (see section 7.5.1). If you are working with OTU abundance tables, you can also specify in the

next dialog the phylogenetic tree reconstructed from the alignment of the most abundant OTUs and the phylogenetic diversity measures you wish to use for this analysis. Finally, choose how many permutations should be performed (the default is set to 99,999).

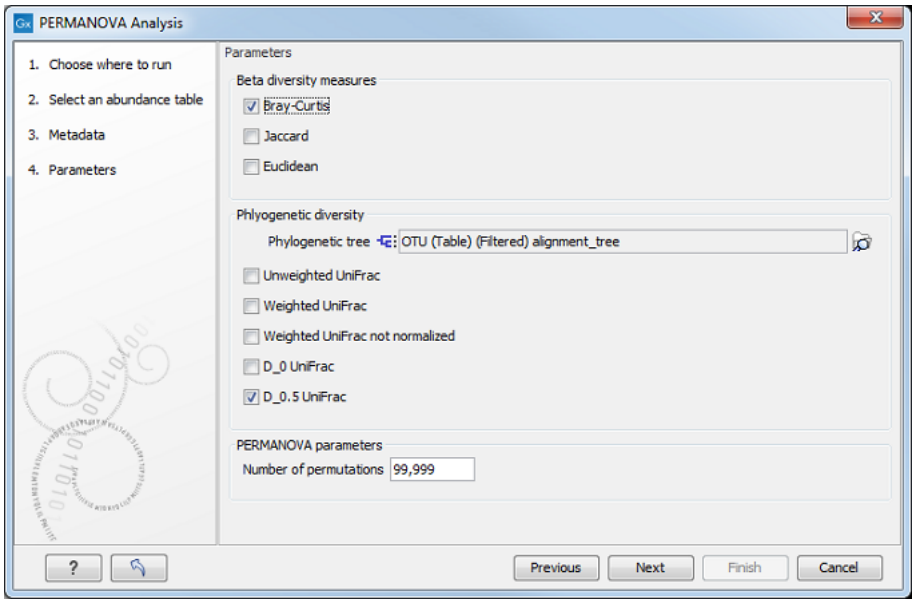


Figure 7.15: *Beta diversity and Phylogenetic diversity measures are included in the PERMANOVA analysis.*

The output of the analysis is a report which contains two tables for each beta diversity measure used:

- A table showing the metadata variable used, its groups and the results of the test (pseudo-f-statistic and p-value)
- A PERMANOVA analysis for each pair of groups and the results of the test (pseudo-f-statistic and p-value). Bonferroni-corrected p-values (which correct for multiple testing) are also shown.

7.7 Differential Abundance Analysis

This tool performs a generalized linear model differential abundance test on samples, or groups of samples defined by metadata. The tool models each feature (e.g., an OTU, an organism or species name or a GO term) as a separate Generalized Linear Model (GLM), where, after performing TMM normalization, it is assumed that abundances follow a Negative Binomial distribution. The Wald test is used to determine significance between group pairs, whereas a Likelihood Ratio test is used in the Across groups (ANOVA-like) comparison. The underlying statistical model is the same as the one used by the Differential Expression for RNA-Seq tool described in details here: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Differential_Expression.html.

Before running **Differential Abundance Analysis**, an abundance table can be aggregated to a desired level, either manually in the abundance table Table View by setting the "Aggregate feature" level in the **Side Panel**, highlighting the rows and clicking the **Create Abundance Subtable** in the bottom, or by using **Refine Abundance Table** (section 7.2).

To run the **Differential Abundance Analysis** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Abundance Analysis** ( ) | **Differential Abundance Analysis** ( )

Select an abundance table with more than one sample as input (e.g., a multi-sample OTU or merged abundance table), and specify if you want to test differential abundance based on metadata defined groups of samples (figure 7.16). It is also possible to correct the results based on a metadata defined group of samples. Finally you can choose whether you want the comparison to be done across groups, between all group pairs or against a control group.

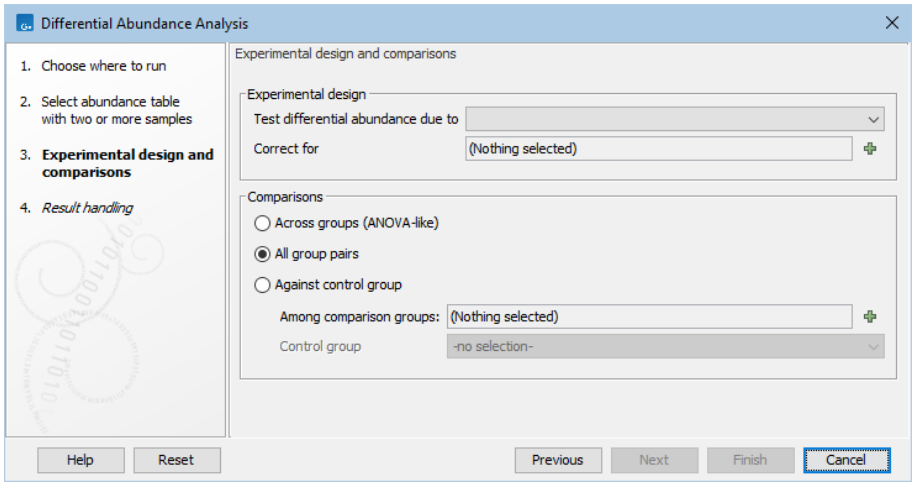


Figure 7.16: *Specify an abundance table and all other parameters.*

The tool generates a Venn diagram for three pairwise comparisons at a time (figure 7.17). You can select which comparisons should be shown using the drop down menus in the side panel. Clicking a circle segment in the Venn diagram will select the samples of this segment in the differential abundance analysis table view. The table summarizes abundances, fold changes, differential abundance p-values, multi-sample corrected p-values, etc.

The values included in the table for each pairwise comparison are:

- **Max group means** For each group in the statistical comparison, the average measured abundance or expression value is calculated. The Max Groups Means is the maximum of the average values.
- **log2 fold change** The logarithmic fold change.
- **Fold change** The (signed) fold change. Genes/transcripts that are not observed in any sample have undefined fold changes and are reported as NaN (not a number). Note: Fold changes are calculated from the GLM, which corrects for differences in library size between the samples and the effects of confounding factors. It is therefore not possible to derive these fold changes from the original counts by simple algebraic calculations.
- **P-value** Standard p-value. Genes/transcripts that are not observed in any sample have undefined p-values and are reported as NaN (not a number).
- **FDR p-value** The false discovery rate corrected p-value.
- **Bonferroni** The Bonferroni corrected p-value.

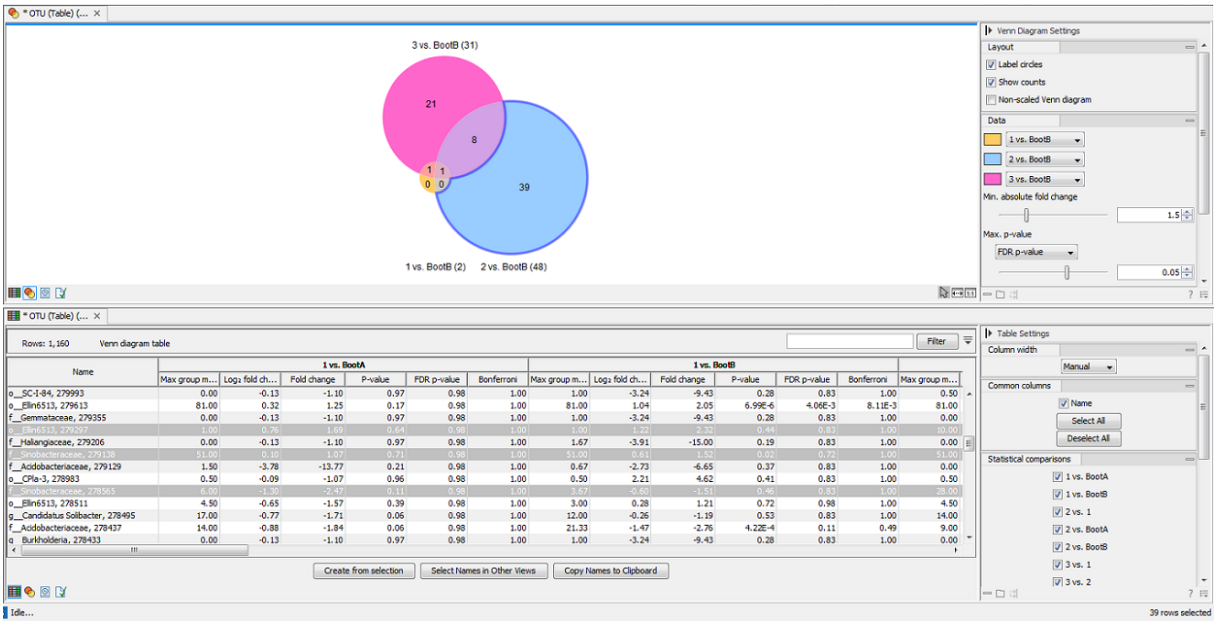


Figure 7.17: *Venn diagram for three comparisons at a time. Selecting a segment will highlight the samples in the differential abundance table opened below in split view.*

It is possible to create a subset list of samples using the Create from selection button. As usual, the table can be adjusted with the right hand side panel options: it is possible to adjust the column layout, and select which columns should be included in the table.

7.8 Create Heat Map for Abundance Table

The Create Heat Map for Abundance Table tool simultaneously clusters samples and features (taxa), showing a two dimensional heat map of taxonomic abundances.

The following filtering and normalization is performed:

- 'log CPM' (Counts per Million) values are calculated for each feature. The CPM calculation uses the effective library sizes as calculated by the TMM normalization.
- Z-score normalization is performed across samples for each feature: the counts for each feature are mean centered, and scaled to unit variance.

For more detail about these steps, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=RNA_Seq_normalization.html.

7.8.1 Clustering of features and samples

Hierarchical clustering clusters taxa by the similarity of their taxonomic profiles over the set of samples, and samples by the similarity of taxonomic composition over the set of features (taxa).

Each clustering has a tree structure that is generated as follows:

1. Letting each taxa or sample be a cluster.

2. Calculating pairwise distances between all clusters
3. Joining the two closest clusters into one new cluster.
4. Iterating 2-3 times until there is only one cluster left (which contains all the taxa or samples).

In the resulting tree, the length of branches reflect the distance between clusters.

To create a heat map, run the **Create Heat Map for Abundance Table** tool, available from:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Abundance Analysis** ( ) | **Create Heat Map for Abundance Table** ( )

Select an abundance table with two or more samples as input (e.g., a multi-sample OTU or merged abundance table) and click **Next**.

Specify a distance measure and a cluster linkage (figure 7.18). The distance measure specifies how distances between two taxa or samples should be calculated. The cluster linkage specifies how the distance between two clusters, each consisting of a number of taxa or samples, should be calculated. For information on how distances and clusters are calculated, see `Normalization and clustering`.

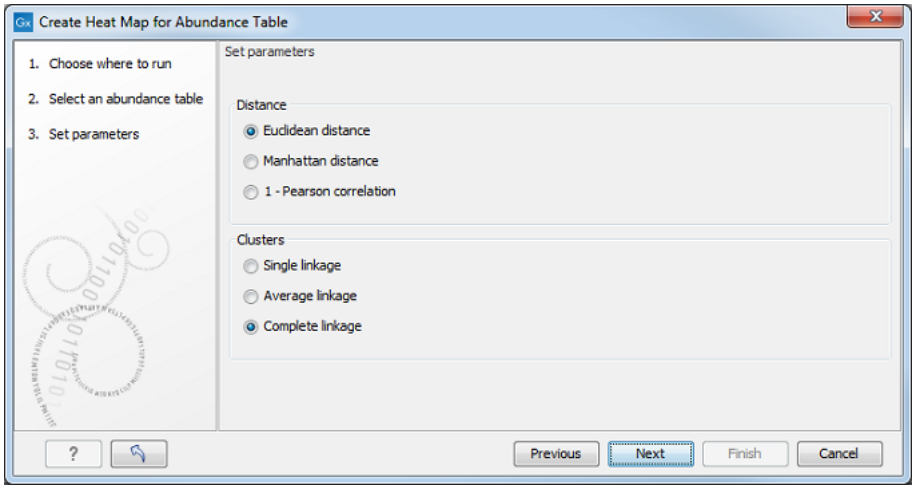


Figure 7.18: *Select an abundance table.*

After having selected the distance measure, set up the feature filtering options (figure 7.19).

Genomes usually contain too many features to allow for a meaningful visualization. Clustering hundreds of thousands of features is also very time consuming. We therefore recommend to reduce the number of features before clustering, using the filter options available:

- No filtering: Keeps all features.
- Fixed number of features:
  - *Fixed number of features*: The given number of features with the highest coefficient of variation (the ratio of the standard deviation to the mean) are kept.
  - *Minimum counts in at least one sample*: Only features with more than this number of counts in at least one sample will be taken into account. Notice that the counts are raw, un-normalized values.

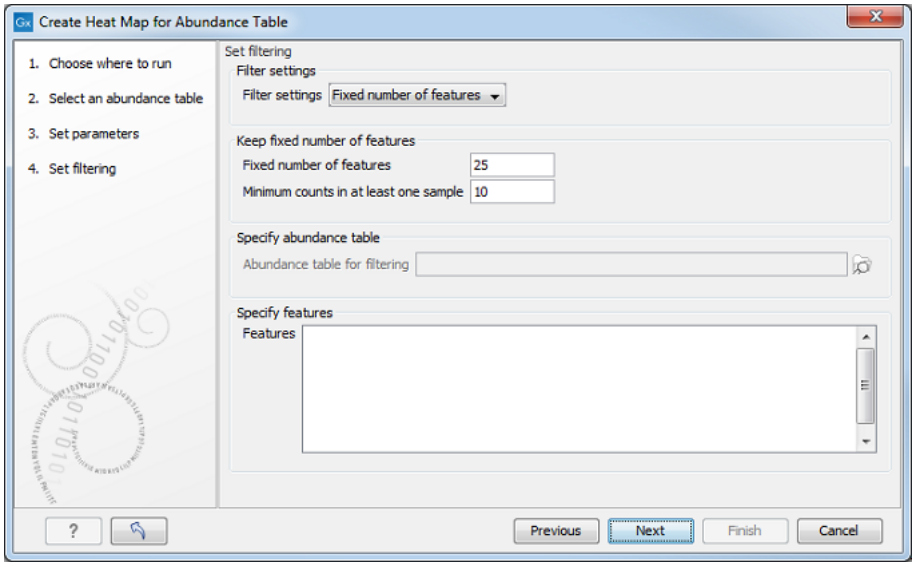


Figure 7.19: *Set filtering options.*

- Abundance table: Specify a subset of an abundance table in case you only want to display the heat map for that particular subset. Note that creating the heat map from the subset abundance table directly can not ensure proper normalization of the data, and it is therefore recommended to use the original abundance table as input and filter using this option.
- Specify features: Keeps a set of features, as specified by plain text, i.e., a list of feature names. Any white-space characters, as well as "," and ";" are accepted as separators.

7.8.2 The heat map view

The tool generates a heat map showing the abundance of each feature in each sample and showing the sample clustering and/or feature clustering as a binary tree over the samples and features, respectively (figure 7.20).

Each column corresponds to one sample, and each row corresponds to a taxon. Samples and features are hierarchically clustered. Available sample metadata is added as an overlay.

The abundance value of each cell in the heat map is available from the table view ( ).

7.8.3 Create heat map for specific taxonomic level

To create a heat map with a particular taxonomic level, you first need to create an aggregated version of your abundance table:

- Open the abundance table.
- In the **Data** section of the **Side panel**, select the desired **Aggregate feature** value, for example *Genus*.
- Select the desired features from the aggregated table, or use Ctrl+A (⌘ +A on Mac) to select all.

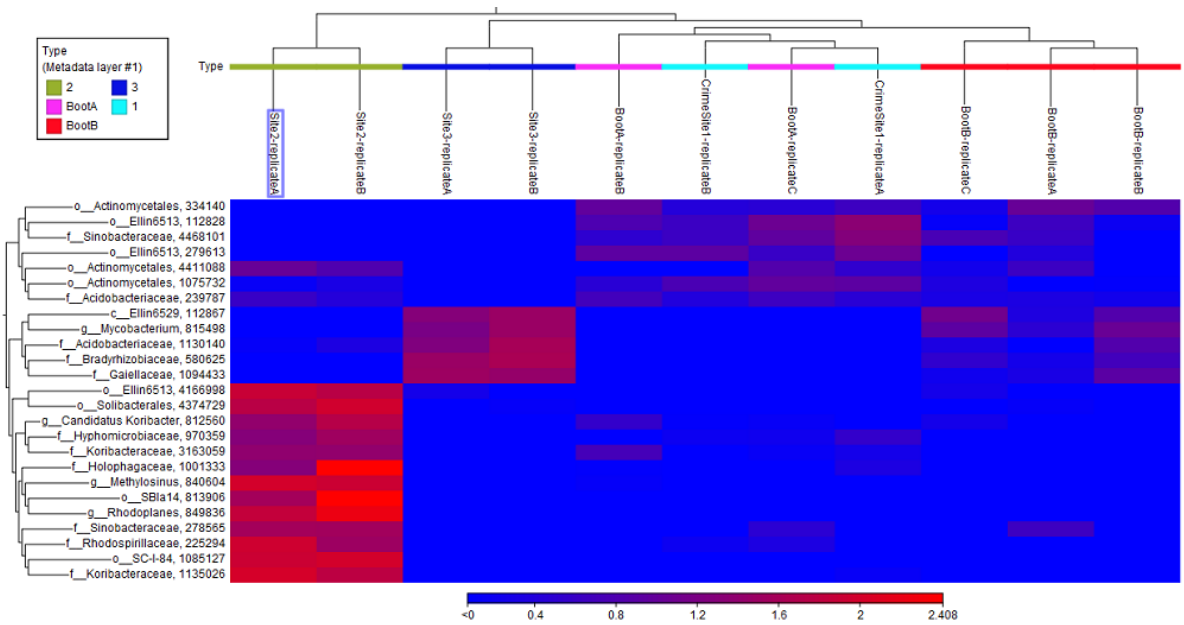


Figure 7.20: *Heat map.*

- Click on **Create Abundance Subtable** at the bottom of the viewing area. This will create an abundance subtable from the selection.
- Save the subtable and use it as input for **Create Heat Map for Abundance Table**.

The resulting heat map will be grouped based on the selected taxonomic level.

7.9 Add Metadata to Abundance Table

It is useful to have abundance tables decorated with sample metadata. This can be done by importing metadata and associating it with the reads before generating an abundance table. To learn more about how to create a metadata table, how to import a metadata table, or how to associate data elements with metadata, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Metadata.html.

If you did not have metadata associated with your reads prior to creating an abundance table, use **Add Metadata to Abundance Table** to add it retrospectively.

To run the **Add Metadata to Abundance Table** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Metagenomics** ( ) | **Abundance Analysis** ( ) | **Add Metadata to Abundance Table** ( )

Choose an abundance table as input. In the next wizard window you can select a file describing the metadata on your local computer (figure 7.21).

The metadata should be formatted in a tabular file format (*.xls, *.xlsx). The first row of the table should contain column headers. There should be one column called "Name" and the entries in this column should match the names of the reads selected for creating the abundance table. This column is used to match row in the table with samples present in the abundance table, so if

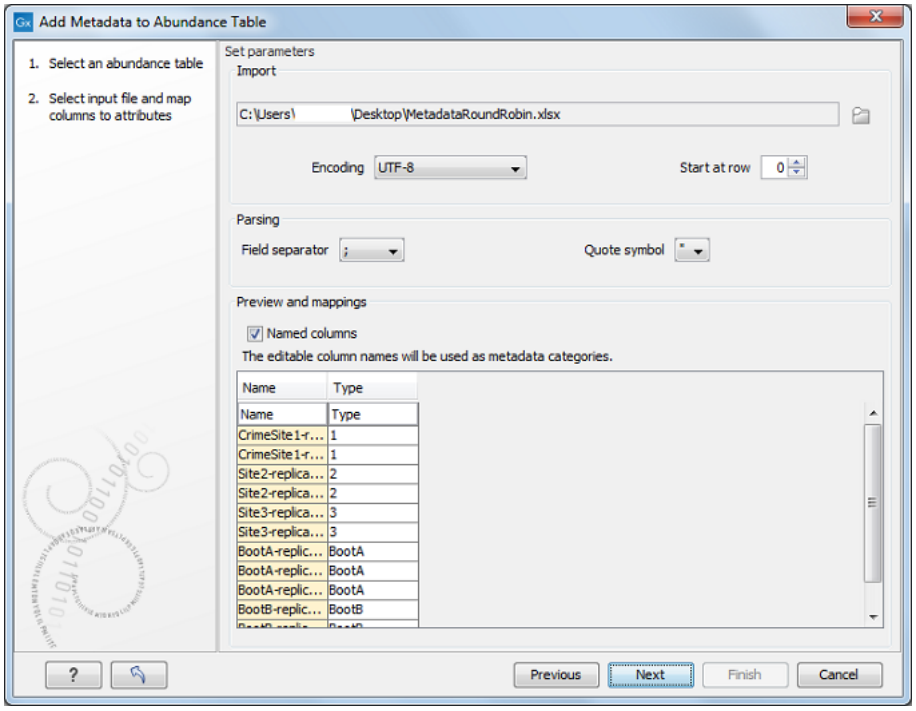


Figure 7.21: *Setting up metadata parameters.*

the names do not match you will not be able to aggregate your data at all. There can be as many other columns as needed, and these information can be used as grouping variables to improve visualization of the results or to perform additional statistical analyses. If you wish to ignore a column without deleting it from your file, simply delete the text in the header row.

Note that when importing an Excel file, formulas will be imported as the formula text and not as the result of the calculation. If you utilize formulas in the metadata file you want to import, you have to flatten the file before importing. This can be done in a number of ways, for instance by exporting to a CSV file (and then importing that instead), or copying and using "Paste Special" in Excel: Start by selecting everything, copy the selection to the clipboard and then execute "Paste Special". On Windows "Paste Special" can be executed by holding Ctrl and Alt and then pressing V. On a Mac "Paste Special" can be executed by holding Ctrl and ⌘ and pressing V. Once the "Paste Special" dialog appears, select "Values" under "Paste" and finally click OK.

Part V

Typing and Epidemiology

Chapter 8

Find the best matching reference

8.1 Find Best Matches using K-mer Spectra

The Find Best Matches using K-mer Spectra tool is inspired by Hasman et al., 2013 and Larsen et al., 2014 and enables identification of the best matching reference among a specified reference sequence list.

Template workflows for typing and epidemiology analysis are available at:

> **Workflows** | **Template Workflows** ( ) | **Microbial Workflows** ( ) | **Typing and Epidemiology** ( )

For more information, see section 2.3.

To identify best matching bacterial genome reference, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Typing and Epidemiology** ( ) | **Find Best Matches using K-mer Spectra** ( )

Select the sequences you want want to find a best match sequence for (figure 8.1).

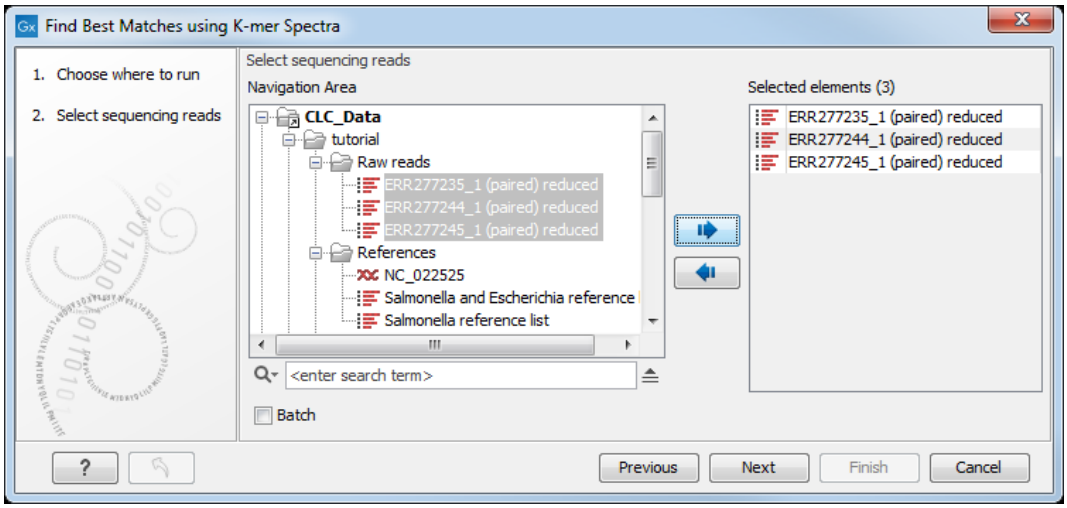


Figure 8.1: *To identify best matching reference, specification of read file is the first step.*

Select then a reference database, and specify the following settings (figure 8.2).

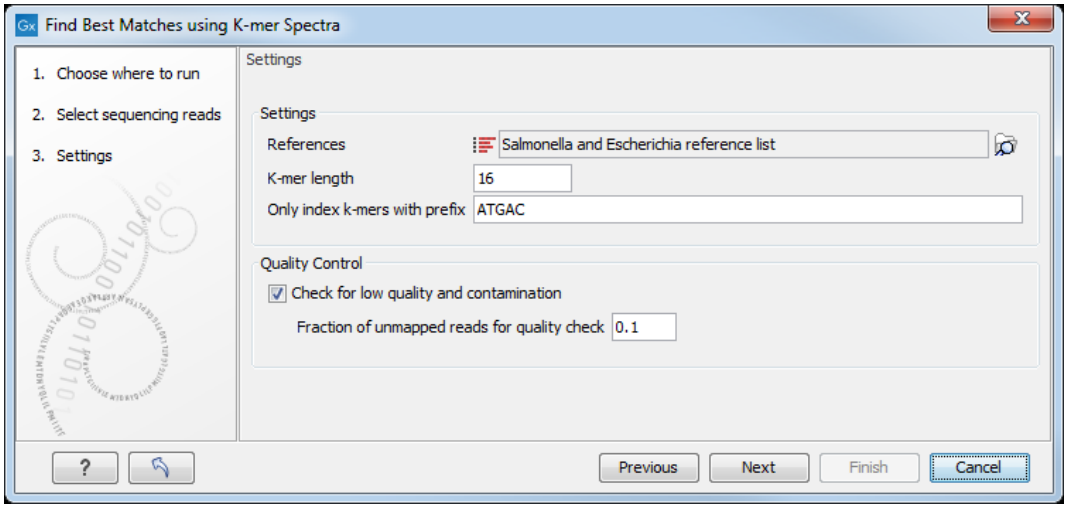


Figure 8.2: *Specify reference list to search across.*

- **References** may be a single- or multiple list(s) of sequences. Sequences with identical entries in the Assembly ID and Latin Name columns are considered as one reference, see section 22. It is for example possible to use the full NCBI's bacterial genomes database, or subset(s) of it.
- **K-mer length** is the fixed number (k) of DNA bases to search across.
- **Only index k-mers with prefix** allows specification of the initial bases of the k-mer sequence to limit the search space.
- **Check for low quality and contamination** will perform a quality check of the input data and identify potential contaminations.
- **Fraction of unmapped reads for quality check** defines the contamination tolerance as the fraction of the total number of reads not mapping to the best reference.

In the last wizard window, the tool provides the following output options (figure 8.3).

- **Output Best Matching Sequence** is the best matching genome within the provided reference sequence list(s).
- **Output Best Matching Sequences as a List** includes the best matching genomes ordered with the best matching reference sequence first. The list is capped at 100 entries. Content is the same as in the Output Report Table.
- **Output Report Table** represents the best matching sequence. It lists all significantly matching references including various statistical values (as described in Hasman et al., 2013 and Larsen et al., 2014). The list is capped at 100 entries and the column headers are defined as such:
  - **Score** Numbers of k-mers from the database seen in the reads.
  - **Expected** The expected value, i.e., what score should been for the Z-score to be 0 and thus the P-value to be 1.
  - **Z** Calculated Z-score.

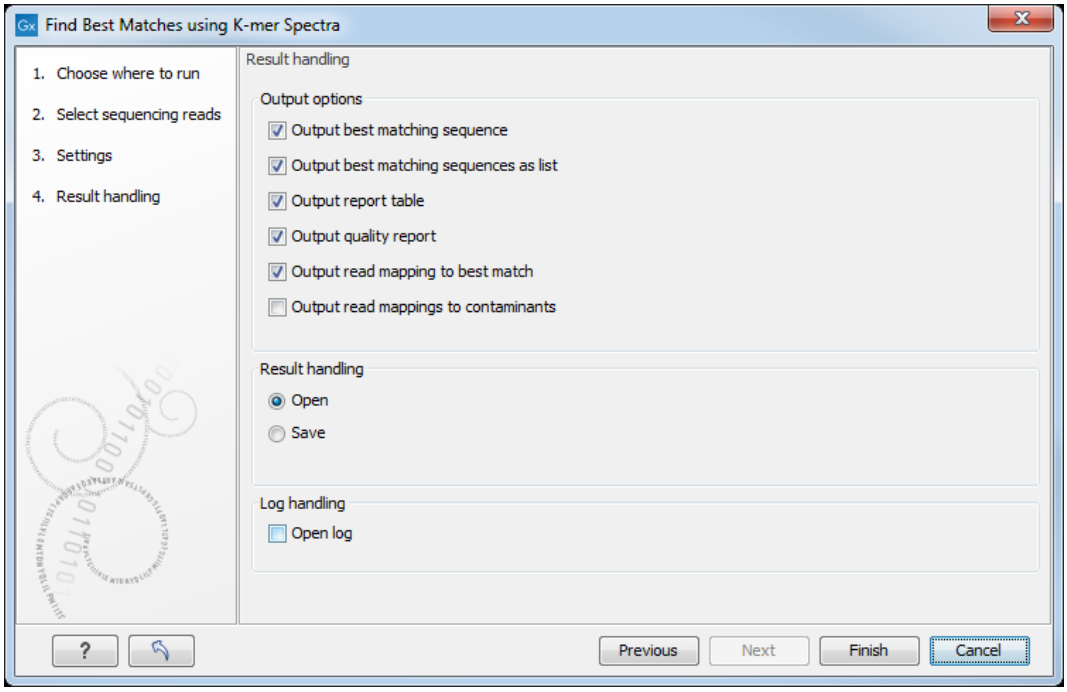


Figure 8.3: *Choose your output option before saving your results.*

  - **P** Z-score translated to two-sided P-value.
  - **P, corrected** P-value with Bonferroni correction.

- **Output Quality Report** gives a report with some statistics on possible contamination and coverage reports for the read mappings. This option is available if the option **Check for low quality and contamination** was selected in the first wizard window. This report contains the metadata:
  - **Best match, % mapped** Percent of reads mapping to the best matching reference.
  - **Contaminating species, % mapped (taxonomy info)** Percent of mapping reads and the most specific accessible taxonomy information for the most probable contaminant.
- **Output read mapping to best match** gives the mapping of the reads to the best matching reference. This option is available if the option **Check for low quality and contamination** was selected in the first wizard window.
- **Output read mapping to contaminants** if a contamination is detected, this generates the mapping of the reads (which do not map to the best reference) to the probable contaminants. This option is available if the option **Check for low quality and contamination** was selected in the first wizard window.

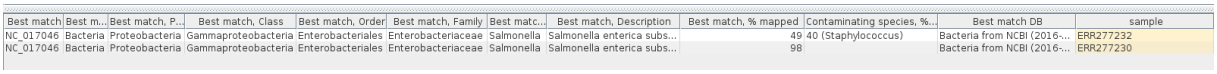
In cases where the tool stops with a warning that good references were not found, you should download a new set of references for the organisms of interest and re-run the workflow.

To add the obtained best match to a Result Metadata Table, see section 20.2.3.

Note that in rare instances, the lists of references found in the **Output Best Matching Sequences as a List** and **Output Quality Report** may differ. The reason is that the former list is compiled based on a "Winner takes all" based count of K-mers which attributes all uniquely found K-mers

only to the reference with the highest Z-score,. The latter list however is produced by removing all reads mapping to the best matching reference and using the remaining reads as a basis for determining the next best match. Thus, in the second round the pool of K-mers has been altered, and some K-mers that determined the Z-score of the original second-best match may have been removed.

Once results from the Find Best Matches using K-mer Spectra tool are added to the Result Metadata Table, extra columns are present in the table, including the taxonomy of the best matching references. In addition, in case the quality control was activated, the table will include the percentage of reads mapping to the best reference and the most probable contaminating species (see figure 8.4).



Best match	Best m...	Best match, P...	Best match, Class	Best match, Order	Best match, Family	Best matc...	Best match, Description	Best match, % mapped	Contaminating species, %...	Best match DB	sample
NC_017046	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacteriales	Enterobacteriaceae	Salmonella	Salmonella enterica subs...	49	40 (Staphylococcus)	Bacteria from NCBI (2016-...	ERR277232
NC_017046	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacteriales	Enterobacteriaceae	Salmonella	Salmonella enterica subs...	98		Bacteria from NCBI (2016-...	ERR277230

Figure 8.4: *Taxonomy of the best matching reference and quality information is shown in the Metadata Result Table.*

8.1.1 From samples best matches to a common reference for all

If several best matches are found across the samples, you probably want to find a common reference sequence to all (or a subset of) the samples. This can be done directly from your Metadata Result Table, by selecting the samples of interest and creating a K-mer Tree based on these samples (see figure 8.5).

1. **Select** in your Metadata result Table the samples to which a common best matching reference should be identified.
2. **Click** on the **Find Associated Data** ( ) button to find their associated Metadata Elements.
3. **Click** on the **Quick Filtering** ( ) button and select the option **Filter for K-mer Tree** to find Metadata Elements with the Role = Trimmed Reads.
4. **Select** the relevant Metadata Element files.
5. **Click** on the **With selected** ( ) button.
6. **Select** the **Create K-mer Tree** action.

Once you have selected the **Create K-mer Tree** action, you can follow the wizard as described in section 9.2. This section will also explain how to understand the tree and continue with subsequent analyses. In short, the common reference is chosen as the genome sharing the closest common ancestor with the clade of isolates under study in the k-mer tree.

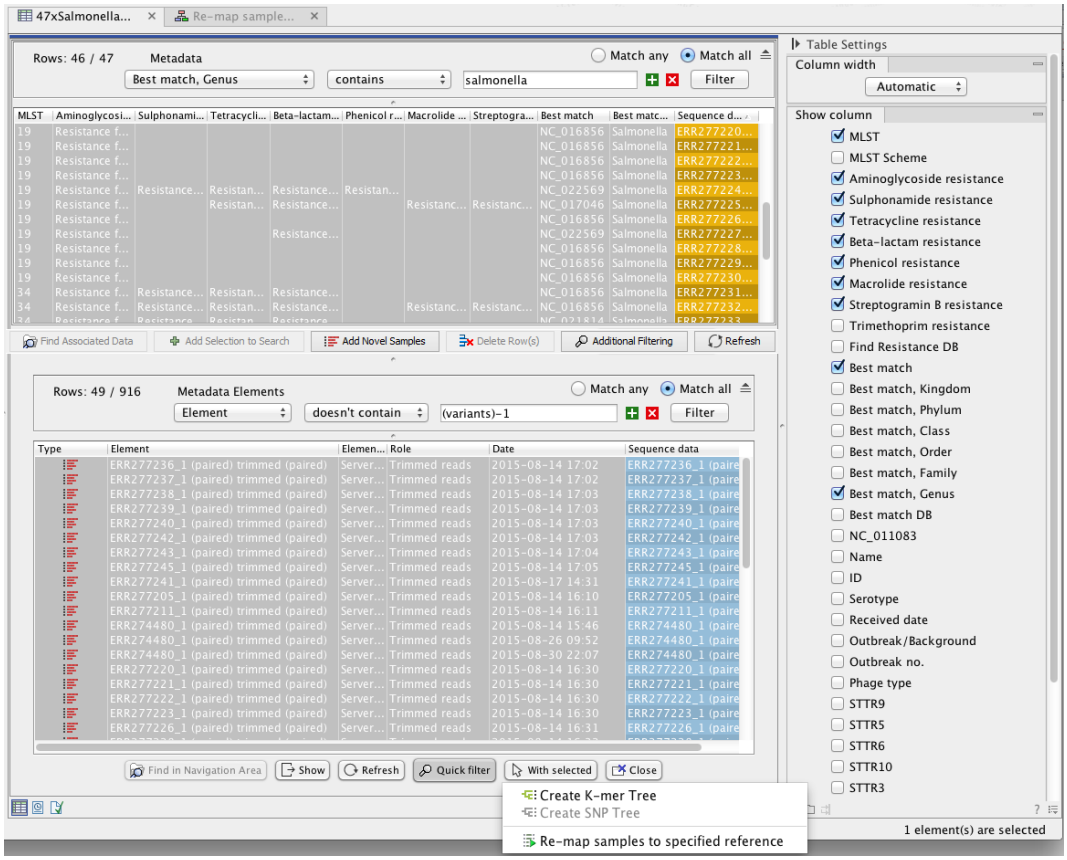


Figure 8.5: *Once samples are selected in the top window, it is easy to find the associated Metadata Element files, Quick Filter towards generation of K-mer tree and finally initiate creation of K-mer tree to be used for identification of common reference sequence. From the same view, it is also possible to run a customized version of the Map to Specified Reference workflow with the selected elements.*

8.2 Find Best References using Read Mapping

The Find Best References using Read Mapping tool maps reads to a reference sequence list to identify the best matching reference i.e., the references for which the input reads hold more evidence.

If a host genome is provided, reads that map better to the host are filtered to not have them count toward results.

To start the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Typing and Epidemiology** () | **Find Best References using Read Mapping** ()

In the first dialog, select the sequences or sequence lists containing the sequencing reads, and click on **Next**.

In the **References** dialog, specify the following (figure 8.6):

- **Treat each sequence as a reference**. Each sequence makes up a separate reference.
- **Treat each assembly ID as a reference**. Sequences with the same assembly ID make up

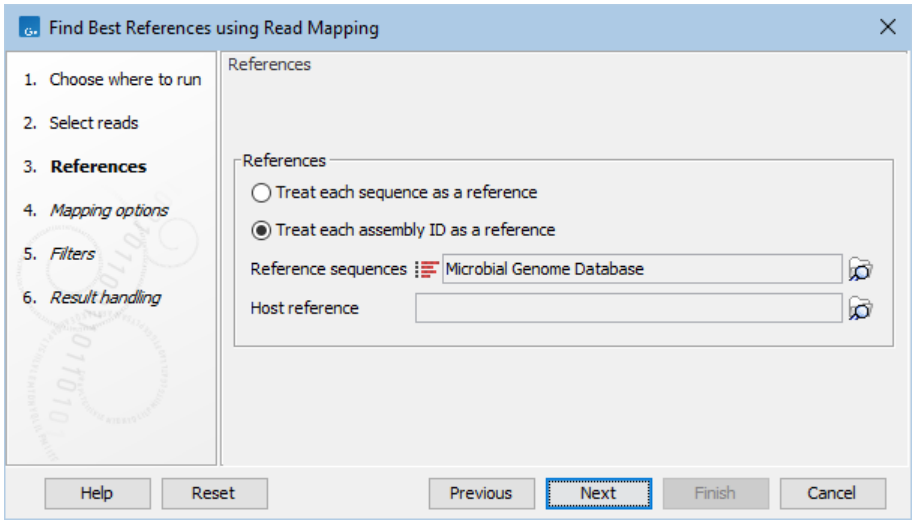


Figure 8.6: *Select references.*

one reference and will be reported as such. This supports segmented references.

- **Reference sequence**. Select the reference sequence list.
  The tool is able to handle duplicate references. If same-name references have identical sequences, only one of these will be included in analysis. If same-name references have different sequences, they will be renamed to ensure unique names.
- **Host reference**. If relevant, provide a host reference to filter reads that map better to the host genome than to the reference sequences.

In the **Mapping options** dialog, specify settings for the read mapping (figure 8.7). The options are identical to those of the Map Reads to Reference tool and are described here: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html.

The **Filters** dialog holds the following options (figure 8.8):

- **Minimum count**. Minimum number of mapped reads required for a reference to be reported.
- **Minimum relative abundance**. Minimum relative abundance compared to most abundant reference required for a reference to be reported.
- **Minimum fraction of reference covered**. Minimum fraction of the reference sequence to be covered by at least one read for a reference to be reported.
- **Minimum average coverage**. Minimum average coverage for a reference to be reported. Average coverage: Number of nucleotides mapped to a reference divided by the reference length.
- **Maximum number of references to report**. The maximum number of references to report. References are ranked according to the number of mapped reads.

In the final step, specify the output:

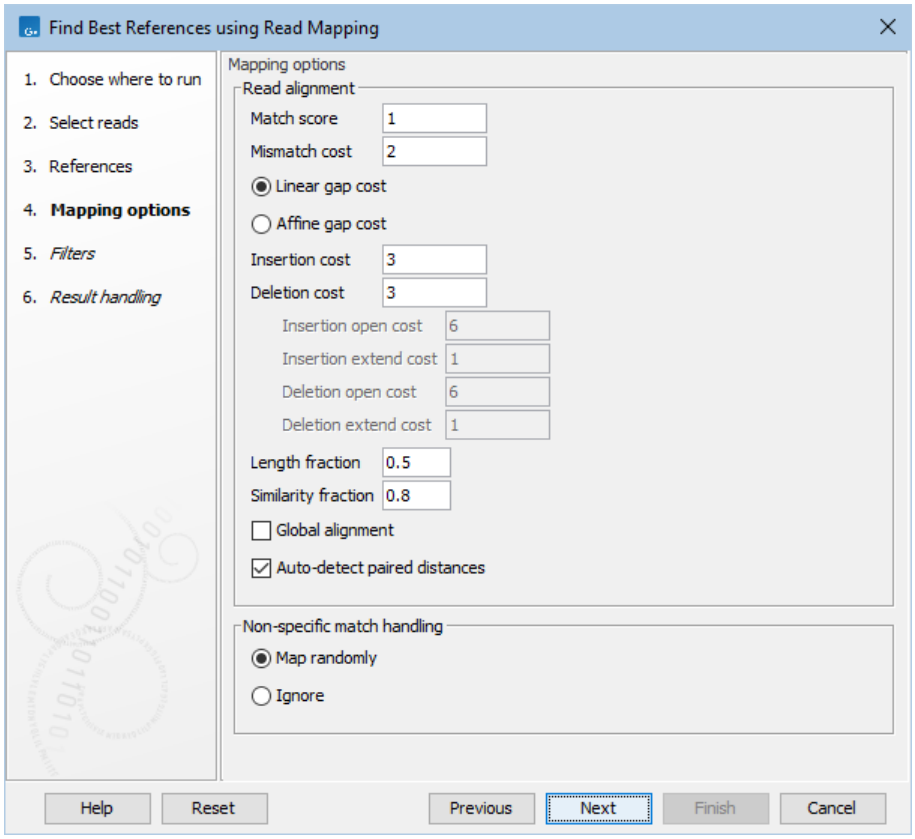


Figure 8.7: *Select mapping options.*

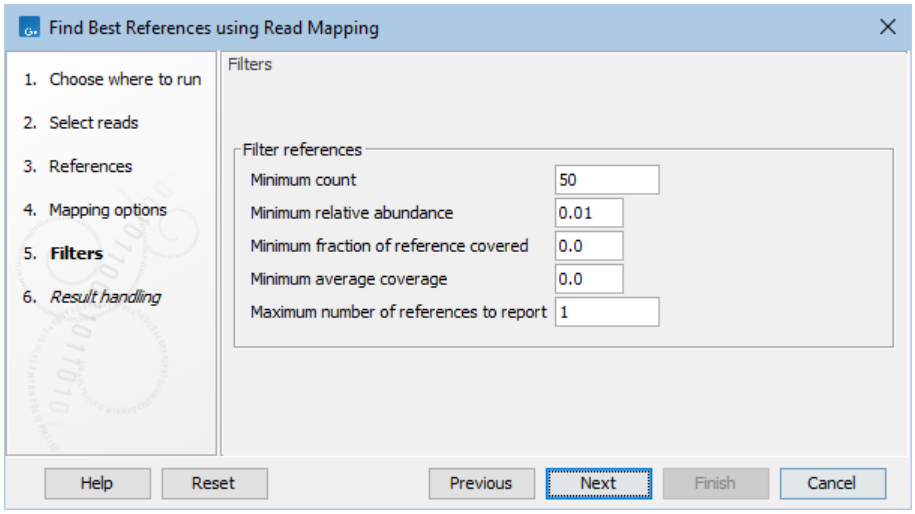


Figure 8.8: *Select filtering options.*

- **Create reference sequence list**. A sequence list with the identified best-match reference sequences.
- **Create reads track**. A track of reads mapped to the reference sequence(s).
- **Create reads track (host)**. A track of reads mapped to the host reference.

- **Create report**. A summary report (section 8.2.1).

8.2.1 The Find Best References using Read Mapping Report

The Find Best References using Read Mapping report contains a summary of the read mapping results and a table of the identified best-match references (figure 8.9).

1 Find Best References using Read Mapping summary

Input reads	269,412
Reads mapped to references	267,008
Reads mapped to host	N/A
Unmapped reads	2,404
Reads mapped to references (%)	99.11
Reads mapped to host (%)	N/A
Unmapped reads (%)	0.89

2 References

Name	Reference	Assembly ID	Reads mapped	Unambiguously mapped reads	Fraction of reference covered	Average Coverage	Taxonomy	Description
Human respiratory syncytial virus A (KY654518)	KY654518	Not available	267,008	266,972	1.00	2,446.62	Orthornavirae; Negarnaviricota; Monjiviricetes; Mononegavirales; Pneumoviridae; Orthopneumovirus; Human orthopneumovirus	Human respiratory syncytial virus A isolate TBp-13-044, complete genome.

Figure 8.9: *The report for Find Best References using Read Mapping.*

8.3 Type with Consensus Refinement

The Type with Consensus Refinement tool is inspired by the Iterative Refinement Meta-Assembler (IRMA) tool [Shepard et al., 2016] developed by the Center for Disease Control and Prevention (CDC) in the USA. The primary use for this tool is to assist in the typing of small, segmented viral genomes such as Influenza viruses. The tool works by iteratively mapping the reads to a set of annotated reference sequences, evaluating the quality of the mappings, and extracting consensus sequences. This process is repeated using the consensus sequences as new references until the mapping metrics stabilize, indicating that further iterations will not improve the results.

After finishing the iterative refinement process, the tool transfers all annotations from the references to the consensus sequences. The CDS annotations are translated into protein sequences. These sequences are then checked for unexpected codons at the start and stop positions as well as unexpected internal stop codons that may indicate frameshifts in the consensus.

8.3.1 Type with Consensus Refinement parameters

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Typing and Epidemiology** () | **Type with Consensus Refinement** ()

In the first dialog, select the reads for analysis, and click on **Next**.

Click **Next** to select a reference database. The reference database is a sequence list in which each sequence must be annotated with a **Segment** attribute. The tool uses the **Segment** information to define which sequences will be considered together when evaluating which of them suits better the input data. The sequences in the reference may also be annotated with a **Type** attribute (e.g. A and B) and a **Subtype** attribute (e.g. H1 and H5). Sequence lists can be updated manually in the Table View or using a table file with `Update Sequence Attributes in Lists`.

A ready-to-use reference data set for typing Influenza A, B, C and D is available through the `Reference Data Manager`.

In the **Mapping options** dialog, specify settings for the read mapping (figure 8.10). The options are identical to those of the `Map Reads to Reference` tool, except for the "Non-specific match handling".

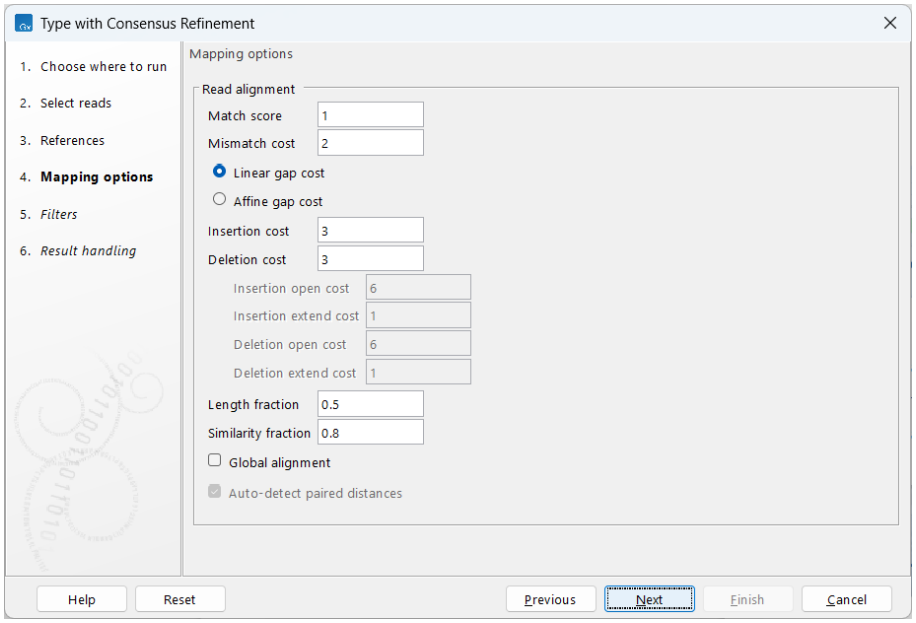


Figure 8.10: *Select mapping options.*

The **Filters** dialog holds the following options (figure 8.11):

- **Minimum count**. Minimum number of mapped reads required for a reference to be reported.
- **Minimum relative abundance**. Minimum relative abundance compared to most abundant reference required for a reference to be reported.
- **Minimum fraction of reference covered**. Minimum fraction of the reference sequence to be covered by at least one read for a reference to be reported.
- **Minimum average coverage**. Minimum average coverage for a reference to be reported, whereas average coverage is the number of nucleotides mapped to a reference divided by the reference length.
- **Maximum number of references to report**. The maximum number of references to report. References are ranked according to the number of mapped reads.

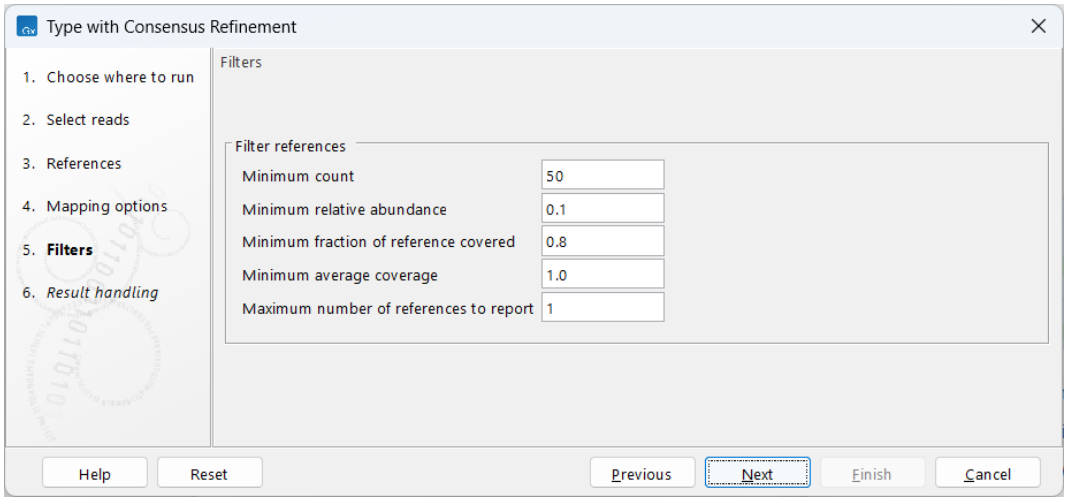


Figure 8.11: *Choose your output option before saving your results.*

Note that all filters apply per sequence, except for the **Minimum relative abundance** and **Maximum number of references to report** filters which consider all sequences with the same **Segment** and **Type** (if present). In the Minimum relative abundance case, if four references for Influenza A segment 3 are available for example, the filter considers the relative abundance among these four sequences and ignores the abundance in the rest of the references. In the Maximum number of references to report case, if the value is set to three, for a virus with eight segments such as Influenza, up to 24 segments may be reported (up to three per segment).

8.3.2 Type with Consensus Refinement outputs

In the final step, specify the output (figure 8.12). In addition to the Consensus Sequences list, the following options are available:

- **Create reference sequence list**. A sequence list with the best-matching reference sequences as identified at the end of consensus sequence building.
- **Create reads track**. A track of the input reads mapped to the consensus sequences.
- **Create report**. A summary report (Type with Consensus Refinement Report).

8.3.3 Type with Consensus Refinement report

The Find Best References using Read Mapping report (figure 8.13) contains the following sections:

- **Typing result**. Short description of the type and subtype for the sample, if these attributes were present on the reference used.
- **Summary**. Read mapping statistics for the whole sample.
- **Consensus**. A table information per segment, including segment, type and subtype, original reference, and mapping quality statistics. If no good match was found for a certain segment, it will still appear in the report, but statistics will be empty or zero.

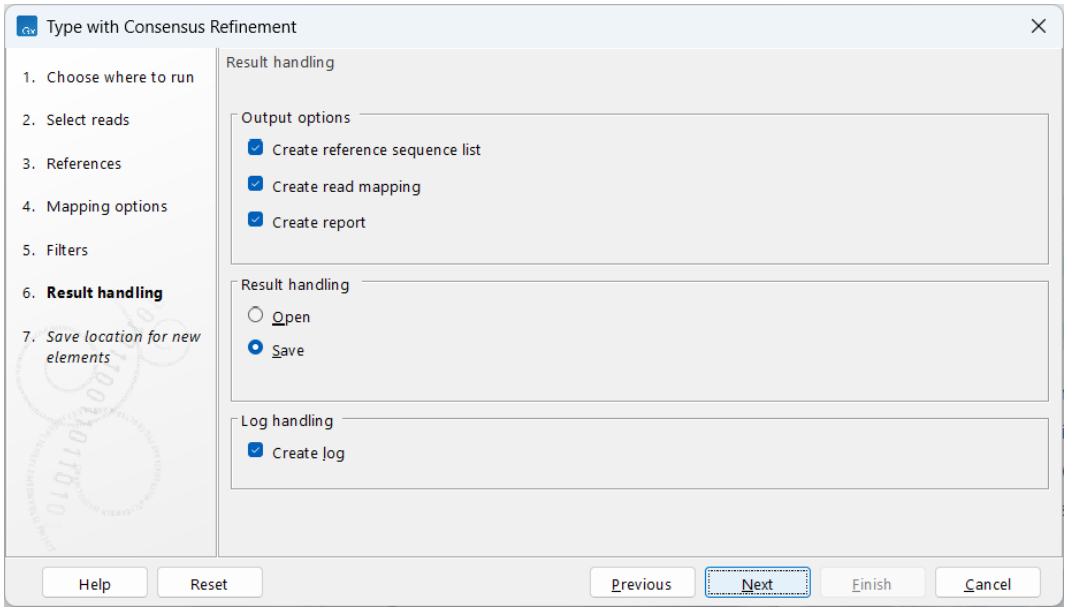


Figure 8.12: *Choose your output options before saving your results.*

- **Annotations**. A table with information about the CDS annotations indicating potential problems, if CDS annotations were present on the reference used.

1 Typing results

Type	A
Subtype	H3 / N2

2 Summary

Input reads	129,416
Reads mapped	128,902
Unmapped reads	514
Reads mapped (%)	99.60
Unmapped reads (%)	0.40

3 Consensus

Type	Segment	Subtype	Reference	Reads mapped	Unambiguously mapped reads	Fraction of consensus covered	Average coverage
A	1		CY002079	19,924	19,924	100.00	118,102.73
A	2		CY003646	12,588	12,588	100.00	74,693.12
A	3		CY003645	15,552	15,550	100.00	96,325.44
A	4	H3	CY002000	23,029	23,029	100.00	183,087.12
A	5		CY006079	13,474	13,474	100.00	121,505.75
A	6	N2	CY002010	21,424	21,422	99.93	204,577.37
A	7		CY002009	16,625	16,625	100.00	225,370.40
A	8		CY002284	6,286	6,286	100.00	98,311.91

4 Annotations

Annotation	Origin	Status
PB2	CY002079	-
PB1	CY003646	Unexpected stop codon: Q
PB1-F2	CY003646	Premature stop codon at position: 87
PA	CY003645	-
HA	CY002000	-
NP	CY006079	-
NA	CY002010	-
M2	CY002009	-
M1	CY002009	-
NS2	CY002284	-
NS1	CY002284	-

Figure 8.13: *The report for Type with Consensus Refinement.*

Chapter 9

Phylogenetic trees using SNPs or k-mers

9.1 Create SNP Tree

The **Create SNP Tree** tool is inspired by Kaas et al., 2014.

To generate a SNP tree, first map reads from the individual samples to a common reference and call variants. The corresponding tools are described at:

- Map Reads To Reference: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html
- Variant detection: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Variant_detection.html

To create a SNP tree, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Typing and Epidemiology** ( ) | **Create SNP Tree** ( )

In the first dialog, select reads tracks or read mappings (figure 9.1).

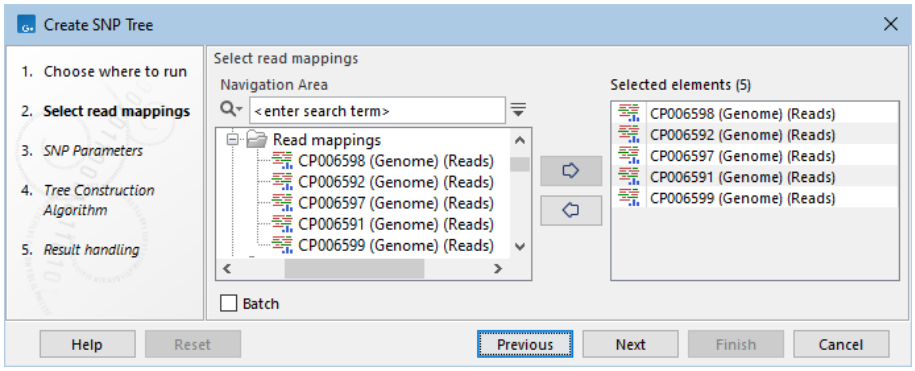


Figure 9.1: *Select read mappings to be included in the SNP tree analysis.*

Next, select **Variant parameters**. These determine which SNPs (single-nucleotide polymorphisms) and MNVs (multi-nucleotide variants) to consider for building the SNP tree:

- **Variant tracks**. Select variant tracks that correspond to the previously selected reads

tracks or read mappings (figure 9.2). The variant tracks determine which positions to potentially include in the SNP tree.

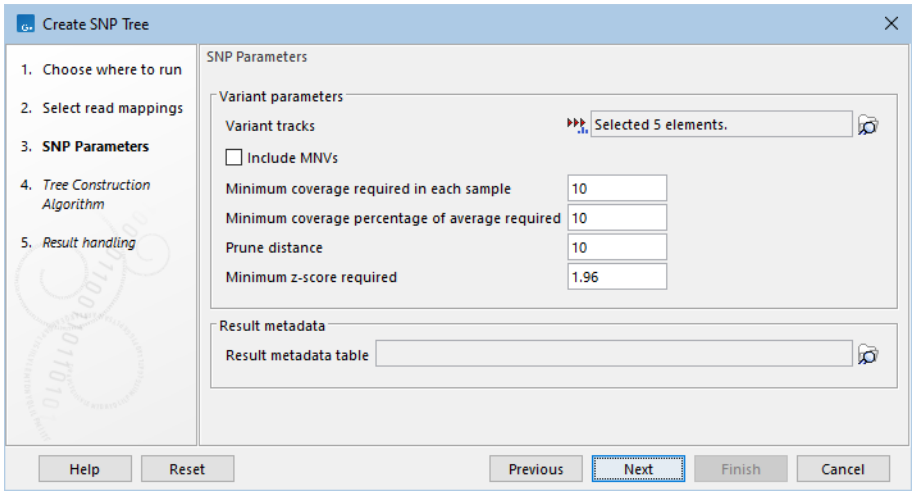


Figure 9.2: *Select variant tracks and specify relevant parameters before generation of a SNP tree.*

- **Include MNVs**. Check this option to include MNVs along with SNPs when building the SNP tree.
- **Minimum coverage required in each sample**. Positions are filtered if at least one sample has coverage below the specified threshold.
- **Minimum coverage percentage of average required**. Positions are filtered if the coverage of at least one sample falls below the specified percentage of the average coverage of that sample.
- **Prune distance**. Minimum number of nucleotides between unfiltered positions. If a position is within this distance of a previously used position it will be filtered.
- **Minimum z-score required**. Defining x as the number of the most prevalent nucleotide at a position and y as the coverage subtracting x, the z-score is calculated as $z = \frac{x-y}{\sqrt{x+y}}$. If the calculated z-score for a given position is less than the specified minimum value the position is filtered.

The initial list of SNP positions is reduced based on the above filters. Of the remaining, only variants with relative frequency above 50% (haploid organisms) will be considered. SNP positions that overlap a deletion in any sample are not considered, because such SNPs are often false positives caused by undetected deletions in repeat regions. Information about reference and alleles is deduced from the read mappings.

Optionally, select the **Result metadata** table with metadata relevant for your samples. This will allow you to decorate the resulting SNP tree with metadata information, see section 9.1.2.

In the next dialog, select the tree construction algorithm (figure 9.3).

- **Neighbor Joining**. Creates a tree with a fast method. In the absence of homoplastic SNPs (a SNP that is acquired independently on different branches of the tree), or positions where

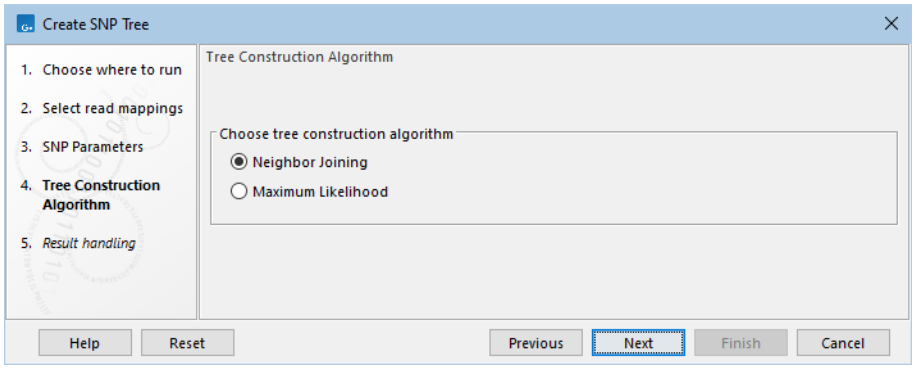


Figure 9.3: *Choose the tree construction algorithm.*

three or more nucleotides are present, then the distances in the tree have the following property: If you move from one sample to another in the tree, the sum of the lengths of the branches traversed equals the number of SNP differences between those samples.

- **Maximum Likelihood**. Calculates the most likely phylogenetic tree under the given evolutionary model. Branch lengths are the number of expected substitutions between samples. For closely related samples these match the number of SNP differences between the samples, as for Neighbor Joining. For more distantly related samples these are expected to exceed the number of SNP differences, and will tend to be systematically too large. This is because distantly related samples are expected to have some positions where multiple substitutions have occurred, and using only SNP positions to build the tree will tend to make samples appear more distantly related than if all positions were used.

If you selected *Maximum Likelihood*, the next dialog covers parameters for this algorithm (see figure 9.4). The parameters are described here: https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Maximum_Likelihood_Phylogeny.html.

In the **Result handling** dialog, specify the output (figure 9.5).

In addition to the SNP tree, the following are available:

- **Create report**. A SNP report with summary of input and results.
- **Create SNP alignment**. Outputs the alignment that is produced as a first step in the algorithm. The alignment consists of columns of concatenated SNPs and columns of constant A, T, C, and G.

  At least 100 constant columns are added to ensure that the equilibrium frequencies of nucleotides more closely resemble those in the reference genome. Additional columns are added until the two most distant sequences in the alignment are 80% identical or until the alignment is the size of the reference genome. This partially mitigates the overestimation of branch lengths when using the Maximum Likelihood tree construction algorithm on SNP positions.

  The alignment can be used as input for the **Model Testing** tool that serves to identify which evolutionary model suits the data best. Based on this, you may want to rerun the **Create SNP Tree** tool with adjusted settings. The **Model Testing** tool is described at https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Model_Testing.html.

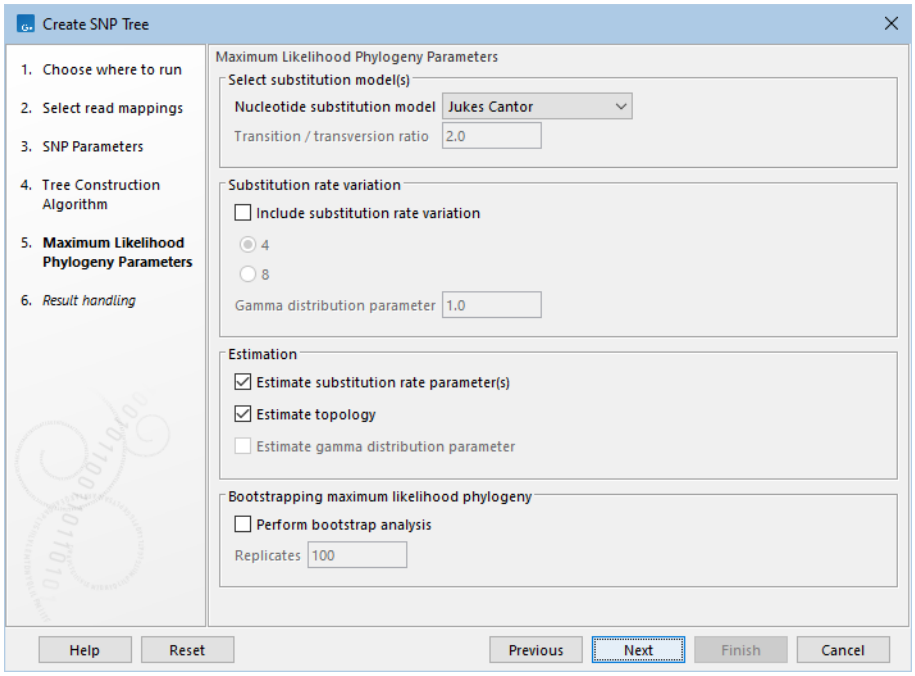


Figure 9.4: *Set parameters for maximum likeihood estimation.*

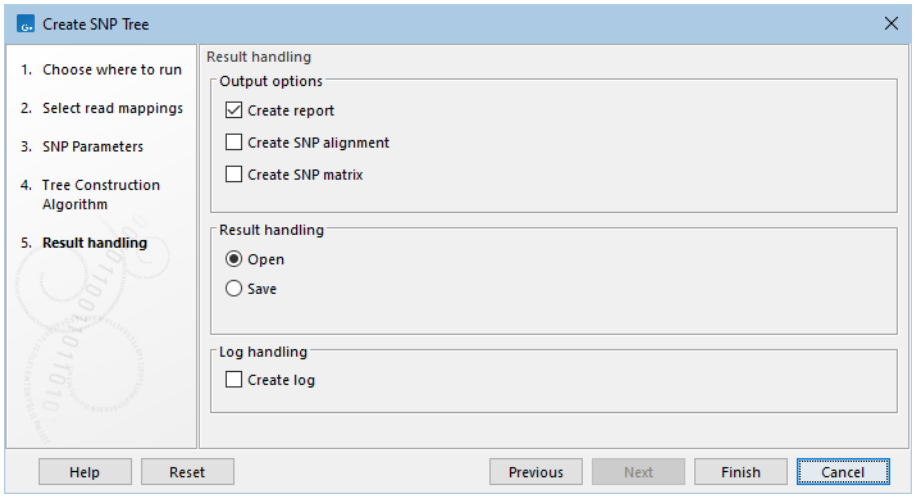


Figure 9.5: *Create SNP Tree output options.*

- **Create SNP matrix**. A matrix containing the number of SNP differences between all pairs of samples.

9.1.1 SNP tree report

The SNP tree report summarizes the result of the applied filtering.

- **Filter Status** (figure 9.6)
  - **Number of different input positions**. Unique SNPs (and MNVs, if selected) in the input variant tracks, pre-filtering.

Description	Count
Number of different input positions	3432
Pruned	1496
Coverage filtered	16
Z-value filtered	1
Deletion filtered	0
Number of input positions used	1919

Figure 9.6: *SNP tree report - Filter Status section.*

  - **Pruned**. Positions filtered due to the prune distance threshold.
  - **Coverage filtered**. Positions filtered based on the two coverage filters.
  - **Z-value filtered**. Positions filtered based on the minimum Z-value threshold.
  - **Deletion filtered**. Positions deleted as at least one sample has a deletion at this position.
  - **Number of input positions used**. Positions that passed all filtering steps and were included in the SNP tree.

- **Ignored positions attributed to read mappings**. Information on the number of positions filtered in the individual read mappings (figure 9.7).

  - **Read mapping**. The name of the read mapping.
  - **Filtered, total**. The number of SNPs from this read mapping that were filtered.
  - **Filtered, only by this**. The number of SNPs that were unique to this read mapping, and were filtered.

    If one or a few samples have a substantially higher number of filtered positions compared to the rest, one might consider rerunning the tree without these to improve the tree resolution.

9.1.2 SNP tree

The SNP tree can be visualized in Tree view (), Table view (), and SNP Tree Variants view ().

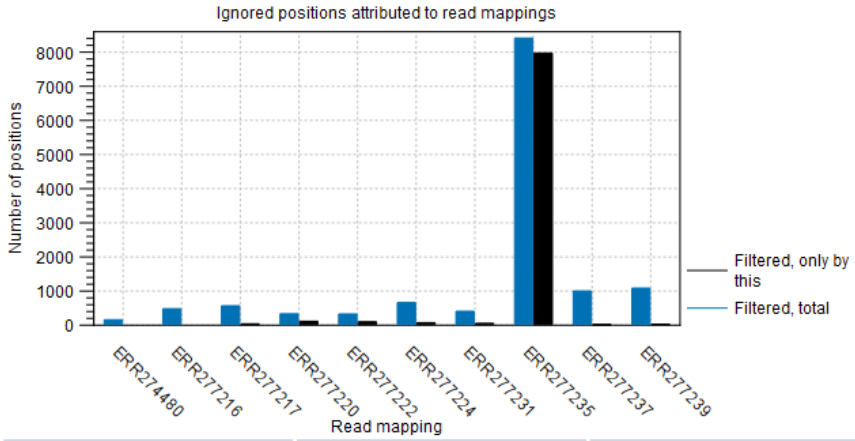
Tree view ()

The SNP tree layout, node and label settings is adjusted from the Tree Settings Side Panel found in the left side of the view area. For details about the settings, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html.

If a Result Metadata Table was provided as input, it is possible to decorate the SNP tree with one or more metadata layers from the Side Panel section **Metadata**, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Visualizing_metadata.html. This enables visual identification of potential correlation between samples, metadata layers, and tree typology (figure 9.8).

SNP Tree Variants view ()

With the SNP Tree Variants view it is possible to inspect the variants relating to a given internal node in the SNP tree. To populate the table with the SNPs of interest, first select the internal node of interest in the Tree view and then go to the **SNP Tree Variants** () view (figure 9.9).



Read mapping	Filtered, total	Filtered, only by this
ERR274480	181	0
ERR277216	504	12
ERR277217	587	58
ERR277220	352	134
ERR277222	351	119
ERR277224	682	96
ERR277231	426	76
ERR277235	8433	7995
ERR277237	1021	29
ERR277239	1105	52

Figure 9.7: *Visualization of the filter effect across data used for generation of SNP tree. One sample shows a higher number of filtered SNPs and could potentially be omitted from a new SNP tree.*

The table lists the following columns:

- **Position**. The position of the SNP on the reference chromosome
- **Chromosome**. The name of the reference chromosome.
- **Subtree or sample specific columns**. The content depends on the Side Panel setting **SNP information**:
  - **As summary**. The table contains one column per subtree with a summary of alleles in the subtree samples e.g., "A (3), G (1)" (figure 9.9 - top image).
  - **By sample**. The table contains one column per sample with sample-specific alleles (figure 9.9 - bottom image).
- **All agree**. Yes/No indicates whether all samples belonging to the internal node have the same variant allele.

9.1.3 SNP Matrix

The SNP Matrix contains the pairwise number of SNP differences between all pairs of samples (see figure 9.10).

Use the Side Panel setting **Comparison gradient** to get an overview of which samples are closely related. Drag the arrows to change the minimum and maximum values of the scale, or click the

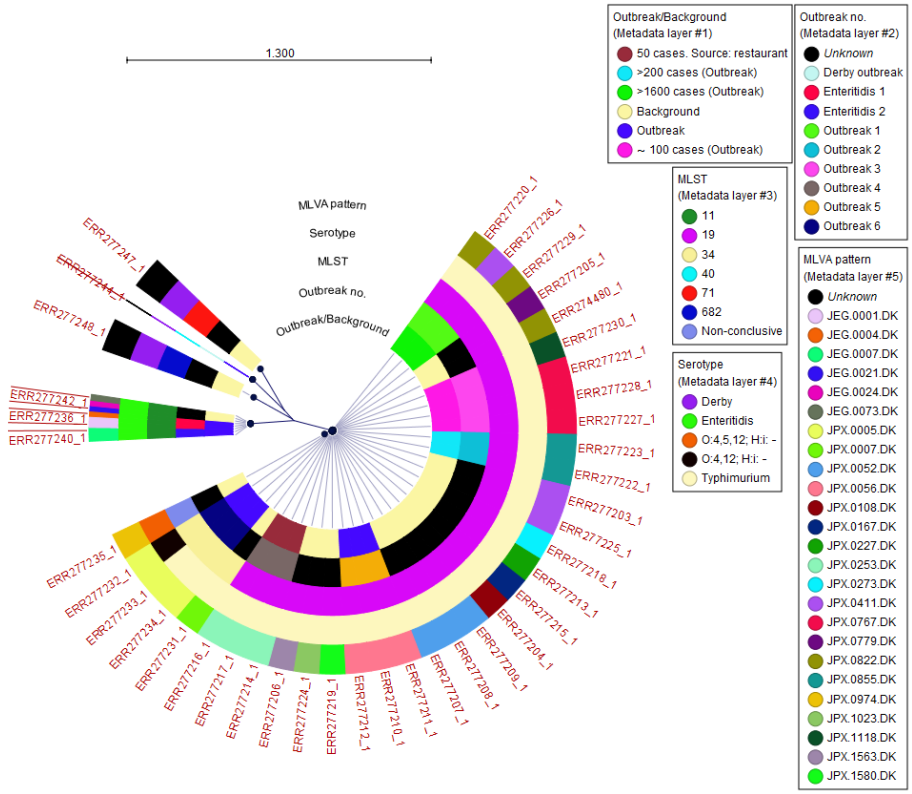


Figure 9.8: *Visualization of SNP tree including selected metadata and analysis result metadata.*

gradient to access the gradient configuration dialog. Use the Lower threshold field to type in a lower threshold value between 0 and the maximum value in the matrix. This results in a distinct coloring of the cells in the matrix which have a value less than the threshold.

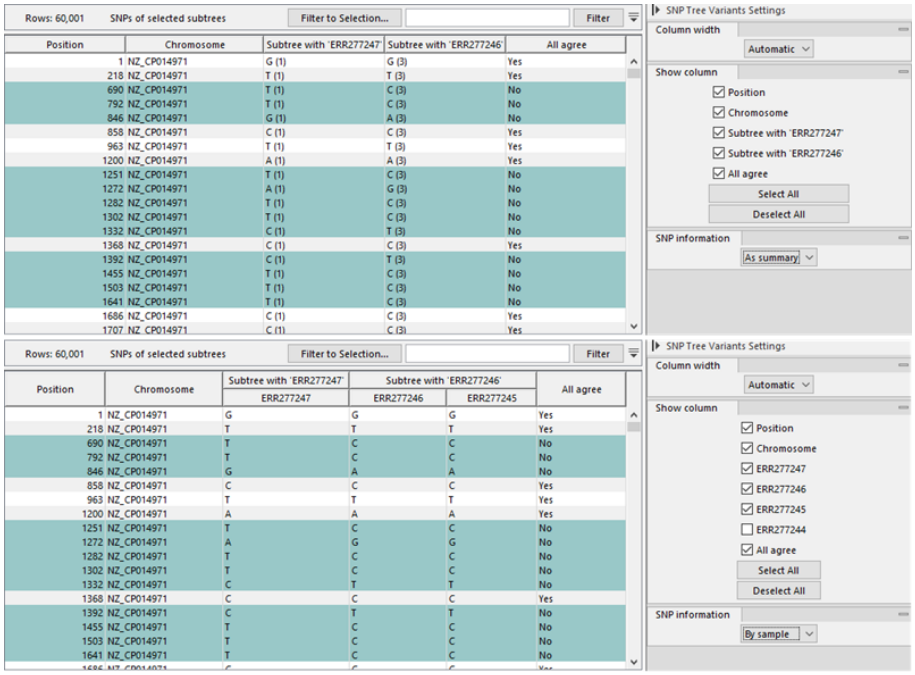


Figure 9.9: *Counts of differences at a given position in the branches of the selected internal node. Top: SNP information "As summary". Bottom: SNP information "By sample".*

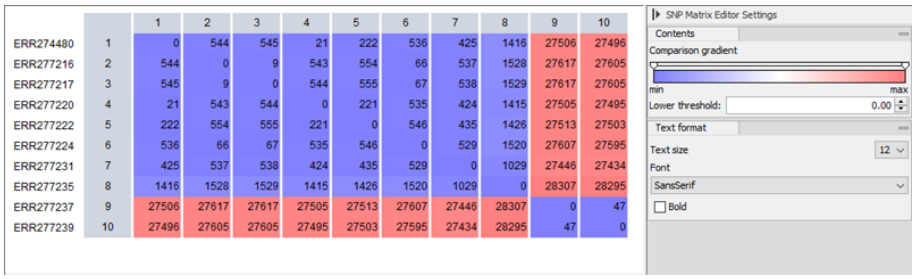


Figure 9.10: *A SNP matrix.*

9.2 Create K-mer Tree

The **Create K-mer Tree** tool may be helpful for identification of the closest common reference across samples. The tool uses reads, single sequences or sequence list as input and creates a distance-based phylogenetic tree. If a sequence list has a read-group it will be treated as a set of reads, otherwise the tool will group the sequences in a sequence list based on their "Assembly ID" annotation or treat the sequences individually when no "Assembly ID" annotation has been assigned. To find out how to assign Assembly ID annotation, please see section 22. There are two ways to initiate creation of a k-mer tree: either from the Result Metadata Table (see chapter 20.2.2), or by running the **Create K-mer Tree** tool from under the Tools menu:

> **Tools** | **Microbial Genomics Module** ( ) | **Typing and Epidemiology** ( ) | **Create K-mer Tree** ( )

Input files can be specified step-by-step like shown in figure 9.11 or by selecting data recursively by right-clicking on the folder name and selecting **Add folder contents (recursively)**. If using the recursive option, remember to double check that files relevant for the downstream analysis are

selected.

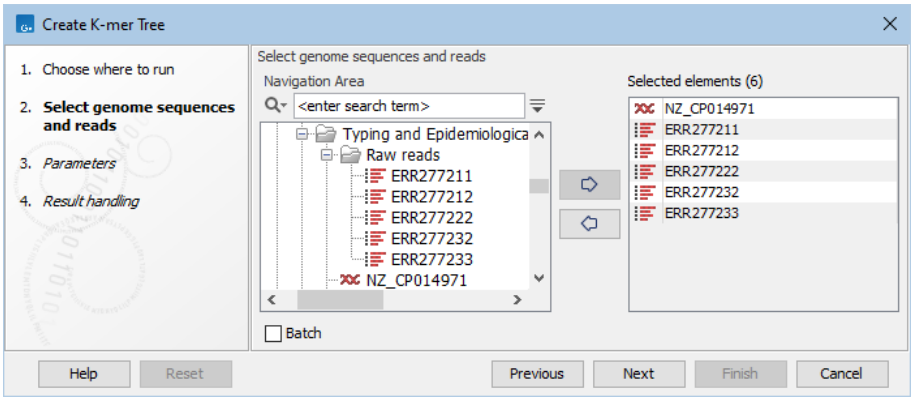


Figure 9.11: *Selection of individual reads and single sequences or sequence list to be included in the K-mer tree analysis.*

Specify the following parameters (figure 9.12):

- **K-mer parameters**
  - **K-mer length** is the fixed number (k) of DNA bases to search across.
  - **Only index k-mers with prefix** allows specification of the initial bases of the k-mer sequence to limit the search space. Reduction of prefix size increases the RAM requirements, and therefore decrease the search speed.
- **Method** may be specified by either of the two statistical methods: **Jaccard Distance** or **Feature Frequency Profile via Jensen-Shannon divergences (FFP)**. You can read more about the Jaccard Distance and FFP at `https://en.wikipedia.org/wiki/Jaccard_index` and `https://en.wikipedia.org/wiki/Alignment-free_sequence_analysis`, respectively.
- **Strand** may be specified as either only the Plus strand or Both strands.
- **Result metadata.** Specify location of the **Result metadata table** file.

The K-mer trees are constructed using a Neighbour Joining method, which makes use of a distance function, either Jaccard Distance or Feature Frequency Profile via Jensen-Shannon divergences (FFP). In both cases, the distance can assume values between 0 (exactly same k-mer distribution) and 1 (completely different k-mer distribution).

Branch lengths depend on the distance function used. Specifically, if one sums up all the branch length of all the branches connecting two leaves, one can get the distance between the two organisms the leaves represent.

9.2.1 Visualization of K-mer Tree for identification of common reference

The k-mer tree below (figure 9.13) includes 46 samples and 44 *Salmonella* genomes. To identify a candidate common reference genome, the tree was visualized using the radial tree topology setting. The common reference is usually chosen as the genome sharing the closest common ancestor with the clade of isolates under study in the k-mer tree. In this case, a reference (acc

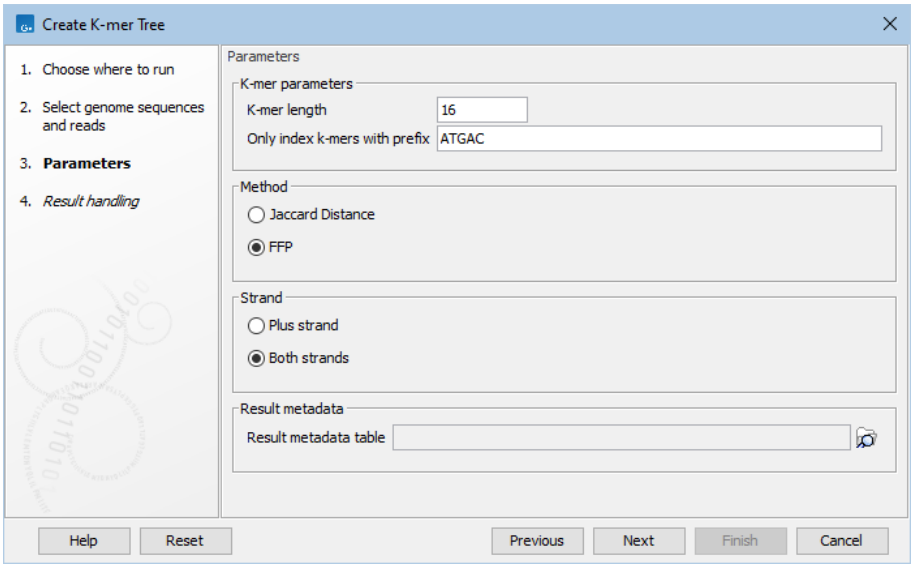


Figure 9.12: *Various parameters may be set before generation of a K-mer tree.*

no NC_011083) located in the centre region of the tree was selected as a common reference candidate.

If the sequence lists (samples and reference genomes) used as input for a k-mer tree contains metadata, the information will be used to decorate the tree.

The scale bar refers to the branch lengths within the tree.

Note that the information in the Taxonomy column of the sequence list needs to be following this format: "Kingdom; Phylum; Class; Order; Family; Genus; Species".

The metadata will also be made available in the K-mer tree table view, where you can manually edit entries in the metadata fields by right clicking on it in the tabular view of the Sequence List. If samples and reference genomes share metadata columns with the same header, these columns will be merged in both the K-mer tree table view and tree view.

Learn more about the overall Tree Settings, including how to decorate trees with metadata, here https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html.

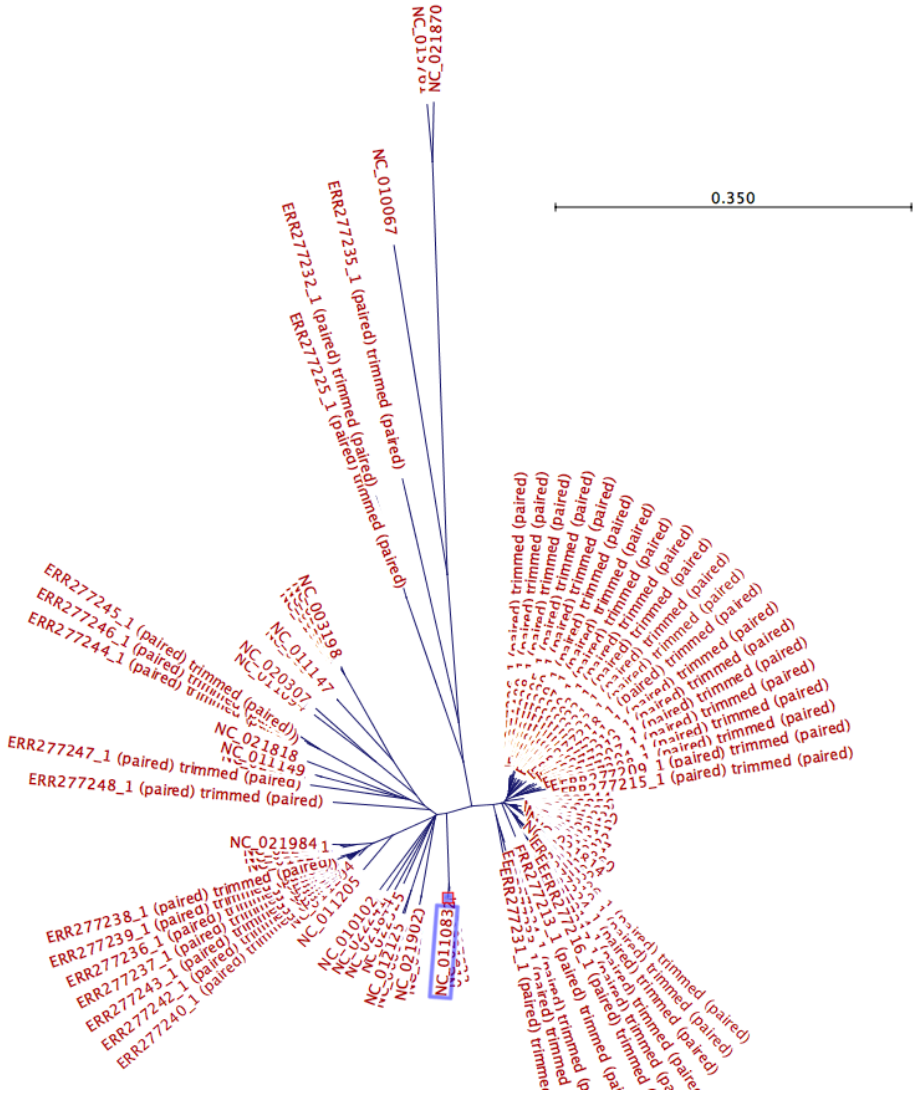


Figure 9.13: *The created K-mer tree is visualized using the radial tree topology setting. The genome reference acc no NC_011083 situated in the center of the tree is selected as a common reference candidate.*

Chapter 10

MLST Scheme Tools

The typing and characterization of pathogenic isolates plays an important part in epidemiology and outbreak analysis. MLST (Multilocus Sequence Typing) makes it possible to efficiently type strains against schemes with known isolates.

Classic MLST analysis types isolates against a small set of gene fragments (typical 500bp fragments for a set of seven house-keeping genes), whereas cgMLST (core-genome MLST) and wgMLST (whole-genome MLST) extends the analysis to thousands of loci, usually containing the complete coding gene sequences for the alleles for a given locus.

This section of the manual describes the MLST Scheme tools, which can be used to work with both cgMLST and wgMLST, as well as classic 7-gene schemes. The MLST Scheme typing can be applied either directly to the NGS reads of an isolate, or to an assembly of an isolate.

10.1 Getting started with the MLST Scheme tools

There are several ways to create MLST Schemes in the Microbial Genomics Module.

- **Create MLST Scheme**. Creates an MLST Scheme from sequence lists of reference genomes or assemblies with CDS annotations (see section 14.1).
- **Download MLST Scheme**. Downloads existing MLST schemes from PubMLST or Institut Pasteur (see section 14.2).
- **Import MLST Scheme**. Imports an MLST scheme from a set of fasta files, sequence type info and optional locus metadata (see section 14.3).

After a scheme has been obtained the following tools can be used together with the schemes.

- **Type With MLST Scheme** (see section 10.4).
- **Add Typing Results to MLST Scheme** (see section 10.5).

These tools are described in more detail in the following sections.

Finally, in order to be able to use the MLST Schemes outside of the Workbench, a MLST Scheme can be exported by clicking on the **Export** button and selecting **MLST Scheme**. To keep the data

manageable, this will export the MLST Scheme into a single zip file containing two text files in tsv format, one with the sequence type definitions and sequence type metadata and the second file containing locus metadata, and one fasta file per locus, corresponding to the format used for importing MLST Schemes.

10.2 MLST Scheme Visualization and Management

An MLST Scheme contains information about:

- The loci that define the regions of interest.
- For each locus, a list of known alleles.
- A list of sequence types, where each sequence type is described by the alleles present at each locus (the profile of the sequence type).

The MLST Scheme has several views. Switching between the views of the scheme is done by clicking the buttons at the lower-left corner of the view.

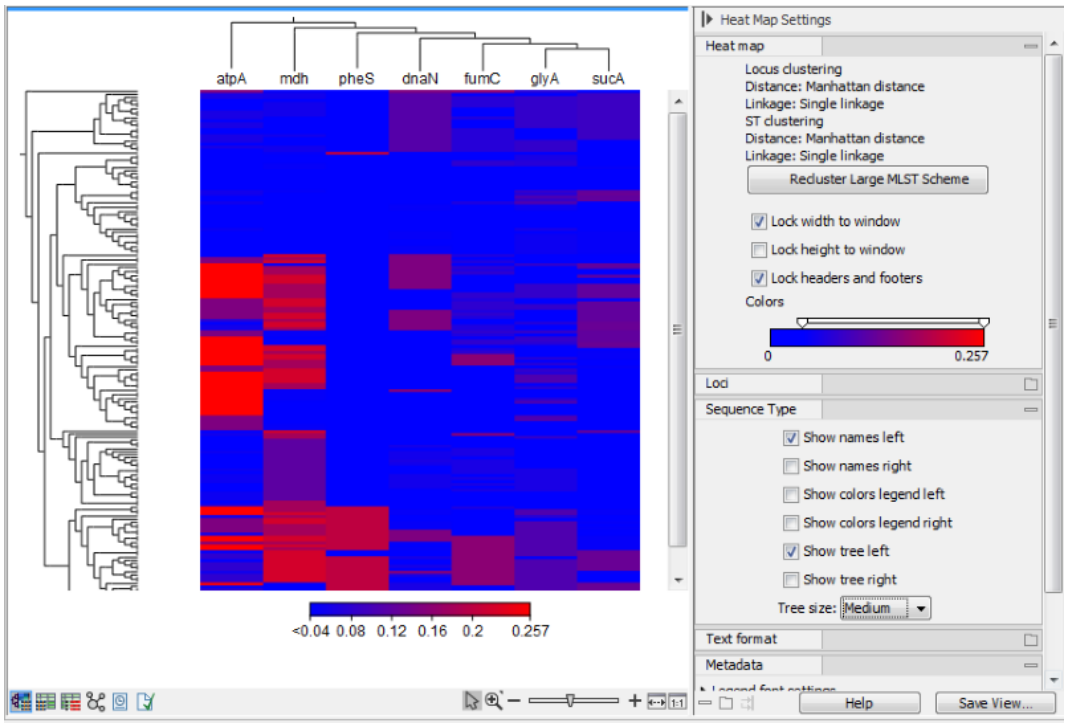


Figure 10.1: *The MLST Scheme Heat Map view.*

The heat map view shows an overview of the scheme (figure 10.1), with the sequence types on the vertical axis, and the loci on the horizontal axis. Each cell in the heat map is colored according to the frequency of the allele in the given locus, that is, a value of 0.9 means 90% of the sequence types have this particular allele. Missing alleles will have a value of zero, alleles not present in any sequence type are not represented by the heat map view. The heat map can optionally be clustered based on the allele frequency. The clustering settings can be specified at scheme creation time, but it is also possible to use the **Recluster MLST scheme** button to update the clustering.

By right-clicking on the heat map, it is possible to either select sequence types or loci in other views or copy sequence or loci names to the clipboard.

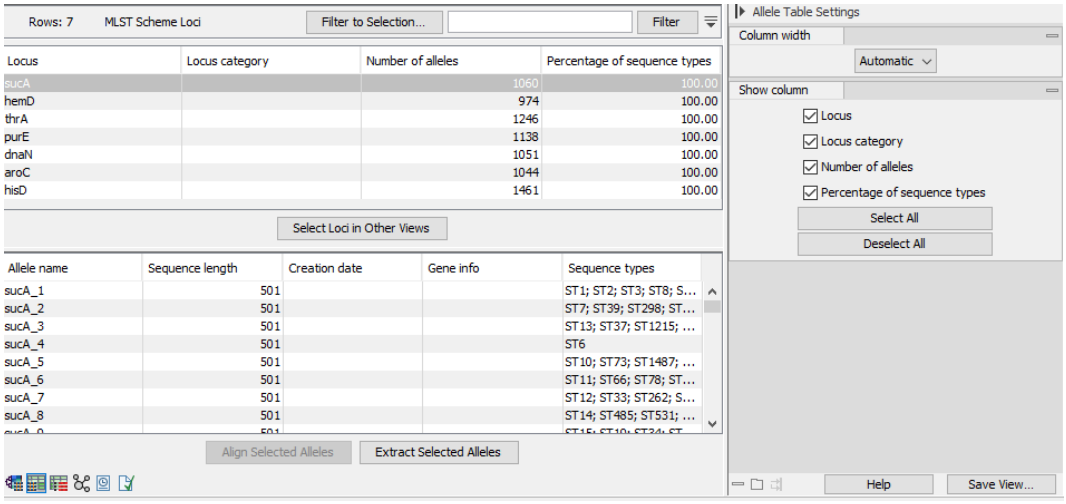


Figure 10.2: *The Allele table view.*

The allele table view (figure 10.2) has an upper table that lists the loci in the scheme. The table contains the following columns:

- **Locus**: the name of the locus
- **Locus category**: shows any virulence or resistance-gene related annotations.
- **Number of alleles**: the total number of alleles for this locus. Not all alleles may be part of a sequence type.
- **Percentage of sequence types**: shows how many of the sequence types have an allele in the given locus. For a strict core genome scheme, all of the sequence types contain all loci.

The lower table lists the alleles for the selected loci. It has the following columns:

- **Allele name**: the name of the allele.
- **Sequence length**: length in nucleotides.
- **Creation date**: when the allele was added.
- **Gene info**: AMR or virulence related information.
- **Sequence types**: the sequence types that contain this allele.

It is possible to **Align Selected Alleles**, which creates a new multiple sequence alignment view or to **Extract Selected Alleles**, which creates a sequence list with the alleles.

The Sequence Type table view (figure 10.3) shows the sequence types in the scheme. It always contains the following columns:

- **ST**: the name of the sequence type

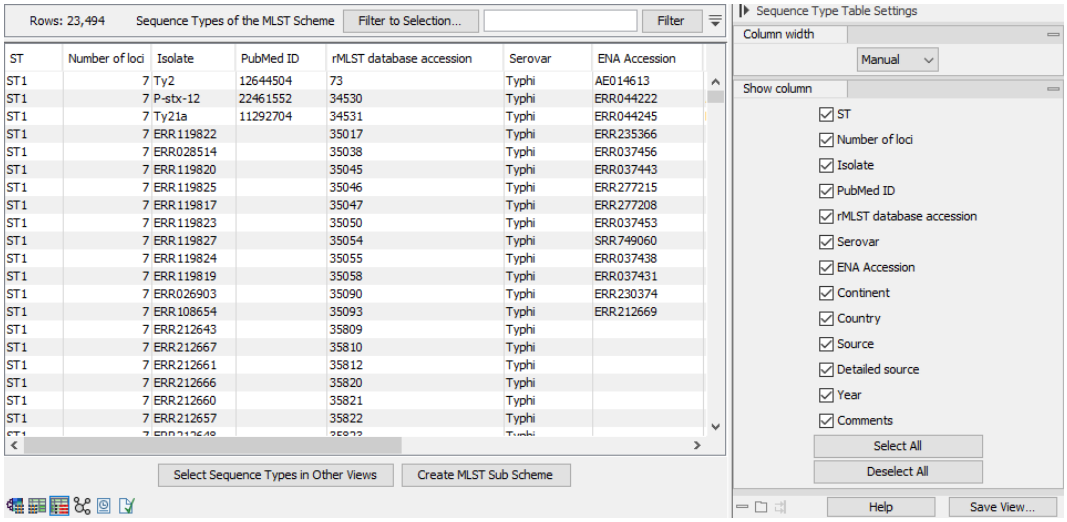


Figure 10.3: *The Sequence Type table.*

- **Number of loci**: the number of loci, that are defined for this sequence type. Strict core genome schemes and classic 7-gene schemes will have the same number of loci for all sequence types.

Several other columns with arbitrary metadata information may be present as well.

At the bottom of the view, two buttons make it possible to **Select Sequence Types in Other Views** and to **Create Large Sub Scheme**.

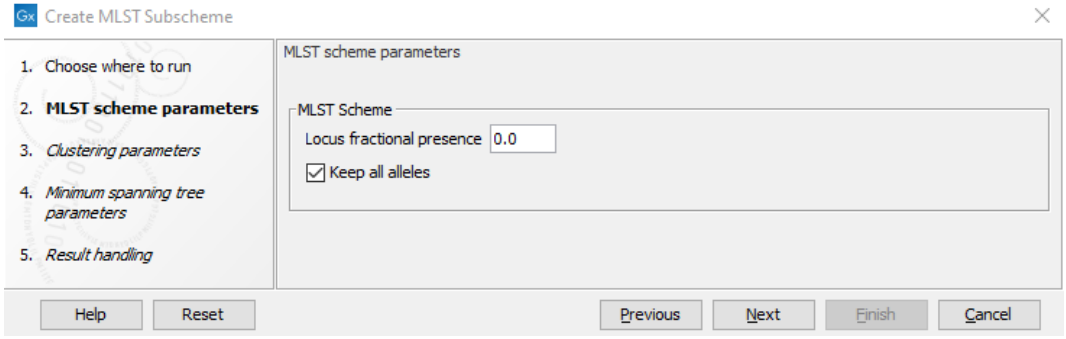


Figure 10.4: *The Create MLST Subscheme options.*

The **Create Large Sub Scheme** has the same options (figure 10.4) as the other scheme creation tools, except for some additional options for pruning the scheme:

- **Locus fractional presence**: the fraction of sequence types required to have an allele specified for a given locus before the locus is added to the new scheme. For instance, a value of 0.95 would mean that the resulting scheme only contains loci present in at least 95% of the selected sequence types (a loose core genome scheme).
- **Keep all alleles**: if this option is deselected, only alleles that are part of at least one sequence type are retained. Alleles from discarded loci will always be removed.

Finally, the MLST Scheme also has a Minimum Spanning Tree view, which is the topic of the next section.

10.3 Minimum Spanning Trees

A minimum spanning tree is a tree connecting all nodes in a graph, in a way such that the sum of edge lengths is minimized, see figure 10.5.

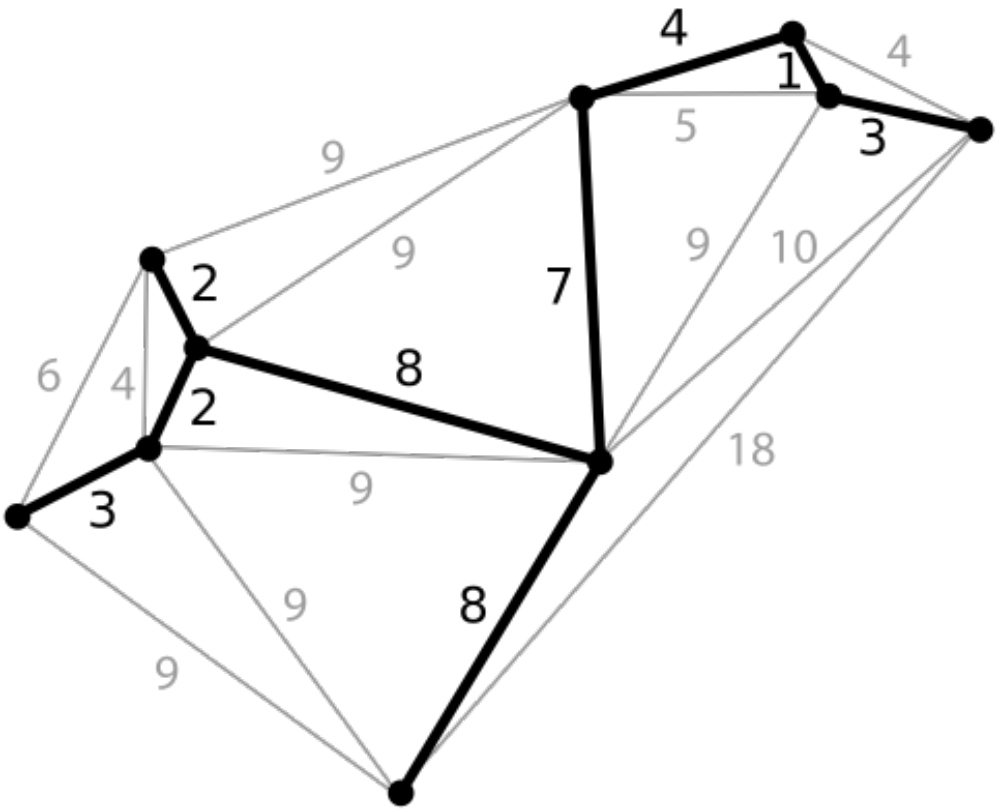


Figure 10.5: *Minimum Spanning Tree construction. (Public domain image from Wikimedia: Minimum spanning tree.svg.)*

Minimum spanning trees can be a bit counter-intuitive. Consider figure 10.6: The distance between A and B is not necessarily larger than the distance from A to C. But we do know that the distance between two nodes is greater or equal to the largest edge connecting them. (e.g. the distance between A and B is at least two, and the distance between A and C is at least two)

Minimum spanning trees are often used to visualize relationships between strains or isolates. But note that MST's are not unique - there are often many possible trees, especially for the classic 7-gene schemes, where there are only a very limited number of possible edge lengths. In order to break the ties when constructing the tree, our MST implementation favor creating connections to nodes that have many low-distance relations in the allelic distance matrix.

Minimum Spanning Trees can be created using the **Create MLST Scheme**, the **Download MLST Scheme**, the **Import MLST Scheme** tools or the **Create MLST Sub Scheme** button of an existing scheme.

10.3.1 The Minimum Spanning Tree view

It is possible to view the minimum spanning tree by selecting the MST icon ( ) at the bottom of the view, see figure 10.7.

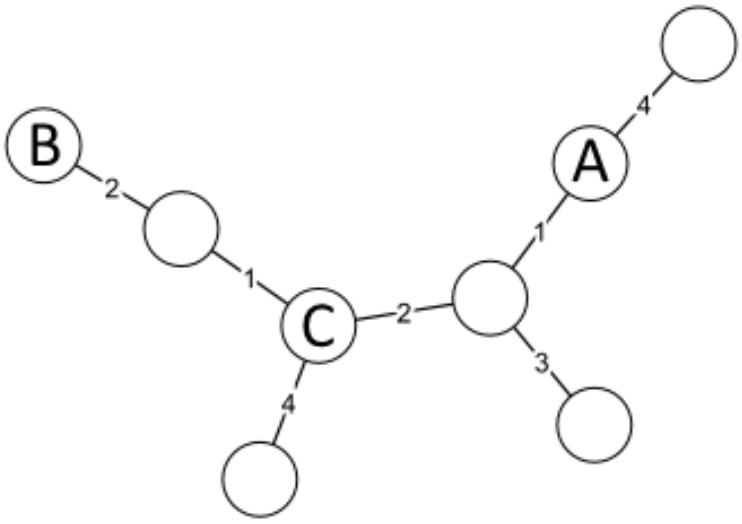


Figure 10.6: *Minimum Spanning Tree distances.*

An initial layout is calculated for a minimum spanning tree during the scheme creation, but it is possible to make changes to the layout.

If a change is made, the layout will be updated using a force-directed layout scheme: fictional forces are assigned to the tree nodes so that non-connected nodes will repel each other, while connected nodes will be held together with a spring force.

It is important to note that branch lengths in a force-directed layout may not be proportional to their ideal distance due to the repulsive force - in fact, the distance can be very different near heavy clusters.

10.3.2 Navigating the Tree view

It is possible to zoom in and out by pressing CTRL (or ⌘ on Mac) and using the scroll-wheel on the mouse.

Nodes can be selected by clicking on them (which toggles them on and off), or by dragging the mouse to create a lasso selection (figure 10.8).

It is possible to clear the current selection by pressing on an empty region of the canvas.

When nodes are selected, they will stay in a fixed position. This can be helpful when manually adjusting the layout, for instance, to prepare the tree for publication (figure 10.9).

The following actions are available from the buttons at the bottom of the view:

- **Select Sequence Types in Other Views**: selected sequence types will be selected in other views that support it. Note that if nodes are collapsed, all the sequence types in a collapsed node will be selected in the other views.
- **Create MLST Sub Scheme**: This makes it possible to create a new scheme based on the

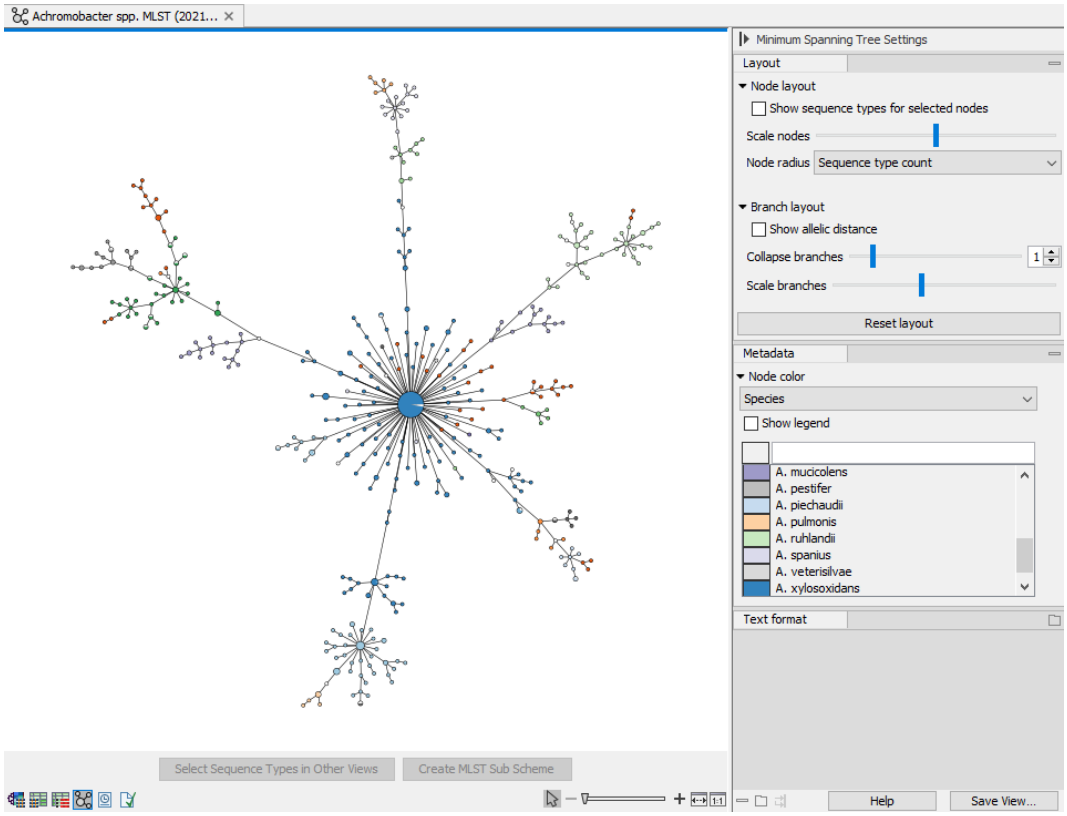


Figure 10.7: *The Minimum Spanning Tree view.*

selected sequence types.

10.3.3 The Layout panel

The following options are available: **Node layout**

- **Show sequence type names**: show the names of the sequence types for the nodes in the tree. If nodes are collapsed, only the first few names will be shown.
- **Scale nodes**: makes the nodes large or smaller.
- **Node radius**: either 'Sequence type count' or 'Isolate count'. A sequence type may have multiple isolates and metadata entries. This setting determines whether the node radius is based on the number of sequence types or isolates.

Branch layout

- **Show allelic distance**: shows the distance between different nodes in the tree. The distance is calculated as the number of loci where the allele assignment differs. Note that loci may have missing assignments. In this case, the distance calculation depends on the choice made when building the scheme (see section 14.1).
- **Collapse branches**: It is possible to reduce the complexity of the tree by clustering together nodes that are within a specific allelic threshold of each other. When setting a threshold, clusters will be formed where all nodes in a cluster are within the specified threshold to

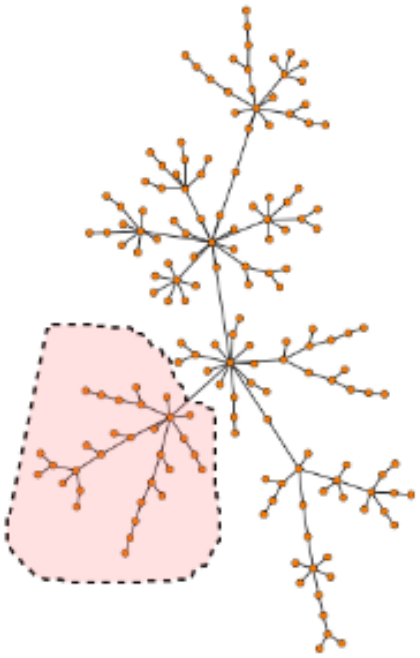


Figure 10.8: *Minimum Spanning Tree lasso selection.*

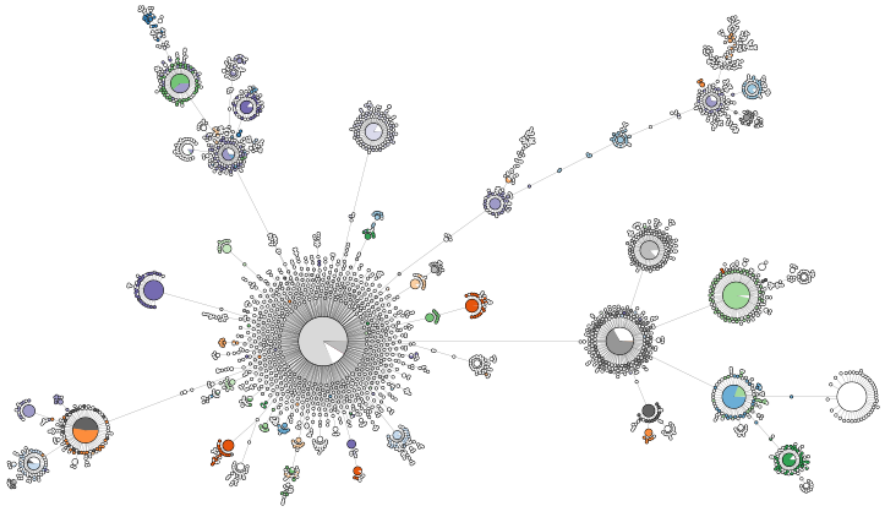


Figure 10.9: *Minimum Spanning Tree with manual layout modifications.*

at least one other node in the cluster (single-linkage clustering). See (figure 10.10) for an example of a tree where nodes have been collapsed.

- **Scale branches**: This parameter can be adjusted to make the branches longer. Note that the force-directed layout is primarily controlled by the repulsive force, so adjusting this parameter will not always have a proportional impact. Also note that due to the large span in allelic distances for cg- and wg-MLST schemes, the layout algorithm tries to fit an ideal branch length that is proportional to the square-root of the allelic distance.

- **Reset layout**: Pressing this button will reset the layout: this is done by first creating an initial radial layout, where no branches are crossed, and then applying a force-directed layout. If the graph is uncollapsed, pressing the **Reset layout** button will reset to the default layout created during scheme building.

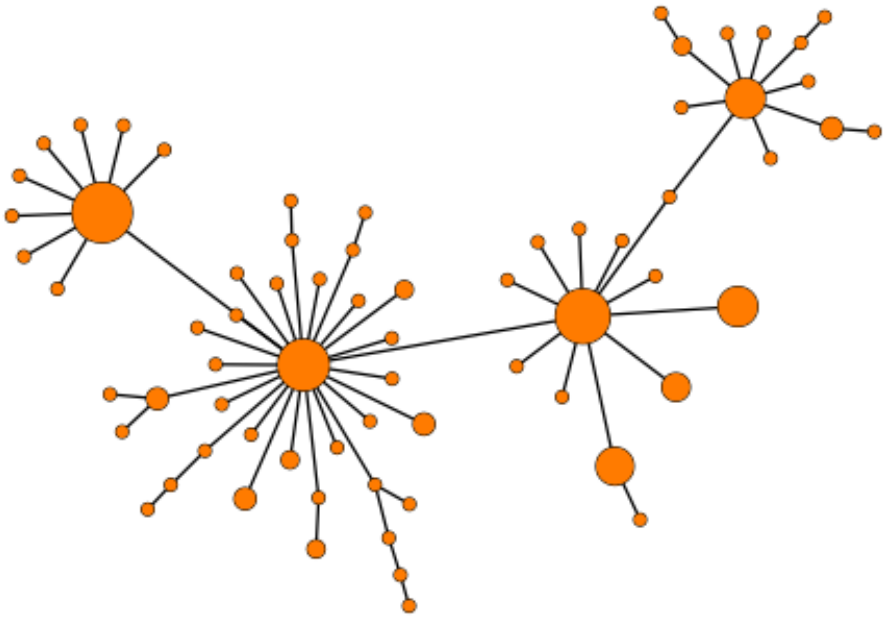


Figure 10.10: *Collapsing nodes in a Minimum Spanning Tree.*

10.3.4 The Metadata panel

The metadata panel makes it possible to color node based on categorical metadata.

Collapsed nodes may have different metadata, in which case the fractional proportions of the different metadata categories will be shown as a pie-chart.

Note that uncollapsed nodes may have different multiple metadata values - this happens when a sequence type is associated with multiple metadata values, for instance from different isolates.

Note that when hovering over a node with the mouse, it is possible to see the distribution of metadata values at the status bar on the bottom of the application window.

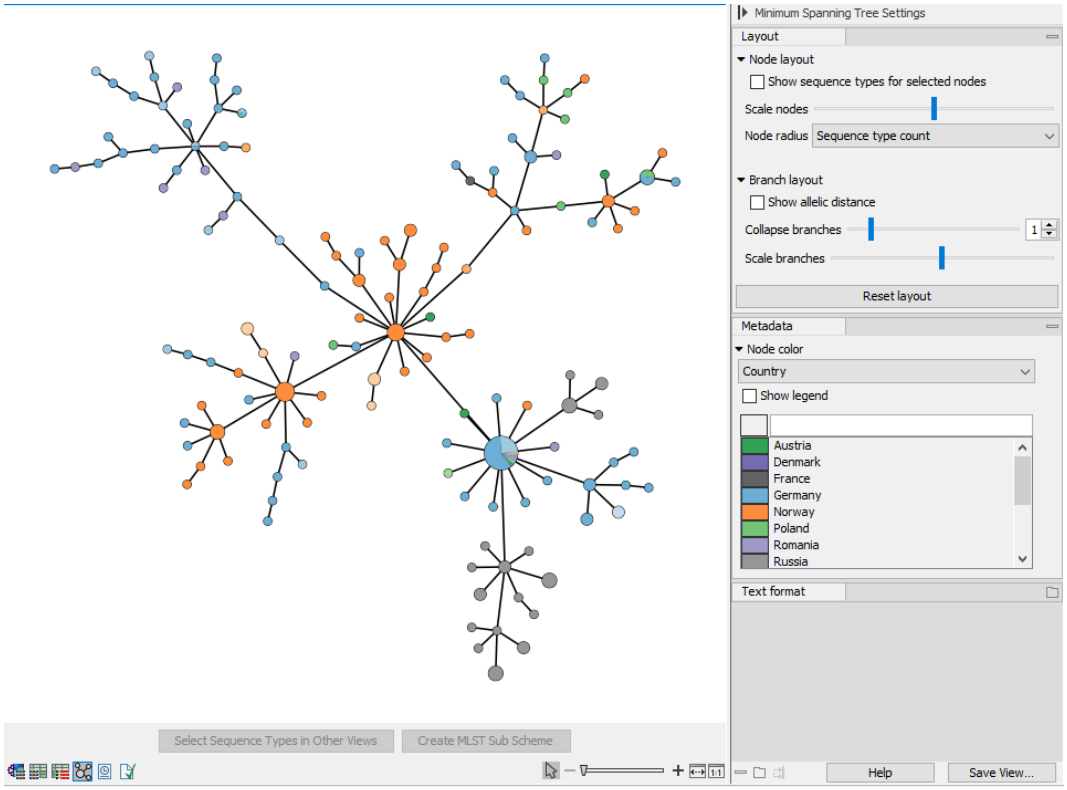


Figure 10.11: *Displaying metadata for Minimum Spanning Trees.*

10.4 Type With MLST Scheme

The **Type With MLST Scheme** tool is used for assigning a sequence type to an isolate.

Type With MLST Scheme is available from:

> **Tools** | **Microbial Genomics Module** () | **Typing and Epidemiology** () | **MLST Typing** () | **Type With MLST Scheme** ()

The tool takes a sequence list as input and will work with either raw NGS reads or an assembled genome. Note that if the input is raw NGS reads, and the tool reports multiple ambiguous sequence types, performing a standard De Novo Assembly might help to reduce noise and provide a more conclusive typing result.

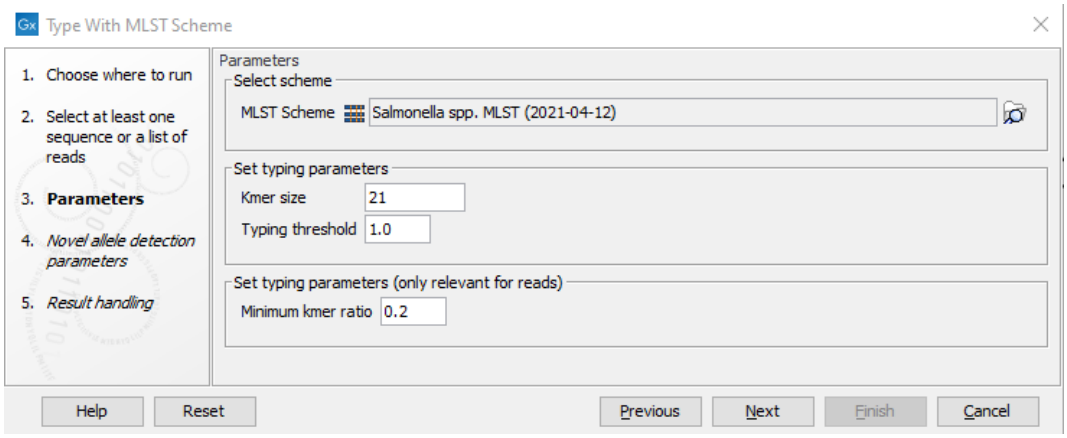


Figure 10.12: *Specifying scheme and typing parameters.*

In the next dialog step (figure 10.12), specify the scheme and the typing parameters. MLST schemes are available from **Download MLST Scheme** (see section 14.2).

Some MLST schemes contain sequence types with ambiguous bases. The Type With MLST Scheme tool does not support ambiguous bases and such sequence types will effectively be ignored.

The tool works by comparing the kmers in the input to the kmers in the alleles for the different loci.

The **Kmer size** determines the number of nucleotides in the kmer - raising this setting might increase specificity at the cost of some sensitivity.

The **Typing threshold** determines how many of the kmers in a sequence type that needs to be identified before a typing is considered conclusive. The default setting of 1.0 means that all kmers in all alleles must be matched. Lowering the setting to 0.99 would mean that on average 99% of the kmers in all the alleles of a given sequence type must be detected before the sequence type is considered conclusive. The typing threshold must be 0.5 or above.

When working with reads, the Type With MLST Scheme tool works by classifying allele calls as high-confidence and low-confidence calls to remove alternative allele calls for the same locus. The **Minimum kmer ratio** threshold gives the possibility to tweak the balance between high-confidence and low-confidence allele calls, e.g. decreasing this number will result in more high-confidence allele calls and thus more ambiguity in how an ST is assigned to the sample, conversely increasing this number will result in fewer high-confidence calls and may lead to no allele being called for a particular locus, which can make sequence type assignments less confident. Specifically, the kmer ratio is calculated as the number of observations for the least occurring kmer in an allele divided by the average number of observations for all kmers.

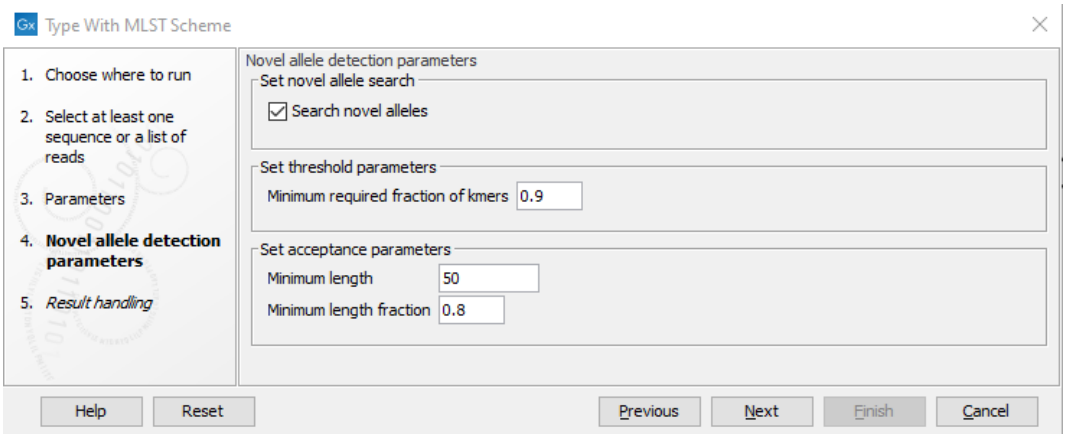


Figure 10.13: *Specifying novel allele detection parameters.*

The next step in the dialog determines how to handle novel alleles (figure 10.13): if the input isolate has loci with alleles that are not part of the scheme, it is possible to still detect the novel alleles. The novel alleles and the resulting new sequence type can then be added to the scheme using the **Add Typing Results to MLST Scheme** tool.

Novel alleles are detected as close hits to existing alleles in a locus. The **Minimum required fraction of kmers** determines how close a match must be: the default setting of 0.9 means that at least 90% of the kmers for an allele in a locus must be identified before the novel allele detection is initiated.

If the input to the tool is raw NGS reads, the tool will assemble the reads containing the kmers for the possible novel allele. If the input is already an assembled genome, the existing alleles for a locus will be mapped to the assembly to extract a novel allele.

After a candidate novel allele has been identified, it is aligned to the other alleles in the locus.

If the scheme has been built with the **Check codon positions** option of the **Create MLST Scheme** tool enabled (see section 14.1), or if the scheme was imported with a specified genetic code (see section 14.2), the start and stop codons in the novel allele sequence are then identified, and the sequence is then trimmed to the start and stop codons that most closely match the length of the existing alleles in the locus. Alleles that contain both a start and a stop codon at the beginning and end, respectively, and pass the acceptance parameters (see below) will be marked as Complete in the output table from the tool.

The acceptance parameters describe the final consistency check: the novel allele must not contain a stop codon, it must be at least the **Minimum length** in nucleotides and have at least a length of the specified **Minimum length fraction** of the shortest allele in the locus before it is accepted.

10.4.1 Type With MLST Scheme results

The **Type With MLST Scheme** tool outputs a report, summarizing the typing (figure 10.14), and a MLST Typing Result element.

The Typing result is based on a comparison of the typing threshold (TT), specified when the tool was launched, and the Average kmer fraction (AKF), an analysis result described later in this section. The Typing result will be one of:

- **Conclusive**: Exactly one sequence type has AKF >= TT.
- **Ambiguous**: More than one sequence type has AKF >= TT.
- **Inconclusive**: No sequence type has AKF >= TT.
- **Not possible**: No sequence type has AFK >= 0.5 (the lower detection limit for the tool).

The Typing information section contains the following:

- **Average kmer fraction**: for all alleles in a given sequence type, we calculate how many of the allele's kmers were detected. This number is the average fraction of the number of kmers detected in all these alleles. It can be seen for the top 5 STs in the report and for all STs in the typing result.
- **Alleles identified**: how many alleles in the sequence type were identified, that is, all kmers in the allele were found in the sample.
- **Alleles called**: how many alleles in the sequence type had at least one kmer found in the sample.
- **Allele count**: the total number of alleles in the sequence type.

1 Typing result

Typing	Conclusive
Sequence type	ST6

2 Typing information

Sequence type	Average kmer fraction	Alleles identified	Alleles called	Allele count
ST6	0.9959796	1,622	1,655	1,661
ST9	0.9213338	1,500	1,531	1,661

3 MLST search summary

3.1 Sample information

Number of kmers matching the scheme	67,478,538
Loci without hits	0
Estimated sample coverage	10
Alleles with all kmers found	1,749
Alleles with kmer fraction of at least 90.00%	175
Novel alleles identified	5
Problematic loci for novel allele detection	-

3.2 Scheme information

Scheme name	Tutorial Scheme
Genetic code	11 Bacterial, Archaeal and Plant Plastid
Check codon positions	Yes
Sequence types in MLST scheme	9
Loci in MLST scheme	1,661
Alleles in MLST scheme	8,373

Figure 10.14: *The Type With MLST Scheme report.*

The sample information and scheme information contains various statistics about the input and scheme.

To add the report information to a Result Metadata Table, see section 20.2.3.

10.4.2 The MLST Typing Result element

The MLST Typing Result element contains several views. Switching between the views of the scheme is done by clicking the buttons at the lower-left corner of the view. The number of Sequence Types shown is limited to 100 or the number of Sequence Types in the scheme, depending on which is lower.

The sequence type table is a tabular view with information about how well the sample matched

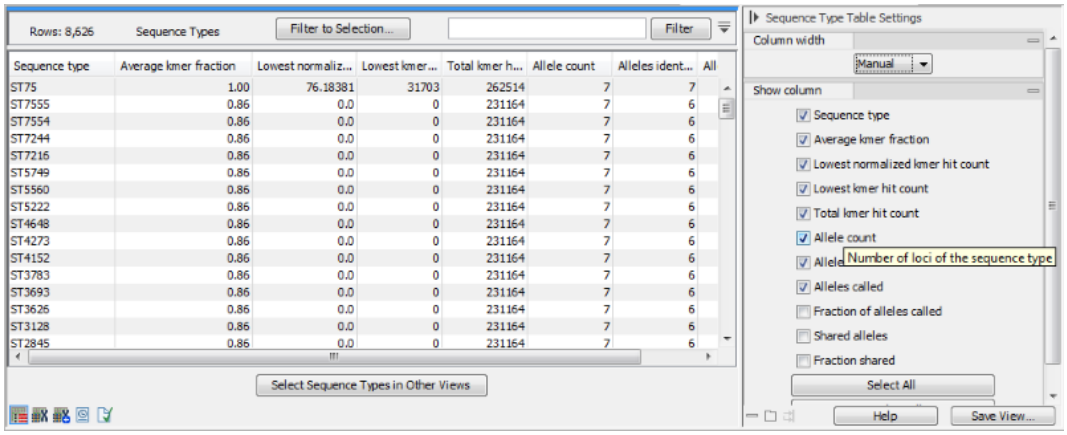


Figure 10.15: *The sequence type table for a MLST Typing Result.*

the sequence types in the scheme (figure 10.15). It contains the following columns:

- **Sequence type**: name of the sequence type
- **Average kmer fraction**: for all alleles in a given sequence type, we calculate how many of the allele's kmers were detected. This number is the average fraction of the number of kmers detected in all these alleles.
- **Lowest normalized kmer hit count**: Normalized kmer hit count for the allele with the lowest normalized hit count of the sequence type.
- **Lowest kmer hit count**: Number of kmer hits for the allele with the fewest hits of the sequence type.
- **Total kmer hit count**: sum of kmer hits for all alleles of the sequence type
- **Allele count**: the total number of alleles in the sequence type.
- **Alleles identified**: the number of alleles in the sequence type where all kmers of the allele were found in the sample.
- **Alleles called**: the number of alleles in the sequence type with at least one kmer found in the sample.
- **Fraction of alleles called**: the ratio between alleles called and the allele count for the sequence type.
- **Shared alleles**: Number of alleles shared with the best scoring sequence type
- **Fraction shared**: Fraction of alleles shared with the best scoring sequence type

The allele table (figure 10.16) contains information about the alleles that were identified in the sample. It contains the following columns:

- **Locus**: the name of the locus.
- **Allele call**: the allele that was identified for that locus. Only the best allele is reported, but if multiple alleles are tied for the first place, they will all be reported.

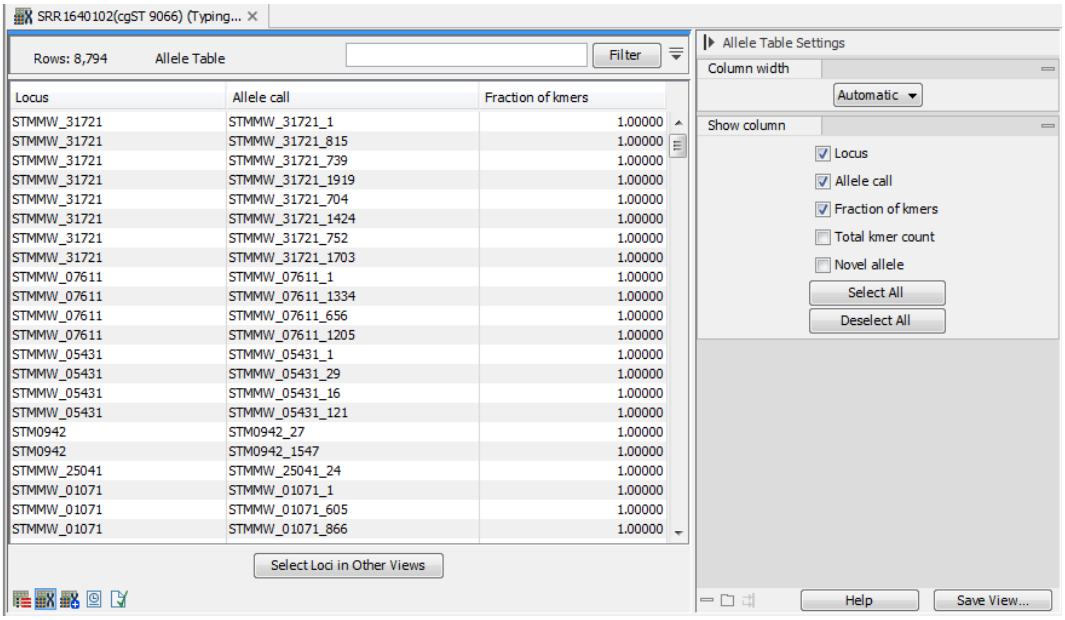


Figure 10.16: *The allele table for a MLST Typing Result.*

- **Fraction of kmers**: the fraction of kmers of the allele that were found in the sample.
- **Total kmer count**: total number of kmer hits for the allele.
- **Novel allele**: contains the string 'Novel' for novel alleles, otherwise this field is left blank.

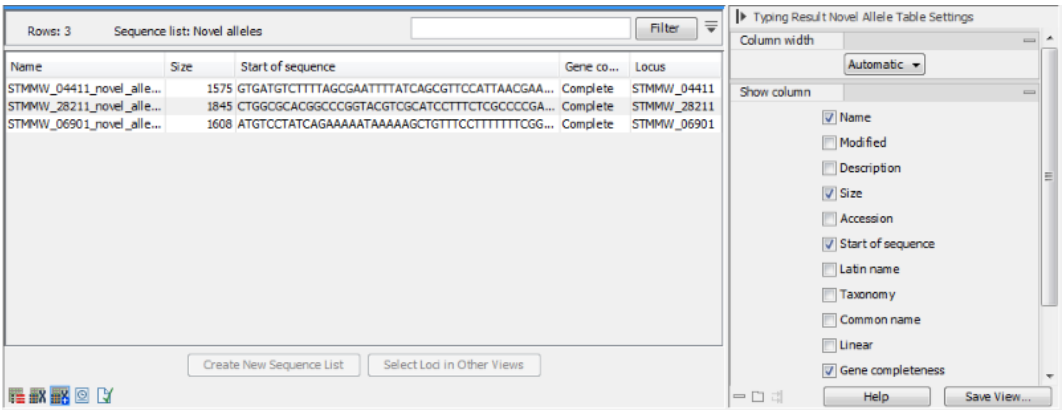


Figure 10.17: *The novel allele view for a MLST Typing Result.*

The novel allele view (figure 10.17) contains the novel alleles that were detected (if searching for novel alleles was enabled during the typing).

It is a sequence list, and it is possible to extract the complete sequences using the **Create New Sequence List** button.

The **Gene completeness** column is the only non-standard sequence list column: if a novel allele starts with a start codon and ends with a stop codon it is considered complete. Note that all novel alleles found with a scheme without a translation code will be incomplete.

10.5 Add Typing Results to MLST Scheme

After typing an isolate, it is possible to add the information to the MLST Scheme. There are several different possibilities when adding a typing result:

The typing result may have matched an existing sequence type completely. In this case, it is still possible and useful to add the typing result to the scheme, in order to add additional isolate metadata for the sequence type which may be from metadata annotations on the sequences or from a metadata table.

The typing result may have matched existing alleles in the scheme but in a new combination not present in any of the existing sequence types. In this case, the typing result will simply be added as a new sequence type.

The typing result may introduce new (novel) alleles to the scheme. In this case, both a new sequence type and one or more alleles are added to the scheme.

Add Typing Results to MLST Scheme is available from:

> **Tools** | **Microbial Genomics Module** ( ) | **Typing and Epidemiology** ( ) | **MLST Typing** ( ) | **Add Typing Results to MLST Scheme** ( )

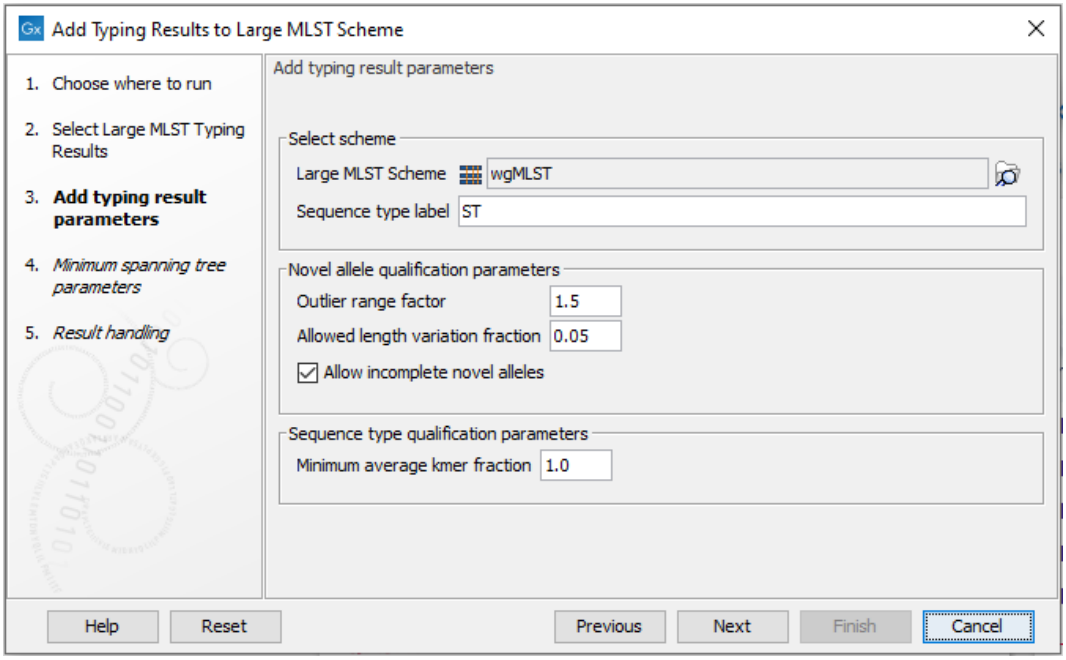


Figure 10.18: *Add typing result parameters.*

After selecting the MLST Typing result to add, the next step is to set up the **Add typing result parameters** (figure 10.18):

- **MLST Scheme**: the scheme that the typing results will be added to. Adding the typing results will not modify the original scheme, but create a new copy with the added types.
- **Outlier range factor**: the allele length outlier definition in terms of the interquartile range of the length distribitution of alleles in a locus. Novel alleles outside this range will not be added to the scheme.

- **Allowed length variation fraction**: The allowed length variation of a novel allele with respect to the median length of alleles in a locus. This allows adding novel alleles with minor length variations irrespective of the outlier definition.
- **Allow incomplete novel alleles**: whether only complete novel alleles (containing both start and stop codon) should be allowed. If incomplete novel alleles are not allowed, a sequence type with incomplete alleles for a locus will be added with missing alleles for that locus. Classic 7-gene schemes typically contain partial gene fragments, and in this case incomplete novel alleles should be allowed.
- **Minimum average kmer fraction**: if this value is larger than zero, a typing result will only be added if the isolate was sufficiently similar to an already existing sequence type in the scheme - or to put it differently - at least one of the sequence types in the MLST Typing Result must have an Average kmer fraction larger than this threshold. Note, that when typing against an empty scheme, this value must be set to zero, to allow for the sequence type to be added. This option is mostly useful when adding a large number of isolates in bulk without manually inspecting them.

After adding new sequence types it is necessary to recreate the Minimum Spanning Tree. The options are the same as described in the **Download MLST** section (section 14.2)

10.6 Identify MLST Scheme from Genomes

This section describes how to perform the identification of the relevant MLST Scheme for a genome sequence or list of genome sequences.

This tool can be used before running the Type With MLST Scheme tool in case you are working with a sample containing a single or multiple unknown species, as in the Type among Multiple Species workflow.

Identify MLST Scheme from Genomes is available from:

> **Tools** | **Microbial Genomics Module** () | **Typing and Epidemiology** () | **MLST Typing** () | **Identify MLST Scheme from Genomes** ()

The input to the tool is a sequence, or a sequence list (figure 10.19).

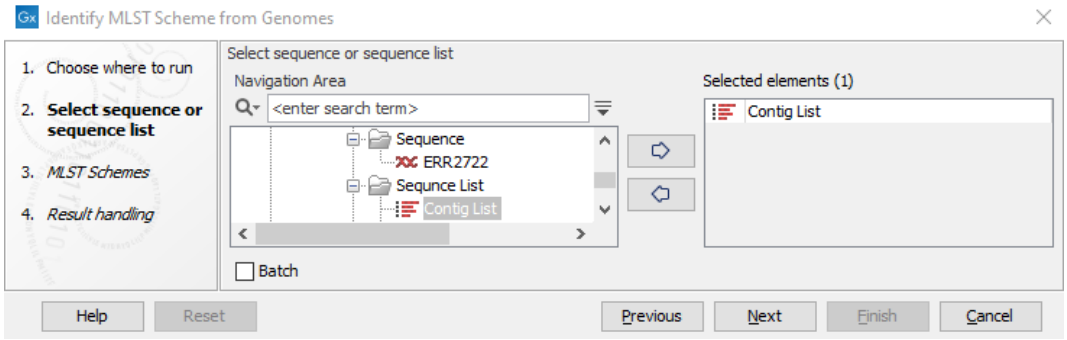


Figure 10.19: *Select relevant genome sequence or sequence list.*

The next step is to select as many MLST schemes as necessary to identify the species present in the input sample (figure 10.20).

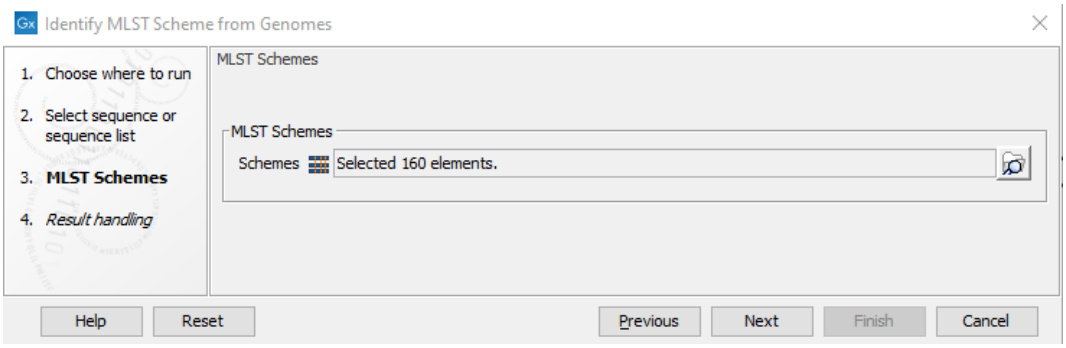


Figure 10.20: *Select relevant MLST scheme(s) to search among.*

To identify the best matching scheme, the tool identifies the 10 most prevalent loci, i.e. loci that occur in most or all of the sequence types. If fewer loci are available, the tool will base the identification on these, thus the tool also works for classic 7-gene MLST schemes, given that they are in the MLST Scheme format.

The k-mers for all alleles for these most prevalent loci are then determined, and the provided references are checked for their presence.

The output of this tool is the MLST scheme that best matches the sequences analyzed. To add the obtained best match to a Result Metadata Table, see section 20.2.3.

The tool will not produce an output if no scheme could be uniquely identified.

Chapter 11

Additional Typing Tools

11.1 Spoligotype Mycobacterium Tuberculosis

Spoligotyping (Spacer Oligonucleotide Typing) is a method for typing Mycobacterium tuberculosis based on the presence/absence of 43 spacer oligonucleotides. Originally, this was a PCR-based assay done in the lab. The traditional method has been adapted for next-generation sequencing (NGS) data, enabling *in silico* spoligotyping directly from whole-genome sequencing data.

Spoligotype Mycobacterium Tuberculosis offers spoligotyping of M. tuberculosis isolates from NGS reads. The tool works by searching for the 43 spacer sequences [Van Embden et al., 2000] in reads obtained from M. tuberculosis samples and counting the number of times a match is found. Presence/absence is determined as a binary code, which is then translated into octal code, lineage, and SIT (Shared International Type) using SpolLineages [Couvin et al., 2020].

11.1.1 Spoligotype Mycobacterium Tuberculosis parameters

To run the **Spoligotype Mycobacterium Tuberculosis** tool, go to:

> **Tools** | **Microbial Genomics Module ()** | **Typing and Epidemiology ()** | **Spoligotype Mycobacterium Tuberculosis ()**

The tool takes sequence lists as input.

The tool will accept any reads, but as spoligotyping targets genomic regions in M. tuberculosis, you are unlikely to get meaningful results if the reads are not from the M. tuberculosis complex or its closely related strains. To avoid matching in off-target sequences, it is best practice to trim reads for adapters, quality, etc. In non-isolate samples it can also be beneficial to extract M. tuberculosis reads first e.g., by using **Taxonomic Profiling** before typing.

In the wizard, minimum coverage thresholds can be set (figure 11.1):

- **Minimum coverage threshold (count).** The minimum number of times a spacer must be found for it to be considered present in the sample.
- **Minimum coverage threshold (%).** The minimum percent of times, relative to the maximum across all spacers, a spacer must be found for it to be considered present in the sample.

Both thresholds must be met for a spacer to be present. To apply only a count threshold, set the

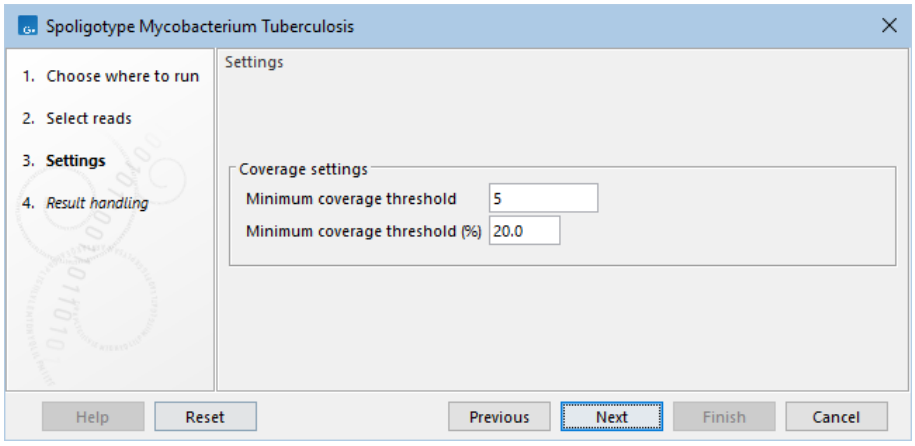


Figure 11.1: *Spoligotype Mycobacterium Tuberculosis parameters.*

percentage to 0. Conversely, to apply only a percentage threshold, set the count to 1.

11.1.2 Spoligotype Mycobacterium Tuberculosis output

The tool outputs a report with the typing results. An example report can be seen in figure 11.2.

- **Spoligotyping result.** This section contains the SpolLineages results based on the spoligotype. Five outputs are reported:
  - **Binary code.** 43 digit binary code, where each spacer is represented as present (1) or absent (0).
  - **Octal code.** 15 digit octal code calculated from the binary code (pattern of triplet of spacers represented as 0-7).
  - **SIT.** Shared International Type of the sample.
  - **Lineage.** The main spoligotype name followed by lineage name and, when available, associated lineage number.
  - **Sublineage.** The spoligotype.
- **Spoligotype pattern.** This section contains a visual representation of the 43 spacers. Spacers present in the sample are represented by 'n's, while 'o's represent the absence of spacers. When combining spoligotyping reports from multiple samples with the Combine Reports tool, the difference in pattern between samples can be easily visualized, see figure 11.3.
- **Coverage.** Coverage statistics for the detected spacers i.e., calculated from the spacers considered present in the sample. Counts are given as the number of times a spacer sequence was detected.
- **Coverage plot.** Bar plot showing the coverage for each spacer. The minimum coverage threshold set when running the tool is visualized as a horizontal line across the plot.

The coverage statistics and plot can be useful to evaluate whether the minimum coverage threshold should be adjusted.

1 Spoligotyping result

Binary code	1111011111111111111100000000000000000110111
Octal code	757777600000331
SIT	Orphan
Lineage	Unknown, Unknown
Sublineage	Unknown

2 Spoligotype pattern

Pattern	••• •0• ••• ••• ••• ••• ••0 000 000 000 000 000 0•• 0•• •

3 Coverage

Coverage statistics are calculated only for spacers considered present according to the minimum coverage threshold.

Minimum	63
Median	80
Maximum	105
Mean	81.08
Standard deviation	10.77

4 Coverage plot

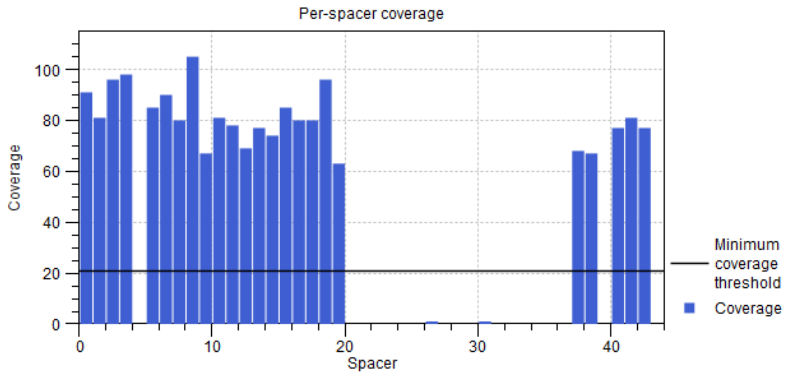


Figure 11.2: *Spoligotype Mycobacterium Tuberculosis report.*

2.2 Spoligotype pattern

The table is based on 5 samples.

Sample name	Pattern
ERR1035264 (Spoligotyping report)	nnn non nnn nnn nnn nnn nno ooo ooo ooo ooo ooo onn onn n
ERR1035281 (Spoligotyping report)	nno nnn nnn nnn nnn nnn nno ooo nnn nnn nno ooo nnn nnn n
ERR1035293 (Spoligotyping report)	onn nnn nnn nnn nnn nnn nno ooo nnn nnn nno ooo nnn onn n
ERR1035306 (Spoligotyping report)	nnn nnn nnn nnn nnn nnn nno ooo nnn nnn nno ooo nnn nnn n
ERR1035316 (Spoligotyping report)	nnn nnn nno oon nnn nnn nno ooo nnn nnn nno ooo nnn nnn n

Figure 11.3: *Spoligotype patterns from several Mycobacterium tuberculosis samples combined using Combine Reports.*

Part VI

Functional and Drug Resistance Analyses

Chapter 12

Functional Analysis

Two of the most widely used definitions of biological function are available in the form of the Gene Ontology (GO) and Pfam databases. While GO is a hierarchy of higher-level functional categories, Pfam (Protein families) classifies proteins into families of related proteins with similar function.

Several tools are available for functional analysis. From a whole metagenome shotgun sequencing dataset as reads, the first step is to assemble the reads using the **De Novo Assemble Metagenome** tool (see section 4). The resulting contigs can then be annotated with coding sequences (CDS) using the **Find Prokaryotic Genes** tool. Given a set of contigs with CDS annotations, the **Annotate CDS with Best BLAST Hit**, the **Annotate CDS with DIAMOND Hits** and the **Annotate CDS with Pfam Domains** tools can be used to annotate all CDS in the annotated contigs with BLAST or DIAMOND hits or Pfam protein families and GO terms, respectively. The database needed for GO annotation can be downloaded using the **Download GO Database** tool, while the Pfam database can be downloaded using the built-in **Download Pfam Database** tool and BLAST databases can be downloaded or created using the built-in **Download BLAST Dabases** and **Create BLAST Database** tools.

Once the contigs are annotated with Pfam annotation, GO terms and/or BLAST hits, the next step will often be to map the original reads back to the annotated contigs, using the built-in **Map Reads to Reference** tool, in order to be able to assess the abundance of the functional annotations. This last step is performed using the **Build Functional Profile** tool (see section 12.7).

All tools described above should be run independently for individual samples (or batched), resulting in a functional profile for each sample. A set of functional profiles can then be joined using the **Merge Abundance Tables** tool (see section 7.1). The functional profile of multiple samples can now be visualized and compared as described in section 5.3.2.

12.1 Find Prokaryotic Genes

The **Find Prokaryotic Genes** tool allows you to annotate a DNA sequence with CDS information. The tool is currently for use with near-complete single prokaryotic genomic and metagenomic data.

The tool creates a gene prediction model from the input sequence, which estimates GC content, conserved sequences corresponding to ribosomal binding sites, start and stop codon usages, and a statistical model (namely, an Interpolated Markov Model) for estimating the probability of

a sequence to be part of a gene compared to the background. The model is then used to predict coding sequences from the input sequence. Note that this tool is inspired by Glimmer 3 (see https://ccb.jhu.edu/papers/glimmer3.pdf).

To maximize the gene prediction accuracy, the gene models should be trained on sequences that belong to the same species or to similar ones. When the input consists of sequences originating from multiple organisms, it is recommended to build a gene model for each organism by choosing the "Learn one gene model for each assembly" option. In assembly grouping, there are multiple options for specifying what should be considered an assembly. For example, when downloading assemblies from the Download Custom Microbial Reference Database tool and the prokaryotic databases from the Download Curated Microbial Reference Database tool, the "Assembly ID" column will be automatically populated and can be used for grouping, see section 22. When assembly information is not known, for example when the input consists of de novo assemblies, the option "Each input element is one assembly can be used". When working with de novo assembled metagenomics sequences, the Bin Pangenomes by Sequence and Bin Pangenomes by Taxonomy tools can be used to group sequences into bins whose sequences are likely to come from the same organism.

To start the analysis, go to:

> **Tools** | **Microbial Genomics Module** () | **Functional Analysis** () | **Find Prokaryotic Genes** ()

In the first dialog, select input sequences. The input should consist of one or few contigs from the same species. If several sequences are provided as input, the model training can be used to specify if the tool should build a separate model for each assembly. The tool can also be run in batch mode.

In the second dialog (figure 12.1), it is possible to configure the tool.

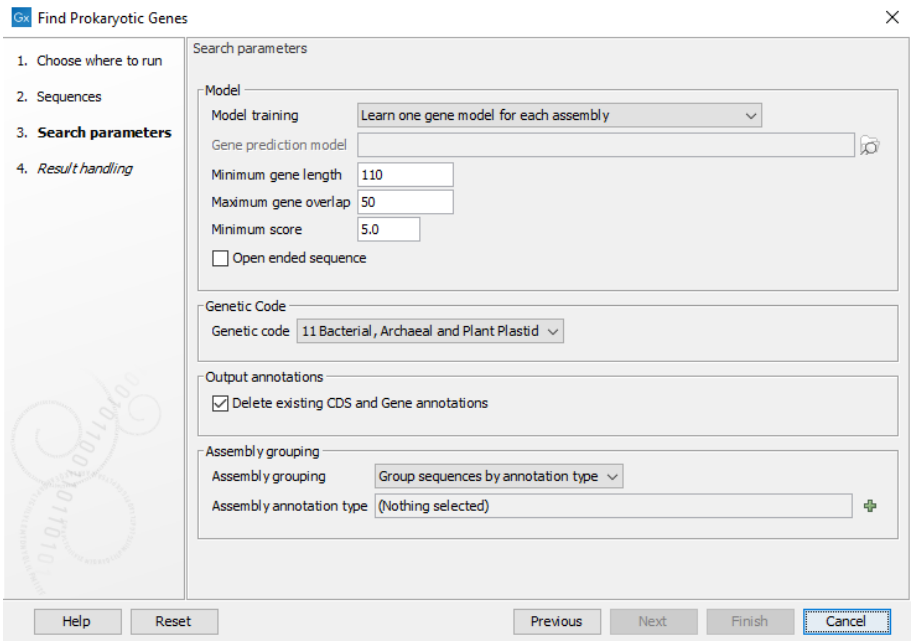


Figure 12.1: *Configuring the Find Prokaryotic Genes tool*

Model

- Learn one gene model: Learns a single model from the data. Assumes that the sequences come from one organism or a group of closely related organisms.
- Learn one gene model for each assembly: Learns a model for each assembly or bin. This option should be used when assemblies can be clearly distinguished, for example when they are separated with one assembly per sequence list or are assigned with with an ID in the Assembly ID column as is the case with output from Bin Pangenomes by Taxonomy, Download Custom Microbial Reference Database and the prokaryotic databases from the Download Curated Microbial Reference Database tool.
- Use a previously trained model and use its default parameters: This option allows to choose a model that has been previously trained and run the analysis with the same parameters used when training the model.
- Use a previously trained model: This option allows to choose a model that has been previously trained. It also allows to modify some parameters.

In all but one of this option, the following parameters can be modified:

- Minimum gene length: in bp, excluding start and stop codons.
- Maximum gene overlap: in bp
- Minimum score: Putative genes with a score below this value will be ignored. The value of a gene score depends on how well the sequence of the gene matches the model. It is computed by taking into account how much the sequence is typical of a coding region (as opposed to background noise or the same coding region read in a different frame), of the prevalence of the start codon, and of the presence of a putative ribosomal binding site near the start codon.
- Open ended sequence: check this option to annotate open-ended sequences, which is particularly useful for annotating small contigs.

Genetic Code The genetic code to use (default to bacterial). This genetic code is used to determine which stop codons should be used and to compute a background distribution for amino-acid usage.

Output annotations Delete Existing CDS and Gene Annotations. This is selected by default in order to avoid having many duplicate annotations. Unchecking is useful if one wants to compare the results with other annotations.

Assembly Grouping

- Each sequence is one assembly: Each sequence in the input elements is treated as one assembly.
- Each input element is one assembly: Each input element is treated as one assembly regardless of annotation types and number of sequences in the input element.
- Group sequences by annotation type: This option allows to choose an annotation type. Each unique label in the input is then treated as one assembly.

The tool will output a copy of the input sequence with CDS and Gene annotations. It is possible to save the gene model(s) used for the analysis when the option "Learn one gene model" was selected earlier. This model can then be reused to annotate other input sequences by setting the "Model Training" option to "Use a previously trained model" or "Use a previously trained model and use its default parameters".

12.2 Annotate with BLAST

The **Annotate with BLAST** tool allows you to annotate a DNA sequence using a set of either protein reference sequences or nucleotide sequences. This tool can be used on sequences without any pre-existing annotations: it is not necessary to annotate the DNA sequences with genes or coding regions.

The tools can be used for various purposes, e.g. transferring annotations from a known reference, annotate the presence of AMR or virulence markers in a genome, or to filter contigs or sequences based on the presence of a set of genes.

If the reference sequences are protein sequences, the **Annotate with DIAMOND** tool may be used instead and is a faster option.

If the input sequences are already annotated with CDS annotations, it is also possible to use the **Annotate CDS with Best BLAST Hit** and **Annotate CDS with Best DIAMOND Hit** tools - see section 12.4 for more information.

To start the analysis, go to:

> **Tools** | **Microbial Genomics Module** () | **Functional Analysis** () | **Annotate with BLAST** ()

The first wizard step (figure 12.2), specifies the reference and search parameters.

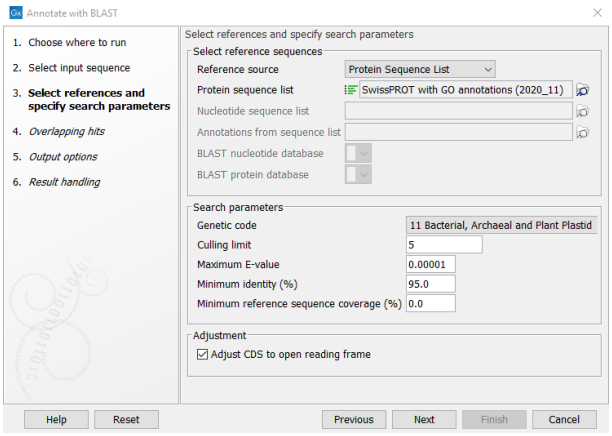


Figure 12.2: *Selecting references and specifying search parameters*

The following sources can be used to annotate the input sequences:

- **Protein sequence list**. The nucleotide input query will be searched against the sequences in the protein sequence list. The nucleotide input will be translated using the chosen genetic code. If the reference protein sequence list contains metadata, this metadata will be transferred to the resulting annotations on the input query sequence.
- **Nucleotide sequence list**. The nucleotide input query will be searched against the sequences in the nucleotide sequence list. If the reference nucleotide sequence list contains metadata, this metadata will be transferred to the resulting annotations on the input query sequence.
- **CDS Annotations (blastx)**. This option uses a nucleotide sequence source with existing annotations as a source. All annotations are extracted, and translated to a protein

database, which is searched similar to the **Protein sequence list** option. All qualifiers on the detected source annotations are transferred to the input query sequence.

- **All Annotations (blastn)**. This option uses a nucleotide sequence source with existing annotations as a source. All annotations are extracted and searched similar to the **Nucleotide sequence list** option. All qualifiers on the detected source annotations are transferred to the input query sequence.
- **BLAST nucleotide database**. BLAST databases can be created using the **Create BLAST database** tool. This option works similar to the **Nuclotide sequence list** option, but can be faster, since the database can be reused. When using this option, the name and description of detected reference sequences are transferred to the input query sequence.
- **BLAST protein database**. BLAST databases can be created using the **Create BLAST database** tool, or downloaded using the **Download BLAST database** tool. This option works similar to the **Protein sequence list** option, but can be faster, since the database can be reused. When using this option, the name and description of detected reference sequences are transferred to the input query sequence.

As can be seen above, metadata (such as GO terms and taxonomy information) is handled differently depending on the database source:

- **Protein / nucleotide sequence list**. The sequence list may contain metadata, which can be inspected in the table view of the sequence list. Such metadata is transferred to the annotations created by this tool.
- **CDS / all annotations**. Annotations are transferred together with any metadata qualifiers the annotations contain.
- **BLAST protein / nucleotide database**. These database types are used for fast annotation with reference sequences and do not allow for metadata. If you require annotation with metadata, for instance when using an RNAcentral database with GO terms in order to build a functional profile, this option can not be used. Instead, the sequence list option must be used, even though it is slightly slower.

The search parameters can be modified using the following settings:

- **Genetic code**. The genetic code used when translating the nucleotide sequences before searching against the protein references.
- **Maximum E-value**. Maximum expectation value (E-value) threshold for accepting hits.
- **Minimum identity (%)**. The minimum percent identity for a hit to be accepted. The percent identity is calculated based on the number of amino acid matches when using protein reference sequences (blastx), and based on the number of nucleotide matches when using nucleotide reference sequences (blastn). Notice. when annotating with a Protein sequence list of clustered sequences such as UniRef50, this should be lowered depending on the level of clustering in the database.
- **Minimum reference sequence coverage (%)**. The minimum length fraction of the reference sequence that must be matched. Notice: this is length fraction per hit (HSP), and should be kept low when searching for non-contiguous matches.

Adjustment can be made to the annotation hits by the following setting:

- **CDS adjustment**. The found annotation hits will be adjusted to begin with a start codon, end with a stop codon and not contain any stop codons in between. The adjustment can extend the annotation to up to 110 percent of the length of the reference gene and will not be shorter than 90 percent of the reference gene length. The frame of the translation may change from the original alignment.

The next step (figure 12.3), determines how to handle when multiple overlapping hits are found on the input query sequence.

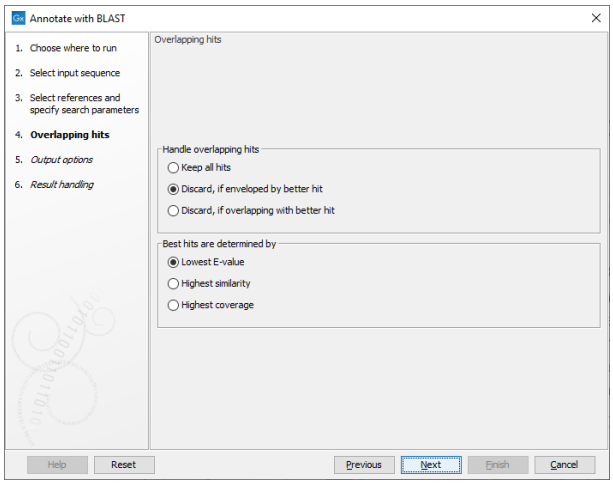


Figure 12.3: *Settings for handling overlapping hits*

The following options are available:

- **Keep all hits**. all hits that meet the search criteria are annotated on the input query sequence.
- **Discard, if enveloped by better hit**. If a hit covers the same region or part of the same region as a better hit, it is discarded.
- **Discard, if overlapping with better hit**. If a hit overlaps the same region as a better hit, it is discarded.

Best hits are determined by:

- **Lowest E-value**. hits with the lowest E-value are kept. Ties are resolved by highest similarity, subsequently highest coverage.
- **Highest similarity**. hits with the highest similarity are kept. Ties are resolved by lowest E-value, subsequently highest coverage.
- **Highest coverage**. hits with the highest coverage are kept. Ties are resolved by lowest E-value, subsequently highest similarity.

The output options step (figure 12.4), has the following options:

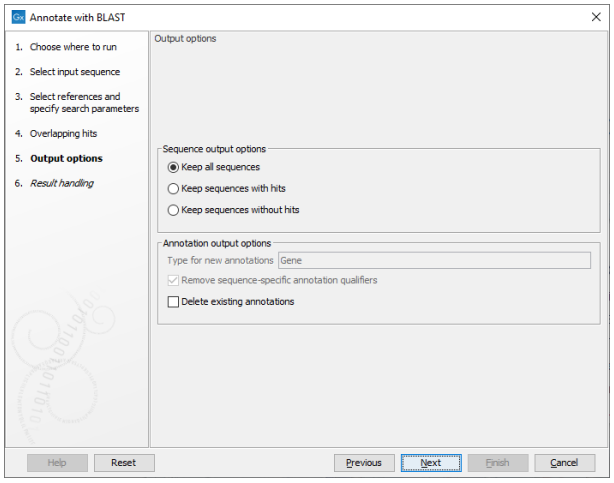


Figure 12.4: *Specifying output options*

- **Type for new annotations**. When using a protein database as source, all new annotations will be of type 'CDS'. However, when using a nucleotide sequence list, or a nucleotide sequence BLAST database, there is no general annotation type to apply. The default output annotation type will be 'Gene', but this can be customized if necessary.
- **Remove sequence-specific annotation qualifiers**. Annotation qualifiers such as 'translation' and 'codon_start' may no longer be accurate on the new annotations. This option removes such qualifiers.
- **Delete existing annotations**. Existing annotations on the input sequences will not be copied to the output sequences.

The following sequence output options are available:

- **Keep all sequences**
- **Keep sequences with hits**. This option can be useful for filtering input sequences for certain regions.
- **Keep sequences without hits**. This option can be useful for comparing sequence lists.

The final step controls which outputs are created. Notice, that reports can be aggregated using the Combine Reports tool.

12.3 Annotate with DIAMOND

The **Annotate with DIAMOND** tool allows you to annotate a DNA sequence using a set of known protein reference sequences. This tool can be used on sequences without any pre-existing annotations: it is not necessary to annotate the DNA sequences with genes or coding regions. For more information about the DIAMOND aligner, see section 12.5.

The tools can be used for various purposes, e.g. transferring annotations from a known reference, annotate the presence of AMR or virulence markers in a genome, or to filter contigs or sequences based on the presence of a set of genes.

For annotating DNA sequences from a set of non-coding reference sequences, the **Annotate with BLAST** tool may be used instead. However, the **Annotate with DIAMOND** tool is in general the fastest option when working with coding regions.

If the input sequences are already annotated with CDS annotations, it is also possible to use the **Annotate CDS with Best BLAST Hit** and **Annotate CDS with Best DIAMOND Hit** tools - see section 12.4 for more information.

To start the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Functional Analysis** () | **Annotate with DIAMOND** ()

The first wizard step (figure 12.5), specifies the reference and search parameters.

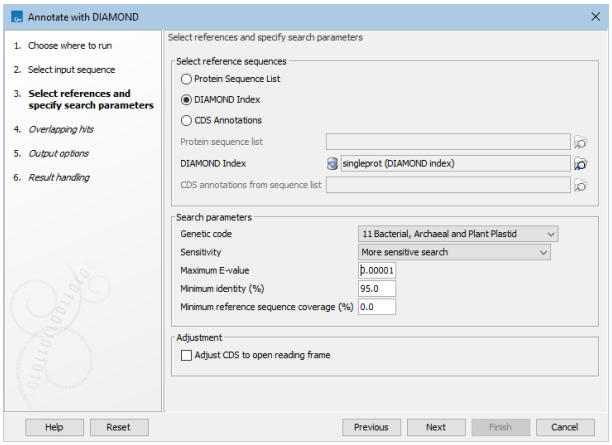


Figure 12.5: *Selecting references and specifying search parameters.*

The following sources can be used to annotate the input sequences:

- **Protein Sequence List**. The nucleotide input query will be searched against the sequences in the protein sequence list. The nucleotide input will be translated using the chosen genetic code. If the reference protein sequence list contains metadata, this metadata will be transferred to the resulting annotations on the input query sequence.
- **DIAMOND Index**. A DIAMOND index can be created using the Create DIAMOND Index tool. This works similar to the Protein Sequence List option, but can be faster since the index can be reused. When using the DIAMOND index, the name and description of detected reference sequences are transferred to the input query sequence.
- **CDS Annotations**. This option uses a nucleotide sequence source with existing annotations as a source. All CDS annotations are extracted and translated to a protein database, which is searched similar to the previous options. All qualifiers on the detected source annotations are transferred to the input query sequence.

As can be seen above, metadata (such as GO terms and taxonomy information) is handled differently depending on the database source:

- **Protein sequence list**. The sequence list may contain metadata, which can be inspected in the table view of the sequence list. Such metadata is transferred to the annotations created by this tool.

- **DIAMOND Index**. If a DIAMOND index was created from a protein sequence list containing metadata, the original metadata will be transferred to the annotations created by this tool.
- **CDS annotations from sequence list**. Annotations are transferred together with any metadata qualifiers the annotations contain.

The search parameters can be modified using the following settings:

- **Genetic code**. The code used when translating the nucleotide sequences before searching against the protein references.
- **Sensitivity**: Select DIAMOND sensitivity:
  - Faster search: The fastest search
  - Fast search: Designed for finding hits of >90% identity
  - Standard search: Designed for finding hits of >60% identity
  - Mid-sensitive search: More sensitive than standard search and faster than sensitive search.
  - Sensitive search: Designed for finding hits of >40% identity
  - More sensitive search: Designed for finding hits of >40% identity with some motif masking disabled
  - Very sensitive search: Designed for finding hits of 40% identity
  - Most sensitive search: The most sensitive search
- **Maximum E-value**. Maximum expectation value (E-value) threshold for saving hits.
- **Minimum identity (%)**. The minimum percent amino acid identity for a hit to be accepted. Notice: when annotating with a Protein sequence list of clustered sequences such as UniRef50, this should be lowered depending on the level of clustering in the database.
- **Minimum reference sequence coverage (%)**. The minimum length fraction of the reference sequence that must be matched. Notice: this is length fraction per hit (HSP), and should be kept low when searching for non-contiguous matches.

Adjustment can be made to the annotation hits by the following setting:

- **CDS adjustment**: The found annotation hits will be adjusted to begin with a start codon, end with a stop codon and not contain any stop codons in between. The adjustment can extend the annotation to up to 110 percent of the length of the reference gene and will not be shorter than 90 percent of the reference gene length. The frame of the translation may change from the original alignment.

The next step (figure 12.6), determines how to handle when multiple overlapping hits are found on the input query sequence.

The following options are available:

- **Keep all hits**: all hits that meet the search criteria are annotated on the input query sequence.

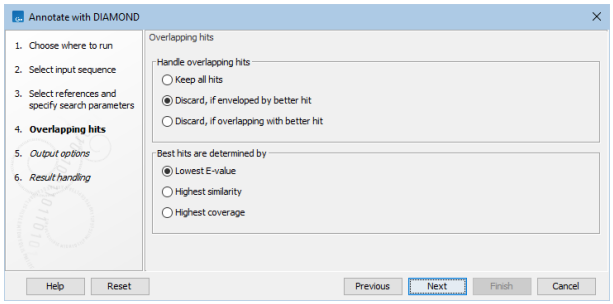


Figure 12.6: *Settings for handling overlapping hits.*

- **Discard, if enveloped by better hit**: If a hit covers the same region or part of the same region as a better hit, it is discarded.
- **Discard, if overlapping with better hit**: If a hit overlaps the same region as a better hit, it is discarded.

Best hits are determined by:

- **Lowest E-value**: hits with the lowest E-value are kept. Ties are resolved by highest similarity, subsequently highest coverage.
- **Highest similarity**: hits with the highest similarity are kept. Ties are resolved by lowest E-value, subsequently highest coverage.
- **Highest coverage**: hits with the highest coverage are kept. Ties are resolved by lowest E-value, subsequently highest similarity.

The output options step (figure 12.7), has the following options:

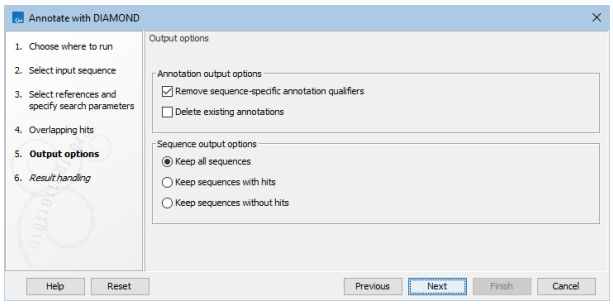


Figure 12.7: *Specifying output options.*

- **Remove sequence-specific annotation qualifiers**. Annotation qualifiers such as 'translation' and 'codon_start' may no longer be accurate on the new annotations. This option removes such qualifiers.
- **Delete existing annotations**. Existing annotations will not be copied to the output sequences.

The following sequence output options are available:

- **Keep all sequences**
- **Keep sequences with hits**. This option can be useful for filtering input sequences for certain regions.
- **Keep sequences without hits**. This option can be useful for comparing sequence lists.

The final step controls which outputs are created. Notice, that reports can be aggregated using the Combine Reports tool.

12.4 Annotate CDS with Best BLAST Hit

The **Annotate CDS with Best BLAST Hit** tool will allow you to annotate a set of contigs containing CDS annotations with their best BLAST hit.

To start the analysis, go to:

> **Tools** | **Microbial Genomics Module** () | **Functional Analysis** () | **Annotate CDS with Best BLAST Hit** ()

Several parameters are available:

- **Genetic code**. The genetic code used for translating CDS to proteins.
- **BLAST database**. A protein BLAST database. Popular BLAST protein databases can be downloaded using the Download BLAST Database tool or created using a the Create BLAST Database tool.
- **Maximum E-value**. Maximum expectation value (E-value) threshold for saving hits.

Metadata from the sequences used to create the BLAST database (such as GO terms or taxonomy information) will not be transferred by this tool. If metadata is relevant, consider using the **Annotate CDS with Best DIAMOND Hit** tool instead.

Note that choosing a very large BLAST database with millions of sequences (e.g. the nt, nr and refeseq_protein databases from the NCBI) will slow down the algorithm considerably, especially when there are many CDS in the input. Therefore, we recommend to use a medium-sized database such as "swissprot". In the wizard, you can choose between databases stored locally () or remotely on the server (). If you create a workflow that you plan to run on a server, you should avoid locking the BLAST database parameter as the chosen database may not exist on the server.

If you select **Create Report**, the tool will create a summary report table. The report is divided in three parts:

- **Input**. Contains information about the size of the contigs and CDS used as input.
- **BLAST database**. The protein BLAST database used in the search, together with its description, location, and size.
- **Output**. The total number (and percent) of CDS that were annotated with their best BLAST hit.

The tool will output a copy of the input file containing the following fields when a hit for a CDS is found (figure 12.8):

- **BLAST Hit**. Accession number of the best BLAST Hit in the BLAST database.
- **BLAST Hit Description**. Description of the matching protein, as present in the BLAST database.
- **BLAST Hit E-value**. The E-value of the match.

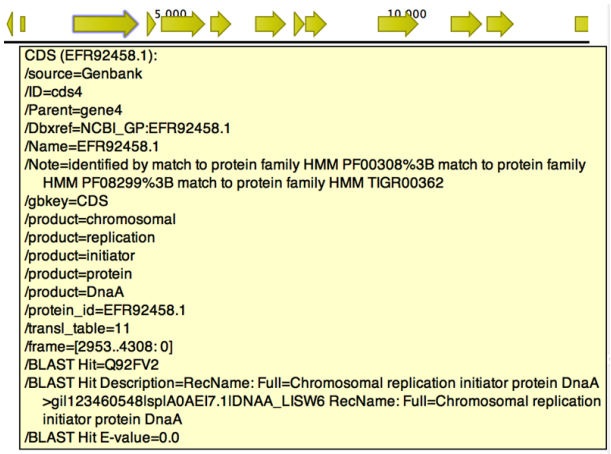


Figure 12.8: *BLAST Best Hit annotations added to gene cds4 of h. pylori.*

The tool can also output an annotation table summarizing information about the annotations added to the sequence list.

12.5 Annotate CDS with Best DIAMOND Hit

Annotate CDS with Best DIAMOND Hit allows you to annotate a set of contigs containing CDS annotations with their best DIAMOND hit. This tool is particularly useful for large data sets, as an alternative to Annotate CDS with Best BLAST Hit.

DIAMOND is a sequence aligner for protein and translated DNA searches, designed for high performance analysis of big sequence data, see https://github.com/bbuchfink/diamond. The key features are:

- Pairwise alignment of proteins and translated DNA at 500x-20,000x speed of BLAST.
- Frameshift alignments for long read analysis.
- Low resource requirements and suitable for running on standard desktops or laptops.

To start the analysis, go to:

> **Tools** | **Microbial Genomics Module** () | **Functional Analysis** () | **Annotate CDS with Best DIAMOND Hit** ()

Select the CDS-annotated contigs to be annotated with DIAMOND hits.

In the **Parameters** dialog page (figure 12.9), set the following

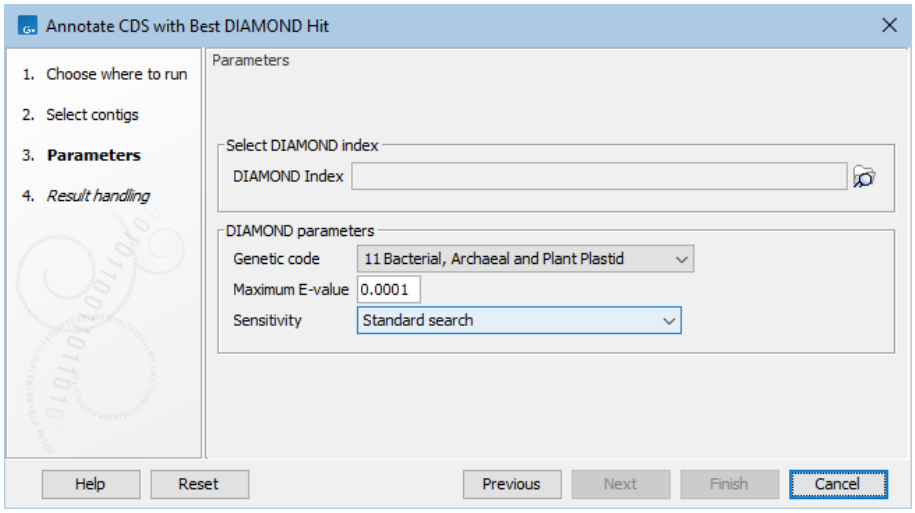


Figure 12.9: *Annotate CDS with Best DIAMOND Hit parameters.*

- **DIAMOND Index**. Select the relevant indexes.

  Indexes can be generated by downloading a database with the Download Protein Database tool (section 17.1) and building and index using the Create DIAMOND Index tool (section 17.4).

- **Genetic code**. The genetic code used for translating CDS to proteins.

- **Maximum E-value**. Maximum expectation value (E-value) threshold for saving hits.

- **Sensitivity**: Select DIAMOND sensitivity:

  - Faster search: The fastest search
  - Fast search: Designed for finding hits of >90% identity
  - Standard search: Designed for finding hits of >60% identity
  - Mid-sensitive search: More sensitive than standard search and faster than sensitive search.
  - Sensitive search: Designed for finding hits of >40% identity
  - More sensitive search: Designed for finding hits of >40% identity with some motif masking disabled
  - Very sensitive search: Designed for finding hits of 40% identity
  - Most sensitive search: The most sensitive search

The tool will output a copy of the input file with the DIAMOND Hit annotations. The tool can also output an annotation table summarizing information about the annotations added to the sequence list. Finally it is possible to generate a report containing information about the input file, the DIAMOND database and the amount of CDS annotated with a DIAMOND hit.

If a DIAMOND index was created from a protein sequence list containing metadata (such as GO terms or taxonomy information), the original metadata will be transferred to the annotations created by this tool.

12.6 Annotate CDS with Pfam Domains

The **Annotate CDS with Pfam Domains** tool will allow you to annotate a set of contigs containing CDS annotations with Pfam and GO terms. To start the analysis, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Functional Analysis** ( ) | **Annotate CDS with Pfam Domains** ( )

The following parameters are available:

- **Genetic code**. The genetic code used for translating CDS to proteins.
- **Pfam database**. The Pfam database. This database can be downloaded using the "Download Pfam Database" tool.
- **Use profile's gathering cutoffs**. Use cutoffs specifically assigned to each family by the curator instead of manually assigning the Significance cutoff.
- **Significance cutoff**. The E-value (expectation value) describes the number of hits one would expect to see by chance when searching a database of a particular size.
- **Remove overlapping matches from the same clan**. Perform post-processing of the results where overlaps between hits are resolved by keeping the hit with the smallest e-value.
- **GO database**. The GO database, used to map between Pfam domains and GO terms. The GO database can be downloaded using the Download GO Database tool ((see section 17.2). If the database is not specified, no GO annotation will be added.
- **GO subset**. A subset of the GO database. Since many GO terms are too general or too specific, several meaningful subsets of GO terms are provided. See https://geneontology.org/docs/download-ontology/.

If you select **Create report**, the tool will create a summary report table. The report is divided in three parts

- **Input**. Contains information about the size of the contigs and CDS used as input.
- **Output**. The total number (and percent) of CDS that were annotated with a Pfam domain or a GO term, as well as the total number of Pfam domains and GO terms added.
- **Pfam database**.The Pfam database used in the search together with its version and size.
- **GO database**. The GO database (or subset) used in the search together with its version, size, and the number of Pfam domains mapping to at least one term.

The tool will output a copy of the input file containing Pfam annotations when a Pfam domain was found in a CDS, as shown in figure 12.10. The annotation contains the following fields:

- **Description**. A description of the Pfam domain.
- **Accession**. The accession number of the Pfam domain.
- **Clan**. The clan that the domain belong to (if any).

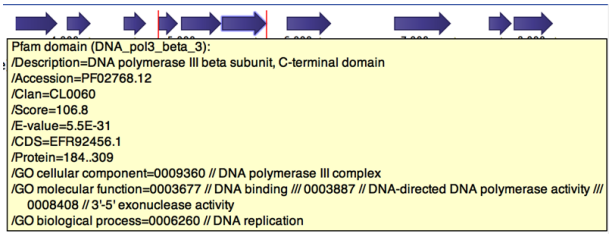


Figure 12.10: *Pfam and GO annotations added to gene cds4 of h. pylori.*

- **Score**. The score
- **E-value**. The E-value of the match.
- **CDS**. The CDS that contains this domain.
- **Protein**. The protein region (in aa coordinates) that encodes for the domain.
- **GO cellular component**. GO terms of the cellular component domain which are related to the Pfam domain.
- **GO molecular function**. GO terms of the molecular function domain which are related to the Pfam domain.
- **GO biological process**. GO terms of the biological process domain which are related to the Pfam domain.

The tool can also output an annotation table summarizing information about the annotations added to the sequence list.

12.7 Build Functional Profile

To compute the number of reads in a sample mapping to regions involved with Pfam domains, or BLAST or DIAMOND hits, you can run the Build Functional Profile tool by going to:

> **Tools** | **Microbial Genomics Module** ( ) | **Functional Analysis** ( ) | **Build Functional Profile** ( )

In the first wizard (figure 12.11), select the read mapping for which you want to build the functional profile.

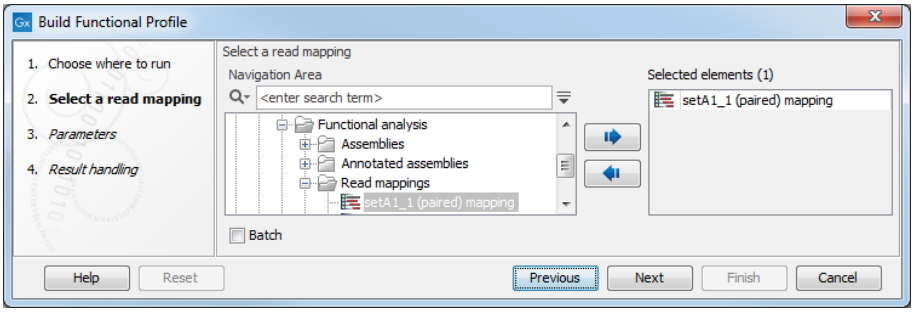


Figure 12.11: *Select a read mapping.*

The parameters that can be set are seen in figure 12.12:

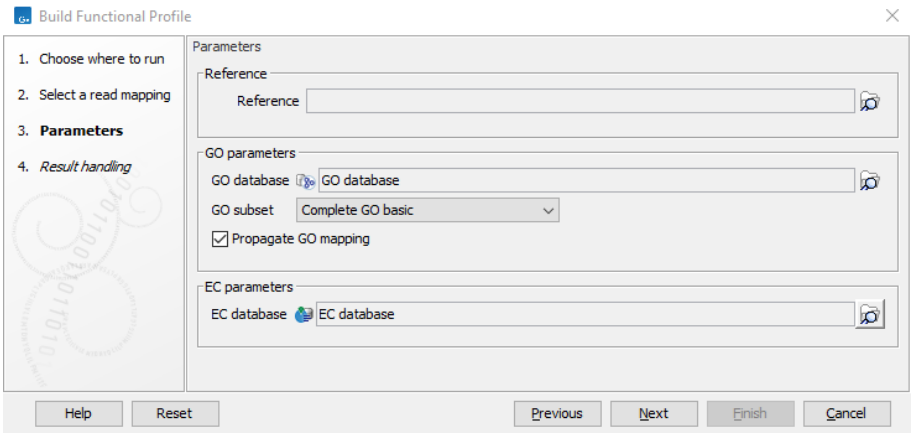


Figure 12.12: *Specify a reference, a GO database and an EC database.*

- **Reference**. A reference set of contigs annotated with Pfam domains, BLAST, or DIAMOND hits. If the read mapping contains an annotated genome, this parameter is optional.
- **GO database**. The GO database. If the reference contains Pfam domains, this database can be used to map from Pfam domains to GO terms. If the BLAST or DIAMOND hits contain GO-terms annotations, this will be also matched against the database and appear in the GO abundance table output. The GO database can be downloaded using the Download Ontology Database tool (see section 17.2).
- **GO subset**. A subset of the GO database. Since many GO terms are too general or too specific, several meaningful subsets of GO terms are provided. See `https://geneontology.org/docs/download-ontology/`.
- **Propagate GO mapping**. When selected, GO terms are mapped to all their ancestor terms. For example, the Pfam domain "CutC" maps to the GO term "0005507 // copper ion binding". If **Propagate GO mapping** is enabled, the tool would also map to more general GO terms such as "0055070 // copper ion homeostasis", "0055076 // transition metal ion homeostasis", and "0065007 // biological regulation".
- **EC database**. The EC database. If the reference contains BLAST or DIAMOND domains, this database can be used to map EC terms. The EC database can be downloaded using the Download Ontology Database (see section 17.2).

You can then select which output elements should be generated figure 12.13.

- **Create Pfam functional profile**. Abundance table obtained by counting reads overlapping Pfam domains.
- **Create GO functional profile**. Abundance table obtained by counting reads overlapping GO terms. Note that a database must be specified in order to build a GO functional profile. For BLAST and DIAMOND hits, the GO-terms specified on the annotations are used, but preexisting GO annotations on pfam domains are ignored by this tool.
- **Create EC functional profile**. Abundance table obtained by counting reads overlapping EC terms. Note that a database must be specified in order to build an EC functional profile.

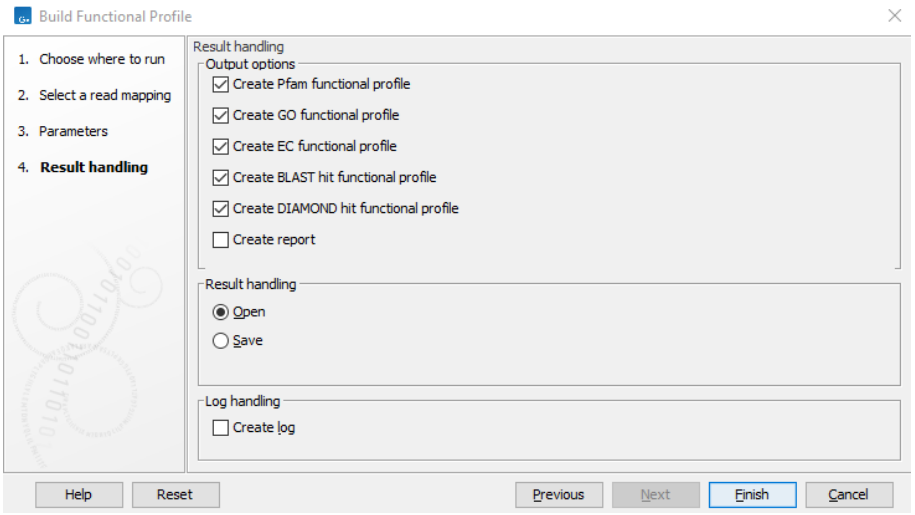


Figure 12.13: *Specify what type of output you want the tool to generate.*

- **Create BLAST hit functional profile**. Abundance table obtained by counting reads overlapping BLAST hits.
- **Create DIAMOND hit functional profile**. Abundance table obtained by counting reads overlapping DIAMOND hits.
- **Create Report**. A report stating statistics about the input reference contigs and read mapping, as well as the number of matches to each feature.

The resulting functional abundance tables store the number of reads corresponding to each Pfam domain, GO term, EC number, best BLAST hit or best DIAMOND hit.

12.7.1 Functional profile abundance table

The functional profile abundance table displays the names of the function, along with their clan, a combined abundance. The table can be visualized using the Stacked bar charts and Stacked area charts function, as well as the Sunburst charts.

- **Table view** ( ) (figure 12.14)

  The table displays the following columns:

  - **ID**: internal ID the abundance tables use for ordering the samples. IDs are unique, while Names are not necessarily, so that when merging abundance tables taxa with the same ID will be combined.
  - **Name**: the name of the taxon, specified by the reference database or the NCBI taxonomy. If the name contains the text "(Unknown)", it indicates that this taxon corresponds to a higher-level node in the taxonomy, and that this node had a significant amount of reads associated to ancestor taxons that are present in the database but were disqualified. This indicates that there was some organism in the sample for which there is no exactly matching reference in the database, but is most likely closely related to this taxon. If the name does not contain the text "(Unknown)", it means that the sample contains this exact taxon, which is present in the database.

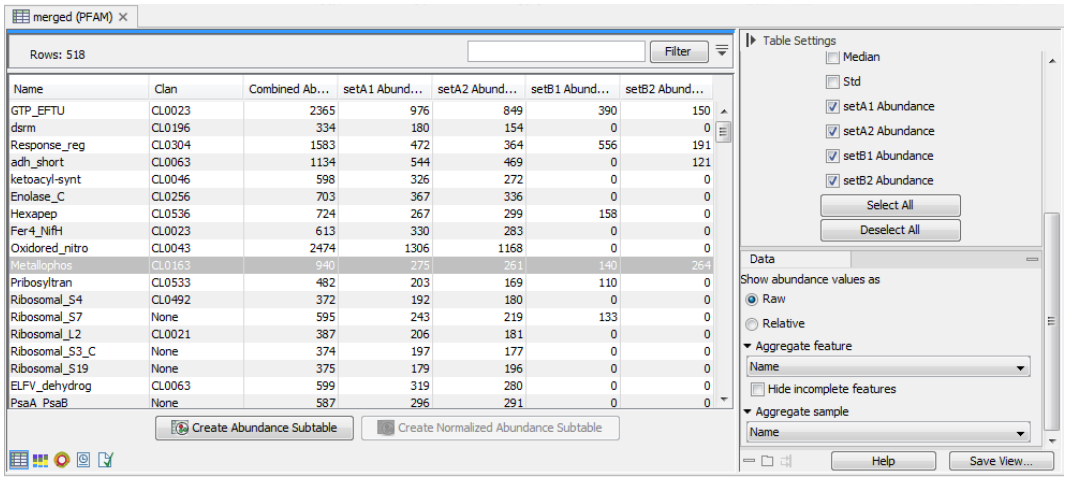


Figure 12.14: *Functional profile abundance table.*

- **Clan**: a collection of related Pfam entries. The relationship may be defined by similarity of sequence, structure or profile-HMM.
- **Combined Abundance**: total number of reads for the function across all samples
- **Min**, **Max**, **Mean**, **Median** and **Std**: respectively minimum, maximum, mean, median and standard deviation of the number of reads for the fucntion across all samples
- **Abundance for the sample**: number of reads for each sample

In the right side panel, under the tab Data, you can switch between raw and relative abundances (relative abundances are computed as the ratio between the coverage for a function in a specific sample and the amount of coverage in the sample). You can also combine absolute counts and relative abundances by selecting the Clan level in the **Aggregate feature** drop-down menu.

Finally, if you have previously annotated your table with Metadata (see section 7.9), you can **Aggregate sample** by the groups previously defined in your metadata table. This is useful when for example analyzing replicates from the same sample origin.

Under the table, the following actions are available:

- **Create Abundance Subtable** will create a table containing only the selected rows.
- **Create Normalized Abundance Subtable** will create a table with all rows normalized on the values of a single selected row. The row used for normalization will disappear from the new abundance table. The normalization scales the abundance table linearly, where the scaling factor is calculated by determining the average abundance across all samples and for each sample scale it to the average for the reference. Note that to be enabled, the selected row for normalization can only have non null abundance values. If you have zero values in some samples for the control, you will need to generate a new abundance table where these samples are not present. If the abundance table is obtained from merging single-sample abundance tables, then the merge should be redone excluding the samples with zero control read counts.

- **Stacked Bar Chart and Stacked Area Chart** ( )

  Choose which chart you want to see using the drop down menu in the upper right corner of the side panel. The charts can be scaled by percentage, where all bars have the same height of 100%, or counts, where the bar heights are proportional to the number of counts.

In the Stacked Bar (figure 12.15) and Stacked Area Charts (figure 12.16), the metadata can be used to aggregate groups of columns (samples) by selecting the relevant metadata category in the right hand side panel. Also, the data can be aggregated at any taxonomy level selected. The relevant data points will automatically be summed accordingly.

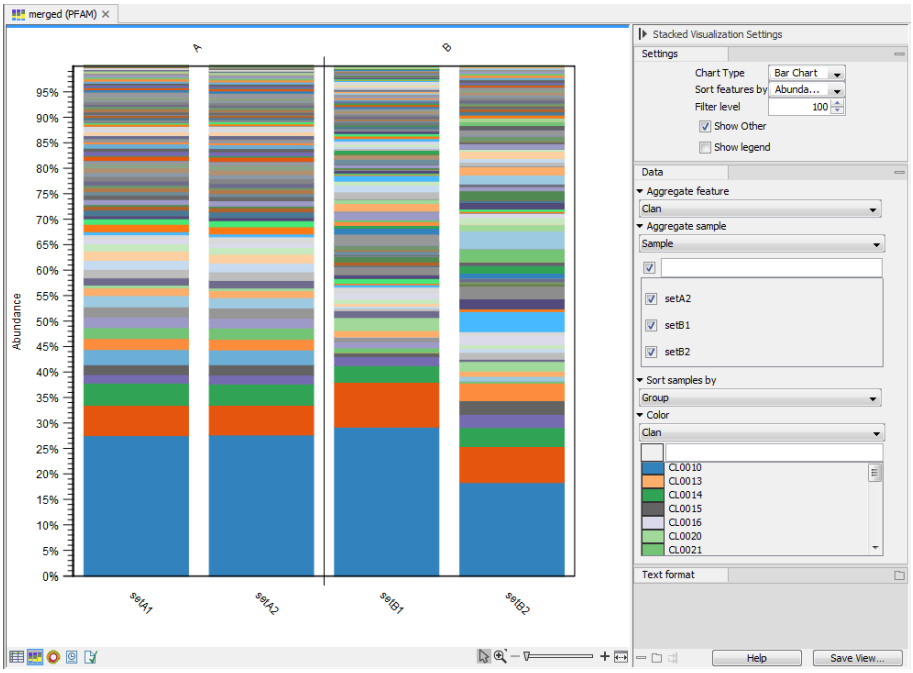


Figure 12.15: *Stacked bar chart.*

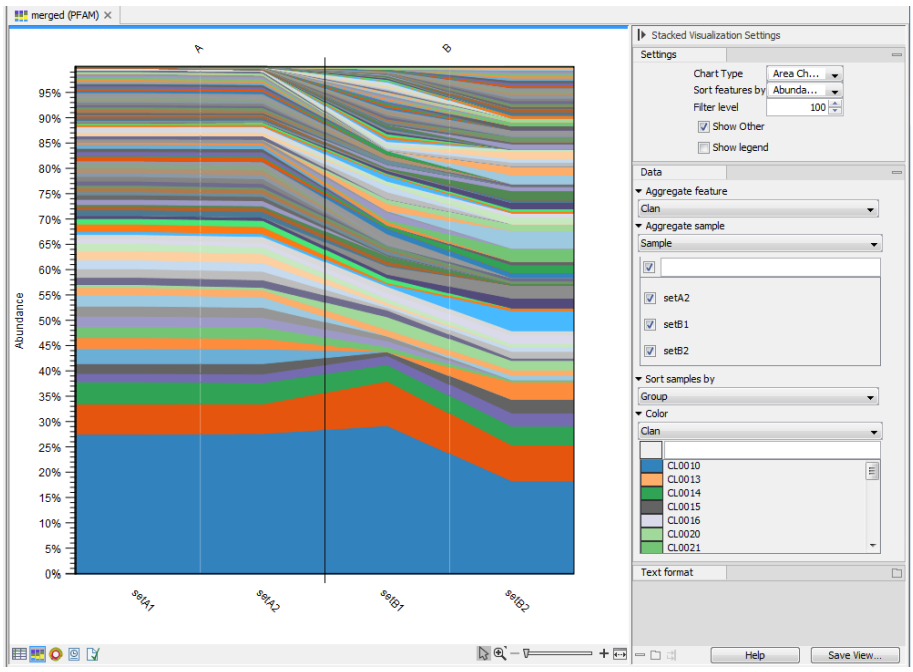


Figure 12.16: *Stacked area chart.*

Holding the pointer over a colored area in any of the plots will result in the display of the corresponding taxonomy label and counts. **Filter level** allows to modify the number of features to be shown in the plot. For example, setting the value to 10 means that the 10 most abundant features of each sample will be shown in all columns. The remaining

features are grouped into "Other", and will be shown if the option is selected in the right hand side panel. One can select which taxonomy level to color, and change the default colors manually. Colors can be specified at the same taxonomy level as the one used to aggregate the data or at a lower level. When lower taxonomy levels are chosen in the data aggregation field, the color will be inherited in alternating shadings. It is also possible to sort samples by metadata attributes, and to show groups of samples without collapsing their stacks, as well as change the label of each stack or group of stacks. Features can be sorted by "abundance" or "name" using the drop down menu in the right hand side panel. Using the bottom right-most button (**Save/restore settings** ( )), the settings can be saved and applied in other plots, allowing visual comparisons across analyses.

- **Zoomable Sunbursts** ( ) The Zoomable Sunburst viewer lets the user select how many taxonomy level counts to display, and which level to color. Lower levels will inherit the color in alternating shadings. Taxonomy and relative abundances (the ratio between the coverage of the species in a specific sample and the total amount of coverage in the sample) are displayed in a legend to the left of the plot when hovering over the sunburst viewer with the mouse. The metadata can be used to select which sample or group of samples to show in the sunburst (figure 12.17).

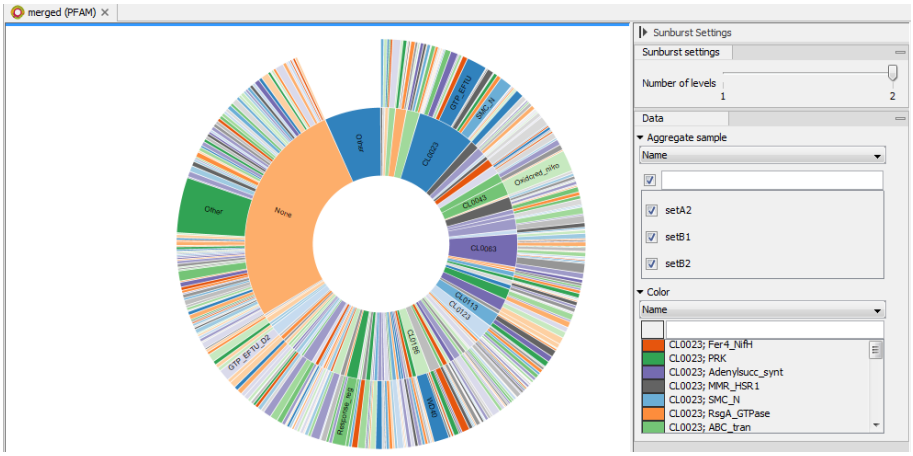


Figure 12.17: *Sunburst view.*

Clicking on a lower level field will render that field the center of the plot and display lower level counts in a radial view. Clicking on the center field will render the level above the current view the center of the view.

12.8 Infer Functional Profile

This tool is currently in beta. Feedback on this plugin is welcome - please get in touch by email (ts-bioinformatics@qiagen.com) and let us know how it can be improved.

For OTU abundance tables it is possible to infer an approximate functional profile using the Infer Functional Profile (beta) tool. In order to run this tool you need a PICRUSt2 Multiplication Table [Douglas et al., 2020] which may be imported using the Import PICRUSt2 Multiplication Table (beta) 17.6 and optionally an EC database that may be downloaded using the Download Ontology Database 17.6 tool.

To infer a functional profile from an OTU table, go to

Tools | Microbial Genomics Module ( ) | Functional Analysis ( ) | Infer Functional Profile (beta) ( )

In the first wizard (figure 12.18), select the OTU table for which you want to build the functional profile. Note that the OTU table must have OTU sequences.

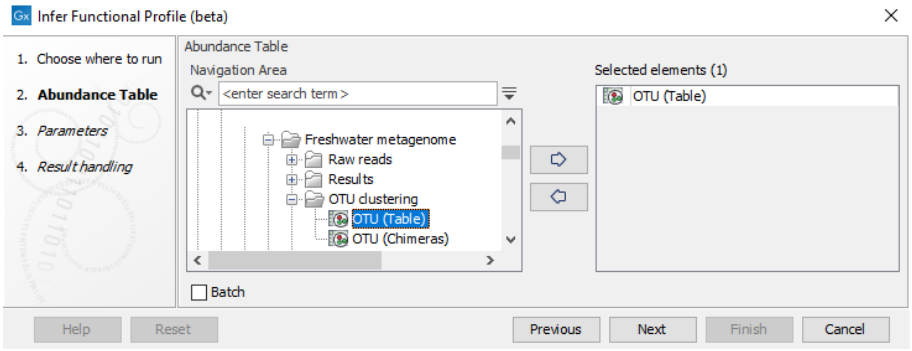


Figure 12.18: *Select an OTU abundance table.*

In the second step of the wizard (figure 12.19) the terms for which to produce a functional abundance profile can be selected. Note that when selecting to create an EC abundance profile, an additional EC database is required.

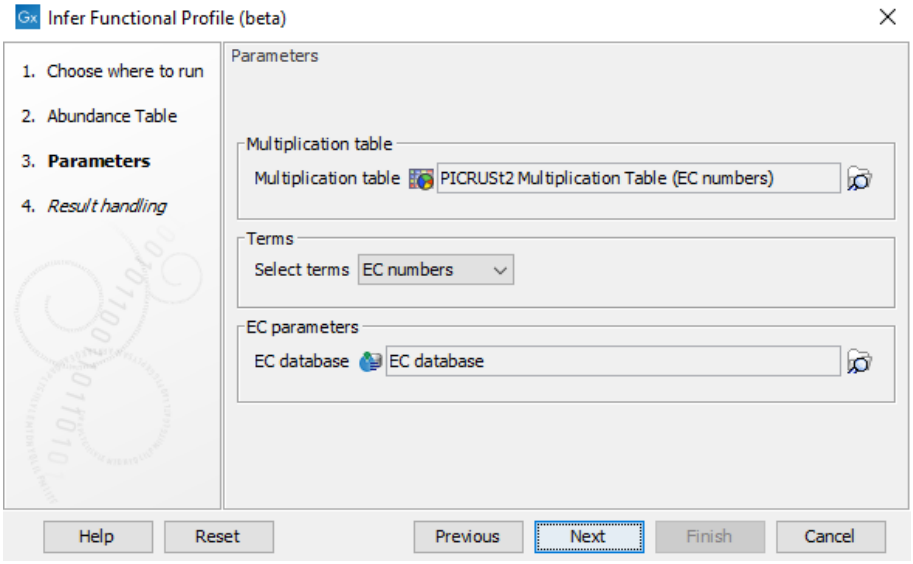


Figure 12.19: *Specify the functional terms for which a functional abundance table shall be inferred.*

The resulting functional abundance tables store the inferred number of reads corresponding to each of the selected terms in a separate table.

Inference of functional abundances from 16S/ITS data PICRUSt2 Multiplication Tables can be imported using the Import PICRUSt2 Multiplication Table (beta) tool. The multiplication tables contain kmer frequency profiles, associated rRNA copy numbers and term multipliers, i.e. how often a certain functional term is encountered on the genomic sequence of the associated rRNA sequence. The Infer Functional Profile (beta) algorithm works by comparing kmer frequency profiles for each identified OTU with the stored kmer frequency profiles in a PICRUSt2 Multiplication Table to find the nearest neighbor to the OTU under consideration in the Multiplication Table. For this nearest neighbor, both the rRNA copy number and term multiplication numbers are available.

From a single OTU the predicted term multiplicity is obtained by dividing the read count for the OTU by the identified rRNA copy number and multiplying it by the identified term multiplication number. To obtain the final inferred term read count, the individual predictions for all OTUs are summed up per term.

The Infer Functional Profile (beta) algorithm is inspired by two published methods PICRUSt2 [Douglas et al., 2020] and Piphillin [Narayan et al., 2020]. Note that PICRUSt2 [Douglas et al., 2020] and Piphillin [Narayan et al., 2020] do not use kmer frequency profiles but alignments (and for PICRUSt2 optimal tree positioning of a reference) for the identification of the nearest neighbor(s), which typically have a higher precision but are also slower to compute. Typically, it is not expected that high precision is required for the identification of the nearest neighbors as most OTUs are most likely not represented exactly in the database and a close neighbor is typically good enough. While this is true for well-represented species, it has been shown in [Douglas et al., 2020] that only a single nearest neighbor may be a bad predictor for the rRNA and term copy numbers. In this respect we expect the Infer Functional Profile (beta) tool to be comparable to Piphillin [Narayan et al., 2020].

12.9 Identify Pathways

The Identify Pathways tool takes a functional abundance table with EC terms or a differential abundance table with EC terms as input and translates these into pathway calls using a pathway database. A pathway database can be obtained with the Download Pathway Database tool, see section 17.3. If the input is an abundance table, the called pathways will correspond to all pathways present in the sample. If the input is a differential abundance table, the called pathways are the pathways that have been up or down regulated between two groups of samples.

The algorithm produces a range of solutions for the pathway calls:

- The **naive solution** where a pathway is called if it contains at least one of the functional terms present in the input table.
- The **minimum solution** where the smallest set of pathways is chosen such that all terms from the input are present in at least one of the chosen pathways. This is similar to MinPath [Ye and Doak, 2009], only that the algorithm used is based on a greedy minimum set cover strategy and thus only finds an approximate solution to the stated minimization problem.
- The **confidence** based solution where each pathway is associated with a confidence call based on randomized evaluations of the minimum set cover strategy. In this way, each pathway call is be associated with a confidence for the presence of that pathway. The naive solution and the minimum solution are the outer goal posts, whereas the confidence based solution gives a smooth metric in-between.

To run the Identify Pathways tool go to

Tools | **Microbial Genomics Module** () | **Functional Analysis** () | **Identify Pathways** ()

Select a functional abundance table or a differential abundance table with EC terms as input and click "Next".

In the **Pathway database** section of the second step of the wizard (figure 12.20), select the required pathway database. A taxonomic range filter for the called pathways can be set to reduce the amount of false positive pathway calls in the case where the metagenomic reads are known to be of a certain type of origin. For example, if the (differential) abundance table has been produced from an OTU table based on ITS regions using Infer Functional Profile (beta), then the taxonomic range would have to be set to **Fungi**. Per default the filter is set to **Disabled** as is appropriate for many whole metagenome and metatranscriptome experiments. Finally, you can choose to include super-pathways in the analysis. This will have an influence on the minimum solution and the confidence scores as super-pathways are constructed of smaller pathways occurring in the pathway database. Since super pathways usually contain a lot of terms, it is more likely that a super-pathway is part of the minimum solution. Also, the super-pathway will tend to have a higher confidence at the cost of a lower confidence for the individual pathways it is composed of

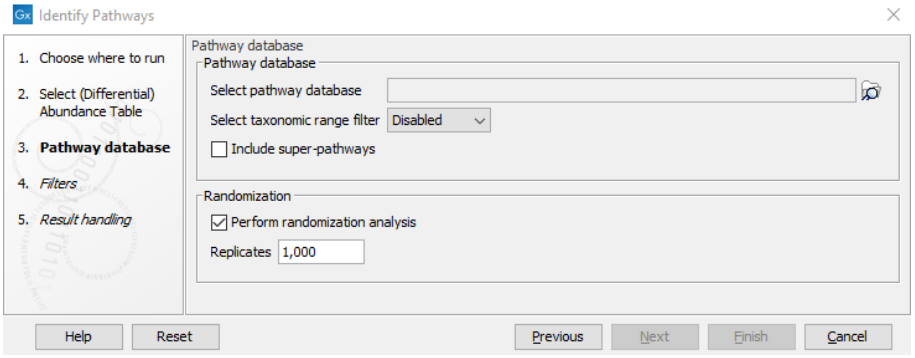


Figure 12.20: *Select the pathway database, taxonomic range and set the randomization parameters.*

In the **Randomization** section of the second wizard step (figure 12.20) it is possible to control the randomization experiment for setting the confidence scores. If **Perform randomization analysis** is selected, the order of pathways in the naive solution is shuffled and the pathways are called sequentially while removing their functional terms until no pathways or no functional terms are left. The number set for Replicates thereby controls how often this is executed and the confidence score becomes the fraction of randomizations in which a pathway is part of the solution. If the setting is deselected an estimate for this number will be given as the confidence of a pathway being present.

In the third wizard step (figure 12.21) it is possible to remove EC terms from the analysis based on the input table.

If the input table is an abundance table, the Abundance table filter section will be relevant. When selecting **Ignore terms with a low abundance value**, EC terms with abundance values below the value given in **Abundance threshold** will be ignored in the pathway calling procedure.

If the input table is a differential abundance table, several filters may be applied, one for each column for a statistical comparison in the differential abundance table. Note that some filters remove EC terms with values **lower** than the specified value in the corresponding field, i.e.

- Max Group Mean
- Absolute fold change
- Absolute log fold change

and other filters remove EC terms with with values **higher** than the specified value in the corresponding field

- P-value
- FDR corrected p-value
- Bonferroni corrected p-value

The filters may be combined freely to achieve the desired level of filtering. It is generally recommended to use a filter on the p-values, either FDR corrected or Bonferroni corrected to remove EC terms whose abundance level does not change between the groups. Based on the remaining terms after filtering, the naive, minimal and confidence based solutions will be calculated.

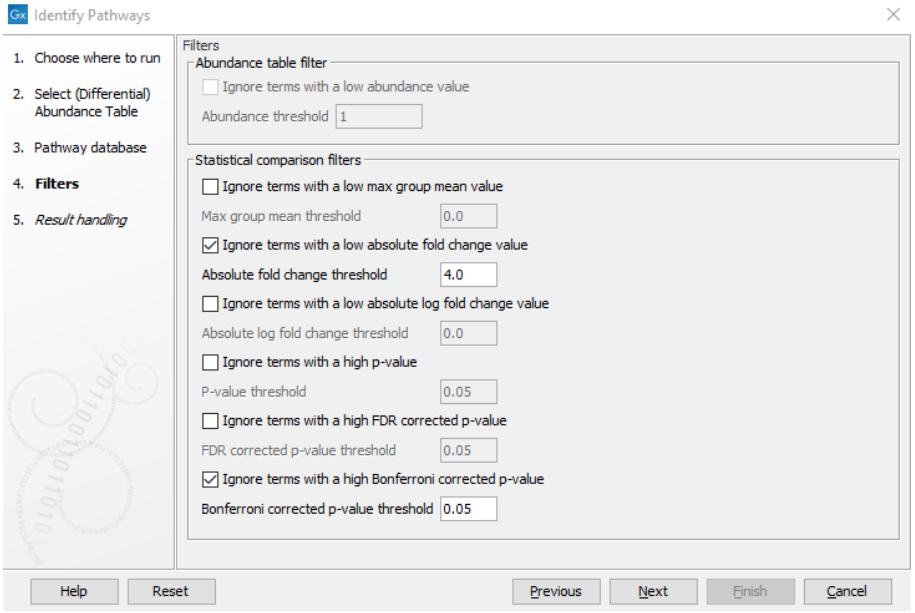


Figure 12.21: *Filter the EC terms based on entries in the abundance table (Abundance table filter) or differential abundance table (Statistical Comparison filter) here shown for a differential abundance table.*

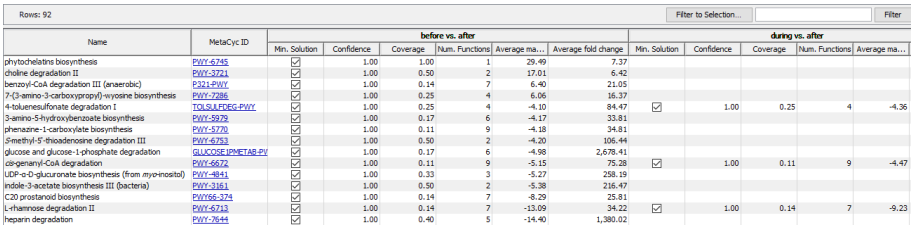
12.9.1 Called Pathways Result

The result of a Identify Pathways run is very similar to the pathway database, see section 17.3.1, in that it has three views:

- The Identified Pathways table ( )
- The Compound table ( )
- The Enzyme table ( )

Note that the latter two are identical to the views in the pathway database, except that the pathways opened from these views are enriched with the data from (differential) abundance tables, see below.

The Identified Pathways Table The result of a Identify Pathways run presented as a table where each row corresponds to a pathway with a pathway name, pathway id and for each sample or comparison, depending on whether an abundance table or a differential abundance table has been used, a number of statistics on the pathway call (figure 12.22).



Rows: 92 — Filter to Selection... — Filter

Name	MetaCyc ID	before vs. after						during vs. after					
		Min. Solution	Confidence	Coverage	Num. Functions	Average ma...	Average fold change	Min. Solution	Confidence	Coverage	Num. Functions	Average ma...	
phytochelatins biosynthesis	PWY-6745	☑	1.00	1.00	1	29.49	7.37						
choline degradation II	PWY-3721	☑	1.00	0.50	2	17.01	6.42						
benzoyl-CoA degradation III (anaerobic)	P321-PWY	☑	1.00	0.14	7	6.40	21.05						
7-(3-amino-3-carboxypropyl)-wyosine biosynthesis	PWY-7286	☑	1.00	0.25	4	6.06	16.37						
4-toluenesulfonate degradation I	TOLSULFDEG-PWY	☑	1.00	0.25	4	-4.10	84.47	☑	1.00	0.25	4	-4.36	
3-amino-5-hydroxybenzoate biosynthesis	PWY-5979	☑	1.00	0.17	6	-4.17	33.81						
phenazine-1-carboxylate biosynthesis	PWY-5770	☑	1.00	0.11	9	-4.18	34.81						
2-methyl-5'-thioadenosine degradation III	PWY-6753	☑	1.00	0.50	2	-4.20	106.44						
glucose and glucose-1-phosphate degradation	GLUCOSE1PMETAB-PWY	☑	1.00	0.17	6	-4.98	2,678.41						
cis-geranyl-CoA degradation	PWY-6672	☑	1.00	0.11	9	-5.15	75.28	☑	1.00	0.11	9	-4.47	
UDP-α-D-glucuronate biosynthesis (from *myo*-inositol)	PWY-4841	☑	1.00	0.33	3	-5.27	258.19						
indole-3-acetate biosynthesis III (bacteria)	PWY-3161	☑	1.00	0.50	2	-5.38	216.47						
C20 prostanoid biosynthesis	PWY66-374	☑	1.00	0.14	7	-8.29	25.81						
L-rhamnose degradation II	PWY-6713	☑	1.00	0.14	7	-13.09	34.22	☑	1.00	0.14	7	-9.23	
heparin degradation	PWY-7644	☑	1.00	0.40	5	-14.40	1,380.02						

Figure 12.22: *The result of the Identify Pathways tool is a table with pathways in the rows and some columns describing the pathway calls for each sample or comparison. Here the result is shown for a differential abundance table.*

There are two general columns to describe the pathways that have been called.

- **Name**: the name of the pathway.
- **MetaCyc ID**: the MetaCyc ID of the pathway, also a link to the corresponding MetaCyc page.

For each sample or comparison, there are four columns summarizing the result of the pathway calling procedure. Note that empty fields in this table mean that a pathway is not part of any solution for a given sample or comparison.

- **Min. Solution**: A check mark indicates whether the pathway is part of the minimal solution (see above), an unchecked checkbox means that the pathway is part of the naive solution (see above) and an empty field means that a pathway has not been identified at all.
- **Confidence**: A confidence score for the pathway to be called, given the EC terms after filtering. If **Perform randomization analysis** has been selected, the confidence score is calculated as the fraction of randomization experiments in which the pathway occurs. If the aforementioned option was not selected, a simple approximation for this number is given as confidence score. Typically, pathways which are part of the minimal solution also have a high confidence score, but not necessarily.
- **Coverage**: Reports the fraction of EC terms that have been identified (not filtered) of all EC terms that are present in that pathway.
- **Num. Functions**: Reports the number of EC terms present in a given pathway.

Depending on whether the input has been an abundance table or a differential abundance table, the result may contain some more columns giving average statistics for the EC terms from the (differential) abundance table for the whole pathway in which they occur. For an abundance table the column **Average abundance** gives the average abundance for all identified (not-filtered) EC terms that are present in the pathway. Similarly, for a differential abundance table the metrics are summarized by averaging over all identified (not-filtered) EC terms in a pathway, specifically the **Average max group mean** and **Average fold change** are reported.

Exporting content of the Identified Pathways table views The Identified Pathways Table can be exported to tabular format. To export the content of the Identified Pathways table view, run export with default parameters. To export the content of the Compound or Enzyme table view, take the following steps:

1. Open the The Identified Pathways Table.
2. Switch to the view you wish to export by clicking on the relevant icon below the view area.
3. While on the relevant view, launch the standard export functionality by clicking on the Export button in the Workbench toolbar or by selecting **Export** under the File menu.
4. Select the tabular format to export the data to.
5. Confirm the data element that has been pre-selected in the **Navigation Area**.
6. Configure the export parameters. Deselect **Export all columns**.
7. Select **Export table as currently shown**.
8. Select where the data should be exported to.
9. Click **Finish**.

12.9.2 The Identified Pathways View

When double clicking on a line in the table or selecting one or several lines and clicking on **Open Pathway View** in the bottom of the table, a simple visualization of the corresponding pathway(s) will be opened in a split view (figure 12.23). This visualization is similar to the pathway view of a pathway database, see section 17.3.2. This object has two views

- The Identified Pathways Graph View ()
- The Text Contents View ()

where the text contents view is the same as for the pathway view.

The Identified Pathways Graph View In this split view, the data from the original (differential) abundance table can be visualized on the EC terms. A minimap in the right upper corner of the side panel simplifies the navigation if many pathways have been opened at the same time. In the **Metric** section, a specific sample or comparison can be chosen for which the data will be visualized on the EC terms, whereas the Metric drop down menu provides a selection of different metrics associated with the EC terms. For an abundance table, the only option is Abundance whereas for a differential abundance table there are the options

- Max group mean
- Log_2 fold change
- Fold change
- P-value

- FDR p-value
- Bonferroni

corresponding to the headers in the differential abundance table.

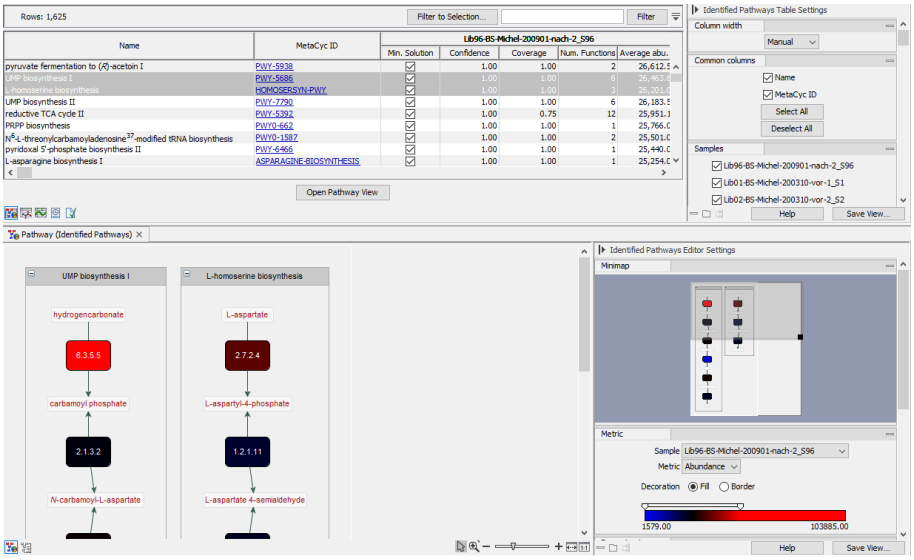


Figure 12.23: *A pathway call result from an abundance table, where the abundances from the input table correspond to the hue of the functional entities in the pathway. The hues or width of the functional terms in the pathway visualization can be set to any metric from the input (differential) abundance table for a given sample or comparison.*

In the lower right corner of the side panel there is a property viewer which displays information about selected EC terms and metabolites.

Chapter 13

Drug Resistance Analysis

Antimicrobial resistance (AMR) - bacteria, viruses, fungi and parasites that no longer respond to drugs commonly used to treat infections - poses a threat to human and animal health worldwide. By identifying resistance genes and markers from sequencing data, antimicrobial resistance detection aims to provide critical information about the resistance of microbial species to various antimicrobial drugs and to help monitor the spread of resistant pathogens.

Find Resistance with ShortBRED and **Find Resistance with PointFinder** facilitates detection of antimicrobial resistances directly from sequencing reads. The tools look for gene families and resistance conferring mutations, respectively.

Find Resistance with Nucleotide Database identifies AMR genes from contigs.

13.1 Find Resistance with PointFinder

Find Resistance with PointFinder identifies known antimicrobial resistance conferring mutations from reads. In contrast to the Find Resistance with Nucleotide Database tool that quantifies the occurrence of entire resistance conferring genes, the aim is to detect the presence of resistance conferring mutations in antimicrobial targets in both susceptible and resistant strains.

The presence of antimicrobial resistance conferring variants can be inferred by mapping reads to a point mutation database containing both wild type and known resistant mutants of antimicrobial target genes.

PointFinder databases can be downloaded using **Download Resistance Database** (section 18.1).

To run Find Resistance with PointFinder, go to:

> **Tools** | **Microbial Genomics Module** () | **Drug Resistance Analysis** () | **Find Resistance with PointFinder** ()

The tool accepts a nucleotide sequence or sequence list as input, followed by the selection of the point mutation database for the relevant pathogen.

For the read mapping step, several parameters are available (figure 13.1):

- **Match score**: score for a match
- **Mismatch cost**: cost for a mismatch

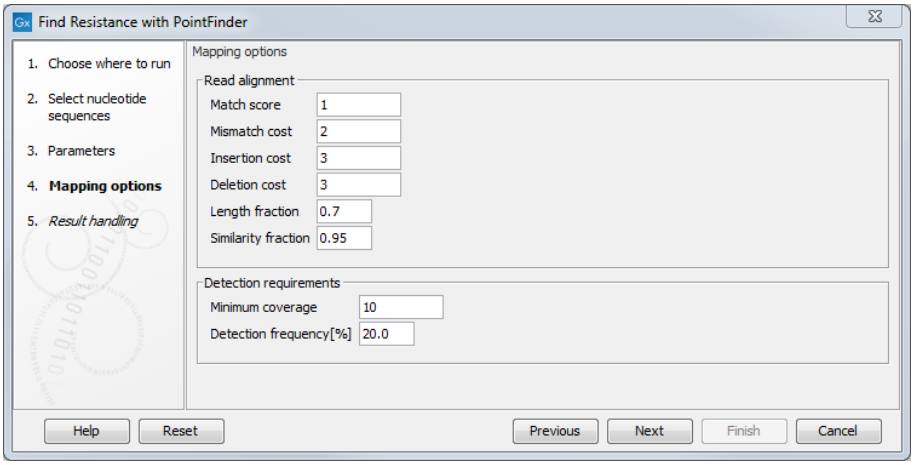


Figure 13.1: *Options.*

- **Insertion cost**: cost for an insertion
- **Deletion cost**: cost for accepting a deletion
- **Length fraction**: minimum length of the mapped portion of the read
- **Similarity fraction**: minimal fraction of matches in the mapped region

The tool first maps the reads to the specified database sequences. Next, the tool analyses the read mappings. For each reference sequence containing the variant, the tool verifies if the reads mapping to that reference or related references (e.g. references with an additional mutation) contain that variant. The result table will only list those entries from the database that have a minimum coverage and exceed the minimum frequency threshold. The tool also adds information:

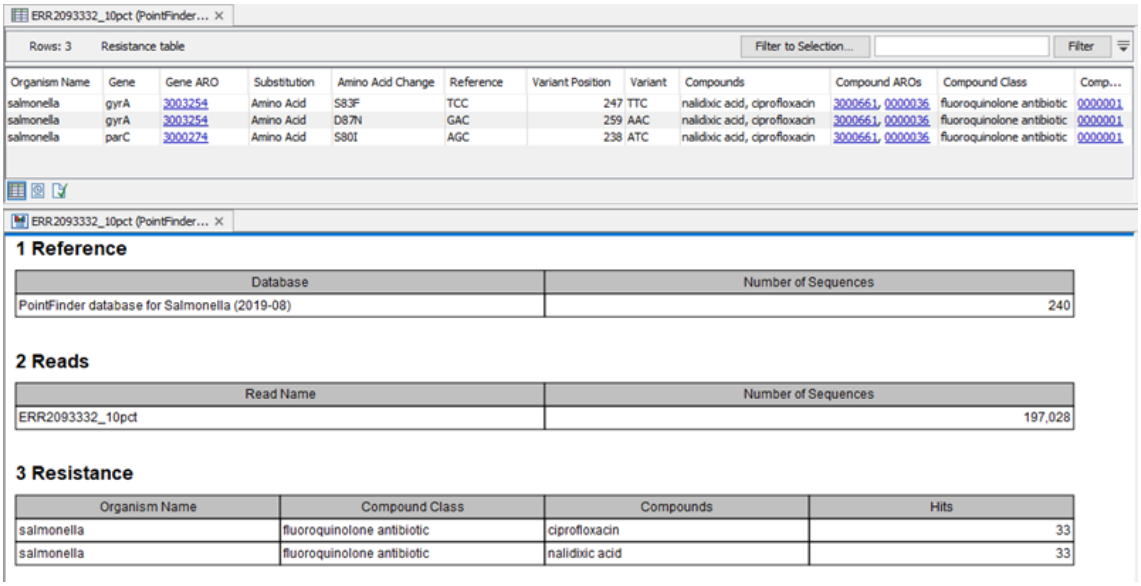
- **Minimum coverage**: the number of reads covering the variant region. The coverage must be complete, e.g. all 3 nucleotides that constitute an amino acid change have to be covered.
- **Detection frequency**: The minimum allele frequency that is required to annotate a variant as being present in the sample.

The tool outputs a table containing information about the variants detected in the reads. The columns available in that table can be seen in figure 13.2 (which also shows an example of report output by the tool):

By enabling the "Output annotated reads" option, the user can obtain a copy of the subset of reads that map to target variants. The reads will be annotated with the following annotations:

- Target: The name of the target reference (from the PointFinder database) where the read mapped to, e.g. "salmonella_gyrA_AAS_S_83_Y_TCC_247_TAC".
- Compound Class: The compound class to which the variant gives resistance, e.g. fluoroquinolone antibiotic.
- Compounds: The compound(s) class to which the variant gives resistance, e.g. colistin.

The resulting report contains information about the reads and the database that was used.



ERR2093332_10pct (PointFinder...

Rows: 3 Resistance table

Filter to Selection... Filter

Organism Name	Gene	Gene ARO	Substitution	Amino Acid Change	Reference	Variant Position	Variant	Compounds	Compound AROs	Compound Class	Comp...
salmonella	gyrA	3003254	Amino Acid	S83F	TCC	247	TTC	nalidixic acid, ciprofloxacin	3000661, 0000036	fluoroquinolone antibiotic	0000001
salmonella	gyrA	3003254	Amino Acid	D87N	GAC	259	AAC	nalidixic acid, ciprofloxacin	3000661, 0000036	fluoroquinolone antibiotic	0000001
salmonella	parC	3000274	Amino Acid	S80I	AGC	238	ATC	nalidixic acid, ciprofloxacin	3000661, 0000036	fluoroquinolone antibiotic	0000001

ERR2093332_10pct (PointFinder...

1 Reference

Database	Number of Sequences
PointFinder database for Salmonella (2019-08)	240

2 Reads

Read Name	Number of Sequences
ERR2093332_10pct	197,028

3 Resistance

Organism Name	Compound Class	Compounds	Hits
salmonella	fluoroquinolone antibiotic	ciprofloxacin	33
salmonella	fluoroquinolone antibiotic	nalidixic acid	33

Figure 13.2: *Table generated by Find Resistance with PointFinder, shown here together with the corresponding report.*

13.2 Find Resistance with Nucleotide Database

The Find Resistance with Nucleotide Database tool is designed for resistance typing of de novo-assembled contig sequences.

Find Resistance with Nucleotide Database is inspired by Zankari et al., 2017 and uses BLAST for identification of acquired antimicrobial resistance genes within whole-genome sequencing (WGS) data.

Nucleotide resistance databases for use with the tool can be downloaded using **Download Resistance Database** (section 18.1).

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Drug Resistance Analysis** () | **Find Resistance with Nucleotide Database** ()

Select the input genome or contigs (figure 13.3).

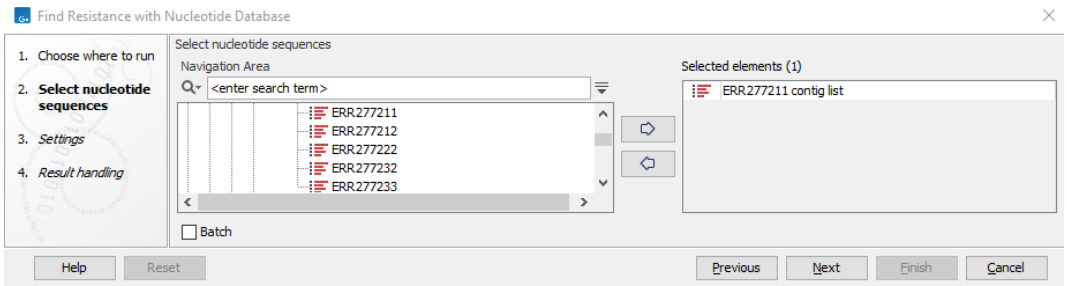


Figure 13.3: *Pre-assembled and complete- or partial genomes simple contig sequences may be used as input for resistance typing.*

You can then specify the settings for the tool (figure 13.4).

- **Database**. Select a nucleotide resistance database.

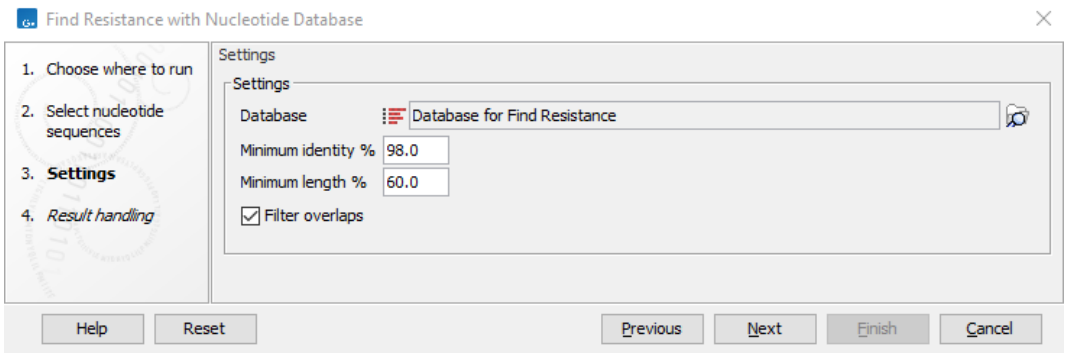


Figure 13.4: *Select database and settings for resistance typing.*

- **Minimum identity %.** The threshold for the minimum percentage of nucleotides that are identical between the best matching resistance gene in the database and the corresponding sequence in the genome.
- **Minimum length %.** The percentage of the resistance gene length that a sequence must overlap to count as a hit for that gene.
- **Filter overlaps.** Extra filtering of results per contig, where one hit is contained by the other with a preference for the hit with the higher number of aligned nucleotides (length * identity).

The output of the Find Resistance with Nucleotide Database tool is a table listing all the possible resistance genes and predicted phenotypes found in the input genome or contigs, as well as additional information such as degrees of similarity between the gene found in the genome and the reference (% identity and query /HSB values) and the location where the gene was found (contig name, and position in the contig). Depending on the type of database used, additional columns with link to resources may also be present in the table. To add the obtained resistance types to your Result Metadata Table, see section 20.2.3.

13.3 Find Resistance with ShortBRED

This tool allows you to detect and quantify the presence of antibiotic resistance (AR) marker genes that are represented by a database of peptide marker sequences. The tool first checks for exact matches against this database, and then runs DIAMOND. The Find Resistance with ShortBRED tool works similarly to the quantify step of ShortBRED, a public bioinformatics pipeline and resource. The tool is based on the ShortBRED-Quantify tool [Kaminski et al., 2015].

Find Resistance with ShortBRED quantifies the presence of Antibiotic Resistance (AR) marker genes in a sample of NGS short reads. It is possible to output a sequence list containing all the input reads which contained a marker (each read in the output is annotated with metadata describing the properties of the marker detected in the read).

Antibiotic Resistance marker databases for use with the tool can be downloaded using **Download Resistance Database** (section 18.1).

To start the tool, go to:

Tools | **Microbial Genomics Module** ( ) | **Drug Resistance Analysis** ( ) | **Find. Resistance with ShortBRED** ( )

The tool accepts a nucleotide sequence or sequence list as input.

In the next dialog (figure 13.5), several parameters are available:

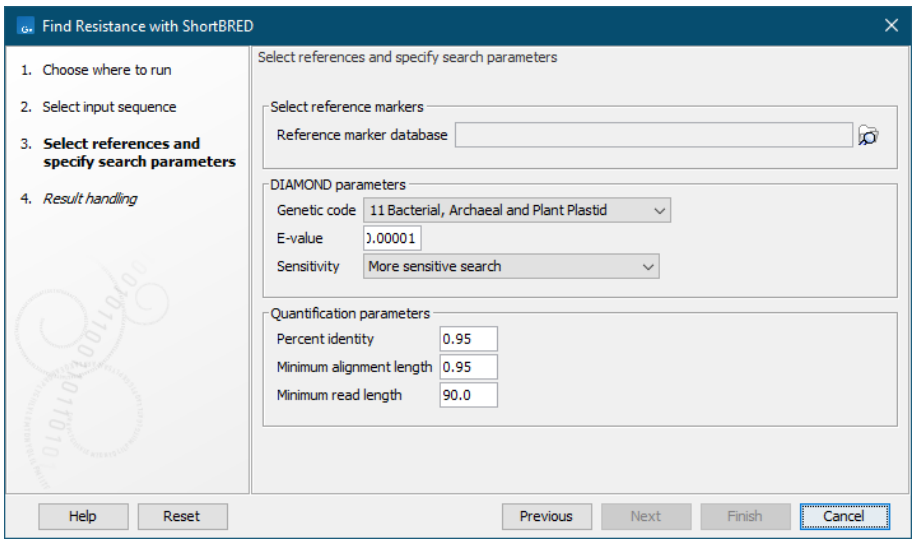


Figure 13.5: *References and search parameters.*

- **Reference marker database**: select the antimicrobial resistance marker database.
- **Genetic code**: select the genetic code to use when translating the nucleotide sequences to proteins.
- **E-value**: specify an expectation value to use as threshold for qualifying hits with DIAMOND. Note that exact matches to a peptide in the antimicrobial resistance marker database will always be reported, even if these would otherwise have an E-value larger than this threshold.
- **Sensitivity**: select DIAMOND sensitivity:
  - Faster search: The fastest search
  - Fast search: Designed for finding hits of >90% identity
  - Standard search: Designed for finding hits of >60% identity
  - Mid-sensitive search: More sensitive than standard search and faster than sensitive search.
  - Sensitive search: Designed for finding hits of >40% identity
  - More sensitive search: Designed for finding hits of >40% identity with some motif masking disabled
  - Very sensitive search: Designed for finding hits of 40% identity
  - Most sensitive search: The most sensitive search
- **Percent identity**: defines a minimum threshold for the percent identity of an alignment. This value is used by Find Resistance with ShortBRED to determine whether a hit found by DIAMOND is sufficiently good to be validated as a true hit (equivalent to the parameter "-id" of the ShortBRED-Quantify tool).

- **Minimum alignment length**: the minimum length of the DIAMOND alignment. This value is used by Find Resistance with ShortBRED to determine whether a hit found by DIAMOND is sufficiently good to be validated as a true hit (equivalent to the parameter "-pctlength" of the ShortBRED-Quantify tool - see citation at the beginning of this section).
- **Minimum read length**: the minimum read length. This value is used by Find Resistance with ShortBRED to determine whether a read is long enough to be processed by Find Resistance with ShortBRED (equivalent to the parameter "-minreadBP" of the ShortBRED-Quantify tool).

The Find Resistance with ShortBRED tool will output a resistance abundance table, a result summary table, an optional report, and an optional sequence list.

The result summary table provides an overview of all the resistance phenotypes in the applied database and reports the number of identified resistance genes and markers, and number of reads assigned to each phenotype.

The optional report output contains general information about the input sample, the marker database used and a short result summary.

The optional sequence list output contains all reads from the input sample which were found to contain one of the AR marker sequences. Each read in the sequence list is annotated with metadata describing the properties of the marker detected in the read.

13.3.1 Resistance abundance table

The Resistance abundance table summarizes the abundance of each marker, i.e., it reports the number of times a given marker is found in the input reads. The abundance is also reported in units of RPKM, referred to as the normalized abundance, which are calculated in the same way as is done by ShortBRED-Quantify. It is possible to aggregate the abundance by gene name and resistance phenotype to get the abundance at each of these levels. The table can be visualized using the Stacked bar charts and Stacked area charts function, as well as the Sunburst charts.

- **Table view** () (figure 13.6)

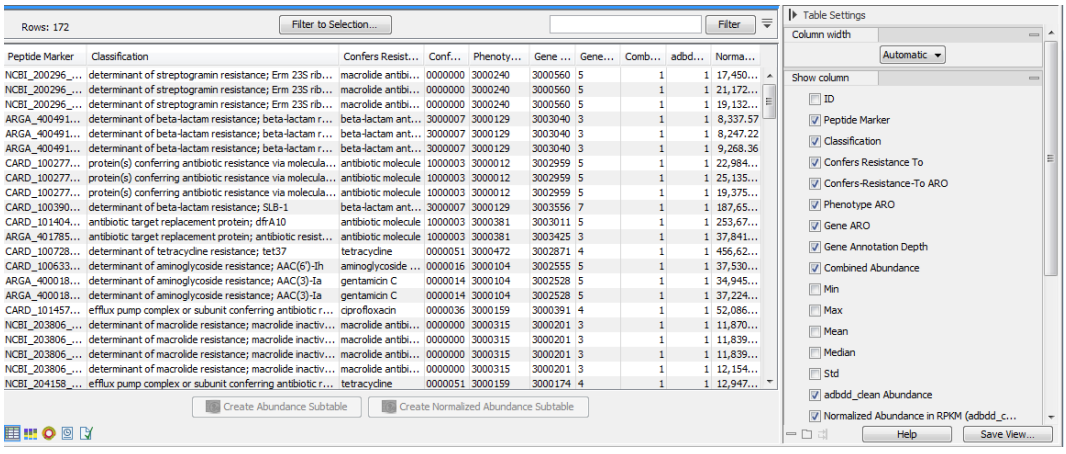


Figure 13.6: *Resistance abundance table.*

The table displays the following columns (note that the columns can vary depending on the marker database used):

- **ID**: internal ID which the abundance tables use for ordering the samples. IDs are unique, so that when merging abundance tables peptide markers with the same ID will be combined.
- **Peptide Marker**: the name of the Peptide Marker as it is given in the AR marker database.
- **Classification**: the Classification of the peptide marker contains the resistance phenotype and the gene name separated by a semi-colon.
- **Confers Resistance To**: the antibiotic class which this marker confers resistance to.
- **Confers-Resistance-To ARO**: the ARO (Antibiotic Resistance Ontology) ID number of the "Confers-Resistance-To" property.
- **Phenotype ARO**: the ARO ID number associated with this particular resistance phenotype.
- **Gene ARO**: the ARO ID number associated with this particular gene.
- **Gene Annotation Depth**: the number of parents in the gene annotation graph in the Antibiotic Resistance Ontology. This indicates the specificity of the annotation. The higher the number, the more specific the annotation is.
- **Combined Abundance**: reports the number of times a given marker is found in the input reads across all samples.
- **Min**, **Max**, **Mean**, **Median** and **Std**: respectively minimum, maximum, mean, median and standard deviation of the number of reads for the taxa across all samples.
- **Name of the sample Abundance** (for example SRR2754560 in the table above): number of reads containing this peptide marker for each sample.
- **Normalized Abundance in RPKM** (name of the sample): Normalized Abundance is reported in units of RPKM (Reads Per Kilobase per Million reads) which are calculated in the same way as is done by ShortBRED-Quantify.

In the right side panel, under the tab Data, you can switch between raw and relative abundances. You can also combine absolute counts and relative abundances by Phenotype and Gene name by selecting the appropriate level in the **Aggregate feature** drop-down menu. Incomplete fatures at a given level of Aggregation can be hidden using the "Hide incomplete features" check box.

Finally, if you have previously annotated your table with Metadata (see section 7.9), you can **Aggregate sample** by the groups previously defined in your metadata table. This is useful when for example analyzing replicates from the same sample origin.

Above and under the table, the following actions are available:

- **Filter to Selection...** to have the table only displaying pre-selected rows in the table.
- **Create Abundance Subtable** will create a table containing only the selected rows.
- **Create Normalized Abundance Subtable** will create a table with all rows normalized on the values of a single selected row. The row used for normalization will disappear from the new abundance table. The normalization scales the abundance table linearly, where the scaling factor is calculated by determining the average abundance across all samples and for each sample scale it to the average for the reference. If you have zero values in some samples for the control, you will need to generate a new abundance table where these samples are not present. If the abundance table is obtained from merging single-sample abundance tables, then the merge should be redone excluding the samples with zero control read counts.

- **Stacked Bar Chart and Stacked Area Chart** ( )

  Choose which chart you want to see using the drop down menu in the upper right corner of the side panel. The charts can be scaled by percentage, where all bars have the same height of 100%, or counts, where the bar heights are proportional to the number of counts. In the Stacked Bar (figure 13.7) and Stacked Area Charts (figure 13.8), the metadata can be used to aggregate groups of columns (samples) by selecting the relevant metadata category in the right hand side panel. Also, the data can be aggregated at any classification level selected. The relevant data points will automatically be summed accordingly.

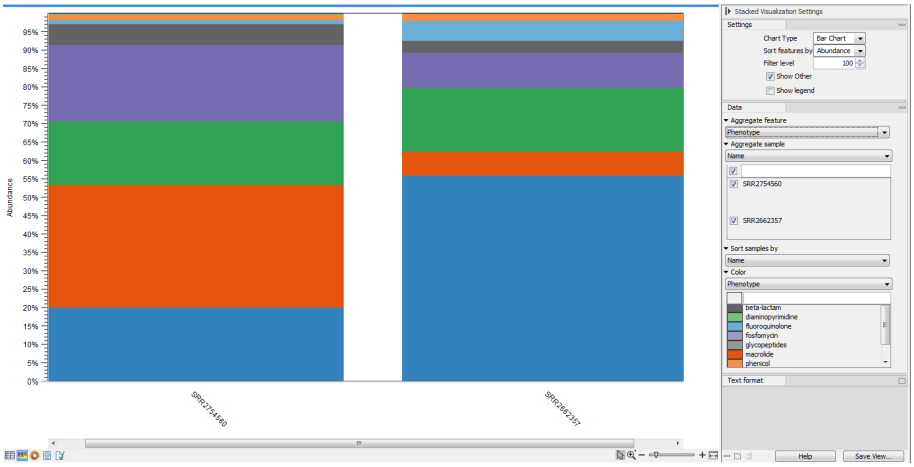


Figure 13.7: *Stacked bar chart.*

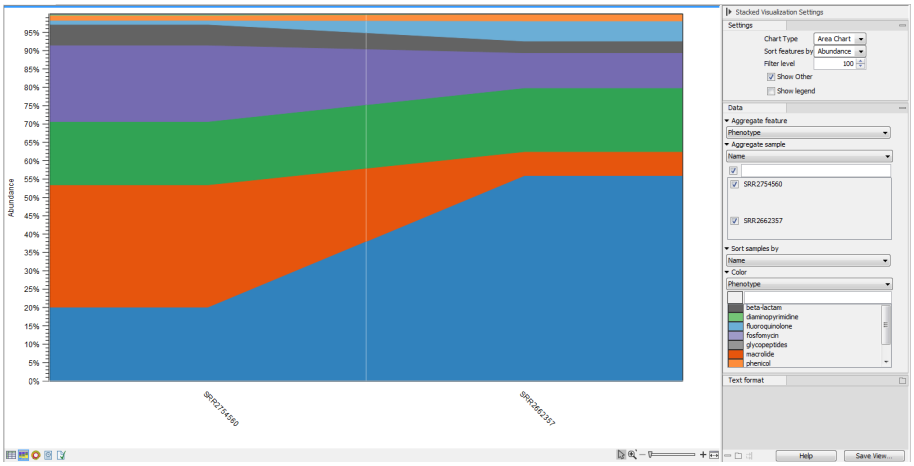


Figure 13.8: *Stacked area chart.*

Holding the pointer over a colored area in any of the plots will result in the display of the corresponding classification label and counts. **Filter level** allows to modify the number of features to be shown in the plot. For example, setting the value to 10 means that the 10 most abundant features of each sample will be shown in all columns. The remaining features are grouped into "Other", and will be shown if the option is selected in the right hand side panel. One can select which classification level to color, and change the default colors manually. Colors can be specified at the same classification level as the one used to aggregate the data or at a lower level. When lower classification levels are chosen in the data aggregation field, the color will be inherited in alternating shadings. It is also possible to sort samples by metadata attributes, and to show groups of samples without collapsing their stacks, as well as change the label of each stack or group of stacks. Features can be

sorted by "abundance" or "name" using the drop down menu in the right hand side panel. Using the bottom right-most button (**Save/restore settings** ( )), the settings can be saved and applied in other plots, allowing visual comparisons across analyses.

- **Zoomable Sunbursts** ( ) The Zoomable Sunburst viewer lets the user select how many classification level counts to display, and which level to color. Lower levels will inherit the color in alternating shadings. Classification and relative abundances are displayed in a legend to the left of the plot when hovering over the sunburst viewer with the mouse. The metadata can be used to select which sample or group of samples to show in the sunburst (figure 13.9).

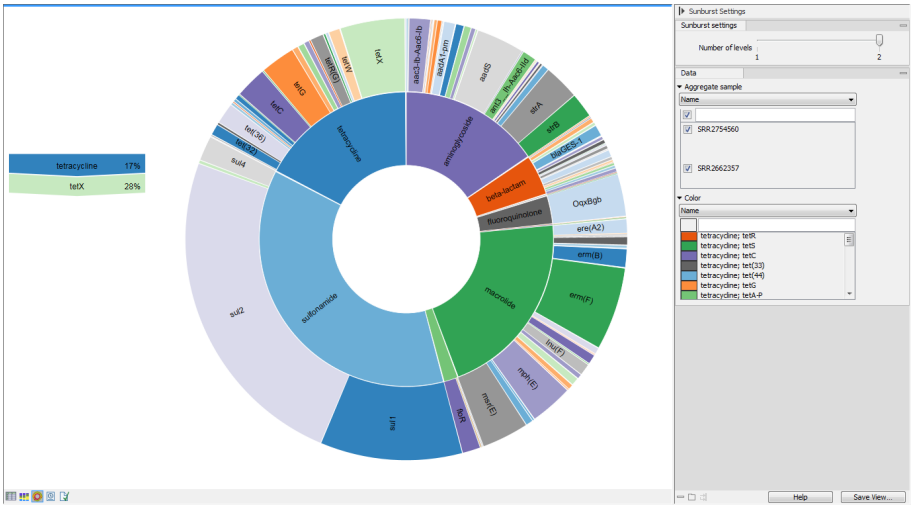


Figure 13.9: *Sunburst view.*

Clicking on a lower level field will render that field the center of the plot and display lower level counts in a radial view. Clicking on the center field will render the level above the current view the center of the view.

13.4 Join Nearby Variants

The **Join Nearby Variants** tool merges variants that are more than one nucleotide apart and within three nucleotides of each other into larger microhaplotypes. Because the tool merges variants regardless of their zygosity and frequency, it is most suited for use on isolates from monoploid organisms, such as most viruses and bacteria.

Merging variants allows amino acid changes to be more precisely determined by the **Amino Acid Changes** tool.

To run the tool, go to

> **Tools** | **Microbial Genomics Module** ( ) | **Drug Resistance Analysis** ( ) | **Join Nearby Variants** ( )

The tool takes a variant track as input.

The following parameters are available (figure 13.10):

- **Sequence track.** The genome against which the variants were called. This is needed to fill in spaces between variants being called with reference nucleotides.

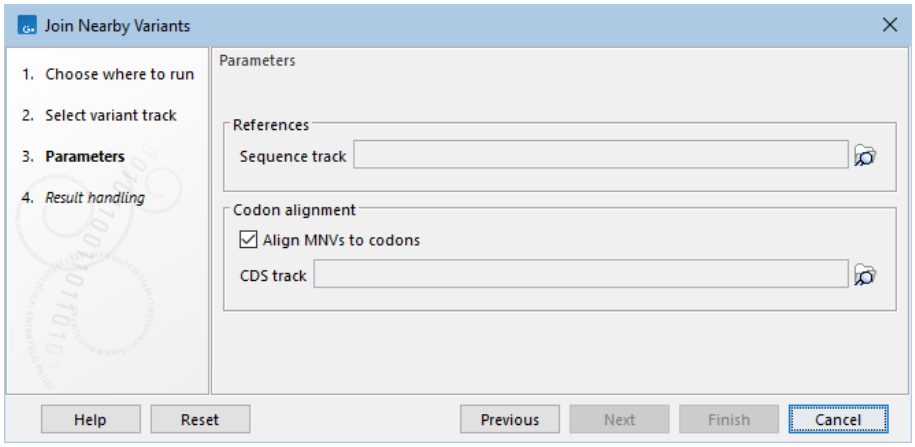


Figure 13.10: *Join Nearby Variants parameter settings.*

- **Align MNVs to codons.** When checked, MNVs are split into smaller MNVs and SNVs at codon boundaries. If a variant overlaps multiple CDS regions that have different reading frames, then the variant will be split in multiple ways: one for each reading frame.
  - **CDS track.** The CDS track is required to determine codon boundaries when aligning MNVs to codons.

Note: The Codon alignment option is primarily intended for matching variants against variant databases from outside the *CLC Genomics Workbench*. Databases of known variants (such as the WHO drug resistance variant database, section 18.2), can have variants given as amino acid substitutions i.e., given in triplets/codon MNVs as opposed to larger substitutions, like when called within the *CLC Workbench*. These longer variants must be split on codon boundaries in order to accurately match the database when using **Filter against Known Variants** or **Annotate from Known Variants**.

Detailed behavior

- Reference variants are removed from the output track.
- Overlapping variants are never merged with a nearby variant, because it is unclear which of the overlapping variants should be merged with the nearby variant.
- Variants that are adjacent are never merged. If adjacent variants were actually one longer variant, they would have been called as such by the variant detection tools (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Variant_Detection_tools.html). Adjacent variants mean that the variants were not present on the same reads and thus should not be inferred to be a single variant.
- When variants are merged, the count, coverage etc. are the minimum of the observed values in the variants being merged. The frequency of the resulting variant is the minimum count divided by the minimum coverage. The zygosity of the resulting variant is unknown if any of the variants being merged have unknown zygosity, heterozygous if any of the variants being merged are heterozygous, and otherwise homozygous.

- The first variant in a group to be merged may be right-shifted to permit merging to occur. For example, if the reference is "ACGTACACACG" and the sample has "ACGTACACT", then there are three ways in which "AC" may be deleted. The tool right-shifts the deletion such that it is closest to the G -> T. The final variant called is then a replacement "ACG -> T".
- Variants will not be merged when they are on different sides of the origin of a circular chromosome, different sides of a splice junction, or either side of a ribosomal slippage (i.e. in a region where the CDS annotation overlaps itself by one or two nucleotides).

Part VII

Databases

Chapter 14

Databases for MLST Schemes

14.1 Create MLST Scheme

The **Create MLST Scheme** tool can be used to create a scheme from scratch.

To run the **Create MLST Scheme** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **MLST Typing** ( ) | **Create MLST Scheme** ( )

As input, the tool requires a set of complete isolate genomes in the form of one or more sequence lists or sequences. At least one of these genomes must be annotated with coding region (CDS) annotations. If these are not available, the Find Prokaryotic Genes tool (see section 12.1) or Annotate with DIAMOND (see section 12.3) can be used to predict and annotate the coding regions.

In the first wizard step shown in figure 14.1 the grouping of sequences into genomic units can be controlled. This is necessary when working with genomes that span several chromosomes or several contigs for the tool to consider these as one unit. The grouping can be controlled by the **Assembly grouping** field:

- Each sequence is one assembly: Each individual sequence is considered a complete assembly of a genome.
- Each input element is one assembly: Each input element, i.e. input sequence or input sequence list, is considered a complete assembly of a genome.
- Group sequences by annotation type: Use annotations to group the assemblies and specify the annotation field with **Assembly annotation type**. Some tools, such as the **Download Custom Microbial Reference Database**, will automatically assign an Assembly ID that can be used for grouping. For a manual assignment of Assembly ID annotations, please see section 22.

After specifying the input, the second step is to set up the basic MLST Scheme creation parameters (figure 14.2).

The **Create MLST Scheme** tool works by extracting all annotated coding sequences (CDS) and clustering them into similar gene classes (loci). It is possible to specify whether we are interested

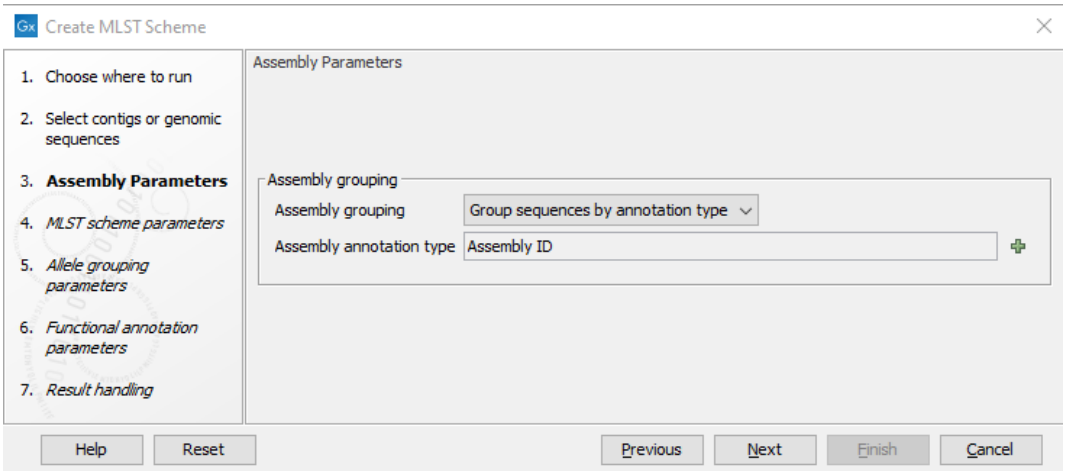


Figure 14.1: *Grouping the input into assemblies.*

in the genes that are present in some genomes (**Whole genome** - must be present in at least 10% of all genomes), most genomes (**Core genome** - must be present in at least 90% of the genomes), or a user-specified **Minimum fraction**.

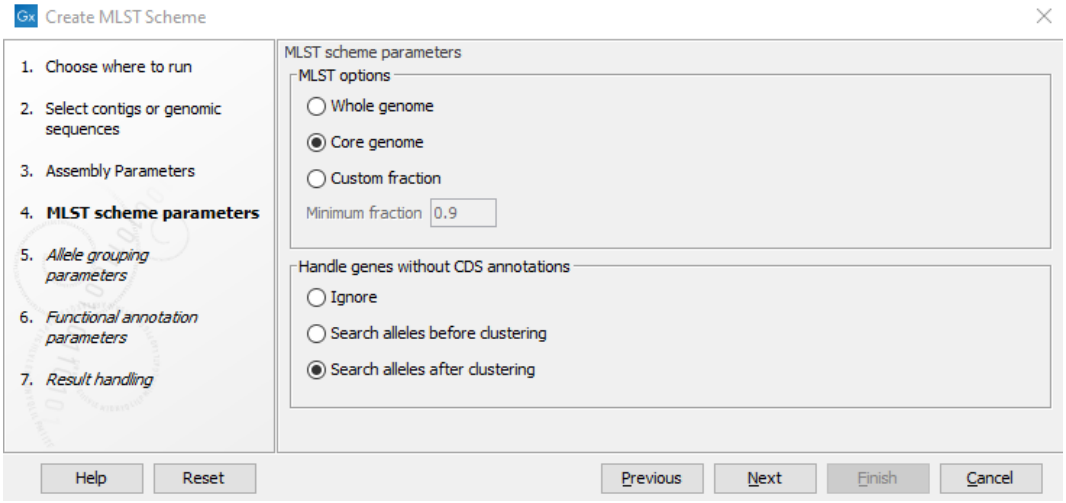


Figure 14.2: *Basic options for creating a MLST scheme.*

The best results are obtained by supplying genomes with proper CDS annotations. The **Handle genes without annotations** option controls how genomes without CDS annotations and how existing CDS may be overridden if a longer CDS from another genome exactly matches the genomic sequence.

- **Ignore**: Only use the existing CDS annotations as a basis for the MLST scheme construction.
- **Search alleles before clustering**: All of the input genomes are blasted (using DIAMOND) against the set of annotated genes, and any new genes will be added as alleles. This is a very slow, but thorough check.
- **Search alleles after clustering**: After clustering the genes, all of the input genomes are blasted (using DIAMOND), but only against the longest protein in each cluster.

The **Allele grouping parameters** step (figure 14.3) specifies how the different genes (CDS annotations) are compared to each other. DIAMOND is used for this clustering. The following can be specified:

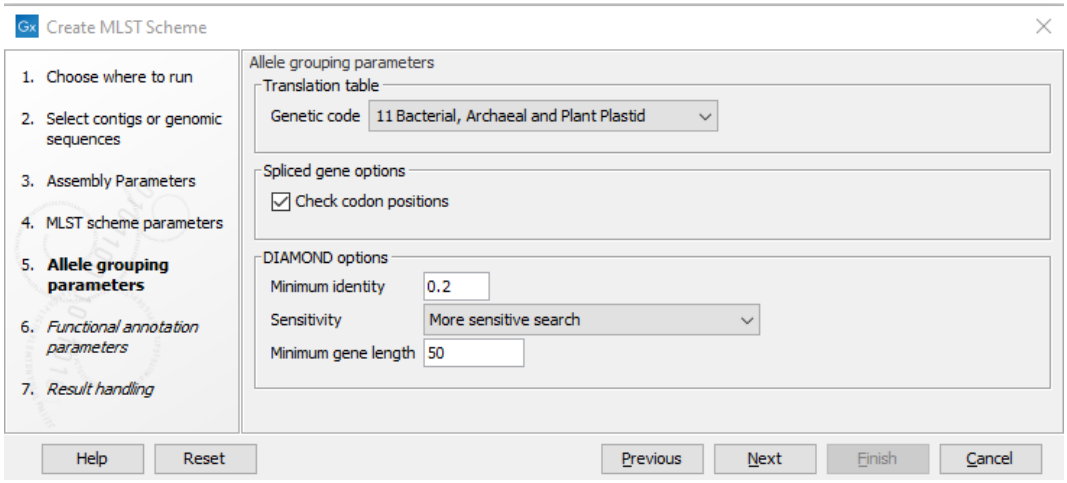


Figure 14.3: *The allele grouping (clustering) options.*

- **Genetic code**: Specify the genomic code to use for the input samples if **Check codon positions** is enabled.
- **Check codon positions**: If this is enabled, coding sequences not starting with a start codon, not ending with a stop codon or containing internal stop codons will be discarded. This can be disabled, for example to allow the construction of MLST schemes with spliced genes where each exon is considered an allele.
- **Minimum identity**: Set the minimum sequence identity before grouping protein sequences.
- **Sensitivity**: Select DIAMOND sensitivity:
  - Faster search: The fastest search
  - Fast search: Designed for finding hits of >90% identity
  - Standard search: Designed for finding hits of >60% identity
  - Mid-sensitive search: More sensitive than standard search and faster than sensitive search.
  - Sensitive search: Designed for finding hits of >40% identity
  - More sensitive search: Designed for finding hits of >40% identity with some motif masking disabled
  - Very sensitive search: Designed for finding hits of 40% identity
  - Most sensitive search: The most sensitive search
- **Minimum gene length**: Set this threshold to remove short genes from the resulting MLST scheme.

Note that after clustering, length outliers of a given cluster are removed by applying Tukey's fences with an interquartile range of 1.5, yet allowing for 5% length variation around the median. For example, for an allele cluster (locus) with allele lengths 51, 51, 51, 51, 53, the latter allele will not be removed although it falls outside the 1.5 IQR (both the first and third quartile are 51) since it is still within 5% of the median, for 48, 51, 51, 54, 63, only the former four will be included.

It is possible to decorate the alleles with information about virulence or resistance. The information can be extracted from either a ShortBRED Marker database or a Nucleotide database. These databases can be accessed using **Download Resistance Database**, see section 18.1, and can be provided as input to the **Create MLST Scheme** tool at this step (figure 14.4).

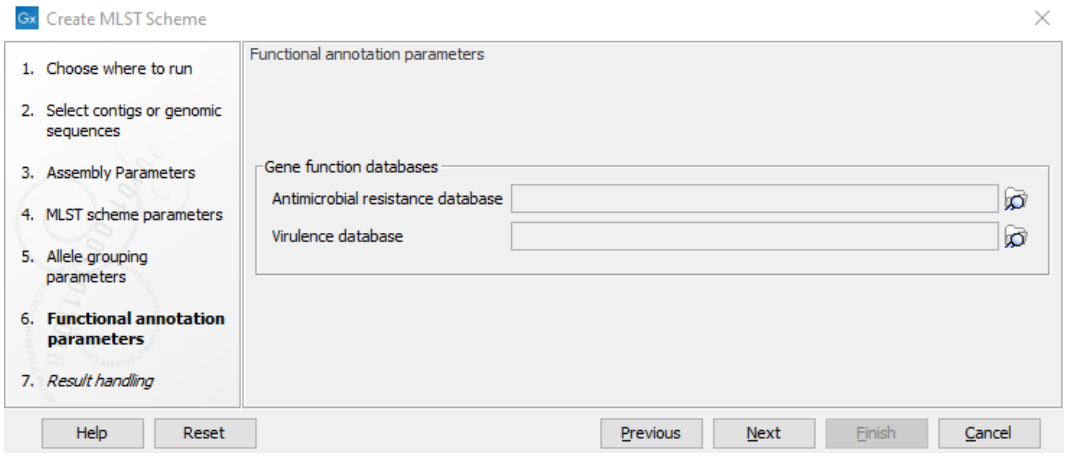


Figure 14.4: *The functional annotation parameters.*

14.2 Download MLST Scheme

The tool supports download of MLST schemes from the following databases:

- **PubMLST** [Jolley et al., 2018] (`https://pubmlst.org/`)
- **Institut Pasteur** (`https://bigsdb.pasteur.fr/`)

It is mandatory to be registered and logged in when downloading from these databases.

14.2.1 Download MLST Scheme parameters

To run the **Download MLST Scheme** tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **MLST Typing** ( ) | **Download MLST Scheme** ( )

Select the scheme you wish to download in the **Scheme to download** drop-down menu (figure 14.5). To jump to specific schemes, click the drop-down menu once and type the first letters of the desired scheme, e.g., type "es" to reach the first Escherichia spp. scheme.

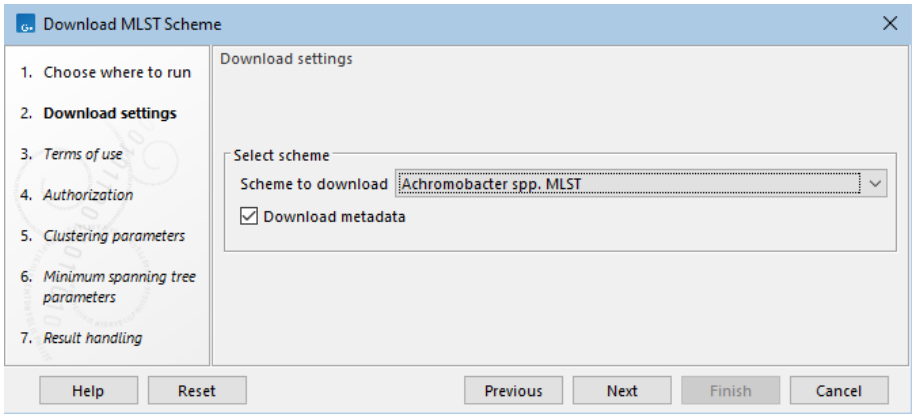


Figure 14.5: *The Download MLST Scheme settings.*

To download and extract metadata for all of the profiles in a scheme, tick the **Download metadata** option. Note that this can make the download take a long time.

Most of the schemes offered for download are classic (7-gene) schemes, but there are also core genome schemes available for several species, e.g.: N. gonorrhoeae, N. Meningitis, C. Jejuni / C. Coli, C. trachomatis, Vibrio cholerae, Listeria monocytogenes.

Some of the schemes may only contain allele and locus definitions and no profiles, i.e., sequence types.

Click on **Next** and accept the terms of use before proceeding to the *Authorization* step.

Authorize access through your account

To download MLST schemes using *CLC Genomics Workbench*, you must first authorize access to download data on you behalf. For this step, you must have a user account with the relevant MLST scheme provider (PubMLST or Pasteur) depending on which scheme you select. You must also have registered with the specific database in your account settings. How to create an account and register for specific databases is explained at `https://pubmlst.org/site-accounts` and `https://bigsdb.pasteur.fr/register/`.

1. Click the **Log in** button (figure 14.6). This will open the relevant login page in an external browser.

2. If you were not already logged in in the browser, you must now do so. Depending on the scheme you are downloading, log in using your PubMLST or Institut Pasteur account (figure 14.7).
   Note: Make sure you are registrered for the specific database you are trying to access. If you have an account, but have not registered for the specific database you are trying to download from, you will not be able to log in.

3. After logging in, you will be asked to authorize *CLC Workbench* to access data on your behalf (figure 14.8). Click "Authorize". This generates an access token and secret for use by *CLC Workbench*. No personal data about the account is shared. A verification code will be displayed after authorizing (figure 14.9).

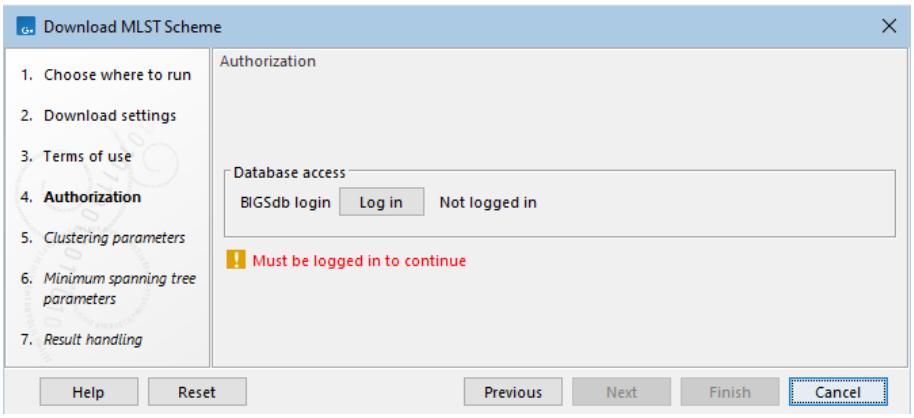


Figure 14.6: *Download MLST Scheme* Authorization *step.*

Figure 14.7: *Log in to your account. In this case the PubMLST account is needed.*

4. Copy the code, return to the Workbench, and paste it into the **Verification code** dialog box (figure 14.9). Click **OK**.

When the verification code has been succesfully entered, you will be logged in and can proceed to the next steps. The browser window can also be closed. For following downloads from the same source, you need not authorize access again. Clicking the **Log in** button should automatically connect to the database on your behalf.

Clicking the **Log out** button will reset the token and secret, allowing you to log in using another account.

Heatmap

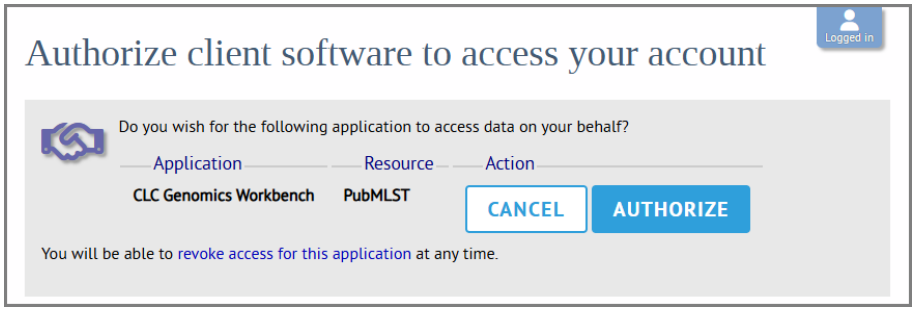


Figure 14.8: *You will be asked to allow the Workbench to access your account. Click "Authorize".*

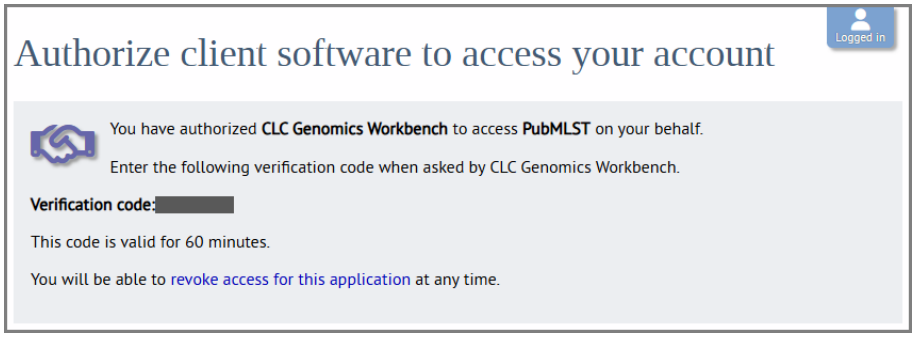


Figure 14.9: *Following authorization a verification code will appear. Copy the code and return to the Workbench.*

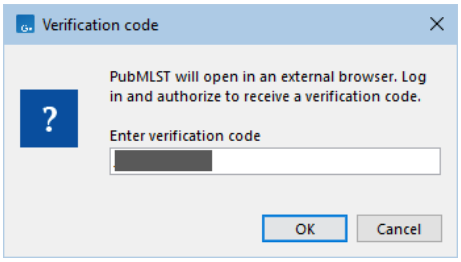


Figure 14.10: *Paste or type the verification code into the dialog box.*

The clustering parameters determine how the heatmap should be clustered (figure 14.11). The heatmap cell values are the observed frequencies of a given allele compared to the other alleles in the same locus.The possible cluster linkages are:

- **Single linkage**: the distance between two clusters is computed as the distance between the two closest elements in the two clusters.
- **Average linkage**: The distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster.
- **Complete linkage**: The distance between two clusters is computed as the distance between the two farthest objects in the two clusters.

The possible distance measures are:

- **Euclidean distance**: the square-root of the sum-of-square differences between coordinates.

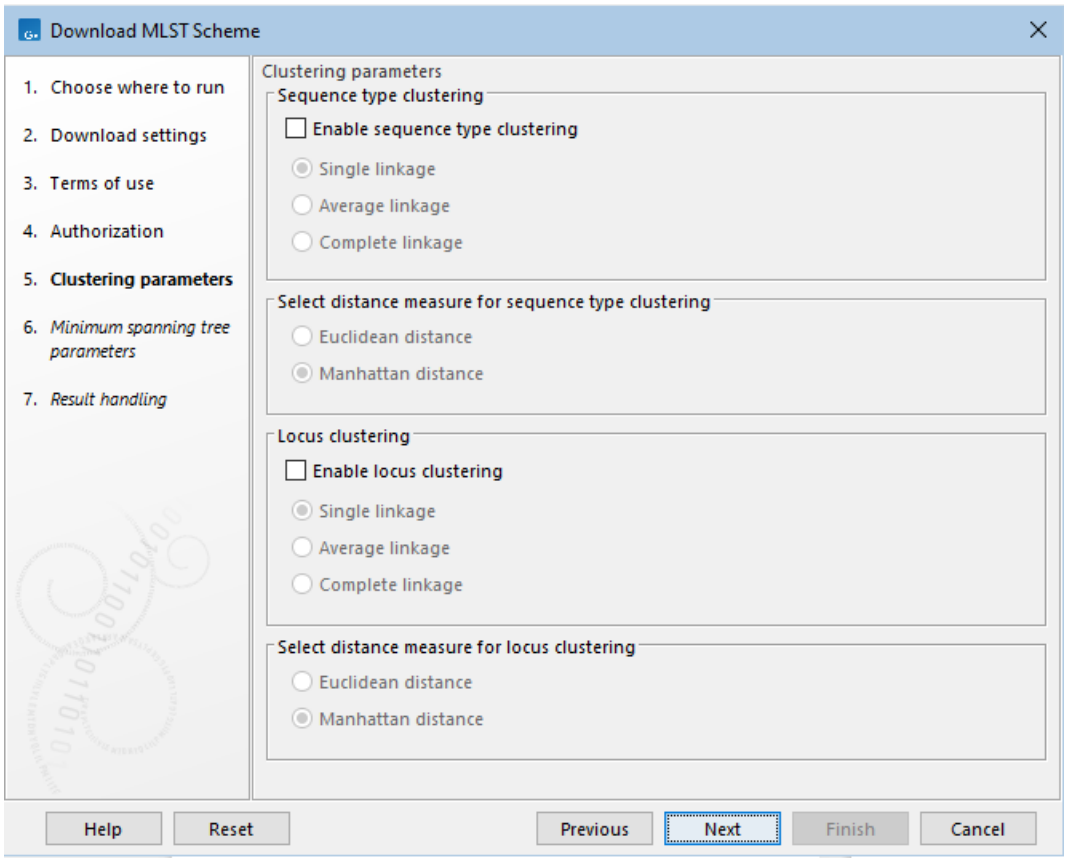


Figure 14.11: *The clustering parameters.*

- **Manhattan distance**: the sum of absolute differences between coordinates.

Note that for schemes with thousands of sequence types and/or loci, the clustering may become very slow and time-consuming.

Minimum spanning tree

The following options are available when creating a minimum spanning tree (figure 14.12):

- **Comparing a known to a missing allele:** the minimum spanning tree is created using a distance matrix, where the distance is calculated between all pairs of sequence types. The distance is calculated as the number of loci where the allele assignment differs. But in some cases, a locus for a sequence type may not have an assigned allele (for instance, for the accessory genes in a wgMLST scheme). If this is the case, the behavior depends on this setting: if 'counted as same alleles' is selected, a locus where at least one allele is missing for the pair being compared will be ignored (it will not count as a difference). On the other hand, if 'Counted as different alleles' is selected, a missing allele being compared to a known allele will increase the distance between the sequence types being compared.
- **Add clonal cluster metadata:** it is possible to assign cluster information to the scheme which will show up as metadata. The clustering is based on the minimum spanning tree, and will be similar to the clustering obtained by using the 'collapse branches' slider in the

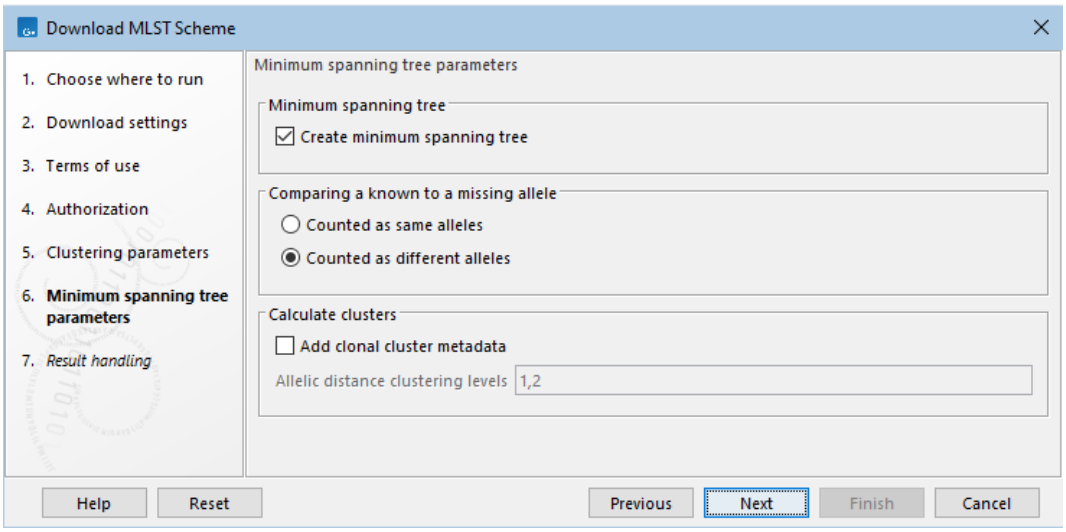


Figure 14.12: *The minimum spanning tree parameters.*

minimum spanning tree view - that is, the clustering will be single-linkage clustering - i.e. all nodes in cluster are within the specified threshold to at least one other node in the cluster. Each cluster will get a name chosen from the sequence type in the cluster with most connections.

- **Add clonal cluster metadata:** specifies the level at which the clustering will be performed. It is possible to specify multiple, comma-separated values. E.g. '100,200' will assign clusters at allelic distances of 100 and 200 - this will create two new metadata columns, cc_100 and cc_200 with the new cluster information.

14.3 Import MLST Scheme

To run the Import MLST Scheme tool, go to:

Tools | **Microbial Genomics Module** ( ) | **Databases** ( ) | **MLST Typing** ( ) | **Import MLST Scheme** ( )

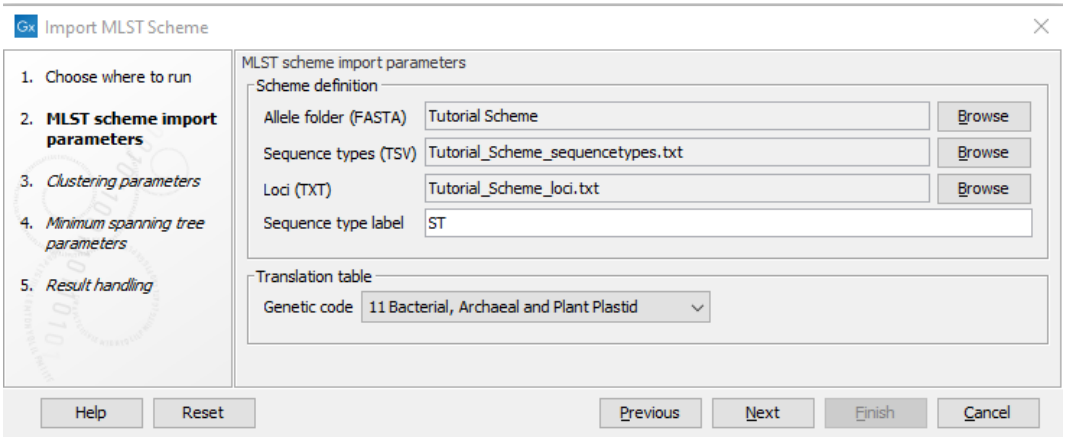


Figure 14.13: *The MLST Scheme import parameters.*

The **Allele folder (FASTA)** must contain a set of FASTA files, one for each locus. The files must have one of the following extensions to be recognized: "fa", "fas", "fsa", "fasta", "tfa". The name of the allele must be the locus name and allele name separated by an underscore, like in this example:

```
>pheS_1
AGAGAAAAGAACGATACTTTCTATATGGCCCGTGATAATCAAGGCAAGCGTGTTGTCTTA
>pheS_2
AGAGAAAAGAACGATACTTTCTATATGGCCCGTGATAATCAAGGCAAGCGTGTTGTCTTA
```

The **Sequence types (TSV)** file must be a tab-separated file listing a sequence type and its alleles in the following format:

```
ST  pheS    glyA    fumC    mdh sucA    dnaN    atpA    clonal_complex
1   30  1   1   1   1   1   1   6
3   6   8   7   3   4   3   1
4   7   9   8   3   5   2   1
6   47  3   10  4   7   2   2
```

It is possible to add arbitrary metadata as additional columns after the loci columns (e.g. the 'clonal_complex' column above). If multiple isolates share the same sequence type, but have different metadata, it is possible to add multiple lines with the same sequence type name and allele ids, but with different metadata entries.

The **Loci (TXT)** file must be a tab-separated file listing a locus name and its corresponding metadata. For this file the only recognized headers are "Locus", "Known name", "Type name", "Locus type" where the name of the locus in the MLST scheme needs to match the name in the Locus column of the annotation file.

```
Locus Known name Type name Locus type
locus5 FALSE Unknown ST1
fliR TRUE fli ST2
flgL TRUE flg ST3
hpaB TRUE hpa ST4
```

The **Clustering parameters** and **Minimum Spanning Tree parameters** are similar to the options for Download MLST Scheme tool (see section 14.2)

The genetic code specified will be used for novel allele detection to make sure each allele starts and ends with an initiation and stop codon, respectively. If "No code specified" is selected, these requirements will not be checked when searching for novel alleles, instead the aligned part of the existing alleles to the assembly is used to define the allelic length. Note that the latter is useful for 7-gene MLST schemes which generally use fractions of genes, but it is also sensitive towards unaligned ends and may return too short alleles in some cases.

Chapter 15

Databases for Amplicon-Based Analysis

15.1 Download Amplicon-Based Reference Database

Amplicon-based reference databases contain a list of representative amplicon sequences and their taxonomy. Databases like these are required for amplicon-based analysis (chapter 5).

The following databases are available:

- **SILVA**. Small subunit (SSU; 16S/18S) and large subunit (LSU; 23S/28S) ribosomal RNA sequences for prokaryotic and eukaryotic taxonomic assignment [Quast et al., 2012] (https://www.arb-silva.de/).
  Note on SILVA taxonomy: SILVA assigns taxonomies only down to the genus level. Therefore, it is best practice to aggregate downstream results to at least this level. To provide an option for more specific taxon assignment, we have appended the organism name of the sequence as a 'species-level' taxonomy. However, since SILVA does not curate organism names, this can lead to inconsistent taxonomies. For example, sequences may share the same species-level taxonomy but differ in higher-level taxonomies or host organism names can be appended as the species. Aggregating results at the genus level is equivalent to running analyses with a SILVA database that excludes species-level information.

- **MiDAS**. 16S ribosomal RNA sequences for prokaryotic and eukaryotic taxonomic assignment of microbes in wastewater treatment and bioenergy systems [**?**] (https://www.midasfieldguide.org/guide).

- **UNITE**. Internal transcribed spacer (ITS) sequences for fungal taxonomic assignment. The database is available in three differently clustered versions, 97% similarity, 99% similarity, and dynamic, in which sequences are dynamically clustered at similarities between 97-99%. Additionally, each of these three databases are available in a version with and without singletons. Singletons are fungal taxons for which only one ITS sequence is available [Kõljalg et al., 2020] (https://unite.ut.ee/).

- **Greengenes2**. Full length 16S ribosomal RNA sequences from the backbone of the Greengenes2 phylogenetic tree. For prokaryotic taxonomic assignment [**?**] (https://greengenes2.ucsd.edu/).

- **RefSeq Prokaryotic 16S**. 16S ribosomal RNA sequences from bacteria and archaea. The

sequences are curated by the NCBI RefSeq Targeted Loci Project (https://www.ncbi.nlm.nih.gov/refseq/targetedloci/).

Some of the above databases are available at different clustering levels of sequence similarities.

To run the tool, go to

Tools | **Microbial Genomics Module** () | **Metagenomics** () | **Databases** () | **Amplicon-Based Analysis** () | **Download Amplicon-Based Reference Database** ()

Select the database needed and specify where to save it. When using this tool, the databases downloaded are automatically formatted.

If you wish to format your own database with your own sequences and a corresponding taxonomy file, use the **Update Sequence Attributes in Lists** tool (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Update_Sequence_Attributes_in_Lists.html) to set the "Taxonomy" field. A clustering level for such custom databases can not be set on the data object directly, but it may be specified as a parameter when running the OTU Clustering tool.

Chapter 16

Databases for Taxonomic Analysis

16.1 Download Curated Microbial Reference Database

The **Download Curated Microbial Reference Database** tool downloads selected references as sequence lists and/or indexes that can be used with downstream analysis tools.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** | **Metagenomics** | **Databases** | **Taxonomic Analyses** | **Download Curated Microbial Reference Database**

In the first window (figure 16.1), select the database you wish to download.

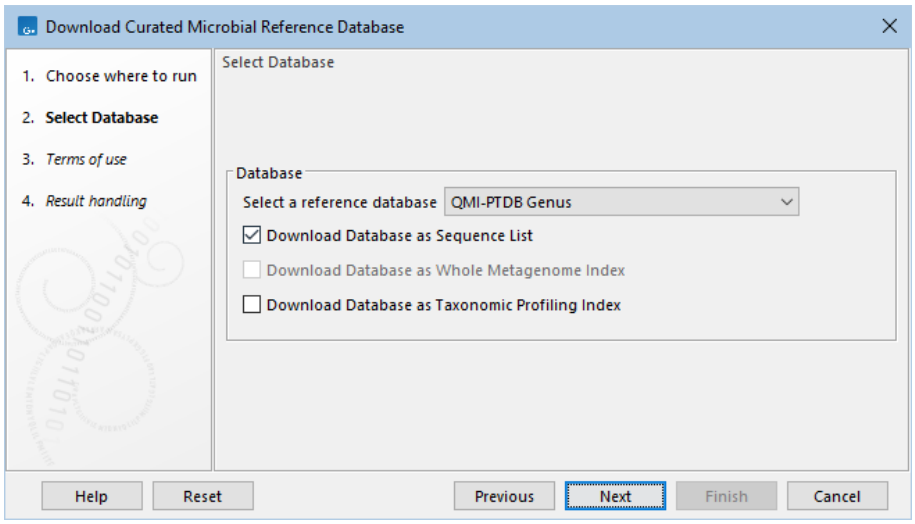


Figure 16.1: *Select the database and output format*

You can choose between several databases:

The sizes of the different formats are indicated below each entry as "size (of) sequence list/taxpro index/metagenome index", respectively. A star indicates that the format is not available for that item.

- **QMM-H**. QIAGEN Microbial Metagenome - Human Host database is a comprehensive microbial reference database for classification of whole metagenome data with **Classify**

Whole Metagenome Data (section 6.3). The database is based on all chromosome level or complete RefSeq genomes of archaea, bacteria, viruses, protozoa and fungi, and UniVec_Core sequences. Genome sequences and annotations are from Genbank (https://www.ncbi.nlm.nih.gov/genbank/). UniVec_Core sequences are from the UniVec Database (http://www.ncbi.nlm.nih.gov/tools/vecscreen/univec/). Human genome assemblies hg38 (GCA_000001405.29) and T2T (GCA_009914755.4) are included as the host reference.
Size: 90.3 GB/*/89.5 GB.

- **QMI-PTDB Genus**. QIAGEN Microbial Insights - Prokaryotic Taxonomy Database is a microbial reference database for taxonomic profiling of bacteria and archaea. The database represents all genera with a varying number of species per genus.
  Genome sequences and annotations are from the NCBI Reference Sequence Database (RefSeq; https://www.ncbi.nlm.nih.gov/refseq/) and have been annotated with taxonomy from the Genome Taxonomy Database (GTDB; https://gtdb.ecogenomic.org).
  The database was created by selecting one representative genome per species, and subsequently reducing the relative number of species per genus to meet the desired database size. For reduction, higher assembly status, lower number of contigs, and longer total length was prioritized. All genomes marked as "reference genome" were retained. So were species commonly included in microbial reference standards.
  When running Taxonomic Profiling with the QMI-PTDB Genus database, 32GB of memory is required.
  Size: 15.7 GB/22.4 GB/*.

- **QMI-PTDB Family**. QIAGEN Microbial Insights - Prokaryotic Taxonomy Database is a microbial reference database for taxonomic profiling of bacteria and archaea. The database represents all families with a varying number of genera per family.
  Genome sequences and annotations are from the NCBI Reference Sequence Database (RefSeq; https://www.ncbi.nlm.nih.gov/refseq/) and have been annotated with taxonomy from the Genome Taxonomy Database (GTDB; https://gtdb.ecogenomic.org).
  The database was created by selecting one representative genome per genus, and subsequently reducing the relative number of genera per family to meet the desired database size. For reduction, higher assembly status, lower number of contigs, and longer total length was prioritized. All genomes marked as "reference genome" were retained. So were species commonly included in microbial reference standards.
  When running Taxonomic Profiling with the QMI-PTDB Family database, 16GB of memory is recommended.
  Size: 4.6 GB/6.5 GB/*.

- **MGnify Unified Human Gastrointestinal Genome (UHGG)**. A database of metagenomic-assembled genomes from human gut samples, curated and hosted by MGnify [Gurbich et al., 2023], EMBL-EBI (https://www.ebi.ac.uk/metagenomics/browse/genomes).
  Size: 3.5 GB/7.8 GB/*.

- **MGnify Chicken Gut**. A database of metagenomic-assembled genomes from chicken gut samples, curated and hosted by MGnify [Gurbich et al., 2023], EMBL-EBI (https://www.ebi.ac.uk/metagenomics/browse/genomes).
  Size: 1.0 GB/2.1 GB/*.

- **MGnify Pig Gut**. A database of metagenomic-assembled genomes from pig gut samples, curated and hosted by MGnify [Gurbich et al., 2023], EMBL-EBI (https://www.ebi.ac.uk/metagenomics/browse/genomes).
  Size: 1.0 GB/2.0 GB/*.
- **Unclustered Reference Viral DataBase (U-RVDB)**. Unclustered Reference Viral Database for virus detection [Goodacre et al., 2018]. The database includes curated viral, virus-related and virus-like nucleotide sequences except bacterial viruses, which are excluded.
  Size: 42.0 GB/*/1.6 GB.
- **Clustered Reference Viral DataBase (C-RVDB)**. Clustered Reference Viral Database for virus detection [Goodacre et al., 2018]. The database includes curated viral, virus-related and virus-like nucleotide sequences except bacterial viruses, clustered at 98% sequence similarity.
  Size: 0.6 GB/2.7 GB/1.27 GB.

When you have made your database selection, choose which format you wish to download.

- **Download Database as Sequence list**. Produces an annotated sequence list.

Depending on the database selection, one of the following will be available:

- **Download Database as Whole Metagenome Index**. This index type is used by the Classify Whole Metagenome Data tool (section 6.3).
- **Download Database as Taxonomic Profiling Index**. This index type is used by e.g., the Taxonomic Profiling tool (section 6.4).

Some of the databases offered are derived work, licensed under a Creative Commons Attribution-ShareAlike (CC BY-SA) license. We offer free access to those without requiring a CLC product license. They can be downloaded using the CLC Genomics Workbench with the Microbial Genomics Module installed in Viewing Mode. The downloaded files can then be exported to non-proprietary formats.

16.2 Download Custom Microbial Reference Database

The Download Custom Microbial Reference Database tool allows you to create a custom database from taxonomies or NCBI assembly IDs. The tool outputs a single sequence list.

To run the tool, go to:

Tools | **Microbial Genomics Module** () | **Databases** () | **Taxonomic Analyses** () | **Download Custom Microbial Reference Database** ()

Under **Customize Database**, select whether to include genomic and/or plasmid sequences (figure 16.2):

- **Include all**. The database will contain both genomic and plasmid sequences.
- **Include only plasmids**. The database will contain only plasmid sequences.

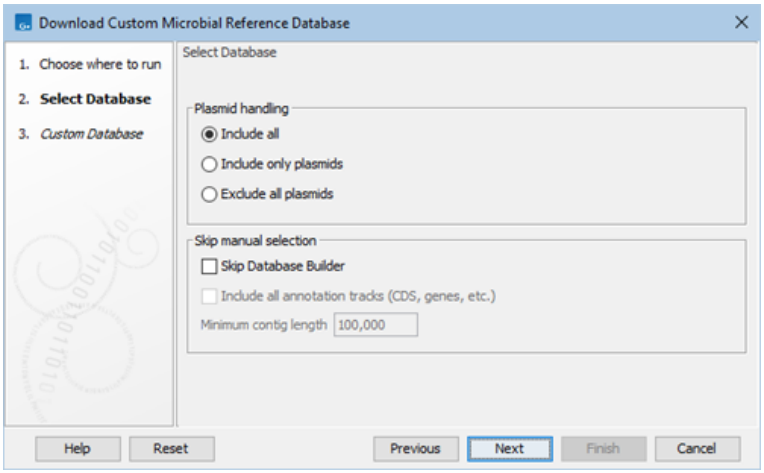


Figure 16.2: *Select type of sequences to include and whether to skip the Database Builder.*

- **Exclude all plasmids**. The database will not contain any plasmid sequences.

Choose whether you wish to skip manual selection:

- **Skip Database Builder**. If checked, a reference database with genomes matching the specified criteria will be downloaded once you click **Finish** from the next wizard step.

  If left unchecked, clicking **Finish** will instead open the **Database Builder** from which you can manually select genomes for download, see section 16.2.1. Genomes that match the specified criteria will be pre-selected.

- **Include all annotation tracks**. Will include CDS, gene, etc. annotations in the downloaded database. The annotations are not needed for taxonomic profiling, but may be required for other applications such as creating MLST schemes.

- **Minimum contig length**. The minimum length of sequences to be included in the database.

Click **Next** to customize the database (figure 16.3):

- **Select source of assemblies**:
  - **Build database from accessions or TaxIDs**. Enables the **ID matching** field, see below.
  - **Build database from taxonomic lineages**. Enables the **Taxonomic matching**, see below.

- **ID matching**. Provide a list of GenBank or RefSeq assembly accessions, or NCBI TaxIDs or species TaxIDs (one per line) to be included in your database.

  If using GenBank or RefSeq assembly accessions, the accessions must follow the assembly accession: 3 letter prefix, (GCA for GenBank assemblies or GCF for RefSeq assemblies) followed by an underscore and 9 digits. For example, GCA_000019425 for the assembly of the DH10B substrain of E. coli. If a version number is included, it will be ignored and the newest version downloaded. The assembly is always downloaded from GenBank.

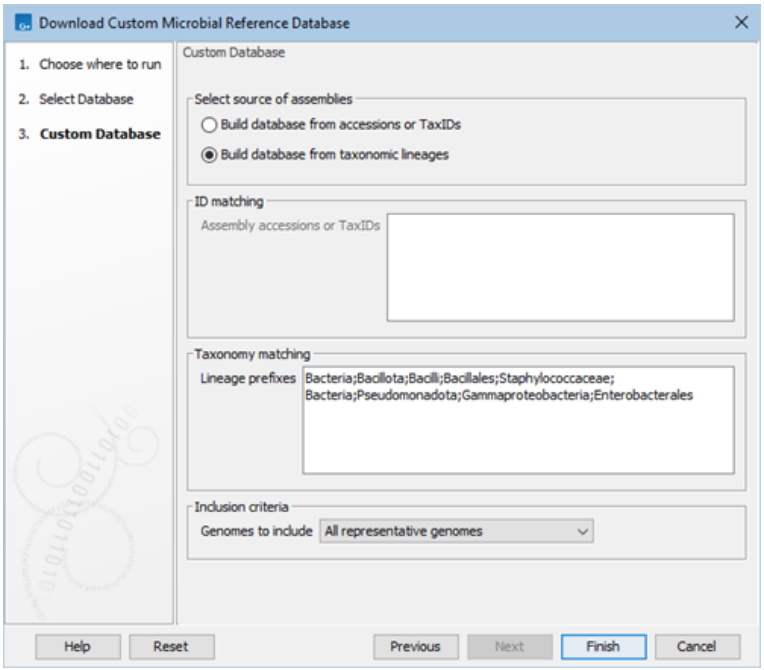


Figure 16.3: *Specify accession or TaxIDs, or taxonomic lineages to include in the database.*

The TaxID is the NCBI taxonomy identifier for the organism from which the genome assembly was derived. The species TaxID is the identifier for the species to which the organism belongs. For a given organism, TaxID and species TaxID will be identical unless the organism was reported at a strain or subspecies level.

- **Taxonomy matching**. Provide a list of taxonomic lineage prefixes (one per line) to include in your database. The lineages should follow the format of 7-step taxonomies. For example entering "Bacteria;Bacillota;Bacilli;Bacillales;Staphylococcaceae;" will include all genera and species genomes in the Staphylococcaceae family. Entering "Bacteria;Pseudomonadota;Gammaproteobacteria;Enterobacterales;" will include all family, genera and species genomes in the Enterobacterales order. The NCBI taxonomy is updated weekly. When searching you should use the updated taxonomy.
- **Inclusion criteria**:
  - **All reference genomes**. All reference genomes in the chosen lineage(s) are included.
  - **All representative genomes**. All representative genomes in the chosen lineage(s) are included.
  - **All reference and representative genomes**. All reference and representative genomes in the chosen lineage(s) are included.
  - **All genomes**. All genomes in the chosen lineage(s) are included.
  - **One per species**. One reference is selected for each species in the chosen lineage(s). The chosen species representative is selected based on ranking with Reference genomes > Representative genomes > Complete genomes > Scaffolds > Contigs. When two or more references share the same rank, the reference with the longest chromosome is selected. Note species are identified using species TaxIDs. This means that assemblies with different species names but the same species TaxIDs are considered as one species.

– **One per genus**. One reference is selected for each genus in the chosen lineage(s). The chosen genus representative is selected based on ranking with Reference genomes > Representative genomes > Complete genomes > Scaffolds > Contigs. When two or more references share the same rank, the reference with the longest chromosome is selected.

Click **Finish**.

If **Skip Database Builder** was selected, all genomes matching the specified criteria will now be downloaded. If the enabled ID or Taxonomy matching field was left empty, no genomes will be downloaded.

If **Skip Database Builder** was left unchecked, a reference database is not downloaded right away. Instead, the Database Builder will open, see section 16.2.1.

16.2.1 Database Builder

Depending on your internet connection, it takes a few seconds to download the content and open the **Database Builder** (figure 16.4).

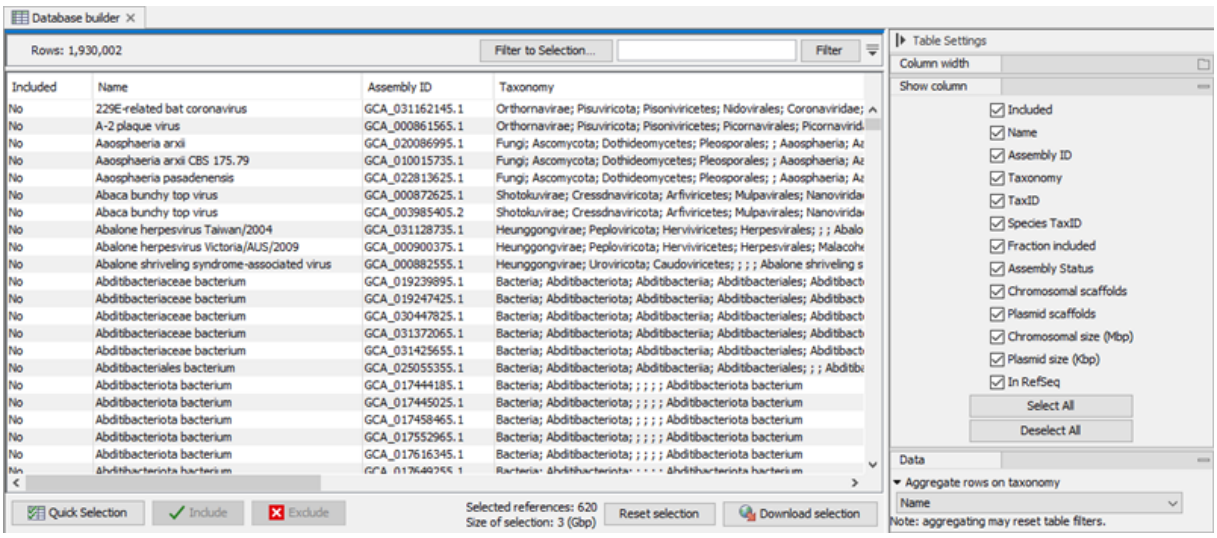


Figure 16.4: *Search, filter and select assemblies to download.*

Assemblies that match the criteria from the Download Custom Microbial Reference Database tool will be pre-selected, indicated by a "Yes" in the *Included* column.

The **Database Builder** table contains additional columns with metadata based on information from GenBank, https://www.ncbi.nlm.nih.gov/genbank/. Use the **Database Builder** functionality described below to customize and define the reference set to be downloaded.

Use the filtering options located at the top right to filter the table. For information on how to use the simple and advanced table filters, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Filtering_tables.html.

From the **Side Panel** on the right, the following option is available:

- **Aggregate rows on taxonomy**. Aggregates results by the selected taxonomic level, e.g.

Order.

Below the table you find buttons for quick selection, including or excluding rows, and download of selected reference subset:

- **Quick selection**. For selection of one of the following predefined subsets, based on information in the *Assembly Status*, *Chromosomal scaffolds*, and *In RefSeq* columns:
  - **Single scaffold complete genomes in RefSeq**. Complete genomes with *Chromosomal scaffolds*= 1; *In RefSeq*= Yes.
  - **Complete genomes in RefSeq**. Complete genomes with *In RefSeq*= Yes.
  - **All complete genomes**. Any Complete genome.

  "Complete genome" refers to the *Assembly Status*. All genomes marked as Complete genome or Chromosome are included in the subsets, as are any reference marked as representative genome (repr) or reference genomes (refr).
- **Include** and **Exclude**. Includes or excludes the selected rows from the subset selection.
- **Reset selection**. Reset selection to match criteria specified in **Download Custom Microbial Reference Database** wizard.
- **Download selection**. For download of the selected reference subset. Brings up a dialog with the following options (figure 16.5):
  - **Include all annotation tracks**. Will include CDS, gene, etc. annotations in the downloaded database. The annotations are not needed for taxonomic profiling, but may be required for other applications such as creating MLST schemes.
  - **Minimum contig length**. The minimum length of sequences to be included in the database.

    The dialog provides an estimate of the memory and disk requirements needed to later run the Taxonomic Profiling tool with the database you are about to download.

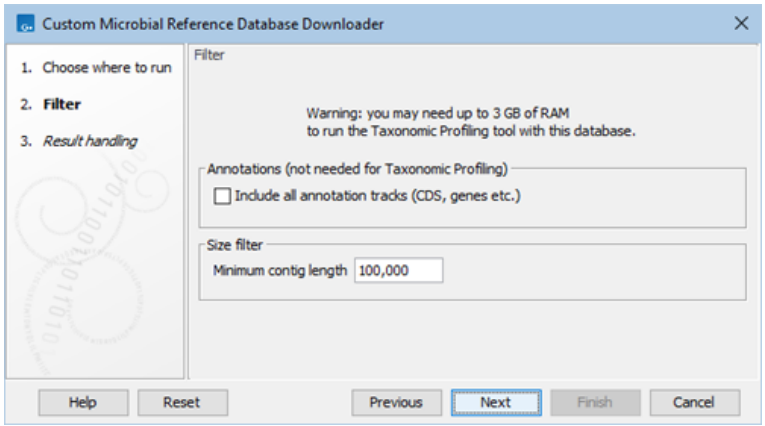


Figure 16.5: *Filter options for download of the selected references.*

16.3 Download Pathogen Reference Database

Download a collection of bacterial assemblies and enrich with metadata from the NCBI Pathogen Detection Project (see https://www.ncbi.nlm.nih.gov/pathogens/).

Tools | **Microbial Genomics Module** ( ) | **Databases** ( ) | **Taxonomic Analyses** ( ) | **Download Pathogen Reference Database** ( )

This will open the following wizard window (figure 16.6):

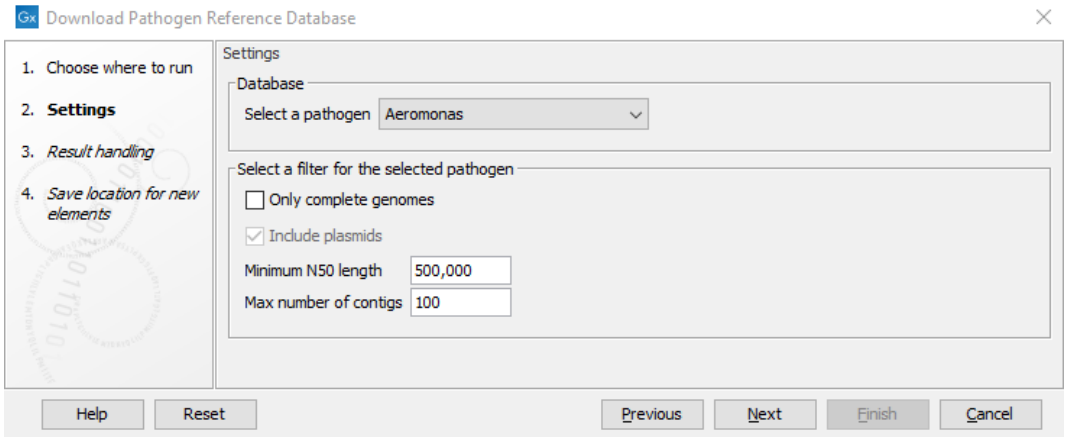


Figure 16.6: *Downloading assemblies and metadata for a selected pathogen from the NCBI Pathogen Detection Project.*

The settings are:

- **Select a pathogen**. Select a pathogen for which to download assemblies and associated metadata.
- **Only complete genomes**. This can be used to switch between complete genomes or to also allow for downloading incomplete assemblies.
- **Include plasmids**. This option can be used to include or exclude plasmids from the downloaded database. Note that if a database of plasmids only is required, the Download Custom Microbial Reference Database tool should be used instead.
- **Minimum N50 length**. This option can be used to remove assemblies with shorter N50 values (the default value is set at 500,000 bp). Short N50 values typically indicate low assembly quality. This option is not available when "Only complete genomes" has been selected.
- **Maximum number of contigs**. This option can be used to remove assemblies with a higher number of contigs (the default value is set at 100). Many contigs typically indicate low assembly quality. This option is not available when "Only complete genomes" has been selected,

Specify a location to save the database. We recommend to create a folder where you can save all the databases and MLST schemes necessary to run some of the CLC Microbial Genomics Module tools.

The resulting database includes a list of different bacterial genome sequences as well as the associated accession numbers, descriptions, taxonomy and size of the sequences. In addition, each reference genome will be annotated with the following metadata (when available):

- serovar
- strain
- taxonomy
- sample collection date
- geographical location
- isolation source
- host
- host disease
- outbreak
- SRA run id
- SRA project id

16.4 Create Whole Metagenome Index

This tool will generate a whole metagenome index from a reference database. This index type is used by the Classify Whole Metagenome Data tool (section 6.3).

It is recommended to mask repetitive sequences in the reference database (host genome excluded) with **Mask Low-Complexity Regions** (section 20.1) prior to running **Create Whole Metagenome Index**.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Databases** () | **Taxonomic Analysis** () | **Create Whole Metagenome Index** ()

Select a sequence list containing the references and the potential host genome of interest.

All sequences must have a taxonomy attribute. Sequences and their associated taxonomy can be downloaded, for example, using **Download Custom Microbial Reference Database** (section 16.2).

The output includes a whole metagenome index file and an optional report. The report provides a summary of the number of features at each taxonomic level, as well as information about the number of sequences and bases that were indexed (figure 16.7). "Number of genomes" is the number of unique assembly IDs. Sequences without assembly IDs are counted as a unique genome per sequence.

Depending on the size of the database, the tool may require a significant amount of free temporary disk space, see (section 1.3).

The tool allows for creating indexes containing up to 65,535 taxonomic nodes, with each node corresponding to a taxonomic classification. For example, the index in figure 16.7 includes a total of 33,368 taxonomic classifications, or nodes.

1 Taxonomic summary

Taxonomic level	Number of classifications
Kingdom	21
Phylum	92
Class	212
Order	409
Family	974
Genus	5,223
Species	26,437

2 Database summary

Number of genomes	63,891
Number of sequences	152,510
Total length (bp)	217,471,214,247

Figure 16.7: *The Create Whole Metagenome Index report.*

16.5 Create Taxonomic Profiling Index

This tool will generate a taxonomic profiling index from a reference database. Taxonomic profiling indexes are used as input for e.g., the Taxonomic Profiling tool (section 6.4) and the Assign Taxonomies to Sequences in Abundance Table tool (section 7.3).

The computation of index files for taxonomic profiling is memory and hard-disk intensive due to the large sizes of reference databases usually employed for this task. The algorithm requires roughly the number of bases in bytes of memory, i.e., approximately the size of the uncompressed reference database; and twice this amount in hard disk space.

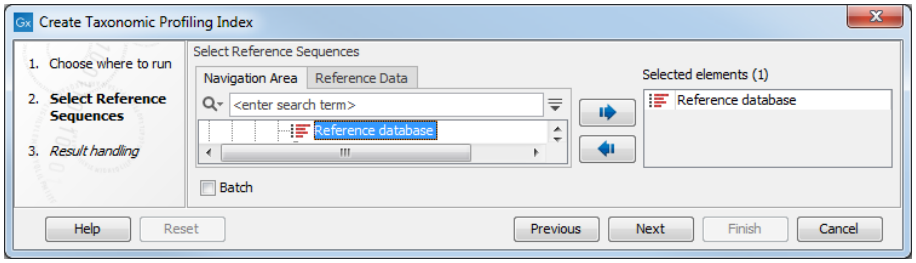


Figure 16.8: *Select sequence lists with the references of interest.*

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **Taxonomic Analysis** ( ) | **Create Taxonomic Profiling Index** ( )

Select one or more sequence lists containing the references of interest. These can be downloaded for example using **Download Custom Microbial Reference Database** (section 16.2).

The tool makes use of Assembly IDs (see section 22) in combination with either Latin name or, if Latin name is not present, Sequence name. The tool will treat sequences as one reference, if they have:

- Identical Assembly ID *and* same Latin name, or
- Identical Assembly ID *and* same unique sequence name

The output is a taxonomic profiling index and an optional report as seen in figure 16.9. The report lists the number of sequence and base pairs that were indexed.

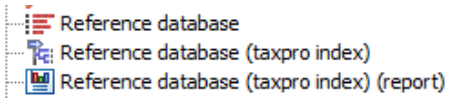


Figure 16.9: *The reference sequences, index and report as seen in the Navigation Area.*

Chapter 17

Databases for Functional Analysis

17.1 Download Protein Database

The **Download Protein Database** allows you to download the following protein databases:

- **Clusters of Orthologous Genes (COG)**
- **SwissPROT (with GO-term annotations)**
- **UniProt (UniRef50 complete with GO & EC annotations)**. Entries are annotated with Enzyme Commission (EC) numbers and GO-terms where available. Notice, that entries with GO-terms and no EC terms are not included.
- **UniProt (UniRef90 subset with GO & EC annotations)**. This is the subset of entries containing Enzyme Commission (EC) numbers and GO terms. Notice, that entries with GO-terms and no EC terms are not included.

These protein databases can be used to create DIAMOND Indexes (see section 17.4), that may be used together with the Annotate CDS with Best DIAMOND Hit tool (see section 12.5) and Annotate with DIAMOND tool (see section 12.3).

Notice, that the **SwissPROT** and **UniProt (UniRef50)** protein reference databases have been annotated with GO associations from the EBI Gene Ontology Annotation (GOA) Database (https://www.ebi.ac.uk/GOA/index).

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Databases** () | **Functional Analysis** () | **Download Protein Database** ()

Choose the database you wish to download from the drop-down menu, and when needed, accept the terms of use before clicking Finish to save the database in the Navigation Area.

17.2 Download Ontology Database

The **Download Ontology Database** tool allows you to download the latest versions of:

- the GO database and of Pfam2GO mappings from the Gene Ontology Consortium (https://geneontology.org/).
- the Enzyme Commission number database from Expasy (https://enzyme.expasy.org/).

The GO database is used to convert Pfam annotation to GO terms by the Annotate CDS with Pfam Domains tool (see section 12.6) and by the Build Functional Profile tool (see section 12.7).

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **Functional Analysis** ( ) | **Download Ontology Database** ( )

If you select **Create Report**, the tool will also generate a summary report table. For each downloaded file, the table will contain the name of the downloaded file, its size, the URL from which is was downloaded, and the number of entries in the file.

17.2.1 The GO Database View

When downloading the GO database, a new object called *GO database* ( ) is created in the Navigation Area.

The GO Database element has a default hierarchical tree view (See figure 17.1). Here it is possible to search the different ontology terms, and see their relations.

It should be noted that the Gene Ontology is not a tree, but a directed acyclic graph - this means that a GO-term may have multiple parents, and thus appear different places in the tree view. Likewise, when searching for a specific GO-term, multiple locations in the tree might be highlighted, even though they refer to the same GO-term.

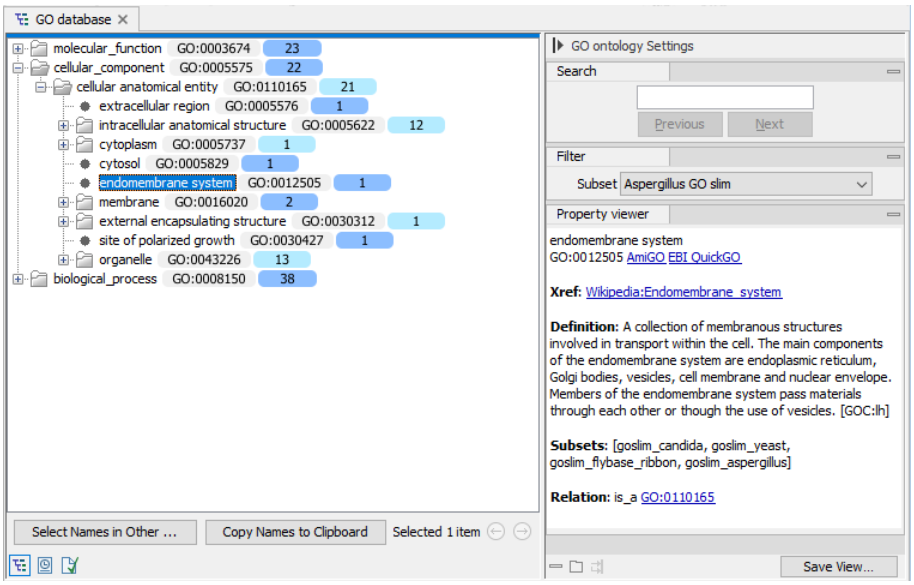


Figure 17.1: *The GO Database View*

In figure 17.2 an example of a search result is shown. Here, there are multiple GO terms matching the search query 'carbon'. The **Previous** and **Next** buttons can be used to navigate to the matching GO terms. Notice, that this particular GO-term (GO:0106148) is shown multiple

times in the tree ('23 instances in tree'). In order to view the different locations in the tree for a matching GO-term, the arrow buttons at the bottom of the view can be used to focus on the different selected elements (see figure 17.3) - note, that this does not change the selection, it only changes the focused area of the tree.

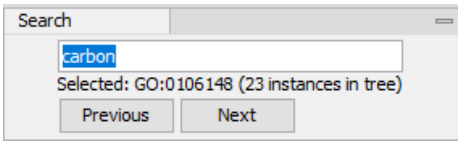


Figure 17.2: *Searching for GO terms.*

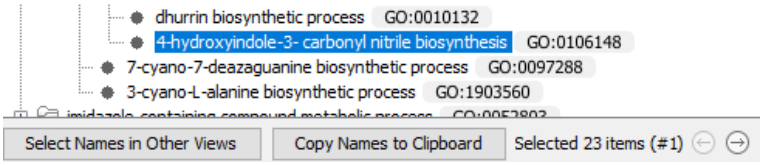


Figure 17.3: *Scrolling through different selected items in the GO view.*

The **Filter** sidepanel section, can be used to restrict to the view to various subsets of the full database ("Slim" subsets). It is also possible to restrict the tree to only the terms that are currently selected.

When clicking on a GO term, the Property viewer in the side panel will show the description, relations, synonyms, and all related links.

It is possible to **Select Names in Other Views** and **Copy Names to Clipboard** (see figure 17.3). Selecting names in other views will match names in other editors that support this - currently, this is only supported in the Differential Abundance element view.

17.2.2 The EC Database View

When downloading the EC database, a new object called *EC database* ( ) is created in the Navigation Area.

Similar to the GO database, the EC database has a hierarchical tree view (See figure 17.4).

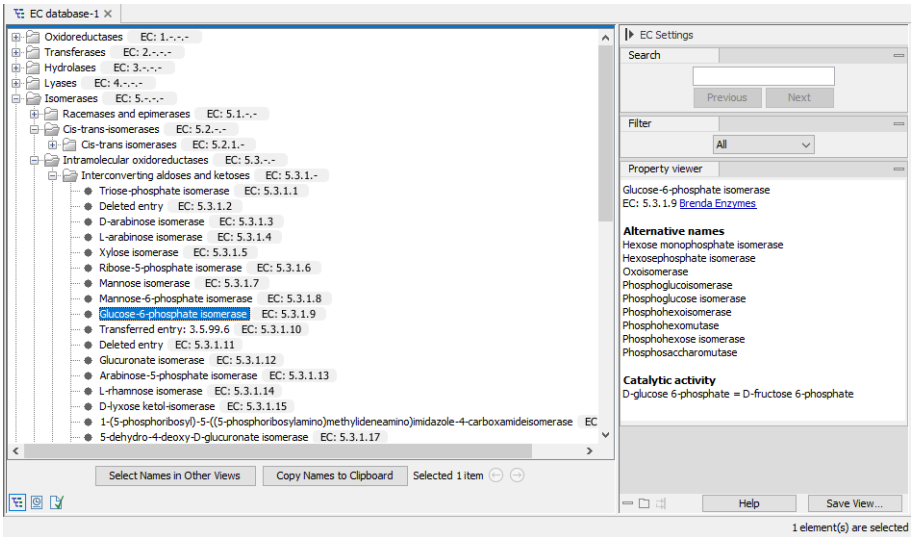


Figure 17.4: *The EC Database View*

The **Search** field can be used to search in EC term names and descriptions. The **Previous** and **Next** buttons can be used to navigate to the matching EC terms.

When multiple items are selected, the arrow buttons at the bottom of the view can be used to focus on the different selected elements - note, that this does not change the selection, it only changes the focused area of the tree.

It is possible to **Select Names in Other Views** and **Copy Names to Clipboard** (see figure 17.4). Selecting names in other views will match names in other editors that support this - currently, this is only supported in the Differential Abundance element view.

The **Filter** side panel section can be used to restrict the view to the terms that are currently selected. This is in particular useful, when EC terms from an another editor (such as the Differential Abundance element view) have been selected.

When clicking on a EC term, the Property viewer in the side panel will show a description together with related links to more information.

17.3 Download Pathway Database

The Download Pathway Database tool allows you to download pathway databases for use with the Identify Pathways tool.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Functional Analysis** ( ) | **Download Pathway Database** ( )

In the first wizard step, select the pathway database to download. Currently only one database is available:

- MetaCyc Pathway Database: A multi-organism database of pathways involved in primary and secondary metabolism, as well as associated metabolites, reactions, enzymes, and genes. The MetaCyc database consists of pathways included in the MetaCyc BioPAX Level 3 file [Caspi et al., 2019] (`https://metacyc.org/`).

17.3.1 The Pathway Database

A pathway database is a clc object that contains information about pathways, specifically which functional terms are present in each pathway, e.g. for MetaCyc each pathway is associated with a set of EC numbers. These EC numbers form the basis for pathway calling from an abundance table or a differential abundance table.

The pathway database has three views:

- The Pathway table ( )
- The Compound table ( )
- The Enzyme table ( )

The Pathway table The pathway table is a list of all pathways in the database and it has two columns, Name and MetaCyc ID, giving the name of a pathway and its MetaCyc ID, which is also a link to the relevant online site. The pathway table has two buttons at the bottom

- **Open Pathway View**: Opens a split view with a visualization of the selected pathways. Note that double clicking on a single line in the Pathway table is identical to selecting that line and clicking on Open Pathway View.
- **Create New Pathway Database**: Creates a new database only containing the selected pathways. This is useful, e.g. for constructing a species specific database

The Compound table The compound table is a list of all compounds, including side compounds, in the database. This table has three columns,

- **Name**: The name of the compound.
- **MetaCyc ID**: The compound's MetaCyc ID which is a link to the compound's description on the MetaCyc homepage.
- **Cellular location**: The location of the compound in the cell. Some names occur several times as their cellular locations are different, e.g. there are protons in the cytosol, outside of the cell and inside the lumen of some organelles.

Furthermore, the table has two buttons in the bottom

- **Open Pathway View from Selected Compound(s)**: Opens a split view with a visualization of the pathways in which the selected compounds occur. Note that double clicking on a single line in the Compund table is identical to selecting that line and clicking on Open Pathway View from Selected Compound(s).
- **Create New Pathway Database**: Creates a new pathway database containing only pathways in which the selected compounds occur.

The Enzyme table The enzyme table is a list of all enzymes in the database. This table has three columns,

- **Name**: the name of the Enzyme. Note that some names occur several times because they reference different genes.
- **MetaCyc ID**: the enzyme's MetaCyc ID which is a link to the enzyme's description on the MetaCyc homepage.
- **Cellular location**: the location of the enzyme in the cell. Some names occur several times as their cellular locations are different, e.g. there are protons in the cytosol, outside of the cell and inside the lumen of some organelles.

Furthermore, the table has two buttons in the bottom

- **Open Pathway View from Selected Enzyme(s)**: Opens a split view with a visualization of the pathways in which the selected enzymes occur. Note that double clicking on a single line in the Enzyme table is identical to selecting that line and clicking on Open Pathway View from Selected Enzyme(s).
- **Create New Pathway Database**: Creates a new pathway database containing only pathways in which the selected enzymes occur.

17.3.2 The Pathway View

When opening one or several pathways in Pathway View, the pathways may be explored visually. Note that the pathway view has two editors

- The Pathway Graph View ()
- The Text Contents View ()

The Pathway Graph View The pathway graph view graphically shows a biochemical pathway in the form of a simplistic textbook style pathway drawing, where each reaction is labeled by one or several EC numbers in black rounded boxes, and reaction arrows connect the reactants, products and EC numbers. As reactants, only the main compounds are shown, side-compounds are ignored in the visualization. All reactants are visible in the property view when selecting an EC number in the pathway view. This information is also displayed as a tool tip when hovering over the EC number.

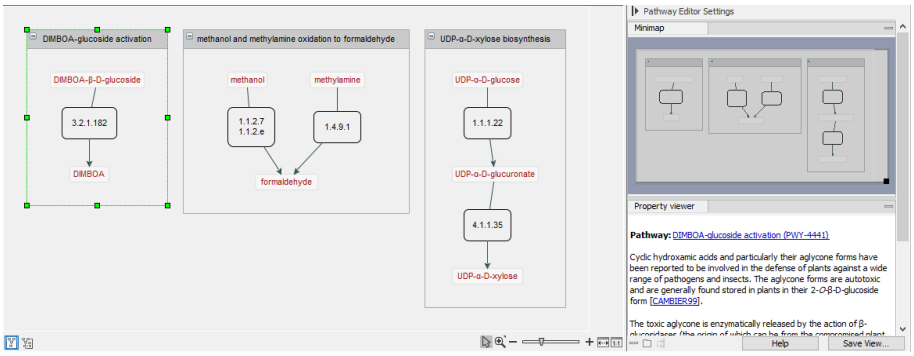


Figure 17.5: *The Pathway view*

Upon clicking an item in the pathway graph additional information about that element will be displayed in the Property Viewer of the side panel.

The Text Contents The text contents view of a pathway contains a textual summary of the pathways including scientific references.

17.4 Create DIAMOND Index

This tool will compute a DIAMOND index from a protein database. These indexes can then be used as input to the Annotate CDS with Best DIAMOND Hit tool (see section 12.5) and Annotate with DIAMOND tool (see section 12.3).

If the input sequences contain metadata, such as GO-terms, these will be transferred to the created index.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **Functional Analysis** ( ) | **Create DIAMOND Index** ( )

In the first dialog, select a protein database downloaded with the Download Protein Database tool (figure 17.6).

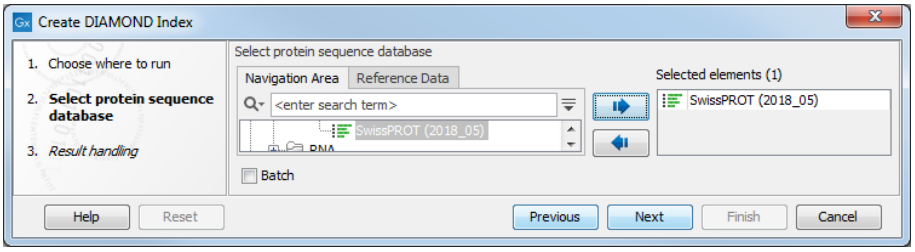


Figure 17.6: *Select a protein database.*

The output is an index file as seen in figure 17.7.

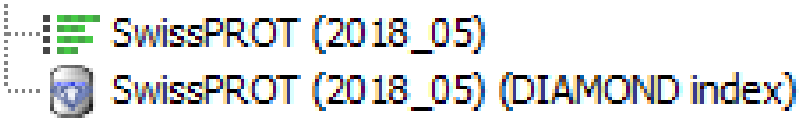


Figure 17.7: *The protein database and its index as seen in the Navigation Area.*

The default view for the DIAMOND index is a tabular overview of the sequence entries and their associated metadata, such as GO-terms (see figure 17.8).

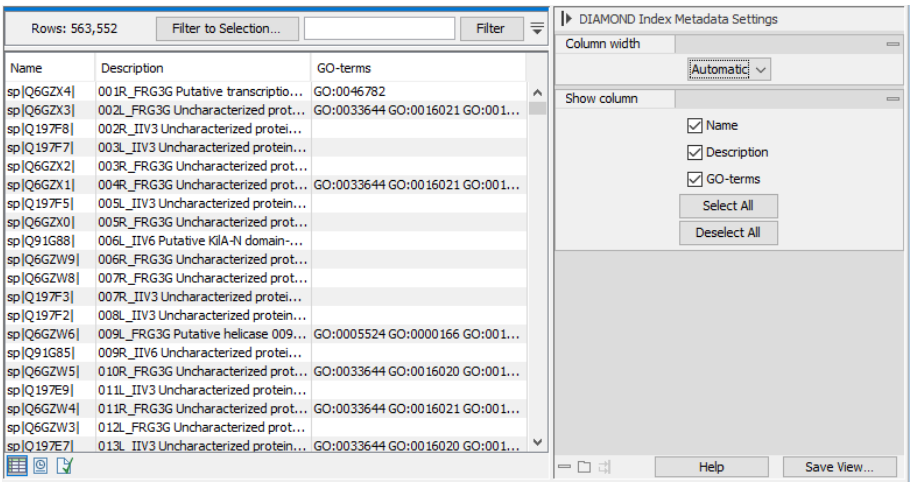


Figure 17.8: *The default view for the DIAMOND index element.*

17.5 Import RNAcentral Database

This tool can be used to import non-coding RNA sequences from RNAcentral, and join the sequences with functional Gene Ontology information.

The imported sequences can then be used together with the Annotate with BLAST tool (see section 12.2) and the Build Functional Profile tool (see section 12.7) to quantify the functional

annotation abundances.

The Import RNAcentral Database tool uses a special FASTA importer that allows for non-standard nucleotides (RNAcentral includes sequences with non-standard IUPAC nucleotide symbols, which are not allowed by our standard FASTA importer).

The tool can also import RNAcentral files with associations to GO-terms, such as 'rnacentral_rfam_annotations.tsv.gz', and match the entries with those in the imported sequence list.

Before running the tool, it is necessary to download the relevant sequences and GO-associations from RNAcentral (https://rnacentral.org/). To get the full set of annotations, we recommend downloading the following files:

RNAcentral FASTA sequences: ftp://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/sequences/rnacentral_active.fasta.gz

RNAcentral GO Associations (from RFAM): ftp://ftp.ebi.ac.uk/pub/databases/RNAcentral/current_release/go_annotations/rnacentral_rfam_annotations.tsv.gz

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Databases** () | **Functional Analysis** () | **Import RNAcentral Database** ()

In the tool dialog (figure 17.9), select the files downloaded as described above.

It is also possible to select whether to include only RNAcentral sequences with matching GO associations, which will reduce the size of the created database.

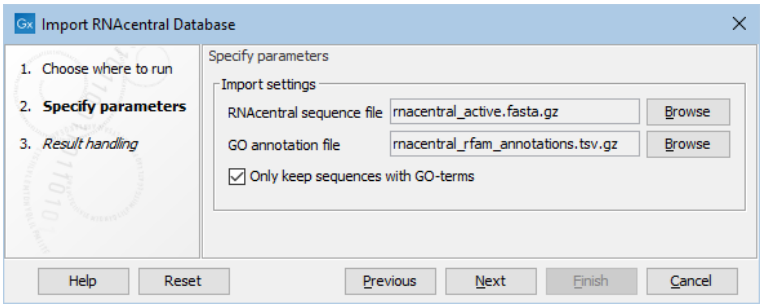


Figure 17.9: *The Import RNAcentral Database tool options.*

RNAcentral identifiers may contain a species-specific suffix (e.g. URS0000000006_1317357 - here 1317357 is an NCBI Taxonomy ID). When we perform the matching of RNAcentral sequences to GO associations these are stripped off and ignored.

17.6 Import PICRUSt2 Multiplication Table

The Import PICRUSt2 Multiplication Table (beta) tool can be used to import multiplication tables from PICRUSt2 [Douglas et al., 2020] in order to perform functional inference for OTU abundance tables using **Infer Functional Profile (beta)** (section 12.8), or to normalize OTU abundance tables by rRNA copy numbers using **Normalize OTU Table by Copy Number (beta)** (section 5.1).

Before running the tool it is necessary to download the relevant data files from the PICRUSt2 github repository (https://github.com/picrust/picrust2/tree/master/picrust2/default_files), specifically three kinds of files are required:

1. Files with 16S, 18S or ITS sequence alignments
    - 16S alignments or prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/pro_ref/pro_ref.fna)
    - 18S alignments for fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/fungi_18S/fungi_18S.fna.gz)
    - ITS alignments for fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/fungi_ITS/fungi_ITS.fna.gz)
2. Files with rRNA copy numbers
    - 16S rRNA copy numbers for prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/16S.txt.gz)
    - 18S rRNA copy numbers for fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/18S_counts.txt.gz)
    - ITS rRNA copy numbers for fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/ITS_counts.txt.gz)
3. Files with functional term counts associated with each type of rRNA
    - EC terms associated with 16S regions in prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/ec.txt.gz)
    - Kegg orthology terms associated with 16S regions in prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/ko.txt.gz)
    - COG terms associated with 16S regions in prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/cog.txt.gz)
    - Pfam domains associated with 16S regions in prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/pfam.txt.gz)
    - TIGRFAM terms associated with 16S regions in prokaryotes (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/prokaryotic/tigrfam.txt.gz)
    - EC terms associated with 18S regions in fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/ec_18S_counts.txt.gz)
    - EC terms associated with ITS regions in fungi (https://github.com/picrust/picrust2/blob/master/picrust2/default_files/fungi/ec_ITS_counts.txt.gz)

Only files corresponding to the same rRNA regions can be combined to obtain a valid PICRUSt2 Multiplication Table, e.g. 16S alignments for prokaryotes, 16S rRNA counts and COG terms associated with 16S regions in prokaryotes.

Note that the rRNA copy numbers for fungi 2 are not consistent for 18S and ITS regions, which may have implications for the normalization and thus also for the functional inference for fungal data.

The tool can import similarly prepared data if other data sources are available. The OTU sequences need not be aligned.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** ( ) | **Databases** ( ) | **Functional Analysis** ( ) | **Import PICRUSt2 Multiplication Table (beta)** ( )

In the tool dialog (figure 17.10), select which type of rRNA and which type of terms you would like to import, then select the corresponding three files downloaded above where

- File with aligned rRNA sequences: takes fasta files as input, e.g. one of the files listed under point 1.
- File with rRNA copy numbers: takes a tab separated text file with two columns as input, where the first column contains the name of an rRNA sequence from the fasta file and the second column the corresponding rRNA copy number, e.g. one of the files listed under point 2 or a fasta file with unaligned rRNA sequences. The file is expected to contain a header.
- File with functional counts: takes a tab separated text file as input. The first column contains the name of an rRNA sequence from the fasta file and the remaining columns contain the corresponding functional counts, where each column is identified with a functional term via the header line, e.g. one of the files listed under point 3.

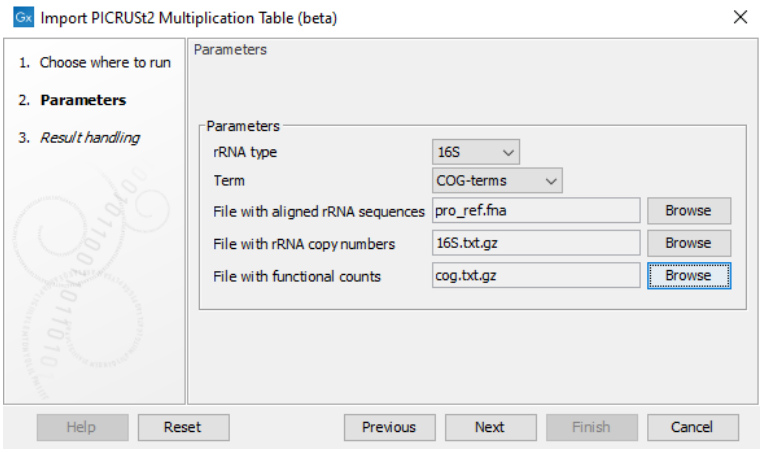


Figure 17.10: *The Import PICRUSt2 Multiplication Table (beta) tool options.*

After import the data in the multiplication table is displayed in a table (figure 17.11) where the name of the rRNA is given under the "Assembly" column, as these tables are typically derived from assemblies with known rRNA content, taxonomy and functional counts. The following four columns list the rRNA copy numbers registered for each type of rRNA, ITS regions will be listed as the selected rRNA type, e.g. 18S or 28S and the number of distinct funtional terms registered for that assembly in the column "Number of terms".

When selecting one or several rows in the upper table, the lower table will show the combined functional counts for the selected row for each of the functional terms individually.

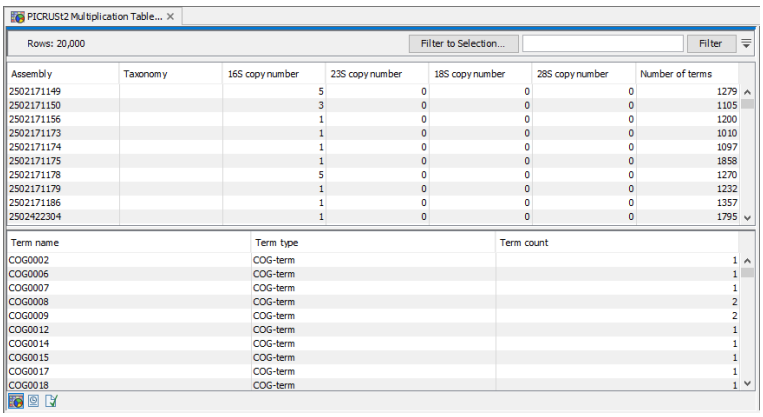


Figure 17.11: *The PICRUSt2 Multiplication Table visualization.*

Chapter 18

Databases for Drug Resistance Analysis

18.1 Download Resistance Database

Download Resistance Database enables download of databases for use with the Find Resistance with Nucleotide Database, Find Resistance with PointFinder and Find Resistance with ShortBRED tools.

To run the tool, go to:

> **Tools** | **Microbial Genomics Module** () | **Databases** () | **Drug Resistance Analysis** () | **Download Resistance Database** ()

The available databases fall into four categories:

ShortBRED Marker Databases

These databases can be used with **Find Resistance with ShortBRED** (section 13.3). The databases are marker databases, containing peptide fragments that uniquely characterize sets of similar proteins, rather than a specific gene.

- **QMI-AR Peptide Marker Database**. The QIAGEN Microbial Insight - Antimicrobial Resistance database is a curated database containing peptide markers derived from the following source databases: CARD [Alcock et al., 2023] (https://card.mcmaster.ca/), ARG-ANNOT [Gupta et al., 2014] (https://www.mediterranee-infection.com/acces-ressources/base-de-donnees/arg-annot-2/), NCBI Bacterial Antimicrobial Resistance Reference Gene Database (previously BioProject PRJNA313047, now integrated in the Pathogen Detection Reference Gene Catalog, https://www.ncbi.nlm.nih.gov/pathogens/refgene/), ResFinder (https://bitbucket.org/genomicepidemiology/resfinder_db/src/master/).
- **CARD Peptide Marker Database**. Peptide markers derived from the Comprehensive Antibiotic Resistance Database [Alcock et al., 2023] (https://card.mcmaster.ca/).
- **ARG-ANNOT Peptide Marker Database**. Protein markers from the ARG-ANNOT database [Gupta et al., 2014] (https://www.mediterranee-infection.com/acces-ressources/base-de-donnees/arg-annot-2/).

Nucleotide Databases

These databases can be used with **Find Resistance with Nucleotide Database** (section 13.2). The databases contain nucleotide gene sequences.

- **QMI-AR Nucleotide Database**. The QIAGEN Microbial Insight - Antimicrobial Resistance database is a curated database containing nucleotide sequences compiled from the following source databases: CARD [Alcock et al., 2023] (https://card.mcmaster.ca/), ARG-ANNOT [Gupta et al., 2014] (https://www.mediterranee-infection.com/acces-ressources/base-de-donnees/arg-annot-2/), NCBI Bacterial Antimicrobial Resistance Reference Gene Database (previously BioProject PRJNA313047, now integrated in the Pathogen Detection Reference Gene Catalog, https://www.ncbi.nlm.nih.gov/pathogens/refgene/), ResFinder (https://bitbucket.org/genomicepidemiology/resfinder_db/src/master/).
- **VFDB**. Nucleotide sequences compiled from the Virulence Factor database core dataset (https://www.mgc.ac.cn/VFs/main.htm).
- **CARD Nucleotide Database**. Nucleotide sequences compiled from the Comprehensive Antibiotic Resistance Database (CARD) (https://card.mcmaster.ca/).
- **ResFinder Nucleotide Database**. Nucleotide sequences from the ResFinder database (https://bitbucket.org/genomicepidemiology/resfinder_db/src/master/).

Point Mutation Databases

These databases are used with **Find Resistance with PointFinder** (section 13.1).

- **PointFinder**. Organism-specific databases containing wild type genes and known resistance conferring mutations [Zankari et al., 2017] (https://bitbucket.org/genomicepidemiology/).

18.2 Reference Data Elements

Some elements are available from the **QIAGEN Sets** tab in the Reference Data Manager (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=References_management.html).

- **WHO Mycobacterium Tuberculosis.** The WHO database is a variant track with all variants from the WHO's Mycobacterium tuberculosis mutation catalogue [WHO, 2023]. Variants called with the variant detection tools (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Variant_Detection_tools.html) can be matched against this database using **Filter against Known Variants** or **Annotate from Known Variants**.

  The variant table contains five columns unique to this reference data element:

  - **Graded variant.** The name(s) of the variant as given by WHO. The name consists of the gene the variant is in, along with the corresponding position and change, either as a nucleotide or amino acid change.
  - **Drug.** The antimicrobial resistance drug(s) for which the variant is graded.
  - **Gene.** The gene with which the variant is associated, as given by WHO.

- **Grade.** The grade of drug resistance determined for the variant. See figure 18.1 for a schematic of the resistance grades.
- **Comment.** In case any comments on the variants resistance are given by WHO, these are listed here.

A single variant may be associated with several of WHO's "Graded variant" names, in which case all are listed, separated by commas. In that case, the "Drug" and "Grade" columns will contain the same number of comma-separated elements, and the order of elements will be equivalent across the three columns.
In some cases of variants being associated with multiple "Graded variant" names, not all of the names are graded for a drug. This leads to instances of N/A in the "Drug" column. In the "Grade" column, this will be listed as "Ungraded". See figure 18.2 for an example of this.

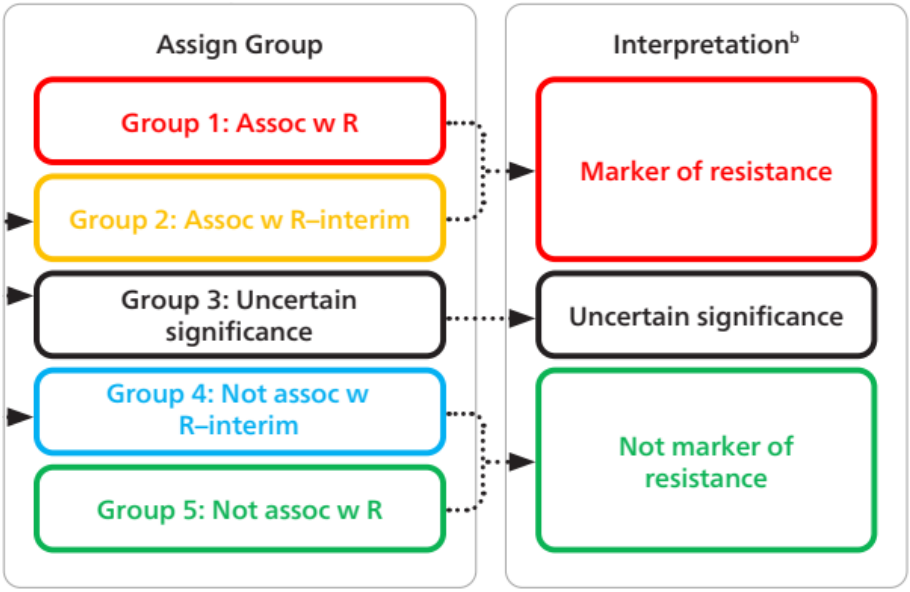


Figure 18.1: *The five levels of resistance as given by WHO (from fig. 1, page 5 of [WHO, 2023]).*

Chromosome	Region	Type	Graded variant	Drug	Grade
NC_000962	4247431..4247440	MNV	embB_c.921C>G, embB_c.924A>G, embB_c.927C>G, embB_p.Met306Ile	N/A, Ethambutol, N/A, Ethambutol	Ungraded, 4) Not assoc w R - Interim, Ungraded, 1) Assoc w R

Figure 18.2: *An example of a single variant with multiple instances of "Graded variant". In this case the third "Graded variant", embB_c.927C>G, is not graded for a drug, so the third "Drug" is N/A and the third "Grade" is Ungraded. The last "Graded variant", embB_p.Met306Ile, is graded for Ethambutol and has received a grade of 1.*

Part VIII

Panel Support

Chapter 19

QIAseq 16S/ITS Demultiplexer

The Panel Support section offers a tool to demultiplex NGS reads of different bacterial variable and fungal ITS regions obtained with the QIAGEN QIAseq 16S/ITS Screening and Region panels. Using this tool, sequences are associated with a particular region when they contain a match to a particular barcode. Sequences that do not contain a match to any of the barcode sequences provided are classified as not grouped.

To run the tool, go to:

> **Microbial Genomics Module** () | **Panel Support** () | **QIAseq 16S/ITS Demultiplexer** ()

In the first dialog, window select the reads you wish to demultiplex and click **Next**. It is possible to run the tool in batch mode.

In the second dialog, select the barcodes used to demultiplex the sequences, from a predefined list (see the list in figure 19.1) or from a custom list. Select only barcodes corresponding to the primers used for library construction.

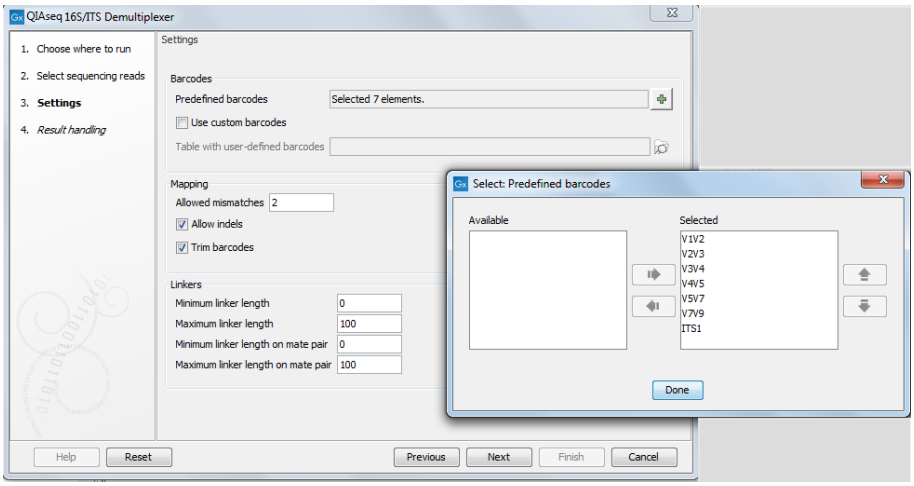


Figure 19.1: *Set the parameters to demultiplex.*

If you choose to use a table of custom barcodes (figure 19.2), you need to specify an Excel or a CSV file previously saved in the Navigation Area. The table will be different when setting barcodes for single or paired reads: for single reads, the first column defines the barcode name,

the second contains the barcode sequence. For paired reads, an additional third column contains the reverse complement of the barcode sequence.

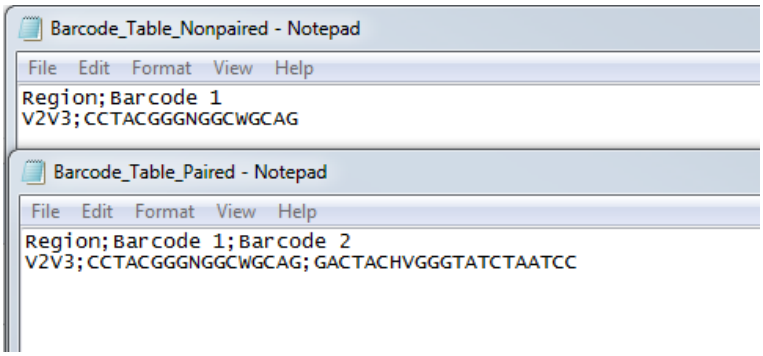


Figure 19.2: *Examples of CSV custom barcodes files for paired and single reads.*

The following parameters for demultiplexing are also available in this dialog:

- Mapping
  - Allow mismatches: decide how many mismatches are allowed between the sequence and the barcode
  - Allow indels
  - Trim barcodes
- Linkers: also known as adapters, linkers are sequences which should just be ignored - it is neither the barcode nor the sequence of interest. For this element, you simply define its length.
  - Minimum linker length
  - Maximum linker length
  - Minimum linker length on mate pair
  - Maximum linker length on mate pair

In the Result handling window, you can choose to Create a report (see figure 19.3), and Save a sequence list of all ungrouped sequences.

The main output are sequence lists for each different regions/barcodes. These sequence lists can be used as input for the Data QC and OTU Clustering workflow, that will generate an output table displaying the OTUs abundances for each region. Note that the Trim Reads step of the workflow will automatically detect and trim the remaining read-through barcodes found on paired-end reads and not discarded by the demultiplexer. However, if you are working with single reads, mate-paired reads or data of low quality, it is recommended to specify a trim adapter list containing all barcodes in the Trim Reads step of the workflow.

1.1 Reads per region: Table

Region	Barcode	Number of reads	Percentage of reads
V1V2	AGRGTTTGATYMTGGCTC-CTGCTGCCTYCCGTA	36,381	7%
V2V3	GGCGNACGGGTGAGTAA-WTTACCGCGGCTGCTGG	124,486	23%
V3V4	CCTACGGGNGGCWGCAG-GACTACHVGGGTATCTAATCC	92,981	17%
V4V5	GTGYCAGCMGCCGCGGTAA-CCGYCAATTYMTTTRAGTTT	61,912	11%
V5V7	GGATTAGATACCCBRGTAGTC-ACGTCRTCCCCDCCTTCCTC	119,161	22%
V7V9	YAACGAGCGMRACCC-TACGGYTACCTTGTTAYGACTT	78,053	14%
ITS1	CTTGGTCATTTAGAGGAAGTAA-GCTGCGTTCTTCATCGATGC	7,028	1%
Ungrouped		31,479	6%

1.2 Reads per region: Barplot

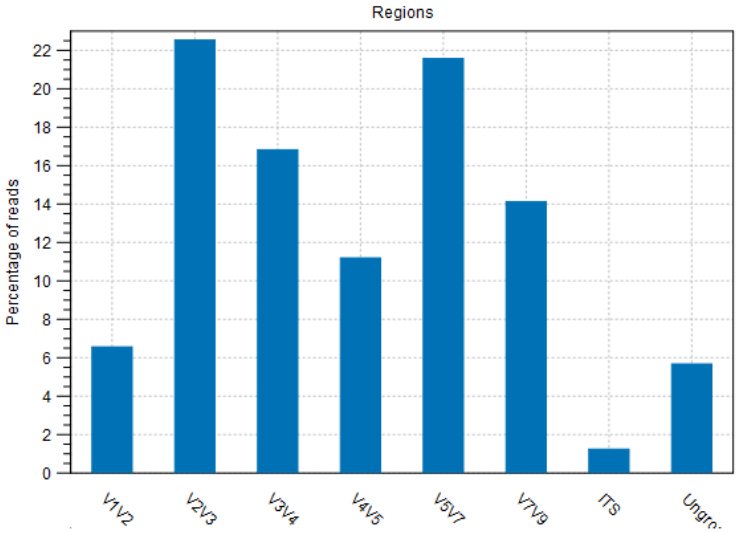


Figure 19.3: *An example of demultiplexing report.*

Part IX

Utility Tools

Chapter 20

Utility Tools

20.1 Mask Low-Complexity Regions

The **Mask Low-Complexity Regions** tool can be used to identify and mask repetitive regions in sequences. In some cases this can remove erroneous matches: for instance, when doing taxonomic profiling, a read with a highly repetitive sequence is likely to match a reference genome purely by chance.

The tool takes any sequence or sequence list as input (including reads and genomes). It will accept both nucleotide and protein sequence input.

To run the tool, go to

Tools | **Utility Tools** ( ) | **Mask Low-Complexity Regions ( )**

The following general options are available (figure 20.1):

- **Window size**: The complexity is evaluated by moving a window along the sequences. This option sets the sliding window size.
- **Window stride**: The number of nucleotides by which the window is moved along the sequence. Increasing this value makes the tool faster, but slightly less accurate when detecting the edges of low-complexity regions.
- **Low-complexity threshold**: This measure is normalized such that a value of 0 corresponds to a trivial sequence (e.g. 'AAAAAAAA'), and 1 corresponds to a random sequence. Higher values mask more of the sequence. Notice, that the report contains sequence examples for different complexity thresholds.

The Sequence filtering options make it possible to specify whether some or all input sequences should be output:

- **Keep all sequences**. No filtering is performed.
- **Keep sequences with low-complexity regions**. Notice, that sequences which have already been masked in a previous run of the tool will not be kept.
- **Keep sequences without low-complexity regions**. Note, that this also keeps sequences where low complexity regions have already been masked in a previous run of the tool.

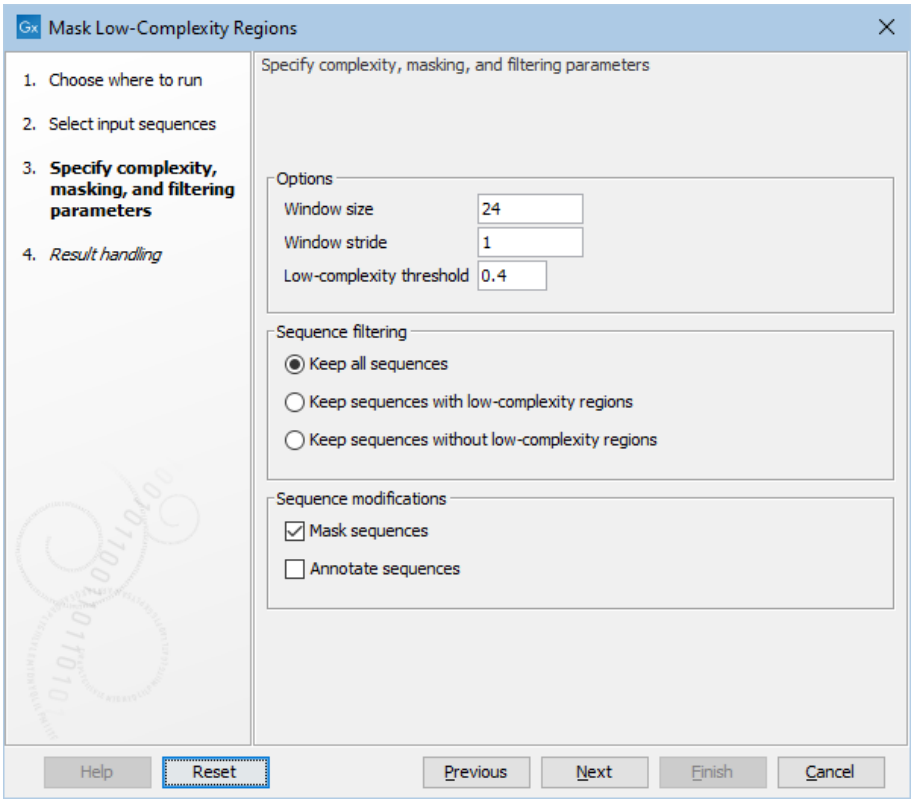


Figure 20.1: *The Mask Low-Complexity Regions options.*

Finally, the Sequence modifications options determine how the output sequences are marked:

- **Mask sequences**: Low-complexity regions are masked with N's (or X's for proteins). Notice, that if a tested window already contains ambiguous symbols (e.g. from a previous run of the tool), it will not be masked.
- **Annotate sequences**: Low-complexity regions are marked with a sequence annotation. Notice, that if a tested window already contains ambiguous symbols (e.g. from a previous run of the tool), it will not be marked.

The tool optionally outputs a report with statistics on the detected regions. The report is described in details below:

20.1.1 Mask Low-Complexity Regions Report

An example of a Mask Low-Complexity Regions report can be seen in figure 20.2.

The summary statistics describes the following measures:

- **Total nucleotides/amino acids**: The total number of nucleotides (or amino acids for protein sequence input) that were processed.
- **Masked nucleotides/amino acids**: The total number of nucleotides (or amino acids for protein sequence input) that were masked or annotated.

- **Ambiguous nucleotides/amino acids**: The total number of ambiguous nucleotides (or amino acids for protein sequence input) that were masked or annotated.
- **Total windows**: The total number of windows that were processed (notice, that these may be overlapping)
- **Masked windows**: The total number of windows that were masked (notice, that these may be overlapping)
- **Ignored windows**: The total number of windows that were ignored (because they already contained ambiguous nucleotides, e.g. from an earlier run)
- **Masked regions**: The total number of regions that were masked (regions are formed by joining all overlapping or adjacent windows into contiguous sections)

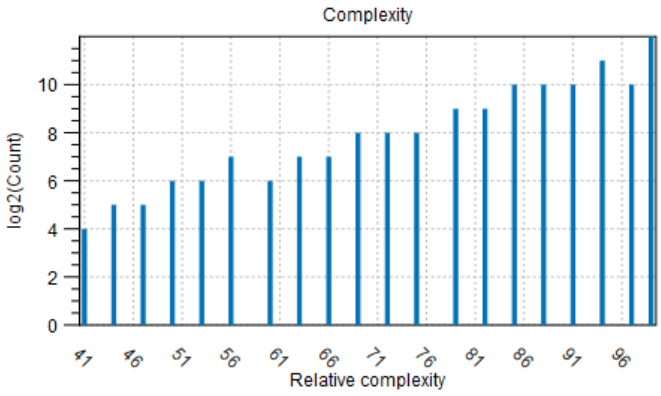
The Mask Low-Complexity Regions complexity overview shows a bar chart with the number of windows for the different complexity levels.

A table provides examples from the input data for different complexity levels: this is shown in order to make it easier to understand what the **Low-complexity threshold** corresponds to.

1 Mask Low-Complexity Regions summary

Total nucleotides/amino acids	523,332
Masked nucleotides/amino acids	0
Ambiguous nucleotides/amino acids	19,220
Total windows	523,309
Masked windows	0
Ignored windows	29,123
Masked regions	0

2 Mask Low-Complexity Regions complexity overview



A complexity of 99 corresponds to a random (uncompressible) sequence.
The table below lists some examples of different complexities from the input data

Complexity	Count	Example
41	72	GTAAAGAGAGAAAGAGAGAGAGAA
44	248	TGTAAAGAGAGAAAGAGAGAGAGA
47	286	CGGTAATAATAATAATAACAATAG
50	536	TTTTTTTTTTGTTTGTTTTAATTA

Figure 20.2: *The Mask Low-Complexity Regions report.*

20.2 Result Metadata

Metadata refers to information about data. In the context of the CLC Microbial Genomics Module, this usually means information about samples. For example a set of reads could come from a particular specimen at a particular time point with particular characteristics. The specimen, time and characteristics would be metadata for that set of reads.

What is metadata used for?

Core uses of metadata in CLC software include:

- Defining batch units when launching workflows in batch mode, described in https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_workflows_in_batch_mode.html.
- Distributing data to the relevant input channels in a workflow when using Collect and Distribute elements, described in https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Control_flow_elements.html.
- Finding and selecting data elements based on sample information (in a CLC Metadata Table). Workflow Result Metadata Tables are of particular use when reviewing results generated by workflows run in batch mode and are described in https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Workflow_Result_Metadata_tables.html.
- Running tools where characteristics of the data elements are relevant. An example is Differential Abundance Analysis, described in section 7.7.

Metadata tables

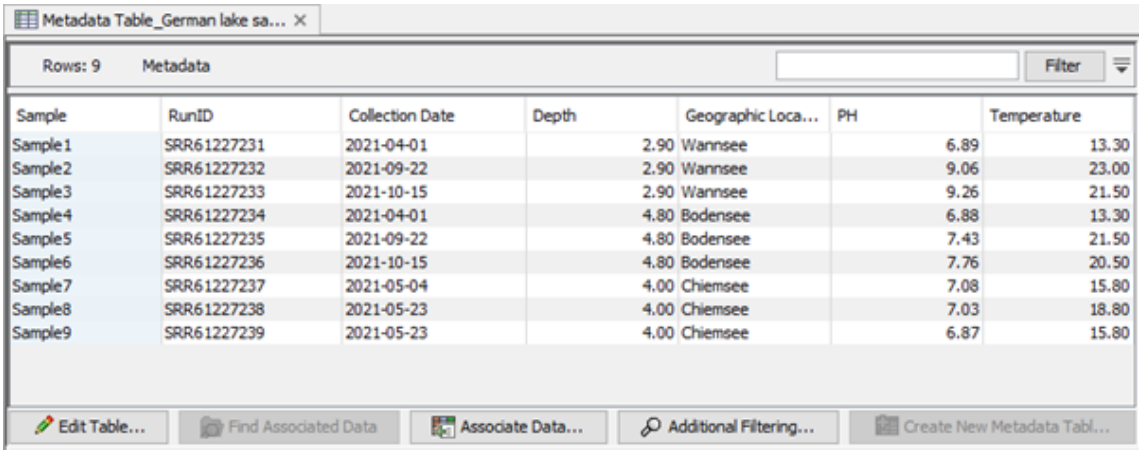
An example of a CLC Metadata Table generated by the *CLC Workbench* is shown in figure 20.3. Each column represents a property of a sample (e.g., identifier, sample depth, geographic location, temperature) and each row contains information relevant to a sample. A single column can be designated the key column. That column must contain unique entries.

Each row can have associations with one or more data elements, such as sequence lists, taxonomic profiling abundance tables, variant tracks, etc.

Creating metadata tables

CLC Metadata Tables can be created in several ways, including:

- Import metadata from an Excel, CSV or TSV format file using the **Import Metadata** () tool. You can associate already imported data with your metadata during import, or do this later. The process of importing metadata and associating data is described in the *CLC Genomics Workbench* user manual, https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Importing_metadata.html.
- Use a workflow to import a sample and its metadata at the same time. A template workflow for importing sequence data with associated metadata can be found in the Preparing Raw



Sample	RunID	Collection Date	Depth	Geographic Loca...	PH	Temperature
Sample1	SRR61227231	2021-04-01	2.90	Wannsee	6.89	13.30
Sample2	SRR61227232	2021-09-22	2.90	Wannsee	9.06	23.00
Sample3	SRR61227233	2021-10-15	2.90	Wannsee	9.26	21.50
Sample4	SRR61227234	2021-04-01	4.80	Bodensee	6.88	13.30
Sample5	SRR61227235	2021-09-22	4.80	Bodensee	7.43	21.50
Sample6	SRR61227236	2021-10-15	4.80	Bodensee	7.76	20.50
Sample7	SRR61227237	2021-05-04	4.00	Chiemsee	7.08	15.80
Sample8	SRR61227238	2021-05-23	4.00	Chiemsee	7.03	18.80
Sample9	SRR61227239	2021-05-23	4.00	Chiemsee	6.87	15.80

Figure 20.3: *A CLC Metadata Table, with the key column highlighted in blue.*

Data folder in the Template Workflows folder under the Workflows menu. It is described in the *CLC Genomics Workbench* user manual, https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Import_with_Metadata.html.

For more ways to create CLC Metadata Tables and information on how to work with CLC Metadata Tables in general, see the Metadata section of the *CLC Genomics Workbench* user manual, https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Metadata.html.

In addition to standard CLC Metadata Tables, the CLC Microbial Genomics Module makes use of a special type of metadata table; the Result Metadata Table. As opposed to CLC Metadata Tables, Result Metadata Tables can be updated with selected types of analysis results e.g., antibiotic resistance. This is described in section 20.2.1.

20.2.1 Create a Result Metadata Table

A Result Metadata Table is generated from a CLC Metadata Table with associated sample data.

To run the tool, go to:

Tools | **Utility Tools** ( ) | **Result Metadata** ( ) | **Create Result Metadata Table** ( )

Select the CLC Metadata Table (figure 20.4) and click on **Next** to specify result handling.

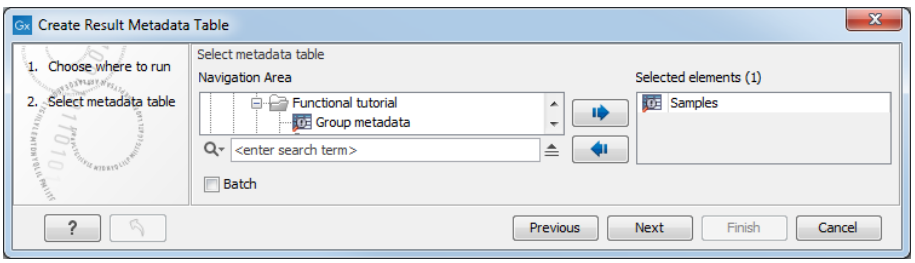


Figure 20.4: *Creation of a Result Metadata Table from a Metadata Table.*

The tool outputs a Result Metadata Table.

When first opened, the Result Metadata Table is empty (figure 20.5).

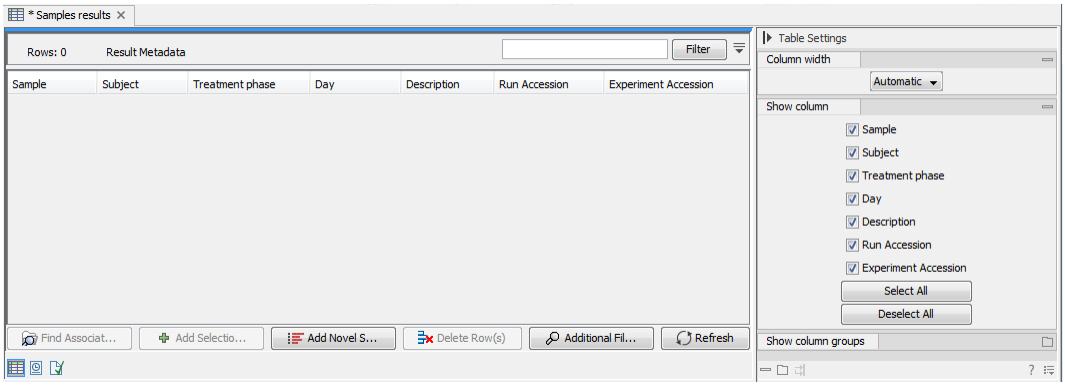


Figure 20.5: *The newly created Result Metadata Table is empty.*

To populate the table with information from the underlying CLC Metadata Table, click on **Add Novel Samples** (). Samples and associated metadata will be listed marked in yellow (figure 20.6). Save the Result Metadata Table to store the sample and metadata information.

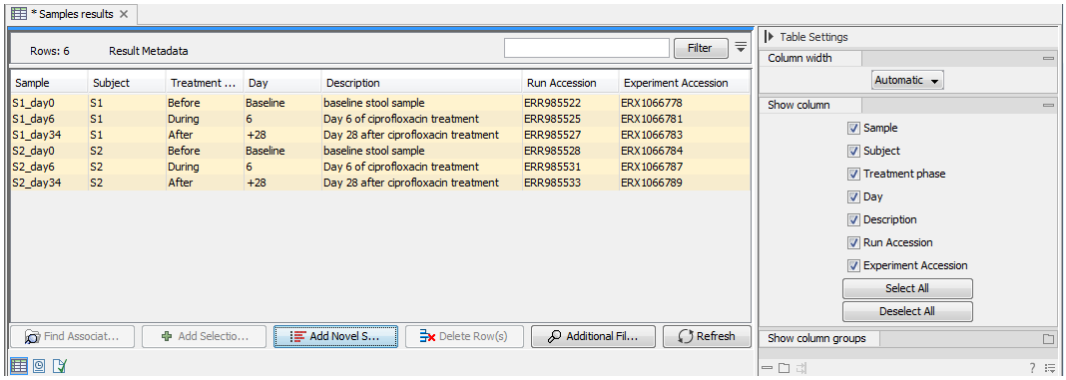


Figure 20.6: *Click on "Add Novel Samples" to add metadata from the underlying CLC Metadata Table to the otherwise empty Result Metadata Table.*

If for some reason Result Metadata rows are not needed, they can be deleted from the table by selecting them and clicking on the **Delete Row(s)** () button.

To find files associated to specific Metadata rows, select the sample row(s) of interest and click on **Find Associated Data** (). This action will list all associated files in a new Metadata Element window located below the Metadata window as shown in figure 20.7.

In most cases, analysis results will be added automatically to the Result Metadata Table when using a properly designed workflow. It is also possible to add manually generated analysis results to the table using the **Extend Result Metadata Table**.

20.2.2 Running an analysis directly from a Result Metadata Table

Analysis results from tools listed in the table of section 20.2.1 are automatically added to the Result Metadata Table as long as it was performed on samples associated with metadata. Content of the Result Metadata Table may be managed in similar ways as other tables in *CLC Genomics Workbench* (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Filtering_tables.html), but it can also be used to start new analyses using the **With selected** () button which provides the option of various downstream

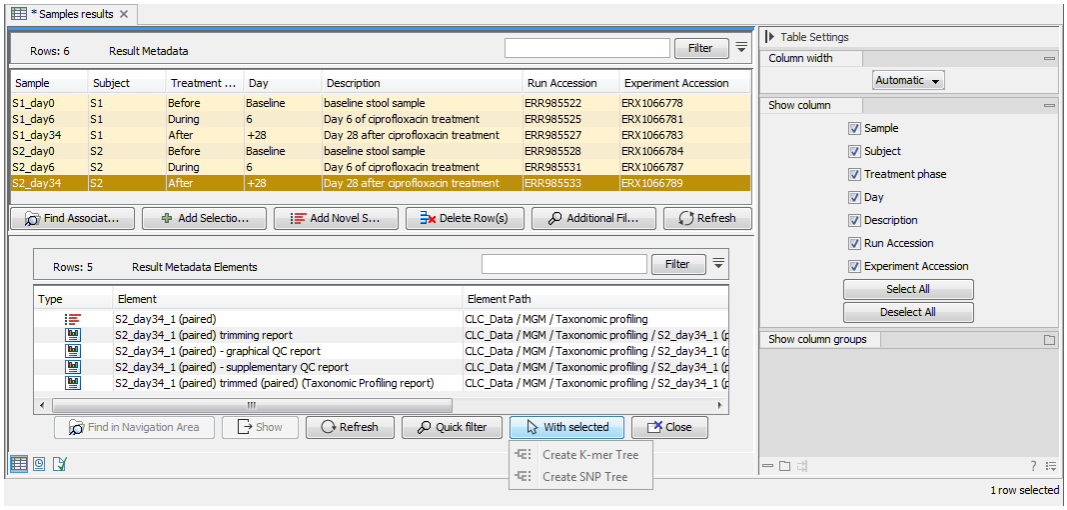


Figure 20.7: *The Metadata Element window at the bottom part of this figure lists all data associated to the selected Result Metadata row shown in the top window. In this example, only the imported read file is associated to the single metadata row. Note! Workflow analysis can be initiated directly on "With selected" Elements.*

analysis of the selected dataset.

To perform an analysis on one or more samples, begins by selecting the relevant rows followed by finding the associated elements by clicking on the **Find Associated Data** ( ) button. All associated elements are then listed in window below called **Metadata Elements**. You can see an example in figure 20.8, where a Metadata Result Table includes 6 rows (Metadata, top view), while 30 elements are found to be associated to these 6 rows (Metadata Elements, bottom view).

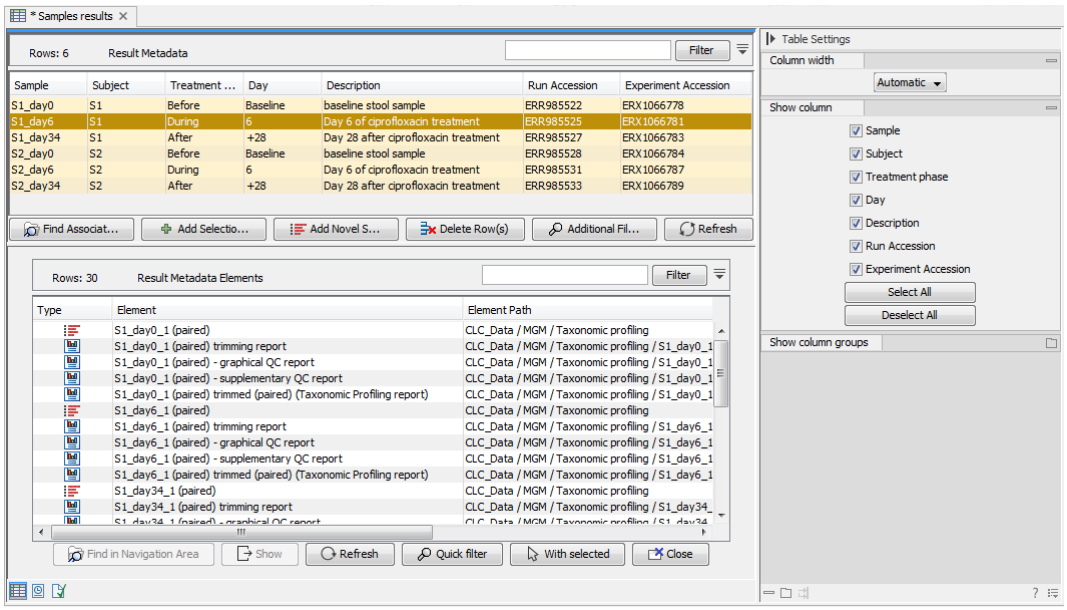


Figure 20.8: *In total, 30 files are associated to the selected 6 sample rows within the Result Metadata Table.*

As the number of samples, metadata and data elements increases over time, and the Result Metadata Table likely will include a mix of analyzed and novel samples, it is helpful to perform

filtering steps to identify the elements you are looking for (see section 20.2.2). Once filtering is done, it is easy to select the remaining rows of data elements and click the **With selected** ( ) button to start tools such as **Create K-mer Tree** and **Create SNP Tree**, or initiate a workflow analyses using an opened and customized version of a workflow.

Filtering in Result Metadata Table

Filtering is generally performed as a two step process: by picking or filtering firstly on the rows of the Result Metadata Table and secondly among the associated Metadata Elements.

Filtering can be done several ways, usually using a combination of the following options:

- Use the traditional table filtering function in top right corner. Filter for text elements, or unroll the banner by clicking on the icon ( ) and use more specific filters options.
- Tables can be sorted according to one or more columns, making it easier to find (and select) the desired elements. One example is to click on the **Role** column to find data elements with the same role.
- In the case of Metadata elements, use the **Quick filter** ( ) button and select the desired filtering option. It is possible to choose among:
  - **Imported** filters down to elements with the "Role" being **Sample data**. This can, for instance, be used for analyzing using an open and validated workflow based on one of the template workflows from the Workflows menu by clicking the **With selected** ( ) button.
  - **Filter for SNP Tree** filters down to elements with the "Role" being either **Read mapping**, **Realigned mapping** or **Variants**. Selection of the elements remaining after this filtering has been applied makes it easy to click the **With selected** ( ) button and initiate the **Create SNP Tree** tool using the selected data as input.
  - **Filter for K-mer Tree** filters down to elements with the "Role" being **Trimmed reads**. Selection of the elements remaining after this filtering has been applied makes it easy to click the **With selected** ( ) button and initiate the **Create K-mer Tree** tool using the selected data as input.
  - **Filter Re-mapped 'name of common reference' for SNP Tree**, option available for elements generated with the **Map to Specified Reference** or manually added with the **Use Genome as Result** and based on a shared reference.

  Applied quick filter selection can be removed by clicking the **Quick filter** ( ) followed by **Clear Quick Filter**.

Filtering in a SNP-Tree creation scenario

To construct a SNP tree, all sample data must have been analyzed (i.e., reads mapped and variants called) using the same reference sequence. If we want to use all the samples that were generated by the Map to Specified Reference workflow on several occasions using a common reference sequence, we use the quick filtering options.

- **Filter** all samples where read mapping and variant calling was performed using a common reference by clicking on the icon ( ) and using the following filter parameters: in the first

drop-down menu, choose the column whose header is the reference sequence of interest; in the second drop-down menu, choose the term "contains"; and in the third window, write "true".

- Select all remaining samples.
- **Click** on the **Find Associated Data** ( ) button. This opens the Metadata elements table underneath the initial Metadata table with a certain amount of elements associated with the samples selected in the Metadata Result Table.
- **Click** on the **Quick Filters** ( ) button in the Metadata Elements Table (bottom view) and select the **Filter Re-mapped 'common reference' for SNP Tree** option.
- **Select** all the remaining elements.
- **Click** on the **With selected** ( ) button and select the **Create SNP Tree** option. The **Create SNP Tree** wizard is displayed (see section 9.1). The read mappings are preselected as input. The variant tracks and the Result Metadata Table are automatically preselected as parameters.

20.2.3 Extend Result Metadata Table

The **Extend Result Metadata Table** tool adds one or more Result objects to the Result Metadata Table. The tool outputs a copy of the source Result Metadata Table. The original source table is not modified.

To manually add results to an existing Result Metadata Table, go to:

Tools | **Utility Tools** ( ) | **Result Metadata** ( ) | **Extend Result Metadata Table** ( )

1. In the first wizard window, select the relevant Result Metadata Table (see figure 20.9) and click **Next**.

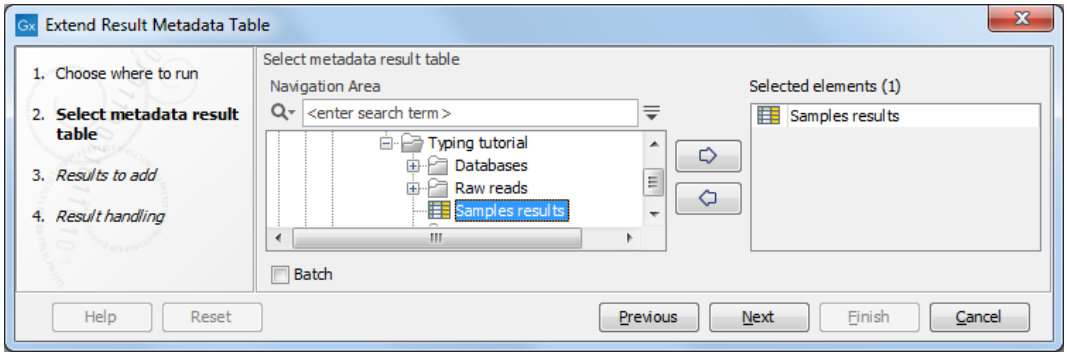


Figure 20.9: *First select the Result Metadata Table you want to add results to.*

2. Now select the relevant Result object(s) to be added to the Result Metadata Table (see figure 20.10) and click **Next**.
3. Finally, select to **Save** and click **Finish**.

The output of this tool is a copy of the Result Metadata Table containing cells updated with the results (figure 20.11).

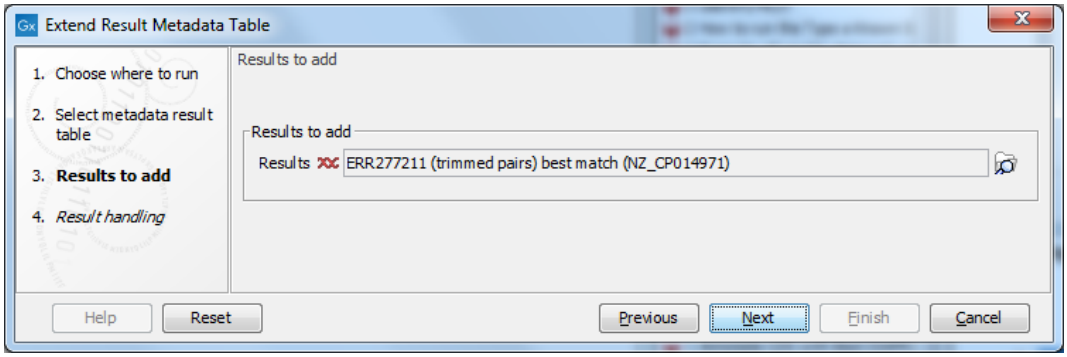


Figure 20.10: *In the second step of this example, the identified best matching sequence for a particular sample is added.*

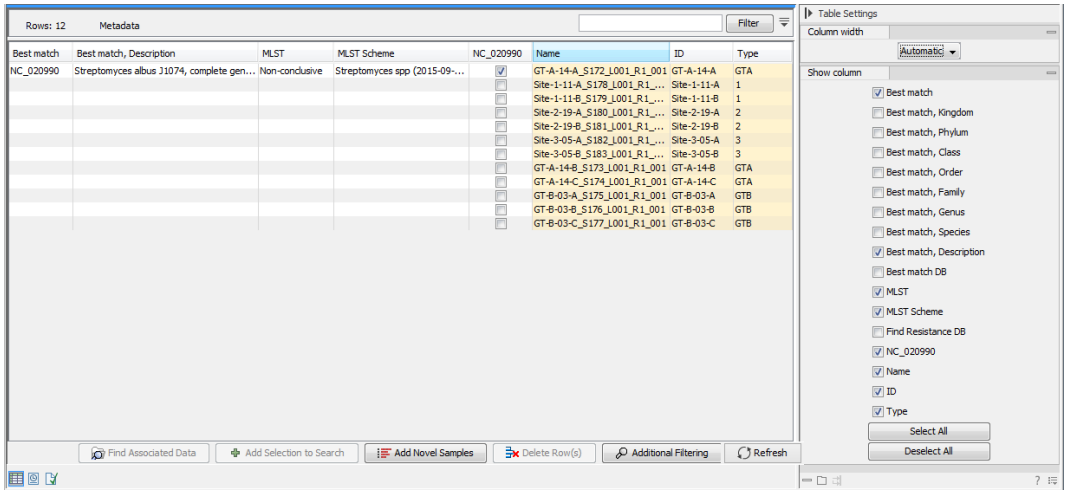


Figure 20.11: *Example with column added to the Result Metadata Table, including the data for the particular sample that was specified in step 1.*

20.2.4 Use Genome as Result

The **Use Genome as Result** tool is part of the **Map to Specified Reference** workflow scenario and is not necessarily intended to be used as a single tool by users. Its function, at the last step of the **Map to Specified Reference** workflow, is double: it adds the name of the reference genome used for the re-mapping to the 'role' of the input files (for example the role "mapping report" will become "NZ_CP014971 mapping report", where NZ_CP014971 is the name of the reference used to re-map). It also creates an extra column in the Result Metadata Table whose header is the name of the common reference that was used for the re-mapping (here NZ_CP014971). This extra column makes it possible to distinguish between read mappings that were generated at different time points as well as in different runs of the workflow, despite using the **same genome reference**.

The tool can take multiple elements as input, and each will have its metadata role changed to include the name of reference sequence in addition to the original role value. Relevant elements can be selected individually, or you can select folders by right-clicking on the folder value and selecting **Add folder contents** (it will select all elements in that folder), or select folder recursively by right-clicking on the folder value and selecting **Add folder contents (recursively)**. In this last case all elements of the folder, including elements contained in subfolders, will be selected (see figure 20.12).

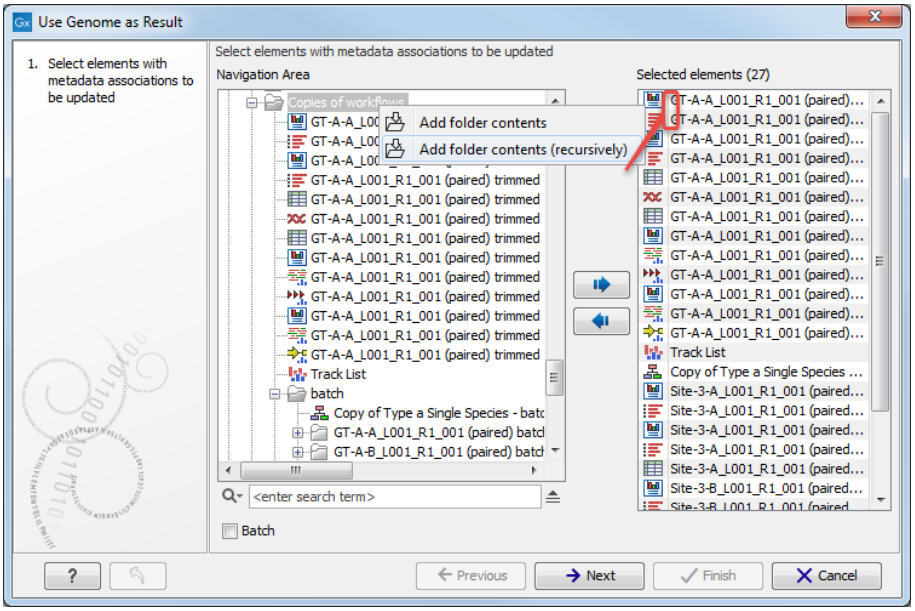


Figure 20.12: *Select recursively elements with metadata associations to be updated. Here the selected elements are the 16 of the folder 'Copies of workflows' as well as the 9 from the subfolder called 'batch'.*

In the second dialog, select the relevant read mapping, i.e., the read mapping that was created using the common reference you want to annotate the roles with (figure 20.13).

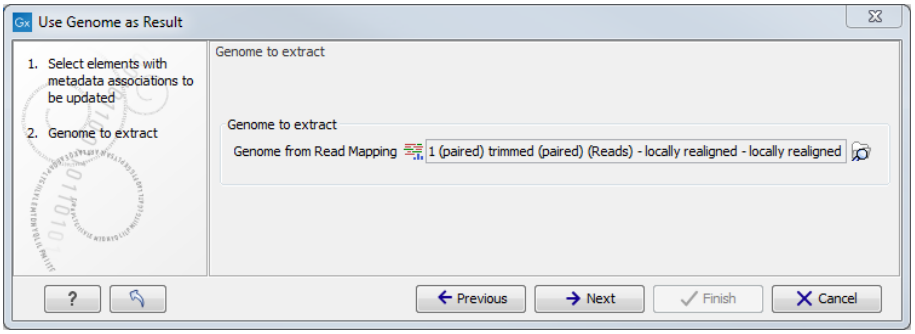


Figure 20.13: *Specification of the read mapping to be associated with genome metadata.*

In addition to changing the role name, the tool creates a new column named after the selected reference sequence in the Result Metadata Table. This column indicates whether data has been analysed using this reference or not (For an example see column "NZ_CP014971" in figure 20.14). To filter for all possible elements that were generated using this sequence as reference data, open the filter banner by clicking on the icon ( ) next to the **Filter** button. In the first drop down menu, choose the column whose header is the reference sequence of interest. In the second drop down menu, choose the term "contains", and in the third window, write "true". This will filter for all the elements with a tick in the reference sequence column, as can be seen on the figure 20.14.

To add the genome output from this tool to a Result Metadata Table, see section 20.2.3.

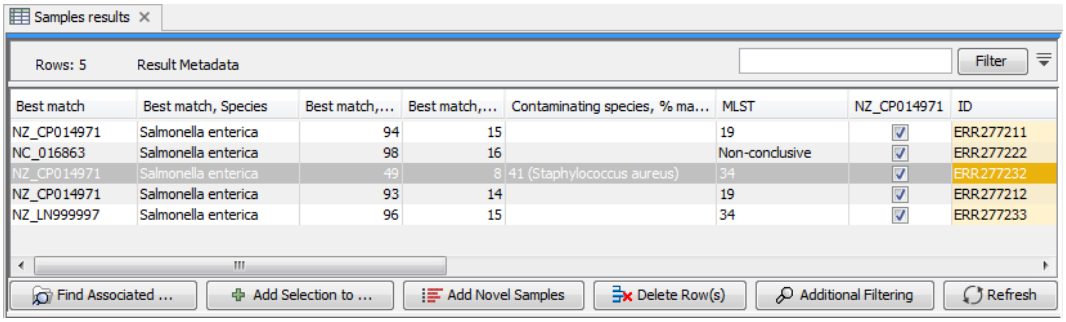


Figure 20.14: *Filter for elements who share a tick in the column newly generated by the Use Genome as Result tool*

Part X

Legacy tools

Chapter 21

Legacy tools

The documentation in this section is for tools that have been deprecated and that will be retired in a future version. In most cases, deprecated tools can be found in the **Legacy Tools** () folder of the Workbench Toolbox, with "(legacy)" appended to their names to highlight their status.

We recommend redesigning workflows containing any of these tools to remove them. Where a new tool has been introduced to take the deprecated tool's place, please try including the new tool.

If you have concerns about the retirement of particular tools in this section, please contact QIAGEN Bioinformatics Support team at ts-bioinformatics@qiagen.com.

21.1 Remove OTUs with Low Abundance

This tool has been superseded by **Refine Abundance Table**, see section 7.2.

Low abundance OTUs can eliminated from the OTU table if they have fewer than a given count across all the samples in the experiment.

To run the tool, go to

Tools | **Legacy Tools** () | **Remove OTUs with Low Abundance (legacy)** () .

Choose an OTU table as input, select the filtering parameters and save the table. The threshold for determining whether an OTU has sufficient abundance is specified by the parameters **minimum combined abundance** and **minimum combined abundance (% of all the reads)**. The algorithm filters out all OTUs whose combined abundance across all samples is less than the minimum combined abundance or whose combined abundance is less than the minimum combined abundance (% of all the reads) across all samples. The default value for the Minimum combined abundance is set at 10.

Part XI

Appendix

Chapter 22

Using the Assembly ID annotation

The **Assembly ID** annotation on sequences is used by many tools of the module to group sequences into meaningful entities, e.g. to group all contigs of a draft assembly. Tools that are aware of this annotation include

- Bin Pangenomes by Sequence, section 6.1.3
- Create K-mer Tree, section 9.2
- Create MLST Scheme, section 14.1
- Create Taxonomic Profiling Index, section 16.5
- Find Best Matches using K-mer Spectra, section 8.1
- Find Prokaryotic Genes, section 12.1

In order to see how these tools utilize the ''Assembly ID'' annotation, please read the tool documentation. In order to assign these annotations to sequences in a sequence list

1. Open the table view of a sequence list.
2. Select all rows corresponding to sequences that form a logical unit.
3. Right-click on the selection and choose **Assign annotations**, see figure 22.1.
4. Select **Assembly ID** from the dropdown menu in the Name field, see figure 22.2.
5. Enter a string in the Value field to uniquely identify the assembly.

For large sequence lists containing many assemblies it may be beneficial to use **Update Sequence Attributes in Lists** (see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Update_Sequence_Attributes_in_Lists.html).

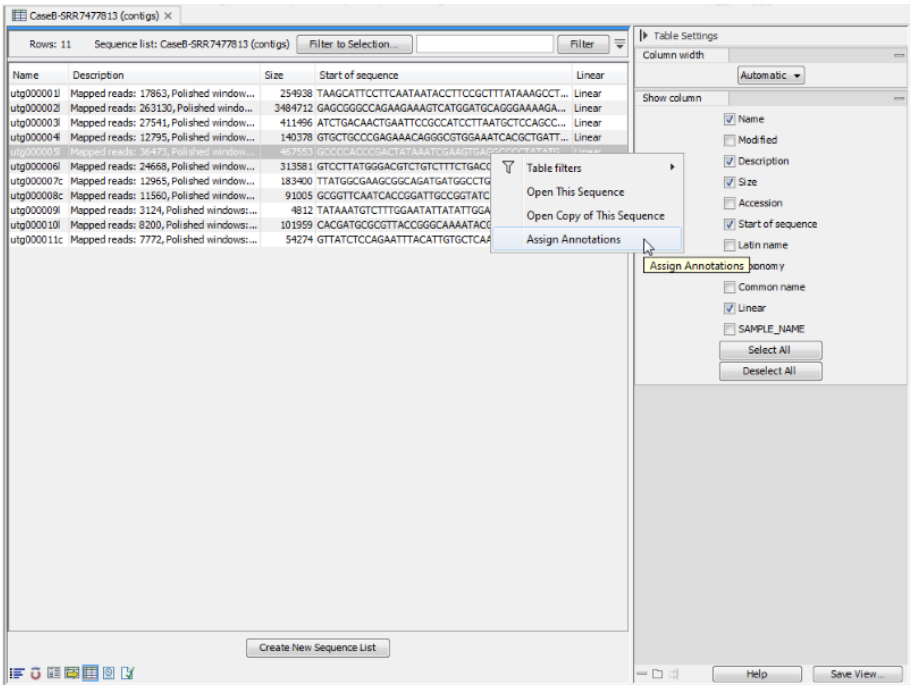


Figure 22.1: *Select the sequences forming a logical unit and right-click on the selection to assign annotations to these sequences.*

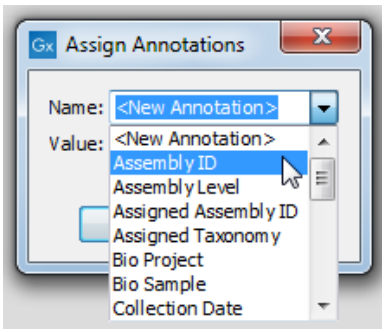


Figure 22.2: *Select Assembly ID from the dropdown menu in the Name field and enter a string to uniquely identify the assembly in the Value field.*

Bibliography

[Alcock et al., 2023] Alcock, B. P., Huynh, W., Chalil, R., Smith, K. W., Raphenya, A. R., Wlodarski, M. A., Edalatmand, A., Petkau, A., Syed, S. A., Tsang, K. K., Baker, S. J. C., Dave, M., McCarthy, M. C., Mukiri, K. M., Nasir, J. A., Golbon, B., Imtiaz, H., Jiang, X., Kaur, K., Kwong, M., Liang, Z. C., Niu, K. C., Shan, P., Yang, J. Y. J., Gray, K. L., Hoad, G. R., Jia, B., Bhando, T., Carfrae, L. A., Farha, M. A., French, S., Gordzevich, R., Rachwalski, K., Tu, M. M., Bordeleau, E., Dooley, D., Griffiths, E., Zubyk, H. L., Brown, E. D., Maguire, F., Beiko, R. G., Hsiao, W. W. L., Brinkman, F. S. L., Van Domselaar, G., and McArthur, A. G. (2023). CARD 2023: expanded curation, support for machine learning, and resistome prediction at the comprehensive antibiotic resistance database. *Nucleic Acids Res*, 51(D1):D690–D699.

[Anderson, 2001] Anderson, M. (2001). A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26(1):32–46.

[Callahan et al., 2016] Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J., and Holmes, S. P. (2016). Dada2: High resolution sample inference from illumina amplicon data repository. *Nature Methods*, 13:581–583.

[Caspi et al., 2019] Caspi, R., Billington, R., Keseler, I. M., Kothari, A., Krummenacker, M., Midford, P. E., Ong, W. K., Paley, S., Subhraveti, P., and Karp, P. D. (2019). The MetaCyc database of metabolic pathways and enzymes - a 2019 update. *Nucleic Acids Research*, 48(D1):D445–D453.

[Chen et al., 2012] Chen, J., Bittinger, K., Charlson, E. S., Hoffmann, C., Lewis, J., Wu, G. D., Collman, R. G., Bushman, F. D., and Li, H. (2012). Associating microbiome composition with environmental covariates using generalized unifrac distances. *Bioinformatics*, 28(16):2106–13.

[Couvin et al., 2020] Couvin, D., Segretier, W., Stattner, E., and Rastogi, N. (2020). Novel methods included in spollineages tool for fast and precise prediction of mycobacterium tuberculosis complex spoligotype families. *Database*, 2020:baaa108.

[Curry et al., 2022] Curry, K. D., Wang, Q., Nute, M. G., Tyshaieva, A., Reeves, E., Soriano, S., Wu, Q., Graeber, E., Finzer, P., Mendling, W., et al. (2022). Emu: species-level microbial community profiling of full-length 16s rrna oxford nanopore sequencing data. *Nature methods*, 19(7):845–853.

[Douglas et al., 2020] Douglas, G. M., Maffei, V. J., Zaneveld, J. R., Yurgel, S. N., Brown, J. R., Taylor, C. M., Huttenhower, C., and Langille, M. G. I. (2020). PICRUSt2 for prediction of metagenome functions. *Nature Biotechnology*, 38(6):685–688.

[Goodacre et al., 2018] Goodacre, N., Aljanahi, A., Nandakumar, S., Mikailov, M., and Khan, A. S. (2018). A reference viral database (rvdb) to enhance bioinformatics analysis of high-throughput sequencing for novel virus detection. *MSphere*, 3(2):e00069–18.

[Gupta et al., 2014] Gupta, S. K., Padmanabhan, B. R., Diene, S. M., Lopez-Rojas, R., Kempf, M., Landraud, L., and Rolain, J.-M. (2014). Arg-annot, a new bioinformatic tool to discover antibiotic resistance genes in bacterial genomes. *Antimicrobial agents and chemotherapy*, 58(1):212–220.

[Gurbich et al., 2023] Gurbich, T. A., Almeida, A., Beracochea, M., Burdett, T., Burgin, J., Cochrane, G., Raj, S., Richardson, L., Rogers, A. B., Sakharova, E., Salazar, G. A., and Finn, R. D. (2023). Mgnify genomes: A resource for biome-specific microbial genome catalogues. *Journal of Molecular Biology*, 435(14):168016. Computation Resources for Molecular Biology.

[Hasman et al., 2013] Hasman, H., Saputra, D., Sicheritz-Ponten, T., Lund, O., Svendsen, C. A., Frimodt-Møller, N., and Aarestrup, F. M. (2013). Rapid whole genome sequencing for the detection and characterization of microorganisms directly from clinical samples. *Journal of clinical microbiology*, pages JCM–02452.

[Jolley et al., 2018] Jolley, K. A., Bray, J. E., and Maiden, M. C. J. (2018). Open-access bacterial population genomics: BIGSdb software, the PubMLST.org website and their applications. *Wellcome Open Res*, 3:124.

[Kõljalg et al., 2020] Kõljalg, U., Nilsson, H. R., Schigel, D., Tedersoo, L., Larsson, K.-H., May, T. W., Taylor, A. F. S., Jeppesen, T. S., Frøslev, T. G., Lindahl, B. D., Põldmaa, K., Saar, I., Suija, A., Savchenko, A., Yatsiuk, I., Adojaan, K., Ivanov, F., Piirmann, T., Pöhönen, R., Zirk, A., and Abarenkov, K. (2020). The taxon hypothesis paradigm - on the unambiguous detection and communication of taxa. *Microorganisms*, 8(12).

[Kaas et al., 2014] Kaas, R. S., Leekitcharoenphon, P., Aarestrup, F. M., and Lund, O. (2014). Solving the problem of comparing whole bacterial genomes across different sequencing platforms. *PLOS ONE*.

[Kaminski et al., 2015] Kaminski, J., Gibson, M. K., Franzosa, E. A., Segata, N., Dantas, G., and Huttenhower, C. (2015). High-specificity targeted functional profiling in microbial communities with shortbred. *PLoS Comput. Biol.*

[Kang et al., 2015] Kang, D., Froula, J., Egan, R., and Wang, Z. (2015). Metabat, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ*, 3:e1165.

[Kelley and Salzberg, 2010] Kelley, D. and Salzberg, S. (2010). Clustering metagenomic sequences with interpolated markov models. *BMC Bioinformatics*, 11:544.

[Larsen et al., 2014] Larsen, M. V., Cosentino, S., Lukjancenko, O., Saputra, D., Rasmussen, S., Hasman, H., Sicheritz-Pontén, T., Aarestrup, F. M., Ussery, D. W., and Lund, O. (2014). Benchmarking of methods for genomic taxonomy. *Journal of clinical microbiology*, 52(5):1529--1539.

[Lu et al., 2017] Lu, J., Breitwieser, F. P., Thielen, P., and Salzberg, S. L. (2017). Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science*, 3:e104.

[Narayan et al., 2020] Narayan, N. R., Weinmaier, T., Laserna-Mendieta, E. J., Claesson, M. J., Shanahan, F., Dabbagh, K., Iwai, S., and DeSantis, T. Z. (2020). Piphillin predicts metagenomic composition and dynamics from DADA2-corrected 16S rDNA sequences. *BMC Genomics*, 21(1):56.

[Nearing et al., 2022] Nearing, J. T., Douglas, G. M., Hayes, M. G., MacDonald, J., Desai, D. K., Allward, N., Jones, C. M., Wright, R. J., Dhanani, A. S., Comeau, A. M., et al. (2022). Microbiome differential abundance methods produce different results across 38 datasets. *Nature communications*, 13(1):342.

[Prjibelski et al., 2020] Prjibelski, A., Antipov, D., Meleshko, D., Lapidus, A., and Korobeynikov, A. (2020). Using SPAdes DE novo assembler. *Curr. Protoc. Bioinformatics*, 70(1):e102.

[Quast et al., 2012] Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., and Glöckner, F. O. (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1):D590–D596.

[Sedlar et al., 2017] Sedlar, K., Kupkova, K., and Provaznik, I. (2017). Bioinformatics strategies for taxonomy independent binning and visualization of sequences in shotgun metagenomics. *Computational Structural Biotechnology Journal*, 15:48–55.

[Shepard et al., 2016] Shepard, S. S., Meno, S., Bahl, J., Wilson, M. M., Barnes, J., and Neuhaus, E. (2016). Viral deep sequencing needs an adaptive approach: Irma, the iterative refinement meta-assembler. *BMC genomics*, 17:1–18.

[Van Embden et al., 2000] Van Embden, J., Van Gorkom, T., Kremer, K., Jansen, R., Van der Zeijst, B., and Schouls, L. (2000). Genetic variation and evolutionary origin of the direct repeat locus of mycobacterium tuberculosis complex bacteria. *Journal of bacteriology*, 182(9):2393–2401.

[WHO, 2023] WHO (2023). *Catalogue of mutations in Mycobacterium tuberculosis complex and their association with drug resistance, second edition*. World Health Organization, Geneva.

[Wood et al., 2019] Wood, D. E., Lu, J., and Langmead, B. (2019). Improved metagenomic analysis with kraken 2. *Genome biology*, 20:1–13.

[Ye and Doak, 2009] Ye, Y. and Doak, T. G. (2009). A Parsimony Approach to Biological Pathway Reconstruction/Inference for Genomes and Metagenomes. *PLoS Computational Biology*, 5(8):e1000465.

[Zankari et al., 2017] Zankari, E., Allesï¿½e, R., Joensen, K. G., Cavaco, L. M., Lund, O., and Aarestrup, F. M. (2017). Pointfinder: a novel web tool for wgs-based detection of antimicrobial resistance associated with chromosomal point mutations in bacterial pathogens. *Journal of Antimicrobial Chemotherapy*, 72(10):2764–68. https://doi.org/10.1093/jac/dkx217.