User manual for

*CLC Microbial Genomics Module 23.0.3*

Windows, macOS and Linux

November 30, 2023

This software is for research purposes only.
## Contents

I Introduction 10

1 Introduction 11

1.1 The concept of CLC Microbial Genomics Module 11
1.2 Contact information 12
1.3 System requirements 12
1.4 Installing modules 13
  1.4.1 Licensing modules 15
  1.4.2 Uninstalling modules 15
1.5 Installing server extensions 16
  1.5.1 Licensing server extensions 19

II Core Functionalities 20

2 Metadata 21

2.1 Create a Result Metadata Table 22
2.2 Running an analysis directly from a Result Metadata Table 24
  2.2.1 Filtering in Result Metadata Table 25
  2.2.2 Filtering in a SNP-Tree creation scenario 26
2.3 Extend Result Metadata Table 26
2.4 Use Genome as Result 27

3 Microbial template workflows 30

3.1 Taxonomic Analysis template workflows 30
  3.1.1 Analyze Viral Hybrid Capture Panel Data 31
  3.1.2 Data QC and Remove Background Reads 37
3.1.3 Data QC and Taxonomic Profiling ........................................ 39
3.1.4 Merge and Estimate Alpha and Beta diversities ....................... 41
3.1.5 QC, Assemble and Bin Pangenomes ..................................... 42
3.2 Amplicon-Based Analysis template workflows ............................. 45
   3.2.1 Data QC and OTU Clustering ........................................ 45
   3.2.2 Detect Amplicon Sequence Variants and Assign Taxonomies ..... 46
   3.2.3 Estimate Alpha and Beta Diversities ................................ 48
3.3 Typing and Epidemiology template workflows ............................ 50
   3.3.1 Compare Variants Across Samples .................................. 50
   3.3.2 Create MLST Scheme with Sequence Types ......................... 54
   3.3.3 Map to Specified Reference ......................................... 56
   3.3.4 Type Among Multiple Species ...................................... 63
   3.3.5 Type a Known Species .............................................. 73
3.4 QIAseq Analysis template workflows ...................................... 83
   3.4.1 Analyze QIAseq xHYB Viral Panel Data ............................... 83
   3.4.2 Find QIAseq xHYB AMR Markers .................................. 88

III Metagenomics ..................................................................... 92

4 De Novo Assemble Metagenome ............................................. 93

5 Amplicon-Based Analysis ...................................................... 95
   5.1 Normalize OTU Table by Copy Number ................................ 95
   5.2 Filter Samples Based on Number of Reads ............................ 97
   5.3 OTU clustering ............................................................. 97
      5.3.1 OTU clustering parameters ....................................... 98
      5.3.2 OTU clustering outputs ........................................... 102
      5.3.3 The OTU abundance table ....................................... 103
      5.3.4 Importing and exporting OTU abundance tables .......... 106
   5.4 Remove OTUs with Low Abundance ................................. 107
   5.5 Align OTUs with MUSCLE ........................................... 108
   5.6 Detect Amplicon Sequence Variants ................................. 109
      5.6.1 Detect Amplicon Sequence Variants parameters ........... 110
## 9 Find the best matching reference

9.1 Find Best Matches using K-mer Spectra ........................................... 156
    9.1.1 From samples best matches to a common reference for all ............ 159
9.2 Find Best References using Read Mapping .................................... 160
    9.2.1 The Find Best References using Read Mapping Report ............... 161

## 10 Phylogenetic trees using SNPs or k-mers

10.1 Create SNP Tree ................................................................. 163
    10.1.1 SNP tree output report .................................................. 167
    10.1.2 Visualization of SNP Tree including metadata and analysis result metadata 169
    10.1.3 SNP Tree Variants editor viewer ..................................... 169
    10.1.4 SNP Matrix ............................................................... 170
10.2 Create K-mer Tree ............................................................... 171
    10.2.1 Visualization of K-mer Tree for identification of common reference .... 173

## 11 MLST Scheme Tools

11.1 Getting started with the MLST Scheme tools ................................. 175
11.2 MLST Scheme Visualization and Management .................................. 176
11.3 Minimum Spanning Trees ...................................................... 179
    11.3.1 The Minimum Spanning Tree view ................................... 179
    11.3.2 Navigating the Tree view ............................................. 180
    11.3.3 The Layout panel ...................................................... 181
    11.3.4 The Metadata panel .................................................. 183
11.4 Type With MLST Scheme ....................................................... 184
    11.4.1 Type With MLST Scheme results .................................... 186
    11.4.2 The MLST Typing Result element ................................... 186
11.5 Add Typing Results to MLST Scheme ........................................ 189
11.6 Identify MLST Scheme from Genomes ....................................... 191

## V Functional and Drug Resistance Analyses

## 12 Functional Analysis

12.1 Find Prokaryotic Genes ....................................................... 194
12.2 Annotate with BLAST .................................................. 197
12.3 Annotate with DIAMOND ............................................. 200
12.4 Annotate CDS with Best BLAST Hit .............................. 204
12.5 Annotate CDS with Best DIAMOND Hit ......................... 205
12.6 Annotate CDS with Pfam Domains ............................... 206
12.7 Build Functional Profile ............................................. 208
12.7.1 Functional profile abundance table .......................... 210
12.8 Infer Functional Profile ............................................. 213
12.9 Identify Pathways .................................................... 215
12.9.1 Called Pathways Result ........................................ 217
12.9.2 The Identified Pathways View ................................ 219

13 Drug Resistance Analysis ............................................. 221
13.1 Find Resistance with PointFinder ................................. 221
13.2 Find Resistance with Nucleotide DB ............................. 223
13.3 Find Resistance with ShortBRED ................................. 224
13.3.1 Resistance abundance table ................................. 226

VI Databases ................................................................. 230

14 Databases for MLST Schemes ....................................... 231
14.1 Create MLST Scheme .............................................. 231
14.2 Download MLST Scheme .......................................... 234
14.3 Import MLST Scheme ............................................. 237

15 Databases for Amplicon-Based Analysis ......................... 239
15.1 Download Amplicon-Based Reference Database ............... 239

16 Databases for Taxonomic Analysis ................................. 241
16.1 Download Curated Microbial Reference Database ............ 241
16.1.1 Extracting a subset of a database ......................... 243
16.2 Download Custom Microbial Reference Database ............ 244
16.2.1 Database Builder ............................................. 246
16.3 Download Pathogen Reference Database ..................... 249
16.4 Create Taxonomic Profiling Index ............................................. 250

17 Databases for Functional Analysis ........................................... 252
  17.1 Download Protein Database .................................................. 252
  17.2 Download Ontology Database ............................................... 252
    17.2.1 The GO Database View ................................................. 253
    17.2.2 The EC Database View ................................................. 254
  17.3 Download Pathway Database ............................................... 255
    17.3.1 The Pathway Database ................................................... 255
    17.3.2 The Pathway View ...................................................... 257
  17.4 Create DIAMOND Index ..................................................... 257
  17.5 Import RNAcentral Database ............................................. 258
  17.6 Import PICRUSt2 Multiplication Table ................................ 259

18 Databases for Drug Resistance Analysis .................................. 263
  18.1 Download Resistance Database .......................................... 263
    18.1.1 ARES Database .......................................................... 265

VII Panel Support ................................................................. 271

19 QIAseq 16S/ITS Demultiplexer .............................................. 272

VIII Tools ............................................................................... 275

20 Tools .................................................................................. 276
  20.1 Extract Regions from Tracks ............................................... 276
  20.2 Mask Low-Complexity Regions .......................................... 277
    20.2.1 Mask Low-Complexity Regions Report ............................. 279

IX Legacy tools .................................................................. 281

21 Legacy tools .................................................................. 282
  21.1 Convert Abundance Table to Experiment ............................. 282
 Appendix  

22 Using the Assembly ID annotation  

Bibliography
Part I

Introduction
Chapter 1

Introduction

Welcome to CLC Microbial Genomics Module 23.0.3 – a software package supporting your daily bioinformatics work.

1.1 The concept of CLC Microbial Genomics Module

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.
Unless specified otherwise, all tools described in this manual can be found in the Microbial Genomics Module folder that will be placed in the Toolbox once the plugin is installed.

1.2 Contact information

CLC Microbial Genomics Module is developed by:

QIAGEN Aarhus
Silkeborgvej 2
Prismet
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: http://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/

1.3 System requirements

CLC Microbial Genomics Module 23.0.3 is for use with CLC Genomics Workbench version 23 or newer. The system requirements for CLC Genomics Workbench are provided at http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html.

The system requirements for CLC Microbial Genomics Module are the same as those for CLC Genomics Workbench, except for the following:

- Tools using DIAMOND and ShortBRED require a CPU that supports the AVX2 or NEON instruction sets.

Special requirements for the MLST Scheme tools

The system requirements for the MLST Scheme tools depend 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 requirements of reference based OTU clustering are dependent 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 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 Metagenome**

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

**Sunburst viewer**


### 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 (-plugins) button in the top Toolbar, or go to the menu option: Utilities | Manage Plugins... (-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.

![Figure 1.1: Plugins and modules available for installation are listed in the Plugin Manager under the Download Plugins tab.](image)

**Accepting the license agreement**

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 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/](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 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 Server connection.** If your organization has a CLC Network License Manager (or CLC License Server), select this option to configure the connection to it.


To download licenses, including evaluation licenses, your machine must have access to the external network. To install licenses on non-networked machines, please see http://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 ( đôi ) 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.](image)

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 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 the appendix ??.

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).
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 are managed, and can be 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

**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 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.

![Management tab](resources.qiagenbioinformatics.com/manuals/clcserver/current/admin/index.php?manual=Starting_stopping_server.html)

Figure 1.6: License management in done under the Management tab tab.
Part II

Core Functionalities
Chapter 2

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:


- Running tools where characteristics of the data elements are relevant. An example is Differential Abundance Analysis, described in section 7.6.

Metadata tables

An example of a CLC Metadata Table in the CLC Microbial Genomics Module is shown in figure 2.1. 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:


In addition to the CLC Metadata Table, the CLC Microbial Genomics Module makes use of a special type of metadata table; the Result Metadata Table. As opposed to the CLC Metadata Table, the Result Metadata Table can be updated with selected types of analysis results e.g., antibiotic resistance. This is described in section 2.1.

### 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:

**Toolbox** | **Microbial Genomics Module** | **Typing and Epidemiology** | **Result Metadata** | **Create Result Metadata Table**
Select the CLC Metadata Table (figure 2.2) and click on **Next** to specify result handling.

The tool outputs a Result Metadata Table.

When first opened, the Result Metadata Table is empty (figure 2.3).

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 2.4).

Save the Result Metadata Table to store the sample and metadata information.

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 2.5.
Figure 2.5: 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.

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.

2.2 Running an analysis directly from a Result Metadata Table

Analysis results from tools listed in the table of section 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 (http://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 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 2.6, 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).

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 2.2.1). 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.
CHAPTER 2. METADATA

Figure 2.6: In total, 30 files are associated to the selected 6 sample rows within the Result Metadata Table.

2.2.1 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 Toolbox 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 (_qs) followed by Clear Quick Filter.

### 2.2.2 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 (qs) 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 (qs) 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 (qs) 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 (qs) button and select the Create SNP Tree option. The Create SNP Tree wizard (described in section 10.1) is displayed. The read mappings are preselected as input. The variant tracks and the Result Metadata Table are automatically preselected as parameters.

### 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:

Toolbox | Microbial Genomics Module | Typing and Epidemiology | Result Metadata | Extend Result Metadata Table

1. In the first wizard window, select the relevant Result Metadata Table (see figure 2.7) and click Next.
2. Now select the relevant Result object(s) to be added to the Result Metadata Table (see figure 2.8) 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 2.9).
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 2.10).

Figure 2.10: 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 2.11).

Figure 2.11: 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 2.12). To filter for all possible elements that were generated using this sequence as reference data,
open the filter banner by clicking on the icon (isers) 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 2.12.

Figure 2.12: Filter for elements who share a tick in the column newly generated by the Use Genome as Result tool
Chapter 3

Microbial template workflows

Template workflows are provided as example workflows. They can be launched as they are from the Toolbox, 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:
  Toolbox | 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.


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

3.1 Taxonomic Analysis template workflows

The Taxonomic Analysis template workflows are found at:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Taxonomic Analysis
3.1.1 Analyze Viral Hybrid Capture Panel Data

The Analyze Viral Hybrid Capture Panel Data template workflow is designed for detecting viruses, calculating abundances and for calling variants for the viruses identified from data generated using hybrid capture panels. The workflow (figure 3.1) performs read trimming, creates a QC report, cleans the dataset of host DNA, calculates viral abundances and maps the reads to the most abundant viral reference for variant calling. Note: After taxonomic profiling, the viral reads are downsampled to maximum 500,000 read pairs. Also note: If your panel does not contain control genes, the workflow should be modified by right clicking on the workflow in the tool box and opening a copy of the workflow. Remove the Map Reads to Human Control Genes workflow element plus its input and output, and save the modified workflow. When you run the modified workflow, human control genes are no longer required.

Figure 3.1: Analyze Viral Hybrid Capture Panel Data workflow.

Preliminary steps to run the Analyze Viral Hybrid Capture Panel Data workflow

Before starting the workflow,

- Download reference viral genomes using either the Download Custom Microbial Reference Database (see section VI), the viral databases from the Download Curated Microbial Reference Database tool or create a database using the Update Sequence Attributes in Lists tool (https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Update_Sequence_Attributes_in_Lists.html)

- Create a taxonomic profiling index for calculating abundance (see section 16.4).

How to run the Analyze Viral Hybrid Capture Panel Data workflow

To run the workflow, go to:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Taxonomic Analysis | Analyze Viral Hybrid Capture Panel Data

1. Specify the sample(s) or folder(s) of samples you would like analyze (figure 3.2) and click Next. Note that if you select several items, they will be run as batch units.
2. Specify the human control genes as a sequence list here (figure 3.3). Alternatively, if your panel does not contain control genes, the workflow should be modified by right clicking on the workflow in the tool box and opening a copy of the workflow. Then remove the Map Reads to Reference tool plus its input and output and save the modified workflow. When you run the modified workflow, human control genes are no longer required.

3. 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.

4. 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 3.4).

5. You can specify a trim adapter list and set up parameters if you would like to trim your sequences (figure 3.5).

The parameters that can be set are:

- **Trim ambiguous nucleotides**: if checked, this option trims the sequence ends based on the presence of ambiguous nucleotides (typically N).
- **Maximum number of ambiguities**: defines the maximal number of ambiguous nucleotides allowed in the sequence after trimming.
Figure 3.4: Choose which of the samples present in the selected folder(s) you want to analyze.

Figure 3.5: Choose trimming settings and optionally add an adapter trim list for trimming sequencing reads.

- **Trim using quality scores**: 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.
- **Trim adapter list**: Specifying a trim adapter list is optional but recommended to ensure the highest quality data for your analysis (figure 3.5).


6. In the next wizard window "Taxonomic Profiling", select the viral reference database index you will use to map the reads (figure 3.6). It is also possible to "Filter host reads". You must then specify the index of the host genome (in the case of human virus, the Homo sapiens GRCh38 for example). Note that if your panel uses human control genes, a taxonomic profiling index of the human genome should be used as "Host index".

7. In the next wizard window, select the viral reference database you will use to find the best matching reference (figure 3.7). The best matching reference will be used for read mapping and variant calling. If you wish to have variant calls annotated with amino acid changes the input database should contain CDS annotations.
8. In the next wizard window, specify the parameters for the **Low Frequency Variant Detection** tool (figure 3.8). Note that variants are filtered after variant detection to coverage $> 30x$ and frequency $\geq 20\%$.

The parameters that can be set are:

- **Required significance**: The required significance level for low frequency variant calls.
- **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.
Figure 3.8: Specify the parameters to be used by the Low Frequency Variant Detection tool.

- **Read direction filter**: 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**: This filter 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.

- **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 (we recommend to create a new folder for it) and click **Finish**.

The output will be saved in the folder you selected. An example of the output can be seen in figure 3.9.
The output generated for each sample is:

- **QC report raw reads**: QC report on the raw reads. Contains information on number of input reads, length, quality and nucleotide distributions.

- **Viral reads**: list of the sequences that were successfully trimmed and mapped to the best reference.

- **TaxPro report**: output from the Taxonomic Profiling tool. Contains information on number of reads mapping to the reference database and the host, if host filtering was enabled.

- **Read mapping human control genes**: mapping of the reads to the human control genes.

- **Read mapping human control genes report**: contains information on the read mapping to the selected controls such as number of reads mapped and read length distributions.

- **Read mapping**: output from the Local Realignment tool, mapping of the reads to the best match reference.

- **Best reference report**: contains information on best reference and the number of reads and unique reads mapped to this reference.

- **Best match sequence**: the sequence that matches the data best according to the Find Best References using Read Mapping tool.

- **Consensus sequence**: a consensus sequence generated from the read mapping. Consensus is not calculated in low coverage regions. These positions are instead replaced with Ns.

- **Low coverage areas**: track of regions of the best match reference genome with coverage < 30x.


- **Abundance table**: contains abundance for all detected viral species.

- **Viral reads**: read mapping to the viral reference database.
• **Annotated variant track**: output from the **Low Frequency Variant Detection** tool. Note that variants are post filtered to coverage > 30x and frequency ≥ 20%.

• **Amino acid track**: only generated if the best match contains CDS annotations.

For each batch analysis run, the following outputs are generated:

• **Combined report**: combines the information from the output report including QC, taxonomic profiling and mapping reports.

• **Merged abundance table**: An table containing abundance for all input samples. See figure 3.10 for an example of an analysis run of two HPV samples.

![Merged abundance table from analysis of two HPV samples](image)

Merged abundance tables can be used as input for various tools:

- Alpha Diversity, see section 7.3
- Beta Diversity, see section 7.4
- Differential Abundance Analysis, see section 7.6
- Create Heat Map for Abundance Table, see section 7.7

### 3.1.2 Data QC and Remove Background Reads

The **Data QC and Remove Background Reads** 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 Data QC and Remove Background Reads workflow, go to:

Toolbox | Template Workflows ([opens new window]) | Microbial Workflows ([opens new window]) | Metagenomics ([opens new window]) | Taxonomic Analysis ([opens new window]) | Data QC and Remove Background Reads ([opens new window])

You can select only one read file to analyze (figure 3.11). Alternatively, multiple files can be run using the batch mode.

In the "Trim Sequences" dialog, you can specify a **trim adapter list** and set up parameters if you would like to trim your sequences from adapters (figure 3.12).

The parameters that can be set are:
CHAPTER 3. MICROBIAL TEMPLATE WORKFLOWS

Figure 3.11: Select the reads. Use Batch mode if you have more than one set of reads.

Figure 3.12: You can choose to trim adapter sequences from your sequencing reads.

- **Quality limit**: defines the minimal value of the Phred score for which bases will not be trimmed.
- **Trim adapter list**: the adapter sequences to trim (if any).

In the Taxonomic Profiling dialog, select the "Species of interest taxpro index" you will use to map the reads (figure 3.13). Here, you can also choose to "Filter background reads". You must then specify the "Background taxpro index" (in the case of human microbiota, the Homo sapiens GRCh38 for example). The reference database 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). The host index and (if using the custom downloader) the microbial reference index are built with the Create Taxonomic Profiling Index tool (section 16.4).

The workflow will output three sequence lists:

- **Trimmed, cleaned reads**: trimmed reads mapping to the species of interest taxpro index of reference genome(s).
Figure 3.13: Select the reference database, and potentially a background taxpro index to remove possible contamination.

- **Background reads**: reads mapping to the background taxpro index.
- **Unmapped reads**: reads not mapping to the species of interest or the background taxpro index.

In addition, it generates three reports: a trimming report, a graphical QC report and a supplementary QC report. All of these should be inspected in order to determine whether the quality of the sequencing reads and the trimming are acceptable. For a detailed description of the QC reports and indication on how to interpret the different values, see [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html).


### 3.1.3 Data QC and Taxonomic Profiling

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

To run the tool, go to:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Taxonomic Analysis | Data QC and Taxonomic Profiling

You can select only one read file to analyze (figure 3.14). Alternatively, multiple files can be run using the batch mode.

In the "Trim Sequences" dialog, 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 3.15).

The parameters that can be set are:

- **Quality limit**: defines the minimal value of the Phred score for which bases will not be trimmed.
Also search on reversed sequence: the adapter sequences will also be searched on reverse sequences.

In the “Taxonomic Profiling” dialog (figure 3.16), choose the list of references that you wish to map the reads against. You could also remove host DNA by specifying a reference genome for the host (in the case of human microbiota, the Homo sapiens GRCh38 for example). The reference database 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). The host index and (if using the custom downloader) the microbial reference index are built with the Create Taxonomic Profiling Index tool (section 16.4).

The abundance table displays the names of the identified taxa (possibly with their underlying assemblies), along with their full taxonomy, the total amount of reads associated with that taxon, the total number of reads associated with the children of 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 6.2.1).

The Taxonomic Profiling report is divided in three sections:
• **Taxonomic Summary** Includes information about which taxonomic levels were found in the data sample and how many different taxa were found on each level.

• **Classification of reads** Summarizes the number of reads in the sample and the number of uniquely mapping reads.

• **Reference database Summary** Information on the applied reference database. A subsection is added in case any duplicates are found in the database - these are not used when constructing the taxonomic profile.

In addition, it generates three reports: a trimming report, a graphical QC report and a supplementary QC report. All of these should be inspected in order to determine whether the quality of the sequencing reads and the trimming are acceptable. For a detailed description of the QC reports and indication on how to interpret the different values, see [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html).


### 3.1.4 Merge and Estimate Alpha and Beta diversities

The **Merge and Estimate Alpha and Beta diversities** 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.

To run the tool, go to:

**Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Taxonomic Analysis | Merge and Estimate Alpha and Beta diversities**

In the first step, select several abundance tables (figure 3.17).

![Select abundance tables](image)

**Figure 3.17: 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 7.3 and section 7.4.

The Merge and Estimate Alpha and Beta Diversities workflow generates the results seen in figure 3.18.

Please refer to section 7.3 and section 7.4 to learn more about interpreting these results.
3.1.5 QC, Assemble and Bin Pangenomes

This workflow guides the investigator through the key steps to analyze whole-genome shotgun metagenomic reads and deconstruct them into 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 (also split in multiple sequence objects). The outputs are two sequence lists objects: one for reads and one for assembled contigs, labelled for bin association. Reports are also output at each step.

To run the workflow, go to:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Taxonomic Analysis | QC, Assemble and Bin Pangenomes

In the first step, one or more reads sequence objects are selected (figure 3.19).

The workflow first performs QC of raw reads using basic quality-based trimming, but fixed-length trimming can also be added. In the "Trim Reads" dialog, 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 3.20).
In the next step, QC-processed reads are assembled in contigs using the De novo Assemble Metagenome tool. Specify 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 3.21).

![Parameters for the De Novo Assembly Metagenome tool.](image)

Reads and contigs are then first binned according to taxonomic association and then based on sequence similarity. The tool is designed to work on contigs assembled from the same set of reads used as input (as in the workflow, see section 3.1.5). The tool will use the result of the De novo assembly configured here, but you can set the minimum contig length needed in the next dialog (figure 3.22).

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

- the file provided as "Reference indexes" is used to find taxonomic information for the reads
- the file provided as "Plasmid reference index" (once 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 custom downloader, the microbial reference index is built with the Create Taxonomic Profiling Index tool (section 16.4).

![Select the references and configure the Bin Pangenomes by Taxonomy.](image)

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 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.
In the next dialog (figure 3.23), configure the parameters for the Bin Pangenomes by Sequence: once again you can set the minimum contig length of the contigs generated by the De Novo Assembly Metagenome tool. You can also choose the maximum of iterations that should be performed, and how to label singletons (bins with only one genome).

The tool will produce the following outputs:


- A Bin Pangenomes by Taxonomy report (figure 3.24). Sequences with the same TaxBin label should have very similar, if not identical, taxonomy.

- A Bin Pangenomes by Sequence (figure 3.25). Sequences with same Bin label are closely related in sequence space.

- A list of reads, and a list of contigs, listing binned (and unbinned) reads and contigs from both binning steps.

Individual bins can be extracted from the sequence and contig lists (when seen as tables, you can find the bin label in the Assembly_ID column) and used for downstream analysis such as reference-based assembly (or re-assembly), functional analysis, typing etc.
CHAPTER 3. MICROBIAL TEMPLATE WORKFLOWS

3.2 Amplicon-Based Analysis template workflows

The Amplicon-Based Analysis template workflows are found at:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Amplicon-Based Analysis

3.2.1 Data QC and OTU Clustering

The Data QC and OTU Clustering workflow consists of 3 tools being executed sequentially (figure 3.26). The only necessary input to run the workflow are the reads you want to cluster. You also have the option to provide a list of the primers that were used to sequence these reads if you wish to perform the adapters trimming step with the Trim Sequences tool.

The first tool is the Trim Reads tool. Together with the sequencing primer list, this tool provides a list of trimmed sequences that will be the input of the Filter Samples Based on Number of Reads tool. The results of the trimming and the filter steps are compiled in two reports. The "filtered" list of reads (devoid of reads of poor quality) will be used for the final tool of the workflow, the OTU clustering tool. This tool will give a report, and an abundance table with the newly created
3.2.2 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 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.

The workflow requires a Trim Adapter List to remove adapters from the reads and a Taxonomic Profiling Index to assign taxonomies to ASVs. If the reads have already had adapters removed, we recommend creating a copy of the workflow (right-click the workflow in the Toolbox and select 'Open Copy of Workflow'), removing the Adapter Trim List input, and saving the modified workflow. When running the modified workflow, the Adapter Trim List can be left out to skip adapter trimming, but 'Quality trimming' and 'Automatic read-through adapter trimming' will still take place.
Launching the workflow

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

Toolbox | 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 Trim Adapter List corresponding to the adapters used for sequencing and click on Next (figure 3.27).
3. Select a Taxonomic Profiling Index and click on Next (figure 3.28).
4. The 'Configure batching' and 'Batch overview' steps can be left as is (figure 3.29), or configured as described in http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Launching_workflows_individually_in_batches.html.
5. Choose the trim length to use for your reads and decide whether to remove chimeras by toggling the 'Remove chimeras' box. Click on Next (figure 3.30).
   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. Finally, select a location to save outputs to and click on Finish.

Figure 3.27: Wizard step for selecting the Trim Adapter List.

Figure 3.28: Wizard step for selecting the Taxonomic Profiling Index.

Tools in the workflow and outputs generated

The Detect Amplicon Sequence Variants and Assign Taxonomies workflow consists of the below mentioned tools. See figure 3.31 for a full overview of the workflow.
3.2.3 Estimate Alpha and Beta Diversities

The Estimate Alpha and Beta Diversities workflow consists of 5 tools and requires only the OTU table as input file (figure 3.32).
Figure 3.31: Layout of the Detect Amplicon Sequence Variant and Assign Taxonomies workflow.

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 OTU abundance table with the OTU Clustering tool or the Data QC and OTU Clustering workflow. 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.

The first tool of the workflow is the **Filter OTUs Based on the Number of Reads**. The output is a reduced abundance table that will be used as input for three other tools:

- **Align OTUs with MUSCLE**, a tool that will produce an alignment used to reconstruct a Maximum Likelihood Phylogeny (see [link](http://resources.qiagenbioinformatics.com/manuals/clgenomicsworkbench/current/index.php?manual=Maximum_Likelihood_Phylogeny.html)), which will in turn output a phylogenetic tree also used as input in the following two tools.

- **Alpha diversity** tool

- **Beta diversity** tool

Running this workflow will therefore give the following outputs: a phylogenetic tree of the OTUs, a diversity report for the alpha diversity and a PCoA for the beta diversity.
3.3 Typing and Epidemiology template workflows

The Typing and Epidemiology template workflows are found at:

Toolbox | Template Workflows | Microbial Workflows | Typing and Epidemiology

3.3.1 Compare Variants Across Samples

The Compare Variants Across Samples can be used to compare samples originating from strains or species sharing a common reference. The workflow takes trimmed, host-filtered reads as input such as output produced by the Data QC and Remove Background Reads, Type a Known Species, Type Among Multiple Species or Analyze Viral Hybrid Capture Panel Data workflow. 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

Toolbox | Template Workflows | Microbial Workflows | Typing and Epidemiology | Compare Variants Across Samples

An overview of the workflow can be seen in figure 3.33.

- Select two or more read sets as input (figure 3.34). The workflow uses internal batching and creates an analysis for each sample as well as a combined variant track and SNP tree.

- Select the reference to use (figure 3.35). The reference should match all the samples.
Select a CDS track associated with the reference (figure 3.36).

In the Result handling window, pressing the button **Preview All Parameters** allows you to preview - but not change - all parameters.
Saving the workflow output will generate the files shown in (figure 3.37) and optionally, a workflow result metadata table. The workflow generates outputs for each batch analysis run as well as a folder for each sample. For each sample, the following is output:

- **Annotated variant track**: output from the Low Frequency Variant Detection tool after coverage and quality filtering. Note that it is possible to export multiple variant track
files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

- **Amino acid track**: amino acid track including amino acid changes resulting from the called variants.

- **Read mapping**: output from the **Local Realignment** tool, mapping of the reads to the specified reference. For increased sensitivity, duplicate mapped reads are removed before local realignment.

- **Track list**: output from the **Create Track List** tool. The track list combines the read mapping, variant, amino acid and CDS tracks. An example can be seen in figure 3.38.

![Figure 3.38: The track list generated for each sample analysis](image)

For each batch analysis run, the following outputs are generated:

- **Variant track list for all samples**: output from the **Create Track List** tool. The track combines the variant tracks for all analyzed samples.

- **Combined QC report**: a combined report built from QC for sequencing reads, Read mapping summary and QC for read mapping. This report contains a summary of all analyzed samples.

- **SNP tree report**: summarizes the consequence of 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 (see figure 3.39).

![Figure 3.39: SNP matrix for pairwise comparisons of all samples included in the analysis](image)
CHAPTER 3. MICROBIAL TEMPLATE WORKFLOWS 54

Figure 3.40: An output SNP tree of 21 samples with leaf nodes colored using metadata annotation

- **SNP tree**: the output tree built from the SNPs called in all samples (see figure 3.40). A number of different visualizations are available, see section 10.1.2. Here, the leaf nodes have been colored according to geographic location of the collected samples.

For more information on the tree tools, see section 10.1.

3.3.2 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 workflow, go to

Toolbox | Template Workflows | Microbial Workflows | Typing and Epidemiology | Create MLST Scheme with Sequence Types

You can select one or more assemblies as input (figure 3.41). At least one of the assemblies must be annotated with CDS regions.

Figure 3.41: Select the high-quality references serving as the basis for the scheme

In "Create MLST scheme" dialog (figure 3.42), 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.

- **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 3.43), 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 3.42), 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.

• **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 3.44) and optionally, a workflow result metadata table.

![Figure 3.44: The output from Create MLST Scheme with Sequence Types](image)

- **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.


### 3.3.3 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 3.45, 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 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 9.1.1.

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 10.2.

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 3.46) that will perform a re-mapping of the reads followed by variant calling.
How to run the Map to Specified Reference workflow

This workflow is intended for read mapping and variant calling of the samples to a common reference. To run the workflow, go to:

Toolbox | 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 3.47) 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 use (figure 3.48) 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.
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 3.49).

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 3.50). Learn about trim adapter lists at [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_adapter_list.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_adapter_list.html).

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 3.51).

The parameters that can be set are:

- **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 3.52) before clicking **Next**.

![Figure 3.52](image)

**Figure 3.52**: *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 3.53):
CHAPTER 3. MICROBIAL TEMPLATE WORKFLOWS

Figure 3.53: Output files from the Map to Specified Reference workflow.


- **Reads Track**: output from the Local Realignment tool.

- **Variant Track**: output from the Basic Variant Detection tool. Note that it is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

You have now the data necessary to create a SNP tree for your samples as explained in section 10.1.

Note that the tool will output, among other files, variant tracks. It is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

### 3.3.4 Type Among Multiple Species

The **Type Among Multiple Species** workflow is designed for typing a sample among multiple predefined species (figure 3.54).

![Figure 3.54: Overview of the template Type Among Multiple Species workflow.](image)

It allows identification of the closest matching reference species among the user specified reference list(s) which 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 user specified resistance database.

The workflow also automatically associates the analysis results to the user specified Result Metadata Table. For details about searching and quick filtering among the sample metadata and generated analysis result data (see section 2.2.1).

**Preliminary steps to run the Type Among Multiple Species workflow**

Before starting the workflow,

- Download microbial genomes using either the Download Custom Microbial Reference Database tool, the prokaryotic databases from the Download Curated Microbial Reference Database tool or the Download Pathogen Reference Database tool (see section VI). Databases can also be created using the Update Sequence Attributes in Lists tool.
- Download the MLST schemes using the Download MLST Scheme tool (see section 14.2).
- Download the database for the Find Resistance with Nucleotide DB 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 2.1).

When you are ready to start the workflows, your navigation area should look similar to the figure 3.55.

![Figure 3.55: Overview of the Navigation area after creating the result metadata table and downloading the databases and MLST schemes necessary to run the workflows.](image)

**How to run the Type Among Multiple Species workflow**

To run the workflow for one or more samples containing multiple species, go to

```
Toolbox | 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 (figure 3.56) 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 use (figure 3.57) and click Next.
3. 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.

4. 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 3.58).

5. 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 3.59). Learn about trim adapter lists at http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_adapter_list.html.

The parameters that can be set are:

- **Trim ambiguous nucleotides**: if checked, this option trims the sequence ends based on the presence of ambiguous nucleotides (typically N).
- **Maximum number of ambiguities**: defines the maximal number of ambiguous nucleotides allowed in the sequence after trimming.
- **Trim using quality scores**: if checked, and if the sequence files contain quality scores from a base-caller algorithm, this information can be used for trimming sequence ends.
Figure 3.58: Choose which of the samples present in the selected folder(s) you want to analyze.

Figure 3.59: You can choose to trim adapter sequences from your sequencing reads.

- **Quality limit**: defines the minimal value of the Phred score for which bases will not be trimmed.

Click **Next**.

6. Choose the **species-specific references** to be used by the **Find Best Matches using K-mer Spectra** tool (figure 3.60). The list can be a fully customized list, the downloaded bacterial genomes from NCBI list (see section **16.1.1**) or a subset of it. Click **Next**.
7. Specify **MLST schemes** to be used for the **Identify MLST Scheme from Genomes** tool so they correspond to corresponding to the chosen reference list(s) (figure 3.61).

8. Specify the **resistance database** (figure 3.62) and set the parameters for the **Find Resistance with Nucleotide DB** tool.

   The parameters that can be set are:
   
   - **Minimum Identity %**: is the threshold for the minimum percentage of nucleotides that
CHAPTER 3. MICROBIAL TEMPLATE WORKFLOWS

Figure 3.62: Specify the resistance database to be used for the Find Resistance with Nucleotide DB tool.

are identical between the best matching resistance gene in the database and the corresponding sequence in the genome.

- **Minimum Length %**: reflect the percentage of the total resistance gene length that a sequence must overlap a resistance gene to count as a hit for that gene. Here represented as a percentage of the total resistance gene length.

- **Filter overlaps**: will perform 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**.

9. Specify the parameters for the **Type with MLST Scheme** tool (figure 3.63).

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.

- **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.

- **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.

Click **Next**.

10. Specify the parameters for the **Fixed Ploidy Variant Detection** tool (figure 3.64) before clicking **Next**.
Figure 3.63: Specify the parameters for MLST typing.

The parameters that can be set are:

- **Required variant probability (%)**: The 'Required variant probability' is the minimum probability value of the 'variant site' required for the variant to be called. Note that it is not the minimum value of the probability of the individual variant. For the Fixed Ploidy Variant detector, if a variant site - and not the variant itself - passes the variant probability threshold, then the variant with the highest probability at that site will be reported even if the probability of that particular variant might be less than the threshold. For example if the required variant probability is set to 0.9 then the individual probability of the variant called might be less than 0.9 as long as the probability of the entire variant site is greater than 0.9.

- **Ignore positions with coverage above**: Ignore positions with a read-coverage larger than this value.

- **Restrict calling to target regions**: Select a region track to specify the regions in which variants should be called.

- **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.
Figure 3.64: Specify the parameters to be used for the Fixed Ploidy Variant Detection tool.

- **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.

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 to this effect) and click on the button labeled **Finish**.

Outputs are generated on a per sample basis and on a summary level. You can find them all in the new folder you created to save them (figure 3.65), but those marked with a (*) have also been added automatically to the New Metadata Result Table (see section 3.3.5 to understand where your results have been saved).

![Figure 3.65: Output files from the Type Among Multiple Species workflow.](image-url)

For each sample, the following outputs are generated:


• (***)**Contaminants report**: lists the best match as well as possible contaminants along with coverage level distributions for each reference genome listed.
• **Best match**: sequence that matches best the data according to the **Find Best Matches using K-mer Spectra** tool.

• **Matches table**: contains the best matching sequence, a list of all (maximum 100) significantly matching references and a tabular report on the various statistical values applied.

• **Read mapping best match**: output from the **Local Realignment** tool, mapping of the reads using the Best Match as reference.

• **Trimmed, cleaned sequences**: list of the sequences that were successfully trimmed and mapped to the best reference.


• **Contig list**: contig list from the **De novo assembly** tool.

• (***)Contig list resistance table**: result table from the **Find Resistance with Nucleotide DB** tool, reports the found resistance.

• (***)Typing report**: output from the **Type with MLST Scheme** tool, includes information on which MLST scheme was applied, the best matching sequence type (ST) as well as an overview table with sample information and a table summarizing the allele calls.

• **Typing result**: output from the **Type with MLST Scheme** tool, includes information on kmer fractions, kmer hit counts and allele count, identified and called.

• **Variant Track**: output from the **Fixed Ploidy Variant Detection** tool. Note that it is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

For each batch analysis run, the following outputs are generated:

• **Combined report**: combines the information from the trim report and MLST typing report.

• **Results metadata table**: a table containing summary information for each sample analyzed and a quick way to find the associated files. In addition, an extra column in the Result Metadata Table called "Best match, average coverage" helps the user to decide if a best match is significant, well covered and of good quality. This is especially helpful when a sample has low quality but is not contaminated.

**Example of results obtained using the Type Among Multiple Species workflow**

The following example includes typing of 2 samples: 1 *Salmonella enterica* (acc no. ERR277212), and 1 *Yersinia ruckeri* (acc no SRR3152422). Using the workflow **Type Among Multiple Species** workflow, analysis results are automatically summarized in the Result Metadata Table as shown in figure 3.66. The analysis results in this example include resistance found for antibiotic inactivation enzyme, name of the best matching reference, applied MLST scheme, detected sequence type and typing status. You could also choose (using options in the Table Setting
window) to display additional information such as the ‘Best Match, Species’ and ‘Find Resistance With Nucleotide DB’ for example.

Analyzing samples in batch will produce a large amount of output files, making it necessary to filter for the information you are looking for. 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 10.1).
- Generation of K-mer Trees for identification of the closest common reference across samples (section 10.2).
- Run validated workflows (workflows that are associated with a Result Metadata Table and saved in your Navigation Area).

Note that the tool will output, among other files, variant tracks. It is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

### 3.3.5 Type a Known Species

The **Type a Known Species** workflow is designed for typing of samples representing a single known species (figure 3.67). It identifies the associated MLST, determines variants found when mapping the sample data against the user specified reference, and finds occurring resistance genes if they match genes within the user specified resistance database.

**Preliminary steps to run the Type a Known Species workflow**
Before starting the workflow,

- Download microbial genomes using either the **Download Custom Microbial Reference Database** tool, the prokaryotic databases from the **Download Curated Microbial Reference Database** tool or the **Download Pathogen Reference Database** tool (see section VI). Databases can also be created using the **Update Sequence Attributes in Lists** tool.

- Download the MLST schemes using the **Download MLST Schemes** tool (see section 14.2).

- Download the database for the **Find Resistance with Nucleotide DB** 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 2.1).

When you are ready to start the workflows, your navigation area should look similar to the figure 3.68 with one MLST scheme and one reference genome.

![Figure 3.68: Overview of the Navigation area after creating the result metadata table and downloading the databases and MLST scheme necessary to run the workflow.](image)

**How to run the Type a Known Species workflow**

To run the workflow, go to:

**Toolbox | 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 (figure 3.69) 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 (figure 3.70) and click **Next**.
3. 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**.

4. 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 3.71).

5. 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 3.72). Learn about trim adapter lists at [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_adapter_list.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_adapter_list.html).

The parameters that can be set are:

- **Trim ambiguous nucleotides**: if checked, this option trims the sequence ends based on the presence of ambiguous nucleotides (typically N).
- **Maximum number of ambiguities**: defines the maximal number of ambiguous nucleotides allowed in the sequence after trimming.
- **Trim using quality scores**: if checked, and if the sequence files contain quality scores from a base-caller algorithm, this information can be used for trimming sequence ends.
Figure 3.71: Choose which of the samples present in the selected folder(s) you want to analyze.

Figure 3.72: You can choose to trim adapter sequences from your sequencing reads.

- **Quality limit**: defines the minimal value of the Phred score for which bases will not be trimmed.

Click **Next**.

6. Choose the **species reference** to be used by the **Map reads to reference** tool (figure 3.73). Click **Next**.

7. Specify the **resistance database** (figure 3.74) and set the parameters for the **Find Resistance with Nucleotide DB** tool.

The parameters that can be set are:
Figure 3.73: Specify the reference for the Map reads to reference tool.

Figure 3.74: Specify the resistance database to be used for the Find Resistance with Nucleotide DB tool.

- **Minimum Identity %**: is 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 %**: reflect the percentage of the total resistance gene length that a sequence must overlap a resistance gene to count as a hit for that gene. Here represented as a percentage of the total resistance gene length.

- **Filter overlaps**: will perform 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 parameters for the **Type with MLST Scheme** tool (figure 3.75).

![Figure 3.75: Specify the parameters for MLST typing.](image)

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.

- **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.

- **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 that needs 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 of 0.99 would mean that on average 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 3.76) before clicking **Next**.

The parameters that can be set are:

- **Ploidy**: The expected ploidy of the species of interest.

- **Required variant probability (%)**: The 'Required variant probability' is the minimum probability value of the 'variant site' required for the variant to be called. Note that it is not the minimum value of the probability of the individual variant. For the Fixed Ploidy Variant detector, if a variant site - and not the variant itself - passes the variant probability threshold, then the variant with the highest probability at that
Figure 3.76: Specify the parameters to be used for the Fixed Ploidy Variant Detection tool.

site will be reported even if the probability of that particular variant might be less than the threshold. For example if the required variant probability is set to 0.9 then the individual probability of the variant called might be less than 0.9 as long as the probability of the entire variant site is greater than 0.9.

- **Ignore positions with coverage above**: Ignore positions with a read-coverage larger than this value.
- **Restrict calling to target regions**: Select a region track to specify the regions in which variants should be called.
- **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**: This filter 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.

Click **Next**.

10. 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 new folder you created (figure 3.77), but those marked with a (*) in the list below will also be added automatically to the Metadata Result table.


• **Read mapping**: output from the Local Realignment tool, mapping of the reads to the specified reference.
Figure 3.77: Output files from the Type a Known Species workflow.

- **Trimmed, cleaned sequences**: list of the sequences that were successfully trimmed and mapped to the best reference.


- **Contig list**: contig list from the De novo assembly tool.

- **(*)Contig list resistance table**: result table from the Find Resistance with Nucleotide DB tool, reports the found resistance.

- **(*)Typing report**: output from the Type with MLST Scheme tool, includes information on the applied MLST scheme, the best matching sequence type (ST) as well as an overview table with sample information and a table summarizing the allele calls.

- **Typing result**: output from the Type with MLST Scheme tool, includes information on kmer fractions, kmer hit counts and allele count, identified and called.

- **Variant Track**: output from the Fixed Ploidy Variant Detection tool. Note that it is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

For each batch analysis run, the following outputs are generated:

- **Combined report**: combines the information from the trim report and MLST typing report.

- **Results metadata table**: An table containing summary information for each sample analyzed and a quick way to find the associated files. In addition, an extra column in the Result
Metadata Table called "Best match, average coverage" helps the user to decide if a best match is significant, well covered and of good quality. This is especially helpful when a sample has low quality but is not contaminated.

**Example of results obtained using the Type a Known Species workflow**

In this section, sequence data from five *Salmonella enterica* samples were typed using a customized *Salmonella enterica* version of the Type a Known Species workflow: the specified reference and the MLST scheme was specified as *Salmonella enterica*. The created Result Metadata Table shows these five samples and associated metadata for sample with accession no ERR277212.

Once the workflow analysis was performed, additional columns listing the analysis results (e.g., MLST, best matching reference and identified resistance genes) have automatically been added to the Result Metadata Table (figure 3.78). To ease the overview regarding applied analysis parameters, columns stating the workflow specified reference list, MLST scheme and resistance gene database have also been added automatically to the table.

![Figure 3.78: View of the updated Result Metadata Table and associated elements once the Type a Known Species workflow has been executed on the Salmonella (acc no ERR277212) sample.](image)

Analyzing samples in batch will produce a large amount of output files, making it necessary to filter for the information you are looking for. 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 10.1).
- Generation of K-mer Trees for identification of the closest common reference across samples (section 10.2).
- Run validated workflows (workflows that are associated with a Result Metadata Table and saved in your Navigation Area).
For more information on filtering and running analysis directly from a Result Metadata Table section, see section 2.2.

Note that the tool will output, among other files, variant tracks. It is possible to export multiple variant track files from monoploid data into a single VCF file with the Multi-VCF exporter. This exporter becomes available when installing the CLC Microbial Genomics Module. All variant track files must have the same reference genome for the Multi-VCF export to work.

3.4 QIAseq Analysis template workflows

The QIAseq Analysis template workflows are found at:

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

The QIAseq template workflows are configured with workflow roles and therefore require reference data sets containing elements with the relevant roles.

To learn about workflow roles in workflows, see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Configuring_input_output_elements.html.

QIAGEN reference data sets are available from QIAGEN Sets Reference Data Library accessible via References ( ) in the top Toolbar. Alternatively, access it from the Utilities menu and select Manage Reference Data ( )


3.4.1 Analyze QIAseq xHYB Viral Panel Data

The Analyze QIAseq xHYB Viral Panel Data template workflow trims reads, performs taxonomic profiling, and calls viral variants.

It is suitable for data 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

QIAGEN reference data set

The QIAseq xHYB Viral Panels reference data set is available from QIAGEN Sets Reference Data Library accessible via References ( ) in the top Toolbar.

Launching the workflow

The Analyze QIAseq xHYB Viral Panel Data workflow is at:
Launch the workflow and step through the wizard.

1. Select the sequence list(s) containing the reads to analyze. Click on Next.

2. Select a reference data set or select "Use the default reference data" to configure the reference data elements individually in subsequent wizard steps (figure 3.79). Click on Next.

3. Choose whether batch units should be defined based on organization of the input data, or by provided metadata (figure 3.80). For information on how to use metadata when running part of a workflow multiple times, see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_part_workflow_multiple_times.html.

4. Next, you can review the batch units resulting from your selections above. Click on Next.

5. Verify or select the viral taxonomic profiling index (figure 3.81) and click on Next.

6. Verify or select the host taxonomic profiling index and click on Next.

7. Select the viral reference database(s). If in the first step you selected the QIAseq xHYB Viral Panels reference set, you can now select which of the available viral reference databases from that set to apply (3.82). If you chose to use the default reference data, select a reference database and click on Next.

8. Verify or select the control genes and click on Next.

9. Specify the trim settings.

10. Specify Taxonomic Profiling settings (figure 3.83).

11. Specify Low Frequency Variant Detection settings, see figure 3.84.

12. Finally, select a location to save outputs to and click on Finish.
Tools in the workflow and outputs generated  The workflow Analyze QIAseq xHYB Viral Panel Data consists of the below mentioned tools. See figure 3.85 for a full overview of the workflow.

The tools used in this workflow are:

- **QC for Sequencing Reads** performs basic QC on the sequencing reads. The output, which here is included in a combined report, can be used to evaluate the quality of the sequencing reads. See [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html).


- **Taxonomic Profiling** provides insight into the taxonomic composition of the sample and
estimates the relative abundance of the detected taxa. See section 6.2. 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 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 [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Map_Reads_Reference.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Map_Reads_Reference.html). This serves as a QC step to verify mapping to the 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 9.2.


- **Remove Duplicate Mapped Reads** removes duplicate reads derived from PCR amplification.
(or other enrichment) during sample preparation from the mapping. See [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Remove_Duplicate_Mapped_Reads.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Remove_Duplicate_Mapped_Reads.html). The output reads track is used as input for **Local Realignment**.


- **Create Mapping Graph** and **Identify Graph Threshold Areas** combined creates a track with regions with coverage below a threshold. For this workflow, the threshold is set to 30. See [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Create_Mapping_Graph.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Create_Mapping_Graph.html) and [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Identify_Graph_Threshold_Areas.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Identify_Graph_Threshold_Areas.html).


The sample-specific outputs provided by this workflow are:

- **QC Report Raw Reads** provides QC metrics on the raw reads.

- **Abundance Table** provides the abundance of each identified taxa, along with their full taxonomy. See section 6.2.1.

- **Read Mapping Human Control Genes** contains the host reads mapped against the control gene reference.

- **Viral Reads** is the list of reads that mapped to the viral taxonomic profiling index.

- **Find Best Reference Report** is a report of the references found using **Find Best Reference**.

- **Best Match Sequence** the "Best match" reference sequence as identified by the Find Best References using Read Mapping tool.

- **Read Mapping** displays the mapping of viral reads against the "Best match" viral reference.
• **Consensus Sequence** contains the viral consensus sequence(s) extracted from the Local Realignment reads track.

• **Annotated Variant Track** contains the list of detected variants left after filtering, annotated with amino acid changes.

• **Amino Acid Track** contains a list of amino acid changes.

• **Low Coverage Areas** provides a list of low coverage regions in the mapping of viral reads against the "Best match" viral reference.

• **Track List** is a collection of the following viral tracks: consensus sequence, reads, variants, amino acid changes, and low coverage regions.


The combined outputs provided by this workflow are:

• **Taxonomic Profiling Report** combines taxonomic profiling report content across samples in the workflow run. See section 6.2.2.

• **Human Control Genes Read Mapping Report** holds information about the mapping of host reads to the human control genes for all samples in the workflow run.

• **Combined Report** combines information from various tools, including QC, taxonomic profiling and mapping reports.

• **Merged Abundance Table** provides abundances for the detected taxa for all samples in the workflow run. See section 6.2.1.

### 3.4.2 Find QIAseq xHYB AMR Markers

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

It is suitable for data generated with the QIAseq xHYB AMR Panel.

**QIAGEN reference data set**

The QIAseq xHYB AMR Panel reference data set is available from QIAGEN Sets Reference Data Library accessible via References in the top Toolbar.

**Launching the workflow**

The QIAseq xHYB AMR Markers workflow is at:

Toolbox | Template Workflows | Microbial Workflows | QIAseq Analysis | Find QIAseq xHYB AMR Markers

Launch the workflow and step through the wizard.
1. Select the sequence list(s) containing the reads to analyze and click on **Next**.

2. Select a reference data set or select "Use the default reference data" to configure the reference data elements individually in subsequent wizard steps (figure 3.86). Click on **Next**.

3. Choose whether batch units should be defined based on organization of the input data, or by provided metadata (figure 3.87). For information on how to use metadata when running part of a workflow multiple times, see [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_part_workflow_multiple_times.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Running_part_workflow_multiple_times.html).

4. Next, you can review the batch units resulting from your selections above. Click on **Next**.

5. Verify or select the reference marker database (figure 3.88). Click on **Next**.

6. Verify or select control genes (figure 3.89). Click on **Next**.

7. Specify the trim settings (figure 3.90) and click on **Next**.

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

---

**Figure 3.86:** Select reference data set.

**Figure 3.87:** Define batch units.

**Figure 3.88:** Select the reference marker database.
Tools in the workflow and outputs generated

The workflow Find QIAseq xHYB AMR Resistance consists of the below mentioned tools. See figure 3.91 for a full overview of the workflow.

The tools included in the workflow are:

- **QC for Sequencing Reads** performs basic QC on the sequencing reads. The output, which here is included in a combined report, can be used to evaluate the quality of the sequencing reads. See [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=QC_Sequencing_Reads.html).


  Together, **Map Reads to Human Control Genes** and **QC for Read Mapping** serve to verify mapping to the human control genes.

- **Find Resistance with ShortBRED** detects and quantifies the presence of antimicrobial resistance marker genes of interest. See section 13.3.

- **Merge Abundance Tables** merges the sample-specific abundance tables to one combined abundance table. See section 7.1.

The outputs provided by this workflow are:

- **Resistance table** is the result table from **Find Resistance with ShortBRED**. The sample-specific output provides the abundance of each detected AMR marker. See section 13.3.1.

- **Read Mapping Human Control Genes** is a sample-specific reads track containing the reads that mapped to the human control genes.

- **Combined Report** contains information on the results from the individual tools for all the samples included in the workflow run.

- **Merged Resistance Table** provides the abundances of the detected AMR marker across samples. See section 13.3.1.
Part III

Metagenomics
Chapter 4

De Novo Assemble Metagenome

Before assembly, adapters should be removed from sequences. This can be done using Trim Reads (http://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 itself should 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.

To run the De Novo Assemble Metagenome tool:

Toolbox | 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 predefined 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.

![Image of De Novo Assemble Metagenome tool](image)

Figure 4.1: Setting parameters for the assembly.

Select **Create report** to create a summary report containing statistics on input reads and output contigs.

**The De Novo Assemble Metagenome output**

The De Novo Assemble Metagenome tool will output a list of contigs. If **Perform scaffolding** was selected, scaffolds will appear at the bottom of the contig list as scaffold_1, scaffold_2, etc.
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:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Amplicon-Based Analysis

For more information on the template workflows, see section 3.2.

### 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 5.3.3), 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

Toolbox | 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

**Toolbox | 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).

![Figure 5.2: Output table from the Filter Samples Based on Number of Reads tool.](image)

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

**Toolbox | 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 ACGACGACGnnnnnnnnnnnnnnnnnnnnnTACTACTAC. 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.

The OTU clustering tool produces several outputs:

- a sequence list of the OTU centroids
- abundance tables with the newly created OTUs and the chimeras. Each table gives abundance of the OTU or chimeras at each site, as well as the total abundance for all samples.
- a report that summarizes the results of the OTU clustering
- if the input data is paired-end, a report about the merging of overlapping reads

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 (http://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.

• **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.

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.

The option "Include all reads" is left unchecked by default. Selecting this option will include the non-merged reads in the OTU clustering analysis. Doing so may increase greatly the running time of the analysis. However, people working with data expected to have non-overlapping paired reads (for example fungal ITS data) should check this option to include all reads in the analysis.
5.3.2 OTU clustering outputs

The OTU clustering tool produces several outputs: a sequence list of the OTU centroids, and two abundance tables with the newly created OTUs and chimeras.

In the OTU merge report - generated if the input reads were paired - the following statistics are given for each sequence list input in the tool: a summary of merged and nonmerged sequences, an alignment score distribution and a merged pairs length distribution.

The tool also produces a report (figure 5.5).

1 Overall summary

<table>
<thead>
<tr>
<th>Input database size</th>
<th>Filtered database size</th>
<th>OTUs based on database</th>
<th>De novo OTUs</th>
<th>Total predicted OTUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>99,322</td>
<td>1,237</td>
<td>974</td>
<td>474</td>
<td>1,448</td>
</tr>
</tbody>
</table>

2 Read summary

<table>
<thead>
<tr>
<th>Number of reads</th>
<th>Filtered reads</th>
<th>Unique reads after filtering</th>
<th>Chimeric reads</th>
<th>Unique chimeric reads</th>
<th>Reads in OTUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>82,371</td>
<td>47,870</td>
<td>6,646</td>
<td>514</td>
<td>172</td>
<td>33,987</td>
</tr>
</tbody>
</table>

3 Sample details

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total number of reads</th>
<th>Filtered or chimeric reads</th>
<th>Reads in OTUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT-A-B_1001_1R1_001 (paired)</td>
<td>5,601</td>
<td>3,340</td>
<td>2,261</td>
</tr>
<tr>
<td>merged fixedLength coverage OK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT-A-C_1001_1R1_001 (paired)</td>
<td>9,370</td>
<td>5,035</td>
<td>4,335</td>
</tr>
<tr>
<td>merged fixedLength coverage OK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT-B-A_1001_1R1_001 (paired)</td>
<td>6,512</td>
<td>3,676</td>
<td>2,836</td>
</tr>
<tr>
<td>merged fixedLength coverage OK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CT-B-B_1001_1R1_001 (paired)</td>
<td>8,181</td>
<td>4,607</td>
<td>3,574</td>
</tr>
<tr>
<td>merged fixedLength coverage OK</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.5: Example of report produced by the OTU clustering tool.

The content of the report is stated below.

- **In the section Overall summary**
  - **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.

- **In the section Read summary**
  - **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.

- In the section **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.

### 5.3.3 The OTU abundance table

The OTU abundance table contains a list of OTUs or chimeras, 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 5.6)

![Figure 5.6: OTU abundance table.](image)

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 7.8), 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 tables 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 table, then the merge should be redone excluding the samples with zero control read counts.

- **Stacked visualization view**
In the Stacked Bar (figure 5.7) and Stacked Area Charts (figure 5.8), 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 taxonomy level selected. The relevant data points will automatically be summed accordingly.

Figure 5.7: Stacked bar of the microbial community at the class level for 4 different samples.

Figure 5.8: Stacked area of the microbial community at the phylum level for 11 different sites.

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 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 5.19).

![Sunburst view of the microbial community showing all taxa belonging to the kingdom bacteria.](image)

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 5.10).

**5.3.4 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 ([biom-format.org](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.

### 5.4 Remove OTUs with Low Abundance

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
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.

5.5 Align OTUs with MUSCLE

To estimate Alpha and Beta diversity, OTUs must initially be aligned with the MUSCLE tool of the CLC Microbial Genomics Module:

Choose an OTU abundance table as input. The next wizard window allows you to set up the alignment parameters with MUSCLE (figure 5.11).

- **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 (use the Launch button to find it in the Toolbox). For more information, see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Maximum_Likelihood_Phylogeny.html.

### 5.6 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:

Toolbox | Template Workflows | Microbial Workflows | Metagenomics | Amplicon-Based Analysis | Detect Amplicon Sequence Variants and Assign Taxonomies workflow

For more information, see section 3.2.2.
5.6.1 Detect Amplicon Sequence Variants parameters

To run the Detect Amplicon Sequence Variants tool, go to

Toolbox | 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 (http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Trim_Reads.html).

Set the Trim and filter parameters (figure 5.12):

- **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 5.12: Trim and filter parameter settings](image)

5.6.2 Detect Amplicon Sequence Variants output

Click Next to select the output (figure 5.13).

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.
CHAPTER 5. AMPLICON-BASED ANALYSIS

Figure 5.13: Detect Amplicon Sequence Variants output options

The ASV report

1 Summary

- **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 sequence 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.

2 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.

Figure 5.14: First sections of the Detect Amplicon Sequence Variants report

- **Summary** (figure 5.14)

- **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 7.8 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 7.2).

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 5.15) 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.

![Image](image.png)

Figure 5.15: 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 tables 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 table, then the merge should be redone excluding the samples with zero control read counts.
Stacked Visualization view

Figure 5.16: The ASV abundance table Stacked Bar Chart with ASVs aggregated on Name shows the relative abundance of ASVs.

Adjust the Side panel setting Aggregate feature to Name (figure 5.16) 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:

- With Chart type you can switch between Bar Chart (figure 5.18) and Area Chart (figure 5.17).
- When sample metadata is applied, use Aggregate sample to aggregate based on metadata attributes e.g. replicates from the same sample origin.

Figure 5.17: 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 5.18).
Figure 5.18: Stacked Bar Chart of a merged ASV abundance tables with assigned taxonomies aggregated on Genus level.

- **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 5.19).

  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 the center of the view.

Figure 5.19: Sunburst view of the microbial community showing all taxa belonging to the kingdom bacteria.

### 5.6.3 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**.
Chapter 6

Taxonomic Analysis

Template workflows for taxonomic analysis are available at:

For more information, see section 3.1.

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, and we provide a template workflow QC, Assemble and Bin Pangenomes, that constructs lists of binned assembled contigs and reads via the two methodologies above and starting from raw reads (see section 3.1.5).

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 5 consecutive steps:

1. Obtain taxonomic information for reads
2. Obtain plasmid information for reads
3. Map reads to contigs
4. Assign taxonomic and plasmid labels to contigs
5. Group and filter contigs according to labels (Contig purity)

To start the tool, go to:

Toolbox | Microbial Genomics Module | Metagenomics | Taxonomic Analysis | Bin Pangenomes by Taxonomy

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

The tool is designed to work on contigs assembled from the same set of reads used as input, previously assembled using the De Novo Assembly Metagenome tool (as in the workflow, see section 3.1.5). 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 indexes" is used to find taxonomic information for the reads
• the file provided as "Plasmid reference index" (once 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 16.4).

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 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 the tool's outputs (figure 6.3):

- 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 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, where bins are ordered in order of completeness (see above).

The standard output of the Bin Pangeneomes 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 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 (similar 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.

To start the tool, go to:

![Toolbox | Microbial Genomics Module | Metagenomics | Taxonomic Analysis | Bin Pangenomes by Sequence](image)

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.](image)

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

- **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.
CHAPTER 6. TAXONOMIC ANALYSIS

6.2 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 of those best-match genomes.

If a host genome is provided, reads that map better to this are removed. For paired reads, when a read pair is broken, either because only one read in the pair matches, or because the distance or relative orientation is wrong, both reads are discarded.
After mapping, the tool performs qualification and quantification of the abundance of each qualified taxon, and finally compiles the results into an abundance table.

The purpose of the **qualification** phase is to determine whether a particular taxon is represented in a sample. The qualification is based on a confidence score that a reference did not receive its reads by pure chance. Any taxon with a confidence score < 0.995 will be ignored and the reads assigned to it 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.

The purpose of the **quantification** phase is to estimate how abundant the qualified taxa are. It is based on the number of reads assigned to that taxon. For data sets with varying read length, the abundance values may optionally be adjusted to correct for a skewed read distribution between taxa (see below).

The detection limit of the tool is now controlled by the single read mapping parameter "Minimum seed length". Increasing this value will give higher precision in the taxa called (true positives), while lowering it will give more taxa called at the cost of precision (more false positives).

To run the Taxonomic Profiling tool, go to

**Toolbox | Microbial Genomics Module | Metagenomics | Taxonomic Analysis | Taxonomic Profiling**

You can select only one read file to analyze (figure 6.6). If the sample to be analyzed is split up into several files, the files need to be merged with the Create Sequence List tool before running the Taxonomic Profiling tool.

![Figure 6.6: Select the reads.](image)

In the "Parameters" dialog, provide the reference database you will use to map the reads (figure 6.7). It is also possible to "Filter host reads". You must then specify the host genome as a taxonomic profiling index (for example Homo sapiens GRCh38 in the case of human microbiota). More generally, the "Host genome index" acts as a background database for reads that are expected to be present in your sample but unimportant to your analysis and may hence obscure the final abundances due to misassignments. For example, if the interest is to only see the bacterial fraction of a metagenome, a viral/fungal background database may be supplied as "host genome" to remove viral/fungal reads from the analysis. The reference database 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). The host index and (if using the custom downloader) the microbial reference index are built with the Create Taxonomic Profiling Index tool (section 16.4).
The read mapping parameters used in Taxonomic Profiling are the standard read mapping parameters (see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html) except for the Minimum seed length, which may be specified explicitly here. The Minimum seed length defines the minimum seed (word) size, i.e., perfect match-length, for a position in the reference to be considered a valid candidate when matching the read. If no seed longer than this length can be found in the database, the read is considered “unmapped”. Increasing the Minimum seed length will give higher precision in the results, while lowering it will give a higher recall rate but with more possible false positives.

The option "Auto-detect paired distances" will generate an estimate of the paired distance range in an additional section of the report outputted by the tool. This range is computed with a sample of the input reads (100,000 reads or all reads if there are fewer in the input file) by mapping with a very wide paired distance parameter (0 to 50,000). The histogram of paired-end distances in the given sample set is computed and from this histogram the bottom 0.5% percentile and top 0.5% percentile are used for the estimated range. If fewer than 10,000 reads are mapped as pairs, the range is calculated using the standard deviation. The estimated distances are used in the qualification phase, i.e. both reads of a read pair are required to map to the same reference sequence within the estimated distance to be assigned to the corresponding taxon.

When enabled, the option "Adjust read count abundances" will weight the reads assigned to a taxon by the number of nucleotides mapped to that taxon, otherwise raw read counts are reported. Specifically, the adjusted abundance is calculated as

\[
\text{Adjusted abundance} = \frac{\text{abundance in nucleotides}}{\text{average mapped read length}},
\]

where the denominator is the average length of all mapped reads. For data sets where all reads have similar length, the adjusted abundance is very similar to the read count assigned to a taxon. Note that occasionally, this weighting may lead to zero reads assigned to a qualified taxon if there are only few, shorter than average reads assigned to this taxon.

In the last dialog, choose from the different output options and Open or Save the results (figure 6.8).
The tool will generate by default an abundance table as well as a report with a list of the taxa and their abundances. You can choose to output additional files such as a sequence list of the reads matching the reference database and those matching the host, as well as the unclassified reads.

6.2.1 Taxonomic profiling abundance table

The Taxonomic profiling abundance table displays the names of the identified taxa, along with their full taxonomy, a coverage estimate, and the number of reads found in the sample that are associated with this taxon. Note that the number of reads in the abundance table is not always equal to the number of reads mapping to the database as the number of reads given in the abundance table may be length adjusted, as explained above, and rounding errors may lead to small discrepancies. Furthermore, some reads may map across different Kingdoms in the reference database. Such reads are listed as mapping to the reference database, but they will not appear in an abundance table as no meaningful taxon can be assigned. The table can be visualized using the Stacked bar charts and Stacked area charts function, as well as the Sunburst charts.

- **Table view**

The table displays the following columns:

- **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 taxa 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.

- **Taxonomy**: the taxonomy of the taxon, as specified by the reference database.

- **Assembly ID**: the id of the assembly (typically genbank assembly accession numbers), as specified by the reference database.
– **Combined Abundance**: total number of reads for the taxa across all samples. This number may be length adjusted as explained above.

– **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** (for example LC1 in the table above): number of reads for each sample (calculated during the quantification phase, see section 6.2)

– **Coverage** (name of the sample): coverage estimate for the 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 of a taxon in a specific sample and the amount of coverage in the sample). You can also combine absolute counts and relative abundances by taxonomic levels by selecting the appropriate taxonomic level in the **Aggregate feature** drop-down menu. Incomplete taxonomies at a given level of Aggregation can be hidden using the "Hide incomplete taxonomy" check box.

Finally, if you have previously annotated your table with Metadata (see section 7.8), 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 tables 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 table, then the merge should be redone excluding the samples with zero control read counts.
Extract Reads from Selection will extract reads uniquely associated to specific rows in the table. In order to do this, you must have opted to output and save the reads matching the reference database when running the Taxonomic Profiling tool.

- **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. In the Stacked Bar (figure 6.10) and Stacked Area Charts (figure 6.11), 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 6.10: Stacked bar chart.](image)

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 6.12).

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.

### 6.2.2 Taxonomic Profiling report

The Taxonomic Profiling report (figure 6.13) is divided into several sections containing general information about the taxonomic profiling run and databases used.
1 Taxonomic summary

<table>
<thead>
<tr>
<th>Taxonomic level</th>
<th>Number of classifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kingdom</td>
<td>1</td>
</tr>
<tr>
<td>Phylum</td>
<td>5</td>
</tr>
<tr>
<td>Class</td>
<td>6</td>
</tr>
<tr>
<td>Order</td>
<td>10</td>
</tr>
<tr>
<td>Family</td>
<td>12</td>
</tr>
<tr>
<td>Genus</td>
<td>14</td>
</tr>
<tr>
<td>Species</td>
<td>15</td>
</tr>
</tbody>
</table>

2 Classification of reads

<table>
<thead>
<tr>
<th></th>
<th>Reads</th>
<th>Uniquely matching reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference database matches</td>
<td>3,306</td>
<td>3,224</td>
</tr>
<tr>
<td>Host matches</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Unclassified reads</td>
<td>12,466</td>
<td>N/A</td>
</tr>
<tr>
<td>Total</td>
<td>15,772</td>
<td>3,224</td>
</tr>
<tr>
<td>Reassigned reads</td>
<td>3</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3 Reference database summary

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sequences</td>
<td>62,647</td>
</tr>
<tr>
<td>Number of basepairs</td>
<td>193,813,930</td>
</tr>
</tbody>
</table>

4 Host database summary

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sequences</td>
<td>25</td>
</tr>
<tr>
<td>Number of basepairs</td>
<td>3,086,286,401</td>
</tr>
</tbody>
</table>

5 Auto-detected paired distances

<table>
<thead>
<tr>
<th>Sequence list</th>
<th>Paired distance estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample A (paired)</td>
<td>398 - 502</td>
</tr>
</tbody>
</table>

Figure 6.13: An example of taxonomic profiling report.

- **Taxonomic summary** includes information about which taxonomic levels were found in the data sample and how many different taxa were found on each level.

- **Classification of reads** summarizes the number of reads in the sample and the number of uniquely mapping reads that match either the reference database, the host genome or were unclassified.

- **Reference database summary** gives an overview of the reference database.

- **Host database summary** (optional section) gives an overview of the host reference database.

- **Auto-detect paired distances** (optional section) gives the paired distance range.
6.2.3 Sorted Reads Files

The taxonomic profiling tool can sort reads based on their assigned position in the databases (figure 6.14): database matches, host matches, and unclassified reads.

Figure 6.14: Database matches table.

The resulting files are sequence lists that can be visualized as table. The host and database matches tables include a value called Mapping score, and calculated for each read at their assigned positions (see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Mapping_parameters.html). In addition, the database matches table provide the Assigned taxonomy for the read, taxonomy that was assigned to the read by mapping it to the database.

If the reads matching the reference database have been saved, it is possible to select a set of rows in the abundance table and to click the "Extract Reads from Selection" button which will extract reads uniquely associated to the specific rows selected in the abundance table.

6.3 Identify Viral Integration Sites

This 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 start the tool, go to:

**Toolbox | Microbial Genomics Module | Metagenomics | Taxonomic Analysis | Identify Viral Integration Sites**

The Identify Viral Integration Sites tool takes one or several single or paired-end read files as input.

After selecting the input reads, it is possible to specify the host and virus references, and adjust the detection parameters, see (figure 6.15).

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.

---

![Figure 6.15: Select references and adjust detection options.](image)

The final step is to specify the output objects, see (figure 6.16). 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.
6.3.1 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 6.17).

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.

It is possible to zoom in on the host genome, by using the mouse-wheel on a section of the host genome, see (figure 6.18). 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 6.19), 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.
6.3.2 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 6.19: The table view for the detected viral breakpoints.

Figure 6.20: The report for the Identify Viral Integration Sites tool.
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 7.2.

To run the Merge Abundance Tables tool:

Toolbox | Microbial Genomics Module | Metagenomics | Abundance Analysis | Merge Abundance Tables

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

In the 'Result handling' wizard step, select Create Report to generate a report with summary information on input and output abundance tables.

7.2 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 6.2.

Creating the required reference index

You create a reference index using Create Taxonomic Profiling Index, see section 16.4. 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:

Toolbox | 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 7.1).

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.

Select Create Report to generate a report with summary information on taxonomic assignment.
The 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 threshold.
    - **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.

### 7.3 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 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 tool, go to:

**Toolbox** | **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 7.2).

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 7.3.1).

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 7.2: Set up parameters for the Alpha Diversity tool.](image)

In the following dialog (figure 7.3), set up the rarefaction analysis parameters.

The rarefaction analyses are done differently depending on the type of abundance table used as input. For OTU and functional abundance tables, where abundances are counts, rarefaction is calculated by sub-sampling the abundances in the different samples at different depths. For whole metagenome taxonomic profiling abundance tables, where abundances are coverage...
estimates, sub-sampling is not possible, so diversity is estimated using a probabilistic model corresponding to our qualification criteria instead (see section 6.2).

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 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 7.4). 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 7.5). 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 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.
CHAPTER 7. ABUNDANCE ANALYSIS

Figure 7.4: An example of Alpha Diversity graph based on phylogenetic diversity.

Figure 7.5: 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 7.6). 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 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.

### 7.3.1 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 = \( D + \frac{f_2^2}{2f_2} \).
  
- **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.

### 7.4 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 7.4.1). 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.

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 tool, open Toolbox | Microbial Genomics Module | Metagenomics | Abundance Analysis | Beta Diversity.

Select an abundance table with more than one sample as input (i.e., an OTU table table, or a merged functional or profiling table) and set the parameters for the beta diversity analysis as shown in figure 7.7.

The output of the tool is a 3D PCoA plot (figure 7.8) that can also be seen as a table or a 2D Principal Coordinate Plot.

If you have problems viewing the 3D plot, please check your system matches the requirements for 3D viewers. See http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=System_requirements.html.

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 7.8) or by abundance values (figure 7.9).

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: Name, Taxonomy or, in the case of OTU abundance tables, Sequence. The drop down menu will then display the items that can be deselected (4) if you want...
CHAPTER 7. ABUNDANCE ANALYSIS

Figure 7.9: Beta diversity results seen as a 3D PCoA, with coloring done according to taxonomic abundance values.

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.

7.4.1 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:**
  \[ E = \sqrt{\sum_{i=1}^{n} (x_i^A - x_i^B)^2} \]

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: \( d(U) = \frac{\sum_{i=1}^{n} b_i |I(p_{A}^i > 0) - I(p_{B}^i > 0)|}{\sum_{i=1}^{n} b_i} \)

• Weighted UniFrac: \( d(W) = \frac{\sum_{i=1}^{n} b_i |p_{A}^i - p_{B}^i|}{\sum_{i=1}^{n} b_i (p_{A}^i + p_{B}^i)} \)

• Weighted UniFrac not normalized: \( d(w) = \sum_{i=1}^{n} b_i |p_{A}^i - p_{B}^i| \)

• D_0 UniFrac: The generalized UniFrac distance \( d(0) = \frac{\sum_{i=1}^{n} b_i |p_{A}^i - p_{B}^i|}{\sum_{i=1}^{n} b_i (p_{A}^i + p_{B}^i)} \)

• D_0.5 UniFrac: The generalized UniFrac distance \( d(0.5) = \frac{\sum_{i=1}^{n} b_i \sqrt{p_{A}^i + p_{B}^i} |p_{A}^i - p_{B}^i|}{\sum_{i=1}^{n} b_i (p_{A}^i + p_{B}^i)} \)

where \( n \) is the number of branches in the phylogenetic tree, \( b_i \) is the length of branch \( i \); \( p_{A}^i \) and \( p_{B}^i \) are the proportion of taxa descending from branch \( i \) for samples \( A \) and \( B \), respectively; and the indicator functions \( I(p_{A}^i > 0) \) and \( I(p_{B}^i > 0) \) 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].

7.5 PERMANOVA Analysis

PERMANOVA Analysis (PERmutational Multivariate ANalysis Of VAriance, also known as non-parametric 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:

Toolbox | Microbial Genomics Module | Metagenomics | Abundance Analysis | PERMANOVA Analysis

Choose an abundance table with more than one sample as input (i.e., an OTU table, or a merged functional or profiling 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 7.10), you can choose which Beta diversity measure to use (see section 7.4.1). 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).

The output of the analysis is a report which contains two tables for each beta diversity measure used:
Figure 7.10: Beta diversity and Phylogenetic diversity measures are included in the PERMANOVA analysis.

- 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.

### 7.6 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: [link](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Differential_Expression.html).

To run the tool:

**Toolbox** | **Microbial Genomics Module** | **Metagenomics** | **Abundance Analysis** | **Differential Abundance Analysis**

Select an abundance table with more than one sample as input (i.e., an OTU table table, or a merged functional or profiling table), and specify if you want to test differential abundance based on metadata defined groups of samples (figure 7.11). 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.
CHAPTER 7. ABUNDANCE ANALYSIS

Figure 7.11: Specify an abundance table and all other parameters.

The tool generates a Venn diagram for three pairwise comparisons at a time (figure 7.12). 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.

![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.](image)

Figure 7.12: 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.

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.

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.

### 7.7 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.


#### 7.7.1 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:

Toolbox | Microbial Genomics Module | Metagenomics | Abundance Analysis | Create Heat Map for Abundance Table

Select an abundance table with two or more samples as input (i.e., an OTU table, a merged abundance table, or a functional profiling table) and click Next.

Specify a distance measure and a cluster linkage (figure 7.13). The distance measure is used to specify 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. Learn more about how distances and clusters are calculated at http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Clustering_features_samples.html.

Figure 7.13: Select an abundance table.

After having selected the distance measure, set up the feature filtering options (figure 7.14).

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.
- 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.
CHAPTER 7. ABUNDANCE ANALYSIS

7.7.2 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 7.15).

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.

7.7.3 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.
- 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.
7.8 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 http://resources.qiagenbioinformatics.com/manuals/clcgeneomicsworkbench/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 tool, go to:

Toolbox | 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 7.16).

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 rows in the table with samples present in the abundance table, so if 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 IV

Typing and Epidemiology
Chapter 8

Introduction to Typing and Epidemiology

Next generation sequencing (NGS) data from whole pathogen genomes is frequently used for enhanced surveillance and outbreak detection of common pathogens. CLC Microbial Genomics Module introduces functionality for molecular typing and epidemiological analysis of bacterial isolates. The module enables the user to perform a range of analyses, or to take advantage of template workflows for routine surveillance or outbreak analysis of a specific pathogen (figure 8.1).

![Figure 8.1: Typing of cultured microbial samples shown as process diagram.](image)

The typing and epidemiology features include streamlined tools for NGS-based Multilocus Sequence Typing (MLST) and resistance typing, as well as fast detection of genus and species. It also includes tools for phylogenetic tree reconstruction based on single nucleotide polymorphisms (SNPs) or inference of K-mer trees from NGS reads or genomes. A new table format, acting as a database, collects typing results and associates these with metadata such as sample information, geographic origin, treatment outcome, etc. Results generated using NGS-MLST and resistance typing can hereby be associated with the original sample metadata. Users can filter
on analysis results and metadata and then select relevant subsets of samples for downstream analysis. Results and metadata available during tree generation can also be used to explore and decorate this epidemiologically relevant information on the phylogenetic tree.

The Typing and Epidemiology tools are described in the following chapters.

Template workflows for typing and epidemiology analysis are available at:

Toolbox | Template Workflows | Microbial Workflows | Typing and Epidemiology

For more information, see section 3.3.
Chapter 9

Find the best matching reference

9.1 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:

    Toolbox | Template Workflows | Microbial Workflows | Typing and Epidemiology

For more information, see section 3.3.

To identify best matching bacterial genome reference, go to:

    Toolbox | Microbial Genomics Module | Typing and Epidemiology | Find Best Matches using K-mer Spectra

Select the sequences you want to find a best match sequence for (figure 9.1).

![Figure 9.1](image)

Figure 9.1: To identify best matching reference, specification of read file is the first step.

Select then a reference database, and specify the following settings (figure 9.2).
• **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 9.3).

• **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.
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 contaminations 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 2.3. **Note** that Best match results are added automatically to Result Metadata Table when using the template Type a Known Species and/or Type among Multiple Species workflow(s) or their customized versions.

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 9.4).

Figure 9.4: Taxonomy of the best matching reference and quality information is shown in the Metadata Result Table.

### 9.1.1 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 9.5).

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 10.2. 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 9.5: 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.

9.2 Find Best References using Read Mapping

To start the tool, go to:

Toolbox | Microbial Genomics Module | Typing and Epidemiology | Find Best References using Read Mapping

The Find Best References using Read Mapping tool takes one or several single or paired-end read sequence lists as input. It then maps simultaneously against a set of references, and optionally a set of host references. The tool will output a list of the references identified. The host references can be used to filter away contaminated reads (e.g. a human reference). The tool can also output the read mappings for both the references and the host.

After specifying the input reads, it is possible to specify the references and adjust the detection parameters, see (figure 9.6). 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.

The following parameters are available:

- **References** Reference sequences as a sequence list or as a sequence track. Notice, that the tool ignores Assembly IDs: each sequence is considered to be its own reference.
Figure 9.6: Select references and filtering options.

- **Host references (for filtering)** Host reference genome as a sequence list or as a sequence track.

- **Minimum number of reads on a reference** At least this many reads must map to a reference before it is considered to be present.

- **Minimum relative abundance to most abundant reference** A reference must have at least this fraction of the reads of the most abundant reference.

- **Minimum fraction of reference covered** This specifies the minimum fraction of nucleotides in the reference, that must be covered by at least one read.

- **Minimum coverage** Filters on coverage, which is the number of nucleotides mapped to the reference divided by the reference length.

- **Maximum number of references to keep** Restricts the number of references to keep. The references are ranked according to the number of hits.

The next step is to specify the read mapping options (figure 9.7). These options are identical to the ones used by the **Map Reads to Reference** tool.

The final step is to select which output objects to produce.

### 9.2.1 The Find Best References using Read Mapping Report

The Find Best References using Read Mapping report contains an overall summary for the sample. Notice, that reports from multiple samples may be combined using the 'Combine Report' tool.
Figure 9.7: Select read mapping options.

Figure 9.8: The report for the Find Best References using Read Mapping.
Chapter 10

Phylogenetic trees using SNPs or k-mers

10.1 Create SNP Tree

The Create SNP Tree tool is inspired by Kaas et al., 2014. There are two ways to initiate creation of a SNP tree: from the Result Metadata Table (see section 2.2) or by running the tool from the Toolbox. Note that you can only create a SNP tree if you have identified a common reference for the different strains you are trying to type, and used it for read mapping and variant calling for each of these samples.

To create a SNP tree from the Toolbox:

Toolbox | Microbial Genomics Module | Typing and Epidemiology | Create SNP Tree

Select the relevant read mappings as shown in figure 10.1.

![Figure 10.1: Select read mappings to be included in the SNP tree analysis.](image)

Alternatively, select data recursively by right-clicking on the folder name and selecting Add folder contents (recursively) (figure 10.2), but remember to double check that files relevant for the downstream analysis are selected. An efficient alternative to these methods is to use the Quick filtering functionality from the Metadata Result Table to filter easily the data and initiate the SNP tree creation.

Select the variant tracks you want to use (figure 10.3). The variant tracks determine which positions to include in the SNP tree. The variant tracks need to have the same reference as the previously selected read mappings. Under normal circumstances you would select one variant
track for each read mapping given in the input step, but that is not a requirement.

The following Parameters may be specified before setting up the algorithm for the construction of the SNP tree (see figure 10.3):

- **SNV parameters**
  - **Variant tracks.** Select the variant tracks you want to use. The variant tracks determine which positions to include in the SNP tree.
  - **Include MNVs** or not, along with SNVs when building the SNP tree.
  - **Minimum coverage required in each sample** on a given position. The position is skipped if at least one sample has coverage below the specified threshold.
  - **Minimum coverage percentage of average required** on a given position. The position is skipped if at least one sample has coverage below this percentage of its own average coverage.
  - **Prune distance** specifies the 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 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.

Ignore positions with deletions. This will ignore SNP positions where at least one sample has a deletion at the given position.

- **Result metadata**
  - Result metadata Table. Specify location of the Result metadata table file.

- **Tree view**
  - Tree view settings. Select a standard tree setting (i.e., None, K-mer Tree Default or SNP Tree Default) or your own custom tree setting. Read more on Tree Settings in general: [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html).

The variant calls and read mapping results are used to determine the SNP positions used in the tree. Note that the variant tracks are only used to determine which positions to include in the SNP tree. Only the position and the type (SNP, and MNV if enabled) are used, whereas any information about reference and allele is ignored. The read mappings are then used to estimate the consensus sequence. Only a variant with relative frequency above 50% (haploid organisms) will be effectively considered.

The initial list of variants is reduced as the following: All but one variant from the initial variant lists that fall within the specified pruning distance (for example 10nt) are ignored. Positions that are not well or not covered in one or more read mappings ("Minimum coverage required in each sample" and "Minimum coverage of average required") are removed. In addition, all SNPs which do not have the minimal z-score are excluded.

Select the tree construction algorithm you want to use (figure 10.4).

![Figure 10.4: Choose the tree construction algorithm.](image)

When selecting the Neighbor Joining method to create the tree, branch lengths are based on the distance between samples. The distance between two samples is computed as "Number of input positions used where the consensus sequence is different" / "Number of input positions used". The distance is therefore a number between 0 (no difference found in the input positions used) and 1 (all input positions used were different). From the tree, one can compute the distance between two samples by summing up all branches connecting them.
When selecting the Maximum Likelihood method to construct the phylogenetic tree, the next step of the wizard will be to specify the details of the evolutionary model to be used and to specify whether bootstrapping should be performed (see figure 10.5).


The first step of the maximum likelihood algorithm is to produce an alignment of the concatenated SNPs which is to be given as a starting point for the maximum likelihood phylogeny. The Create SNP Tree tool can optionally output this SNP alignment (see figure 10.6).

It may be beneficial to use the SNP alignment as input for the Model Testing tool (described in detail here http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Model_Testing.html.

This tool can quantify which evolutionary model best suits the data best. You can then rerun the
Create SNP Tree tool with the settings set as suggested by the Model Testing tool.

The maximum likelihood tool starts by creating a starting tree using the Neighbor Joining method. Then it proceeds to calculate the most likely phylogenetic tree under the given evolutionary model.

### 10.1.1 SNP tree output report

The **Create SNP Tree** tool can optionally output a report that summarizes the consequence of the applied filtering settings, as well as a summary of ignored positions attributed to the different read mappings.

The first section of the report, "Filter Status", contains the actual counts found during analysis. The different categories are described below:

- **Number of different input positions** corresponds to the number of unique positions based on SNPs (and MNVs if selected).
- **Pruned** count reflects the number of positions pruned based on the specified filter value (see above).
- **Coverage filtered** count reflects the number of positions filtered based on the two specified filter values (see above).
- **Z-value filtered** count reflects the number of positions filtered based on the specified filter value (see above).
- **Number of input positions used** the final number of positions that passed all filtering steps and used for creating the SNP tree.

The second section consists of a graphical and a tabular representation of the number of positions that was filtered because of the individual read mappings. The tabular view has three columns:

- **Read mapping.** The name of the read mapping.
- **Filtered, total.** The total number of filtered positions contributed by this read mapping, i.e. how many positions in this read mapping where either the coverage was too low, or the z-score was too low. **Note** that several read mappings can contribute the same position.
- **Filtered, only by this.** The number of filtered positions contributed only by this read mapping, i.e. the number of positions that was filtered by this read mapping, and not filtered by any other read mappings.

In the example below, we are creating a tree based on three read mappings, mapping 1, mapping 2 and mapping 3.

- Mapping 1 has low coverage on positions a, b and c.
- Mapping 2 has low coverage on positions a, b, d and e.
- Mapping 3 has low coverage on positions a, b and e.
In this case

- Mapping 1 will have a "Filtered, total" count of 3 (positions a, b and c), and a "Filtered, only by this" count of 1 (position c).
- Mapping 2 will have a "Filtered, total" count of 4 (positions a, b, d and e) and a "Filtered, only by this" count of 1 (position d).
- Mapping 3 will have a "Filtered, total" count of 3 (positions a, b and e) and a "Filtered, only by this" count of 0.

We will get a table like this:

<table>
<thead>
<tr>
<th>Read mapping</th>
<th>Filtered, total</th>
<th>Filtered, only by this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping 1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Mapping 2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Mapping 3</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

An example of report content shown in figures 10.7, 10.8, 10.9 and 10.10.

1 Filter Status

<table>
<thead>
<tr>
<th>Description</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of different input positions</td>
<td>120201</td>
</tr>
<tr>
<td>Pruned</td>
<td>20847</td>
</tr>
<tr>
<td>Coverage filtered</td>
<td>18507</td>
</tr>
<tr>
<td>Z-value filtered</td>
<td>18191</td>
</tr>
<tr>
<td>Number of input positions used</td>
<td>62556</td>
</tr>
</tbody>
</table>

Figure 10.7: Report on Filter Status for the created SNP tree.

2 Ignored positions attributed to read mappings

Figure 10.8: Visualization of the filter effect across data used for generation of SNP tree.

In the applied example, ERR277235_1 shows significantly higher number of positions ignored than the other samples (see figures 10.8 and 10.10), and one might consider rerunning the tree without this sample in an attempt to get a higher resolution in the tree.
Figure 10.9: Listing filtering results of the first 10 rows of the 47 rows of the data sets. Each row show the number of ignored positions generally across all data set as well as number of ignored positions when filtering on the particular data set only.

Figure 10.10: Listing filtering results of the last 14 rows of the 47 rows of the data sets. Each row show the number of ignored positions generally across all data set as well as number of ignored positions when filtering on the particular data set only. The tabular information indicates that sample acc no ERR277235_1 shows significantly lower quality than the other samples.

10.1.2 Visualization of SNP Tree including metadata and analysis result metadata

The overall tree layout, node- and label settings (figure 10.11) is defined by selected Tree Settings (see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html).

The creation of a SNP tree allows for the associated Result Metadata Table to be specified. Thus it is possible to visualize the associated metadata as well as analysis result metadata on the phylogenetic tree. This makes it easier to find potential correlations across samples and verify, such as how well MLST typing and the tree topology fit together.

10.1.3 SNP Tree Variants editor viewer

It is possible to inspect the actual difference in nucleotide counts at a given internal node. This can be done by selecting the internal node of interest (figure 10.12) and then switching from the Show Tree element view to the Show SNP Tree Variants element view in the lower left corner of the visualization area.

Once in the SNP Tree Variant element view, the table lists the actual SNP input positions used for creation of the SNP tree. The number of rows (in figure 10.13 this is 62,556) corresponds to the number of input positions stated in the filter status table of the SNP Tree report (see figure 10.7).
10.1.4 SNP Matrix

The Create SNP Tree tool can optionally output a matrix containing the pairwise number of SNP differences between all pairs of samples included in the analysis (see figure 10.14).

It is possible to easily get an overview of which samples are closely related by using the colored
Figure 10.13: Counts of differences at a given position in the branches of the selected internal node.

Figure 10.14: A SNP matrix.

**Comparison gradient** in the side panel. The minimum and maximum values of the Comparison gradient scale can be adjusted by drag-and-drop such that the colors of the matrix cells change accordingly. Clicking on the Comparison gradients color scale a drop down menu appears where it is possible to select a different color scheme. Finally, it is possible to 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. It is possible to export the SNP Matrix to any table format file (*.csv, *.xls, *.xlsx).

**Note:** it is not possible to import from a *.csv format to a SNP Matrix - rather export the SNP Matrix to a *.clc format and then import it.

### 10.2 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 2.2), or from the Toolbox.

To run the Create K-mer Tree from the toolbox:
Input files can be specified step-by-step like shown in figure 10.15 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.

![Create K-mer Tree](image)

**Figure 10.15:** Selection of individual reads and single sequences or sequence list to be included in the K-mer tree analysis.

Specify the following parameters (figure 10.16):

- **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](https://en.wikipedia.org/wiki/Jaccard_index) and [https://en.wikipedia.org/wiki/Alignment-free_sequence_analysis](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.

- **Tree view.** Select a standard tree setting (i.e., None, K-mer Tree Default or SNP Tree Default) or your own custom tree setting. Read more on creating your customized tree settings: [http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html).

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.
10.2.1 Visualization of K-mer Tree for identification of common reference

The k-mer tree below (figure 10.17) 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 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](http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Tree_Settings.html).
Figure 10.17: 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 11

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.

Whereas classic MLST analysis types isolates against a small set of gene fragments (typical 500bp fragments for a set of seven house-keeping genes), 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.

11.1 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.
- Download MLST Scheme. Downloads existing MLST schemes from PubMLST.
- Import MLST Scheme. Imports an MLST scheme from a set of fasta files, sequence type info and optional locus metadata.

After a scheme has been obtained the following tools can be used together with the schemes.

- Type With MLST Scheme.
- Add Typing Results to MLST Scheme.

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.

11.2 MLST Scheme Visualization and Management

A 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.

The heat map view shows an overview of the scheme (figure 11.1), 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.

Figure 11.1: The MLST Scheme Heat Map view.
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 11.2: The Allele table view.](image)

The allele table view (figure 11.2) 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 11.3) shows the sequence types in the scheme. It always contains the following columns:

- **ST**: the name of the sequence type
Figure 11.3: 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 11.4: The Create MLST Subscheme options.

The **Create Large Sub Scheme** has the same options (figure 11.4) 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.
11.3 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 11.5.

![Minimum Spanning Tree](https://wikimedia.org/Minimum_spanning_tree.svg)

Minimum spanning trees can be a bit counter-intuitive. Consider figure 11.6: 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.

11.3.1 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 11.7.
Figure 11.6: 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.

### 11.3.2 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 11.8).

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 11.9).

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
selected sequence types.

11.3.3 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 11.8: Minimum Spanning Tree lasso selection.

Figure 11.9: Minimum Spanning Tree with manual layout modifications.

at least one other node in the cluster (single-linkage clustering). See (figure 11.10) 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.

![Collapsing nodes in a Minimum Spanning Tree](image)

**Figure 11.10: Collapsing nodes in a Minimum Spanning Tree.**

### 11.3.4 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.
11.4 Type With MLST Scheme

The Type With MLST Scheme tool is used for assigning a sequence type to an isolate.

To run the Type With MLST Scheme tool choose:

Toolbox | 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.
In the next dialog step (figure 11.12), specify the scheme and the typing parameters. 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.

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.

The novel allele detection parameters

![Figure 11.13: Specifying novel allele detection parameters.](image)

The next step in the dialog determines how to handle novel alleles (figure 11.13): 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.

### 11.4.1 Type With MLST Scheme results

The **Type With MLST Scheme** tool outputs a report, summarizing the typing, and a MLST Typing Result element.

The report will contain overall information about the typing results and whether the typing was conclusive or not.

The typing information contains the following information:

- **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.

- **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.

The sample information and scheme information contains various statistics about the input and scheme.

### 11.4.2 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 the sequence types in the scheme. 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.
1 Typing result

<table>
<thead>
<tr>
<th>Typing</th>
<th>Conclusive</th>
<th>Sequence type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ST6</td>
</tr>
</tbody>
</table>

2 Typing information

<table>
<thead>
<tr>
<th>Sequence type</th>
<th>Average kmer fraction</th>
<th>Alleles identified</th>
<th>Alleles called</th>
<th>Allele count</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST6</td>
<td>0.9959798</td>
<td>1.622</td>
<td>1.555</td>
<td>1.851</td>
</tr>
<tr>
<td>ST9</td>
<td>0.9213338</td>
<td>1.500</td>
<td>1.531</td>
<td>1.851</td>
</tr>
</tbody>
</table>

3 MLST search summary

3.1 Sample information

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of kmers matching the scheme</td>
<td>67,478,538</td>
</tr>
<tr>
<td>Loci without hits</td>
<td>0</td>
</tr>
<tr>
<td>Estimated sample coverage</td>
<td>10</td>
</tr>
<tr>
<td>Alleles with all kmers found</td>
<td>1,749</td>
</tr>
<tr>
<td>Alleles with kmer fraction of at least 90.00%</td>
<td>175</td>
</tr>
<tr>
<td>Novel alleles identified</td>
<td>5</td>
</tr>
<tr>
<td>Problematic loci for novel allele detection</td>
<td>-</td>
</tr>
</tbody>
</table>

3.2 Scheme information

<table>
<thead>
<tr>
<th>Scheme name</th>
<th>Tutorial Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genetic code</td>
<td>11 Bacterial, Archaeal and Plant Plasid</td>
</tr>
<tr>
<td>Check codon positions</td>
<td>Yes</td>
</tr>
<tr>
<td>Sequence types in MLST scheme</td>
<td>9</td>
</tr>
<tr>
<td>Loci in MLST scheme</td>
<td>1,851</td>
</tr>
<tr>
<td>Alleles in MLST scheme</td>
<td>8,373</td>
</tr>
</tbody>
</table>

Figure 11.14: The Type With MLST Scheme report.

- **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 11.16) 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.

• **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 11.17: The novel allele view for a MLST Typing Result.](image)

The novel allele view (figure 11.17) 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.

### 11.5 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.

To run the **Add Typing Results to MLST Scheme** tool choose:

```
Toolbox | Microbial Genomics Module | Typing and Epidemiology | MLST Typing | Add Typing Results to MLST Scheme
```

After selecting the MLST Typing result to add, the next step is to set up the **Add typing result parameters** (figure 11.18):

- **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.
Figure 11.18: Add typing result parameters.

- **Outlier range factor**: the allele length outlier definition in terms of the interquartile range of the length distribution 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)
11.6 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.

To run the Identify MLST Scheme from Genomes tool choose:

Toolbox | 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 11.19).

![Figure 11.19: Select relevant genome sequence or sequence list.](image)

The next step is to select as many MLST schemes as necessary to identify the species present in the input sample (figure 11.20).

![Figure 11.20: Select relevant MLST scheme(s) to search among.](image)

The output of this tool is a MLST scheme that best matches the sequences analyzed. To add the obtained best match to a Result Metadata Table, see section 2.3. 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 2.3.
The tool will not produce an output if no scheme could be uniquely identified. 

*Note* that results are added automatically to the Result Metadata Table when using the template Type among Multiple Species workflow.
Part V

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 Databases 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 5.3.3.

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 http://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 Pangenes by Sequence and Bin Pangenes 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:

Toolbox | 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
• 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 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:

Toolbox | 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](image)

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.

- **Annotations from sequence list** when using **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.

- **Annotations from sequence list** when using 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:** Sequence lists 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 a 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](image)

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](image)

- **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 analysis, go to:

Toolbox | Microbial Genomics Module | Functional Analysis | Annotate with DIAMOND

The first wizard step (figure 12.5), specifies the reference and search parameters.

As can be seen above, metadata (such as GO terms and taxonomy information) is handled differently depending on the database source:
- **Protein sequence list**: Sequence lists 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 annotations**: Annotations are transferred together with any metadata qualifiers the annotations contain.

- **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.

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.

- **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.

- **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:

- **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:

```
Toolbox | 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 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 refseq_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.

![BLAST Best Hit annotations added to gene cds4 of h. pylori.](image)

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

The **Annotate CDS with Best DIAMOND Hit** tool will allow 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. 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.

The version of the DIAMOND binaries bundled with the tool is v0.9.26.127. For questions or comments about DIAMOND, see [https://github.com/bbuchfink/diamond](https://github.com/bbuchfink/diamond).

To start the tool, go to:

**Toolbox | Microbial Genomics Module | Functional Analysis | Annotate CDS with Best DIAMOND Hit**

Several parameters are available (figure 12.9):
• **Genetic code parameters.** The genetic code used for translating CDS to proteins.

• **DIAMOND parameters.**
  - **Maximum E-value.** Maximum expectation value (E-value) threshold for saving hits.
  - **DIAMOND Index.** Select the relevant indexes. Indexes are generated by 1/ downloading a database with the Download Protein Database tool (section 17.1) and 2/ building the index using the Create DIAMOND Index tool (see section 17.4).
  - **Standard search**, mainly designed for short reads alignments, i.e., for finding significant matches of >50 bits on 30-40aa fragments.
  - **Sensitive search**, for longer sequences.
  - **More sensitive search**, where results in DIAMOND come close to the quality of BLAST searches.

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:

Toolbox | 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 [http://geneontology.org/docs/download-ontology/](http://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:

![Figure 12.10: Pfam and GO annotations added to gene cds4 of h. pylori.](image)

- **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).
• **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:

**Toolbox | 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.](image)

The parameters that can be set are seen in figure 12.12:

• **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 [http://geneontology.org/docs/download-ontology/](http://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.

• **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)

![Figure 12.14: Functional profile abundance table.](image)

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.

- **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 function 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 7.8), 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 tables 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 table, 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. 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.

  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 (保存/恢复设置)), the settings can be saved.
and applied in other plots, allowing visual comparisons across analyses.

- **Zoomable Sunbursts** (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.](image)

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

**Toolbox | 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.

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.

The resulting functional abundance tables store the inferred number of reads corresponding to
Figure 12.18: Select an OTU abundance table.

Figure 12.19: Specify the functional terms for which a functional abundance table shall be inferred.

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 Pipihillin [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

**Toolbox | 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).

There are two general columns to describe the pathways that have been called.

- **Name**: the name of the pathway.
Chapter 12: Functional Analysis

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.

- **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
- Log₂ fold change
- Fold change
- P-value
- FDR p-value
- Bonferroni

 corresponding to the headers in the differential abundance table.

In the lower right corner of the side panel there is a property viewer which displays information about selected EC terms and metabolites.
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.
Chapter 13

Drug Resistance Analysis

Antimicrobial resistance (AMR) is an emerging threat to human and animal health worldwide. It is therefore of great importance to identify antimicrobial resistance genes from isolate or metagenomic sequencing data both for monitoring purposes and to gain actionable insight. To detect resistances from sequencing reads directly, the tools Find Resistance with ShortBRED and Find Resistance with PointFinder can be used for probing the presence of resistance gene families or resistance conferring mutations, respectively. In order to detect AMR genes from assembled sequencing reads, the tool Find Resistance with Nucleotide DB is available.

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 DB 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 Pointfinder 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:

Toolbox | 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 PointFinder 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
- **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.
13.2 Find Resistance with Nucleotide DB

Identification of antimicrobial resistance genes is important for understanding the underlying mechanisms and the epidemiology of antimicrobial resistance. The Find Resistance with Nucleotide DB tool may be used for resistance typing of pre-assembled, complete or partial genomes simple contig sequences assembled using the de novo assembly algorithm of CLC Genomics Workbench (see http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=De_novo_assembly.html).

Alternatively, use the Type a Known Species or Type among Multiple Species template workflows as described in chapter 3.

Find Resistance with Nucleotide DB is inspired by Zankari et al., 2017 and uses BLAST for identification of acquired antimicrobial resistance genes within whole-genome sequencing (WGS) data. The tool detects resistance conferring genes, whether they break down (e.g. beta-lactamases) or expel (e.g. efflux-pumps) antimicrobial compounds.

Nucleotide resistance databases for use with the tool can be downloaded using Download Resistance Database (section 18.1).

To perform resistance typing, go to:

Toolbox | Microbial Genomics Module ( ) | Drug Resistance Analysis ( ) | Find Resistance with Nucleotide DB ( )

Select the input genome or contigs (figure 13.3).

You can then specify the settings for the tool (figure 13.4).

- **Database** Select a nucleotide resistance database.
- **Minimum Identity %** is 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.
CHAPTER 13. DRUG RESISTANCE ANALYSIS

Figure 13.3: Pre-assembled and complete- or partial genomes simple contig sequences may be used as input for resistance typing.

Figure 13.4: Select database and settings for resistance typing.

- **Minimum Length %** reflect the percentage of the total resistance gene length that a sequence must overlap a resistance gene to count as a hit for that gene. Here represented as a percentage of the total resistance gene length.

- **Filter overlaps**: will perform 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 DB 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 2.3.

### 13.3 Find Resistance with ShortBRED

This tool allows you to detect and quantify the presence of antibiotic resistance (AR) marker genes by running DIAMOND with a database of peptide marker sequences which represent the
genes of interest. The Find Resistance with ShortBRED tool works similarly to the quantify step of ShortBRED, a public bioinformatics pipeline and resource. To learn more about the ShortBRED-Quantify tool [Kaminski et al., 2015], see http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004557.

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:

Toolbox | 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:

- **Reference marker database**: select the antimicrobial resistance marker database.
- **Genetic code**: select the genetic code to use when DIAMOND translates the nucleotide sequences to proteins.
- **E-value**: specify an expectation value to use as threshold for qualifying hits with DIAMOND.
- **Sensitivity**: select DIAMOND sensitivity setting.
- **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).

![Figure 13.5: References and search parameters.](image)
• **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.](image)

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 annotation depth of the gene name in the Antibiotic Resistance Ontology. The higher the number the more specific is the annotation.

- **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 features 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 7.8), 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 tables 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 table, then the merge should be redone excluding the samples with zero control read counts.
• **Stacked Bar Chart and Stacked Area Chart** (Figure 13.7 and Figure 13.8)

Choose which chart you want to see using the drop down menu in the upper right corner of the side panel. 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.

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 ( epilepsy)), the settings can be saved and applied in other plots, allowing visual comparisons across analyses.

- **Zoomable Sunbursts** ( epilepsy) 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).

![Sunburst view](image)

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.
Part VI

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 choose:

Toolbox | 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
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.

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:

- **Genetic code**: specifies 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**: determines the minimum sequence identity before grouping protein sequences.

- It is also possible to specify the sensitivity of the search (**Standard search**, **Sensitive search**, and **More sensitive search**) - increasing the sensitivity makes the search more thorough, but also much slower. The default for this parameter is **Sensitive search**.

- **Minimum gene length**: is used 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 the Download Resistance Database tool (e.g. QMI-AR for resistance or VFDB for virulence) and can be provided as input to the **Create MLST Scheme** tool at this step (figure 14.4).
14.2 Download MLST Scheme

To run the Download MLST Scheme tool choose:

Toolbox | Microbial Genomics Module ( ) | Databases ( ) | MLST Typing ( ) | Download MLST Scheme ( )

Use the Schemes to download selector (figure 14.6) to choose which schemes to download from PubMLST.

Most of the schemes offered for download by PubMLST 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 offered by PubMLST may only contain allele and locus definitions and no sequence types.

The Download metadata option makes it possible to download and extract metadata for all of the isolates for a given species in PubMLST. Note that this is a potentially very slow operation.

The clustering parameters determine how the heatmap should be clustered (figure 14.7). The heatmap cellvalues are the observed frequencies of a given allele compared to the other alleles in the same locus. The possible cluster linkages are:
Figure 14.6: The schemes available for download.

- **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.

- **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.

The following options are available when creating a minimum spanning tree (figure 14.8):

- **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 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 choose:

![Import MLST Scheme](image)

**Figure 14.9: The MLST Scheme import parameters.**

The **Allele folder (FASTA)** must contain a set of FASTA files 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
AGAGAAACGATACCTTCTATATGGCCCGTGATAATCAAGGCAAGCGTGTTGTCTTA
>pheS_2
AGAGAAACGATACCTTCTATATGGCCCGTGATAATCAAGGCAAGCGTGTTGTCTTA
```

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     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.

<table>
<thead>
<tr>
<th>Locus</th>
<th>Known name</th>
<th>Type name</th>
<th>Locus type</th>
</tr>
</thead>
<tbody>
<tr>
<td>locus5</td>
<td>FALSE</td>
<td>Unknown</td>
<td>ST1</td>
</tr>
<tr>
<td>fliR</td>
<td>TRUE</td>
<td>fli</td>
<td>ST2</td>
</tr>
<tr>
<td>flgL</td>
<td>TRUE</td>
<td>flg</td>
<td>ST3</td>
</tr>
<tr>
<td>hpaB</td>
<td>TRUE</td>
<td>hpa</td>
<td>ST4</td>
</tr>
</tbody>
</table>

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

OTU reference databases contain representative OTU sequences and their taxonomy. They are needed to perform reference-based OTU clustering. Three popular reference OTU databases, clustered at various similarity percentages, can be downloaded using the Download Amplicon-Based Reference Database tool:

- **SILVA**: SSU (16S/18S) and LSU (23S/28S) rRNA sequences for prokaryotic and eukaryotic taxonomic assignment (https://www.arb-silva.de/no_cache/download/archive/current/Exports/).

- **MiDAS**: 16S rRNA sequences for prokaryotic and eukaryotic taxonomic assignment of microbes in wastewater treatment and bioenergy systems (https://www.nature.com/articles/s41467-022-29438-7).

- **UNITE**: ITS sequences for fungal taxonomic assignment (https://unite.ut.ee/repository.php).

- **Greengenes2**: Full length 16S rRNA sequences from the backbone of the Greengenes2 phylogenetic tree. For prokaryotic taxonomic assignment [McDonald et al., 2022] (https://greengenes2.ucsd.edu/).

Some of the above databases are available at different clustering levels of sequence similarities.

To run the tool, go to

Toolbox | 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 reference databases as single sequence lists and/or taxonomic profiling indices with the necessary annotations required for the tools in the Typing and Epidemiology and Metagenomics sections of the Microbial Genomics Module.

To run the tool, go to:

Toolbox | 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](image)

You can choose between several databases

- QMI-PTDB Genus: QIAGEN Microbial Insights - Prokaryotic Taxonomy Database is a micro-
bial 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/](https://www.ncbi.nlm.nih.gov/refseq/)) and have been annotated with taxonomy from the Genome Taxonomy Database (GTDB; [https://gtdb.ecogenomic.org](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.

- **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/](https://www.ncbi.nlm.nih.gov/refseq/)) and have been annotated with taxonomy from the Genome Taxonomy Database (GTDB; [https://gtdb.ecogenomic.org](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.

- **Unified Human Gastrointestinal Genome (UHGG)**: A database for taxonomic and functional profiling of human gut samples curated and hosted by EMBL-EBI [Almeida et al., 2021]. The database includes metagenome assembled genomes from human gut samples.

- **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.

- **Clustered Reference Viral DataBase (C-RVDB)**: Clustered Reference Viral Database for virus detection. Viral entries are clustered at 98% by CD-HIT-EST.

- **ViraCura™ HPV REF**: A curated database of Human Papillomavirus reference strains. It contains unmodified viral reference genomes and associated record information from NCBI databases.

- **ViraCura™ HPV VAR**: A curated database of Human Papillomavirus variants of reference strains. It contains unmodified viral reference genomes and associated record information from NCBI databases.

- **ViraCura™ ANIMAL PV**: A curated database of Animal Papillomavirus. It contains unmodified viral reference genomes and associated record information from NCBI databases.
• MPXV: A curated database of Monkeypox virus reference strains. It contains unmodified viral reference genomes and associated record information from NCBI databases, as well as metadata and customized taxonomic nomenclature.

• MOCOVA: A curated database of Monkeypox outgroup reference strains (Molluscum contagiosum, Cowpox, Variola, and Vaccinia). It contains unmodified viral reference genomes and associated record information from NCBI databases, as well as metadata and customized taxonomic nomenclature.

You can then chose to download the database as an annotated sequence list and/or as a taxonomic profiling index.

The Curated Microbial Reference Databases are optimized for balance in the taxonomic representation across the taxonomy, i.e. the oversampling of some branches of the taxonomy is removed by using representative sequences. This has the consequence that some assemblies may not be particularly good assemblies, yet they are included as they constitute the best current representative of the given branch in the taxonomy. For this optimized database you can choose to download the 22g database, or one that is optimized for running the Taxonomic Profiling tool on a laptop computer with 16GB of main memory. The 16g version of the curated database contain a smaller number of assemblies, in order to be able to run on a system with 16GB of main memory.

Note: some of the databases offered are derived works, 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 using the freely available viewing mode of the CLC Genomics Workbench.

16.1.1 Extracting a subset of a database

After download, it is possible to select a subset of sequences and saving the reduced list in a new sequence list. This can reduce subsequent analysis runtime significantly.

For example, from a collection of bacterial genomes that include multiple representatives of each genus, you can extract a genus specific subset of sequences to a new list:

1. Open the downloaded bacterial genomes database.

2. Switch to tabular element mode.

3. Filter towards the desired genus (figure 16.2).

4. Select all remaining rows.

5. Click the Create New Sequence List button.

6. Save the subset reference list.

Another way to extract a subset of a database is to make use of the Split Sequence List tool. For more information, see https://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/current/index.php?manual=Split_Sequence_List.html.
The downloaded NCBI bacterial genomes database was filtered for Salmonella data. A subset of 44 out of 2,253 sequences matched this search criterion.

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. It is possible to download the sequence list with the necessary annotations required for the tools in the Typing and Epidemiology and Metagenomics sections of the Microbial Genomics Module.

To run the tool, go to:

Toolbox | Microbial Genomics Module | Databases | Taxonomic Analyses | Download Custom Microbial Reference Database

In the first window (figure 16.3) in the section Customize Database, select one of the three options:

- Include all: The database will contain both genomic and plasmid sequences.
- Include only plasmids: The database will contain only plasmid sequences.
- Exclude all plasmids: The database will not contain any plasmid sequences.

Here you can also choose to skip the database builder. If you choose to skip, all references matching the criteria stated in the next step will be downloaded. Selecting skip database builder will also enable the options to include all annotation tracks (annotation tracks are not needed for taxonomic profiling applications, but required when creating MLST schemes). An additional filter, Minimum contig length, may also be specified. Note that if you have not selected to skip the database builder, these options can be set later when downloading the database.

Click Next to customise the database you wish to generate (as in figure 16.4). You have the following two options:

- Build database from accessions or species TaxIDs: This enables the ID matching field.

Here you can provide a list of Species TaxIDs or GenBank or RefSeq assembly accessions.
Figure 16.3: Select the source of the database

(one per line) that must 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 GenBank assembly accession can usually be accessed using the search function or from the GenBank sequence on NCBI.

• Build database from taxonomic lineages: This enables the Linages prefixes field. Here you can 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 references in the Staphylococcaceae family. Entering "Bacteria;Pseudomonadota;Gammaproteobacteria;Enterobacteriales;" will include all family, genera and species references in the Enterobacteriales order. The NCBI taxonomy is updated weekly. When searching you should use the updated taxonomy.

The size and completeness of the database can be tuned with Inclusion criteria which select representative references based on the following options:

- 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 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.

It is also possible to leave the Linages prefixes and ID matching fields empty. In this case the database will not have any references pre-selected. Selection can then be done manually in the builder.

![Figure 16.4: Select options for building the database](image)

If **Skip Database Builder** was enabled, all references matching the specified criteria will be downloaded and the running of the tool is complete and you will not see the Database Builder as described in the next section.

### 16.2.1 Database Builder

If you left **Skip Database Builder** unchecked, assembly data is not downloaded right away. Instead, the Database Builder will open (figure 16.5).

The Database Builder table contains a range of metadata. This takes a bit to download, why opening the table may take a little while. The metadata is based on information from GenBank, [http://www.ncbi.nlm.nih.gov/genbank/](http://www.ncbi.nlm.nih.gov/genbank/).

With the DatabaseBuilder you can further customize the reference set to be downloaded:

1. Use the **Quick selection** button below the table to immediately download one of the following predefined subsets:

   - **Single scaffold complete genomes in RefSeq**
Figure 16.5: Search, filter and select assemblies to download

- Complete genomes in RefSeq
- All complete genomes

Each reference in the table will be labeled with one of the statuses listed above. In addition, some references are marked as representative genomes for a clade (repr) or as reference genomes (refr). We include references that are marked as Complete genome, Chromosome, representative genome and/or reference genome in above subsets.

2. Aggregate the table to a specified taxonomic group using the drop down menu in the "Data" palette of the side panel. Use the category "Name" to de-aggregate the table.

3. Use filter(s) and select row by dragging or pressing Ctrl+A to keep only the rows you are interested in, and click on the button **Include** to stage the selected references for downloading, which is indicated by a checkmark in the "Included" column.

4. Alternatively, press Ctrl+A then click on **Include** to include all rows first. Then set one or several filters, and use the button **Exclude** on the remaining rows. Clear the filter(s) by clicking on the red buttons next to each filter set. The rows not filtered away in the second step should still be checked.

5. Use the **Reset** button to reset the selection. This restores the builder to the initial state with only pre-selected reference included. If no references were pre-selected this will exclude all.

Once the table has all the desired references included, which is indicated by a checkmark, click **Download selection**. Close to the button, you can check how many references are selected and see an estimate of the total size of the selection.

The dialog shown in figure 16.6 allows you to include all annotation tracks (annotation tracks are not needed for taxonomic profiling applications, but required when creating MLST schemes). An additional filter, Minimum contig length, may also be specified (this option is not available when downloading the curated database). It also warns about the memory and disk requirements that will be needed to later run the Taxonomic Profiling tool with the database you are about to download.
Figure 16.6: *Filter options for download of the selected references*

Once a database has been downloaded, it is possible to extract a subset following the instructions described in 16.1.1.
16.3 Download Pathogen Reference Database


This will open the following wizard window (figure 16.7):

![Figure 16.7: Downloading assemblies and metadata for a selected pathogen from the NCBI Pathogen Detection Project.](image)

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

Once a database has been downloaded, it is possible to extract a subset following the instructions described in section 16.1.1.

### 16.4 Create Taxonomic Profiling Index

This tool will compute a taxonomic profiling index from a reference database. Indexes are then used as input to the Taxonomic Profiling tool. 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 (as indicated by the Download Custom Microbial Reference Database tool), i.e., approximatively the size of the uncompressed reference database; and twice this amount in hard disk space.

To run the tool, go to:

**Toolbox | Microbial Genomics Module | Databases | Taxonomic Analysis | Create Taxonomic Profiling Index**

In the first dialog, select one or several nucleotide databases, for example downloaded with the Download Custom Microbial Reference Database tool or the prokaryotic databases from the Download Curated Microbial Reference Database tool (figure 16.8). Note that sequences with identical Assembly ID annotations are treated as the same reference, see section 22.

The output is an index file and a report as seen in figure 16.9. The report list the number of sequence and basepairs that were indexed.
CHAPTER 16. DATABASES FOR TAXONOMIC ANALYSIS

Figure 16.8: Select reference sequences.

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:

Toolbox | 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 (http://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:

Toolbox | 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 it 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](image)

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.

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).
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:

```
Toolbox | 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 (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 Compound 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](image)

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:

Toolbox | 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:

**Toolbox | 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.](image)

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) 12.8 or to normalize OTU abundance tables by rRNA copy numbers using Normalize OTU Table by Copy Number (beta) 17.6.

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:
CHAPTER 17. DATABASES FOR FUNCTIONAL ANALYSIS

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)

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)

3. Files with functional term counts associated with each type of rRNA

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:

Toolbox | 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.

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 functional 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 DB, Find Resistance with PointFinder and Find Resistance with ShortBRED tools.

To run the tool, go to:

Toolbox | Microbial Genomics Module | Databases | Drug Resistance Analysis | Download Resistance Database

The following databases are available (figure 18.1):

Figure 18.1: Known variants in these common pathogens can be downloaded with the Download Resistance Database.
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.


- **CARD** This database contains peptide markers derived from the Comprehensive Antibiotic Resistance Database (CARD) (https://card.mcmaster.ca/).

- **ARG-ANNOT** This database contains a set of protein marker sequences based on the ARG-ANNOT database.

Nucleotide Databases These databases can be used with Find Resistance with Nucleotide DB (section 13.2). The databases contain nucleotide gene sequences.


- **VFDB** This database contains nucleotide sequences compiled from the Virulence Factor database, core dataset (http://www.mgc.ac.cn/VFs/download.htm).

- **CARD** This database contains nucleotide sequences compiled from the Comprehensive Antibiotic Resistance Database (CARD) (https://card.mcmaster.ca/).

- **ResFinder** The output is a sequence list with modifiable metadata relating the sequences to drug resistance information (https://bitbucket.org/genomicepidemiology/resfinder_db/src/master/). In the table view, users can review and customize the metadata associated with the sequences.

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/).
Integrated Databases

- **ARES Database** The ARES Database is an interactive interface to ARESdb: [https://www.sciencedirect.com/science/article/pii/S1672022919300920](https://www.sciencedirect.com/science/article/pii/S1672022919300920), a curated database of computationally inferred antimicrobial resistance genes and their respective predictive performances, both established and provided by Ares Genetics. With the ARES Database it is possible to produce sequence lists which are compatible with the Find Resistance with Nucleotide DB and Find Resistance with PointFinder tools, but also with other tools taking annotated sequence lists as input.

### 18.1.1 ARES Database

The ARES Database has three table views:

- a Nucleotide Marker table for gene markers. From this view, it is possible to extract a sequence list which may be used with the Find Resistance with Nucleotide DB tool.
- a Protein Marker table for gene markers.
- a Point Mutation Marker table for Single Nucleotide Polymorphism (SNP) markers. From this view, it is possible to extract a sequence list which may be used with the Find Resistance with PointFinder tool.

The database also comes with an overview report summarizing its content.

**The Nucleotide Marker table** The Nucleotide Marker Table lists AMR marker genes together with AMR related information annotated with CARD Antibiotic Resistance Ontology (ARO) accession numbers (figure 18.2).

![Figure 18.2: A Nucleotide Marker Table.](image)

The table contains the following columns:

- **Gene Name.** The name of the resistance gene or a generic unique name starting with "group_" if no relevant gene information is available.
- **Species.** The name of the species for which a given resistance gene was found. The presence of a gene marker in a species different from the one listed in this column may still indicate AMR, see performance table below.
- **Marker Id.** An identification number for the peptide sequence of the nucleotide sequence.
• **Phenotype.** A phenotype prediction for this gene marker.

• **Phenotype ARO.** The ARO accession number corresponding to the phenotype.

• **Compound Class.** A list of compound classes, corresponding to the compounds in Compound Name.

• **Compound Class ARO.** The ARO accession numbers corresponding to the Compound Class.

• **Compound Name.** A list of antibiotic compounds to which the gene marker confers resistance.

• **Compound ARO.** The ARO accession numbers corresponding to the Compound Name.

• **Start of sequence.** The beginning of the gene marker sequence.

• **Length.** The length of the gene marker in number of nucleotides.

Click on [Create a Nucleotide Sequence List](#) to create a sequence list with unique sequences that can be used as a database for the Find Resistance with Nucleotide DB tool. Note that the underlying sequences can be the same for two different species.

**The Protein Marker table**  The Protein Marker Table lists the protein products corresponding to the sequences of the Nucleotide Marker Table (figure 18.3). The Marker Id shows the connection between nucleotide and protein markers. Note that selections in the nucleotide and protein tables are synchronized when both views are open simultaneously.

![Figure 18.3: A Protein Marker Table.](#)

The table contains the following information:

• **Gene Name.** The name of the resistance gene or a generic unique name starting with "group_" if no relevant gene information is available. Note that this does not necessarily correspond to the "group_" name of the nucleotide sequence as the same Marker Id can occur for multiple nucleotide sequences.

• **Species.** A list of species for which a given resistance gene was found in the database. The presence of a gene marker in a species different from the ones listed in this column may still indicate AMR, see the Performance Table View section below.

• **Marker Id.** An identification number for the peptide sequence of the nucleotide sequence.

• **Phenotype.** A phenotype prediction for this gene marker.

• **Phenotype ARO.** The ARO accession number corresponding to the phenotype.
• **Compound Class.** A list of compound classes, corresponding to the compounds in Compound Name.

• **Compound Class ARO.** The ARO accession numbers corresponding to the Compound Class.

• **Compound Name.** A list of antibiotic compounds to which the gene marker confers resistance.

• **Compound ARO.** The ARO accession numbers corresponding to the Compound Name.

• **Start of sequence.** The beginning of the gene marker sequence.

• **Length.** The length of the gene marker in number of nucleotides.

From this table, it is possible to **Create a Protein Sequence List,** i.e., an annotated protein sequence list of gene markers for the proteins and species of interest. Furthermore, the button **Performance Indicators** opens another table with the performance statistics obtained from experimental data.

**The Point Mutation Marker table** The Point Mutation Marker table gives an overview of resistance conferring single-nucleotide polymorphisms (SNP's) and their performance data (figure 18.4).

![Figure 18.4: A Point Mutation Marker Table.](image)

The columns of the Point Mutation Marker Table are:

• **Gene Name.** The name of the resistance gene or a generic unique name starting with “group_” if no relevant gene information is available.

• **Species.** A list of species for which a given resistance gene was found in the database. The presence of a point mutation in a species different from the ones listed in this column may still indicate AMR, see the Performance Table View section below.

• **Reference.** The reference nucleotide at the position of the variant.

• **Variant position.** The location of the variant in the nucleotide sequence.

• **Variant.** The allele nucleotide at the position of the variant.

• **Amino Acid Change.** This describes the change on the protein level.

• **Phenotype.** The phenotype expected from the presence of this point mutation.

• **Phenotype ARO.** The ARO accession number corresponding to the phenotype.
- **Compound Class.** A list of compound classes, corresponding to the compounds in Compound Name.

- **Compound Class ARO.** The ARO accession numbers corresponding to the Compound Class.

- **Compound Name.** A list of antibiotic compounds to which the point mutation confers resistance.

- **Compound ARO.** The ARO accession numbers corresponding to the Compound Name.

From this table it is possible to create a sequence list which can be used as a database for the Find Resistance with PointFinder tool by pressing the Create Point Mutation Database button and each mutation is associated with performance data which can be accessed by clicking the button Performance Indicators.

**The Performance Indicators table**  Microbial susceptibility to various compounds may depend on a combination of multiple genetic factors and/or be specific to certain organism. As such, the presence of a single gene or variant marker is not a clear indication of whether an isolate is resistant to compound or class or class of compounds. The ARES Performance Indicators table (figure 18.5) provides a measure for how well a given gene marker is expected to perform for a given compound and species. The gene marker statistics is not only calculated for the species the marker was originally identified in, but for other species as well.

![Figure 18.5: A Performance Table.](image)

The ARES Performance Indicators tables were derived by identifying the gene markers in a large collection of isolates with known resistance profiles. Based on the presence of the gene markers in the isolates, confusion matrix statistics (https://en.wikipedia.org/wiki/Confusion_matrix) were calculated for a large number of marker, species, and compound combinations. In confusion matrix terms, resistant isolates are considered ‘positives’ and susceptible isolates are considered ‘negatives’. For example, a resistant isolate with a given gene marker is considered a ‘true positive’, and a resistant isolate without a given gene marker is a ‘false negative’.

- **Gene Name.** The name of the resistance marker evaluated in the light of antimicrobial susceptibility test data. Note, that these can be either complete gene markers or single-nucleotide polymorphisms (see the Type column).

- **Species.** The name of the species used in the antimicrobial susceptibility test.
- **Compound.** The name of the compound used in the antimicrobial susceptibility test.

- **Compound Class.** The compound class for the compound these statistics are calculated for.

- **Sensitivity.** The number of resistant isolates with the marker divided by the total number of resistant isolates in the confusion matrix.

- **Specificity.** The number of susceptible isolates without the marker divided by the total number of susceptible isolates in the confusion matrix.

- **Accuracy.** The (Number of resistant isolates with the marker + Number of susceptible isolates without the marker) divided by the total number of isolates.

- **ppv.** The positive predictive value is calculated as the number resistant isolates with the marker divided by the number of isolates with the marker.

- **npv.** The negative predictive value is calculated as the number of susceptible isolates without the marker divided by the number of isolates without the marker.

- **P-value.** A P-value calculated using Fisher’s exact test based on the confusion matrix numbers.

- **Type.** This indicates whether it is a gene coding sequence (CDS) or a single-nucleotide polymorphism (SNP) marker.

The table with performance data is synchronized with the protein table and a selection made in the former will trigger a selection in the latter.

**The ARES Database Overview Report**  The ARES database overview contains some summary statistics on the content of the ARES database (figure 18.6).

![Figure 18.6: An example of ARES Database Overview Report.](image)

- **Number of Isolates.** The number of isolates that have been experimentally tested for resistance against multiple compounds
- **Number of SNPs.** The number of single-nucleotide polymorphisms (corresponding to the entries in the Point Mutation Marker Table)

- **Number of Genes.** The number of gene markers (corresponding to the entries in the Protein Marker Table)

- **Number of Associations.** This is the number of entries in the ARES Performance tables described in the previous section

- **Number of Significant Associations (P-values < 0.05).** The number of performance table entries where the Fisher’s exact test P-value was less than 5%. The values are not corrected for multiple testing.
Part VII

Panel Support
Chapter 19

QIAseq 16S/ITS Demultiplexer

The Panel Support section offers a new 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, you can choose barcodes, i.e., the stretch of nucleotides used to demultiplex the sequences, from a predefined list (see the list in figure 19.1) or from a custom list.

![Figure 19.1: Set the parameters to demultiplex.](image)

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

<table>
<thead>
<tr>
<th>Region</th>
<th>Barcode</th>
<th>Number of reads</th>
<th>Percentage of reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1V2</td>
<td>AGCTTTGATTAAGCCTGTCGTCCTGCA</td>
<td>39,381</td>
<td>7%</td>
</tr>
<tr>
<td>Y2V3</td>
<td>GCCGGAGCGGTGAGTAA-</td>
<td>124,486</td>
<td>23%</td>
</tr>
<tr>
<td>Y3V4</td>
<td>CCTACTGGAAGCGAATGCTGTCCTGCA</td>
<td>92,991</td>
<td>17%</td>
</tr>
<tr>
<td>Y4V5</td>
<td>GTGYAGCGCGGGCGTAA-</td>
<td>61,912</td>
<td>11%</td>
</tr>
<tr>
<td>Y5V7</td>
<td>GATTAGATACCCCGTGAGTC-</td>
<td>119,151</td>
<td>22%</td>
</tr>
<tr>
<td>Y7V9</td>
<td>YACGGACGACAGAGAACCC-</td>
<td>78,053</td>
<td>14%</td>
</tr>
<tr>
<td>ITS</td>
<td>GATTAGATACCCCGTGAGTC-</td>
<td>7,028</td>
<td>1%</td>
</tr>
<tr>
<td>Ungrouped</td>
<td></td>
<td>31,479</td>
<td>6%</td>
</tr>
</tbody>
</table>

1.2 Reads per region: Barplot

Figure 19.3: An example of demultiplexing report.
Part VIII

Tools
Chapter 20

Tools

20.1 Extract Regions from Tracks

The Extract Regions from Tracks tool facilitates specific extraction of mapped reads covering the particular regions specified by the annotation track, e.g., in this case the regions specified by an MLST scheme. The generated Track list enables visualization of the mapped reads per MLST loci. As the tool focuses on visualization of the mapping against the locus, the coverage up- and downstream of the loci does not reflect the actual coverage.

The tool is initiated by:

Toolbox | Microbial Genomics Module | Tools | Extract Regions from Track

The input file to the tool is a Reads Track, Genome Track or and/or Annotation Track (figure 20.1) while the specification of the regions to be extracted is specified by the NGS MLST Annotation Track (figure 20.2) generated when initially running the Identify MLST tool.

Figure 20.1: The first input file to the Extract Regions from Tracks tool is a Reads Track, Genome Track or and/or Annotation Track.

The extracted mapped reads are visualized in the track list shown in figure 20.3. Through the navigation tool at top right, it is possible to switch among the various MLST regions defined for the analyzed species.
20.2 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, choose

Tools | Mask Low-Complexity Regions.

The following general options are available (figure 20.4):

- **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.

![Figure 20.4: The Mask Low-Complexity Regions options.](image)

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.

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:

### 20.2.1 Mask Low-Complexity Regions Report

An example of a Mask Low-Complexity Regions report can be seen in figure 20.5.

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total nucleotides/amino acids</td>
<td>523,332</td>
</tr>
<tr>
<td>Masked nucleotides/amino acids</td>
<td>0</td>
</tr>
<tr>
<td>Ambiguous nucleotides/amino acids</td>
<td>19,220</td>
</tr>
<tr>
<td>Total windows</td>
<td>523,309</td>
</tr>
<tr>
<td>Masked windows</td>
<td>0</td>
</tr>
<tr>
<td>Ignored windows</td>
<td>29,123</td>
</tr>
<tr>
<td>Masked regions</td>
<td>0</td>
</tr>
</tbody>
</table>

### 2 Mask Low-Complexity Regions complexity overview

![Complexity Graph](image)

A complexity of 99 corresponds to a random (uncompressible) sequence.
The table below lists some examples of different complexities from the input data.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Count</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>72</td>
<td>STAAAGAGAGAAAGAGAGAGAGAA</td>
</tr>
<tr>
<td>44</td>
<td>248</td>
<td>TGTAAAGAGAAAGAGAGAGAGA</td>
</tr>
<tr>
<td>47</td>
<td>286</td>
<td>CGGTAATAATAATAATAACATAG</td>
</tr>
<tr>
<td>50</td>
<td>536</td>
<td>TTTTTTTTTTTGTTTTTTTTAAATTA</td>
</tr>
</tbody>
</table>

Figure 20.5: The Mask Low-Complexity Regions report.
Part IX

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 Convert Abundance Table to Experiment

This tool has been deprecated and will be retired in a future version of the software. It has been moved to the Legacy Tools folder of the Toolbox, and its name has "(legacy)" appended to it. The functionality of this tool has become obsolete as you can now run statistical test on the abundance table as is using the tools Differential Abundance Analysis and Create Heat Map for Abundance Table.

To run the Convert Abundance Table to Experiment (legacy) tool, go to:

Toolbox | Legacy Tools | Convert Abundance Table to Experiment (legacy)

Choose an abundance table as input, and define which metadata group is to be considered as factor.

The tool outputs a table labeled (experiment). The first column is the name of the group used as factor in the analysis (its taxonomy and an ID number). For each feature (e.g. an OTU, an organism, a Pfam family or a GO term) there will be the following data:

- **Range**: The difference between the highest and the lowest expression value for the feature over all the samples.
- **IQR**: The inter-quartile range of the values for a feature across the samples, that is, the difference between the 75%-ile value and the 25%-ile value.
- **Difference**: The difference for a two-group experiment between the mean of the expression
values across the samples assigned to group 2 and the mean of the expression values across the samples assigned to group 1. Thus, if the mean expression level in group 2 is higher than that of group 1 the ‘Difference’ is positive, and if it is lower the ‘Difference’ is negative. For experiments with more than two groups the ‘Difference’ contains the difference between the maximum and minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...).

- **Fold change**: For a two-group experiment the ‘Fold Change’ tells you how many times bigger the mean expression value in group 2 is relative to that of group 1. If the mean expression value in group 2 is bigger than that in group 1 this value is the mean expression value in group 2 divided by that in group 1. If the mean expression value in group 2 is smaller than that in group 1 the fold change is the mean expression value in group 1 divided by that in group 2 with a negative sign. Thus, if the mean expression levels in group 1 and group 2 are 10 and 50 respectively, the fold change is 5, and if the mean expression levels in group 1 and group 2 are 50 and 10 respectively, the fold change is -5. For experiments with more than two groups, the ‘Fold Change’ column contains the ratio of the maximum of the mean expression values of the groups to the minimum of the mean expression values of the groups, multiplied by -1 if the group with the maximum mean expression value occurs before the group with the minimum mean expression value (with the ordering: group 1, group 2, ...).

- **Taxonomy**: The taxonomy of the feature, if specified. For OTUs, this is specified by the reference database when a database entry was used as Reference.

- **Expression values and Means for each site**: The expression values represents the number of reads belonging to the feature in a specific sample, and Means is the mean of these expression values.

You can create a sub-experiment by selecting only some of the features from your experiment table (the most abundant ones for example).
Part X

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 6.1.2
- Create K-mer Tree, section 10.2
- Create MLST Scheme, section 14.1
- Create Taxonomic Profiling Index, section 16.4
- Find Best Matches using K-mer Spectra, section 9.1
- 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 http://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


