Tutorial

Whole Metagenome Functional Analysis

March 1, 2022

Sample to Insight

QIAGEN Aarhus · Silkeborgvej 2 · Prismet · 8000 Aarhus C · Denmark
digitalinsights.qiagen.com · ts-bioinformatics@qiagen.com

Whole Metagenome Functional Analysis

This tutorial will take you through the different tools available in CLC Microbial Genomics Module available for *CLC Genomics Workbench* to perform a whole metagenome functional analysis pipeline.

Introduction The main goal of the tutorial is to demonstrate the assembly of metagenomes derived from two different groups of samples and the subsequent investigation of functional differences. It serves as a template for performing a comparative investigation into the functional composition and diversity of microbial communities. The tools provide a way of looking at different samples in aggregate views and to drill down into differentiating functional categories that result from the comparative analysis.

This tutorial is designed to work with metagenomic data (DNA-Seq) while the "Metatranscriptome analysis" tutorial works with metatranscriptomic data (RNA-Seq). Although these tutorials conceptually have a lot in common, there are differences in some key aspects of the analysis, e.g. gene finding is used in this tutorial while searching a reference database of protiens with DIAMOND is used in the in the metatranscriptome tutorial an approach that would also be applicable to this tutorial instead of gene finding.

Prerequisites For this tutorial, you will need *CLC Genomics Workbench* 12.0 or higher with CLC Microbial Genomics Module installed.

Overview In this tutorial we will go through a suite of useful components in pipelines for analyzing whole-metagenome NGS data from microbial communities.

- First, we will import "raw" NGS sequencing data into the workbench and prepare the samples for analysis.
- We will then assemble the reads using **De Novo Assemble Metagenome** into contigs.
- We will map the reads to the assembled contigs using **Map Reads to Contigs**.
- The tool **Bin Pangenomes by Sequence** will assign reads the bin of the contig they belong to.
- With **Find Prokaryotic Genes**, we will identify genes and coding DNA sequences (CDS) on the contigs.
- Subsequently, functional annotation of the CDS with Gene Ontology (GO) terms and Pfam domains will be performed with **Annotate CDS with Pfam Domains**.
- Based on the annotations, we will construct a Gene Ontology profile using the **Build Functional Profile** tool for measuring functional diversity.
- We will also create a multi-sample abundance table using **Merge Abundance Tables**.
- Finally, we will set up the data for additional statistical analyses and visualizations.

General tips

- Tools can be launched from the Workbench Toolbox, as described in this tutorial, or alternatively, click on the Launch button (🚀) in the toolbar and use the Quick Launch tool to find and launch tools.
- Within wizard windows you can use the **Reset** button to change settings to their default values.
- You can access the in-built manual by clicking on **Help** buttons or going to the "Help" menu and choosing "Plugin Help" | "CLC Microbial Genomics Module Help".

Downloading and importing the data For this tutorial we will make use of the mock dataset generated by Lindgreen et al., 2016. The dataset contains four samples, divided in two groups (A and B). Group A is enriched in bacteria that perform photosynthesis and nitrogen fixation, while group B is enriched in pathogenic bacteria. The goal of the analysis in this tutorial is to find those functional differences from whole-metagenome sequencing data.

For sake of speed, the dataset used for this tutorial is a small subset of reads that is related to the following functional categories:

- **Photosynthesis**, identified by the GO id 0015979, which defines it as *"the synthesis by organisms of organic chemical compounds, especially carbohydrates, from carbon dioxide (CO_2) using energy obtained from light rather than from the oxidation of chemical compounds."*
- **Nitrogen Fixation**, identified by the GO id 0009399, which defines it as *"the process in which nitrogen is taken from its relatively inert molecular form (N_2) in the atmosphere and converted into nitrogen compounds useful for other chemical processes, such as ammonia, nitrate and nitrogen dioxide."*
- **Pathogenesis**, identified by the GO ids 0009405 and 0009403, which are defined as *"the set of specific processes that generate the ability of an organism to cause disease in another"* and *"the chemical reactions and pathways resulting in the formation of toxin, a poisonous compound (typically a protein) that is produced by cells or organisms and that can cause disease when introduced into the body or tissues of an organism,"* respectively.

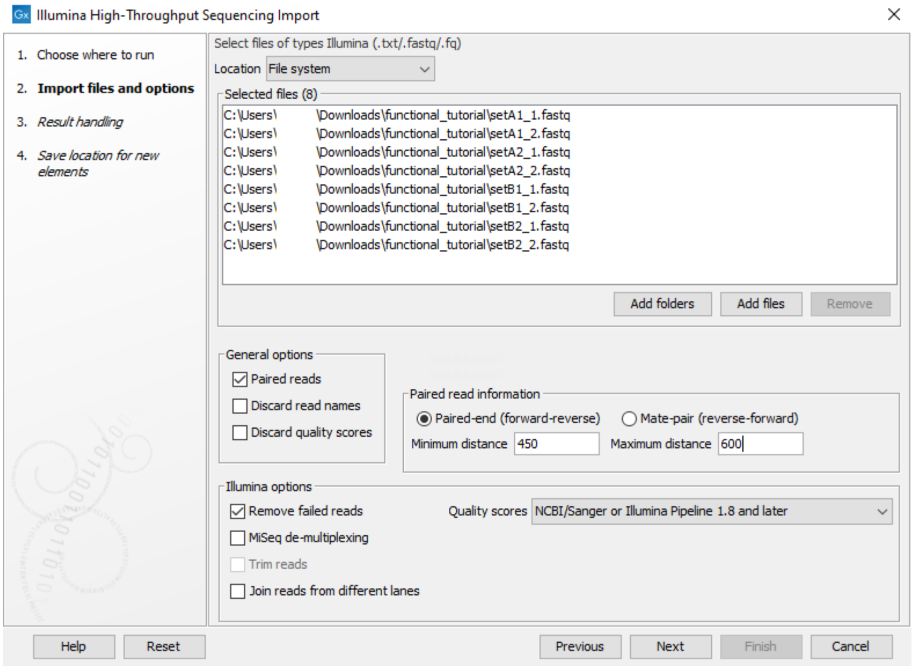
The tutorial data consists of the following files:

- **Sequence data**: 4 pairs of fastq files (two for group A and two for group B). The files contain simulated paired-end sequencing reads.
- **Metadata**: "Group metadata.xls". A spreadsheet containing metadata information about the samples and the group they belong to.
- **Pfam database**: "Pfam-A v29 - Tutorial subset.clc". A small subset of Pfam v29 [Bateman et al., 2004, Finn et al., 2016] containing only the Pfam domains relevant to this tutorial.
- **GO database**: "GO database - Tutorial subset.clc". A small subset of the GO (Gene Ontology) database [Ashburner et al., 2000, The Gene Ontology Consortium, 2015] and Pfam2GO mappings (which allow to infer GO terms from Pfam domains). This database contains only terms relevant for this tutorial.

Please note that this tutorial contains only a small subset of the Pfam and GO databases, which should *only* be used for this tutorial. For applying the functional analysis pipeline to real datasets, please download the *complete* databases: the complete versions of the Pfam and GO databases can be easily obtained using the **Download Pfam Database** and **Download GO Database** tools, respectively. If you are not sure where to find these tools in the toolbox, use the Launch button (🚀).

Now that the prerequisites have been described, it is time to start importing the input data.

1. Download the sample data from our website: http://resources.qiagenbioinformatics.com/testdata/functional_tutorial.zip and unzip it.
2. Start your CLC Workbench and create a folder for storing input data and results, named for example "Functional analysis".
3. Go to **Import** | **Illumina** to import the 8 sequence files (ending with "fastq") (figure 1).



Illumina High-Throughput Sequencing Import

1. Choose where to run
2. **Import files and options**
3. *Result handling*
4. *Save location for new elements*

Select files of types Illumina (.txt/.fastq/.fq)

Location File system

Selected files (8)

```
C:\Users\        \Downloads\functional_tutorial\setA1_1.fastq
C:\Users\        \Downloads\functional_tutorial\setA1_2.fastq
C:\Users\        \Downloads\functional_tutorial\setA2_1.fastq
C:\Users\        \Downloads\functional_tutorial\setA2_2.fastq
C:\Users\        \Downloads\functional_tutorial\setB1_1.fastq
C:\Users\        \Downloads\functional_tutorial\setB1_2.fastq
C:\Users\        \Downloads\functional_tutorial\setB2_1.fastq
C:\Users\        \Downloads\functional_tutorial\setB2_2.fastq
```

General options
- ☑ Paired reads
- □ Discard read names
- □ Discard quality scores

Paired read information
- ● Paired-end (forward-reverse) ○ Mate-pair (reverse-forward)
- Minimum distance 450 Maximum distance 600

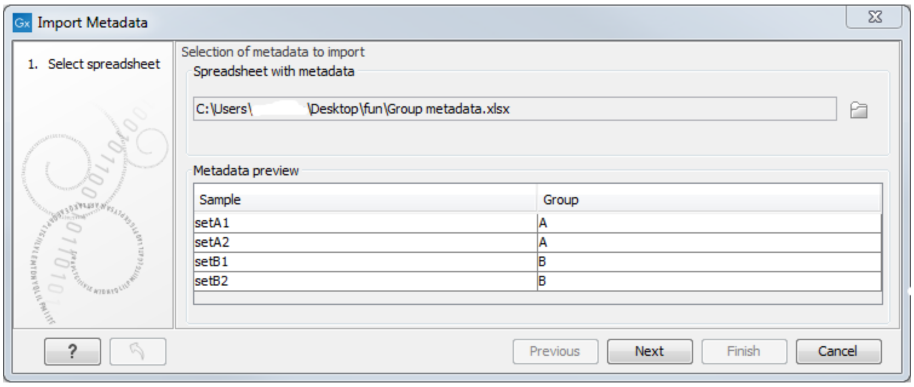
Illumina options
- ☑ Remove failed reads
- □ MiSeq de-multiplexing
- □ Trim reads
- □ Join reads from different lanes

Quality scores NCBI/Sanger or Illumina Pipeline 1.8 and later

Figure 1: *Import paired-end reads for the four samples.*

- The import type under Options is set to **Paired reads**.
- **Paired-end (forward-reverse)** is selected.
- **Discard read names**, **Discard quality scores** and **MiSeq de-multiplexing** are not checked.
- Set the minimum distance to 450 and the Maximum distance to 600.

4. Click **Next** and select the location where you want to store the imported sequences. You can check that you have now 4 files labeled as "paired".
5. Import the metadata by clicking **Import | Import Metadata** on top of the Navigation Area.



Selection of metadata to import

Spreadsheet with metadata

C:\Users\ \Desktop\fun\Group metadata.xlsx

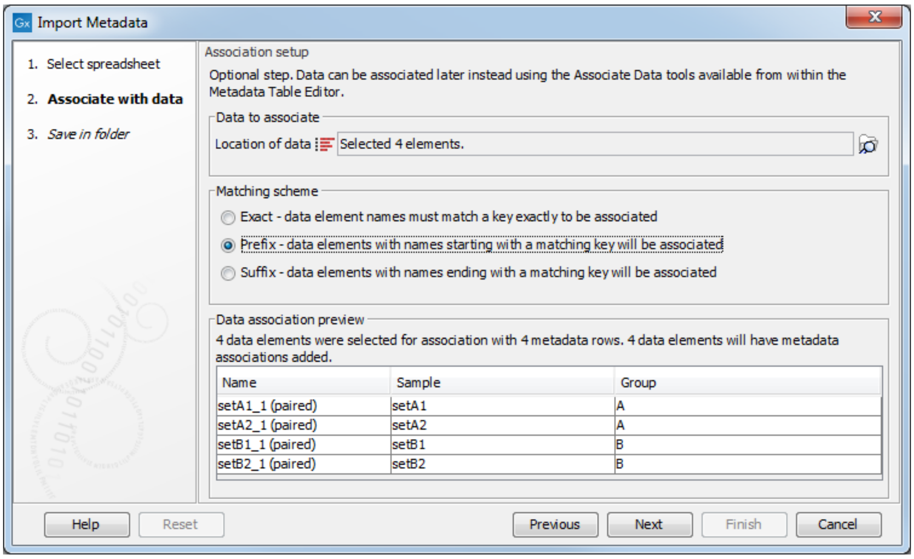
Metadata preview

Sample	Group
setA1	A
setA2	A
setB1	B
setB2	B

Figure 2: *Import the metadata.*

6. A wizard opens (figure 2). Select the spreadsheet Group metadata.xls in the first field. The content fills the Medatata preview table at the bottom of the dialog. Click **Next**.

7. Now select the four reads imported earlier. Check **Prefix** for the Data association preview table to fill in and thereby indicate that the association was successful (figure 3). Click **Next**.



Association setup

Optional step. Data can be associated later instead using the Associate Data tools available from within the Metadata Table Editor.

Data to associate

Location of data: Selected 4 elements.

Matching scheme

○ Exact - data element names must match a key exactly to be associated

● Prefix - data elements with names starting with a matching key will be associated

○ Suffix - data elements with names ending with a matching key will be associated

Data association preview

4 data elements were selected for association with 4 metadata rows. 4 data elements will have metadata associations added.

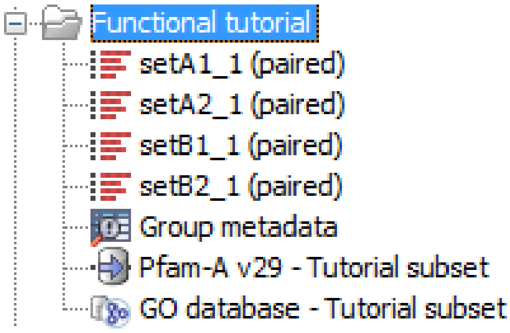
Name	Sample	Group
setA1_1 (paired)	setA1	A
setA2_1 (paired)	setA2	A
setB1_1 (paired)	setB1	B
setB2_1 (paired)	setB2	B

Figure 3: *The metadata and the reads are now associated.*

8. Save the metadata in the folder created earlier.

9. Import the databases ''*GO database - Tutorial subset.clc* ()'' and ''*Pfam-A v29 - Tutorial subset.clc* ()'' by using the Standard Import button on top of the Navigation Area.

All of the data needed to get started is now imported and you should have the objects depicted in figure 4. You are now ready to begin the analysis.



Functional tutorial
- setA1_1 (paired)
- setA2_1 (paired)
- setB1_1 (paired)
- setB2_1 (paired)
- Group metadata
- Pfam-A v29 - Tutorial subset
- GO database - Tutorial subset

Figure 4: *All files are now imported.*

Assembling, binning and annotating

In this section, we will assemble the reads into contigs and annotate them with functional information. Since we are working with downsampled data, we will pool the samples to obtain a better binning result for the metagenomics assembly. In general, pooling should only be done for similar samples, for example technical replicates.

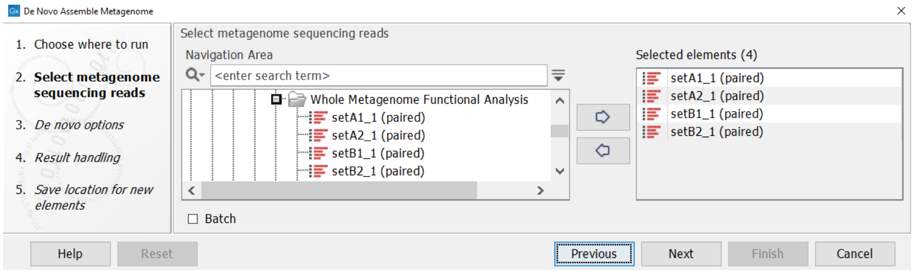
De novo assemble metagenomes

1. Create a new folder, for example "Assembly", to store the results. We are now ready to assemble the reads into contigs using the De Novo Assemble Metagenome tool:

   Metagenomics () | **De Novo Assemble Metagenome** ()

2. Select all four samples as input (figure 5). Do **not** select batch.

3. Click **Next**.



De Novo Assemble Metagenome

1. Choose where to run
2. **Select metagenome sequencing reads**
3. *De novo options*
4. *Result handling*
5. *Save location for new elements*

Select metagenome sequencing reads

Navigation Area

<enter search term>

Whole Metagenome Functional Analysis
- setA1_1 (paired)
- setA2_1 (paired)
- setB1_1 (paired)
- setB2_1 (paired)

Selected elements (4)
- setA1_1 (paired)
- setA2_1 (paired)
- setB1_1 (paired)
- setB2_1 (paired)

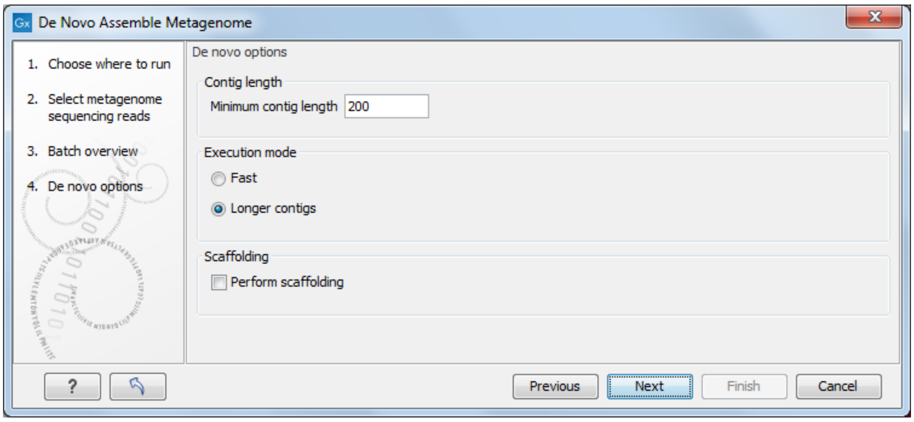
☐ Batch

Figure 5: *Reads for de novo metagenome assembly.*

4. In the "De novo options" dialog, make sure **Minimum contig length** is set to 200, choose the **Longer contigs** execution mode and make sure the **Perform scaffolding** checkbox is not checked (figure 6). Click **Next**.

5. Choose to save the results in the "Assembly" folder.

The tool will output one assembly and a report if this was enabled.



De Novo Assemble Metagenome

1. Choose where to run
2. Select metagenome sequencing reads
3. Batch overview
4. De novo options

De novo options

Contig length

Minimum contig length 200

Execution mode

○ Fast

● Longer contigs

Scaffolding

□ Perform scaffolding

Figure 6: *Select parameters for de novo metagenome assembly.*

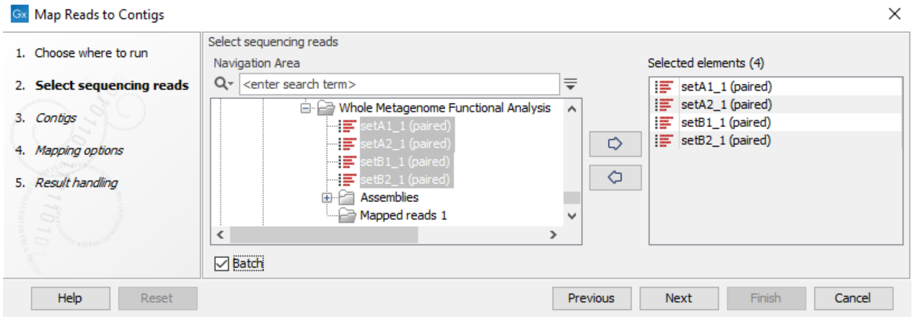
Map reads to contigs

1. Create a new folder, for example "Mapped reads 1", to store the results. We are now ready to map the reads to back to the contigs using the **Map Reads to Contigs** tool:

2. From the Toolbox, choose:

   De Novo Sequencing ( ) | **Map Reads to Contigs** ( )

3. Select the reads as input and select the **Batch** option (figure 18). Click **Next**



Map Reads to Contigs

1. Choose where to run
2. **Select sequencing reads**
3. *Contigs*
4. *Mapping options*
5. *Result handling*

Select sequencing reads

Navigation Area

<enter search term>

- Whole Metagenome Functional Analysis
  - setA1_1 (paired)
  - setA2_1 (paired)
  - setB1_1 (paired)
  - setB2_1 (paired)
  - Assemblies
  - Mapped reads 1

Selected elements (4)

- setA1_1 (paired)
- setA2_1 (paired)
- setB1_1 (paired)
- setB2_1 (paired)

☑ Batch

Figure 7: *Select the reads as input and select the Batch option.*

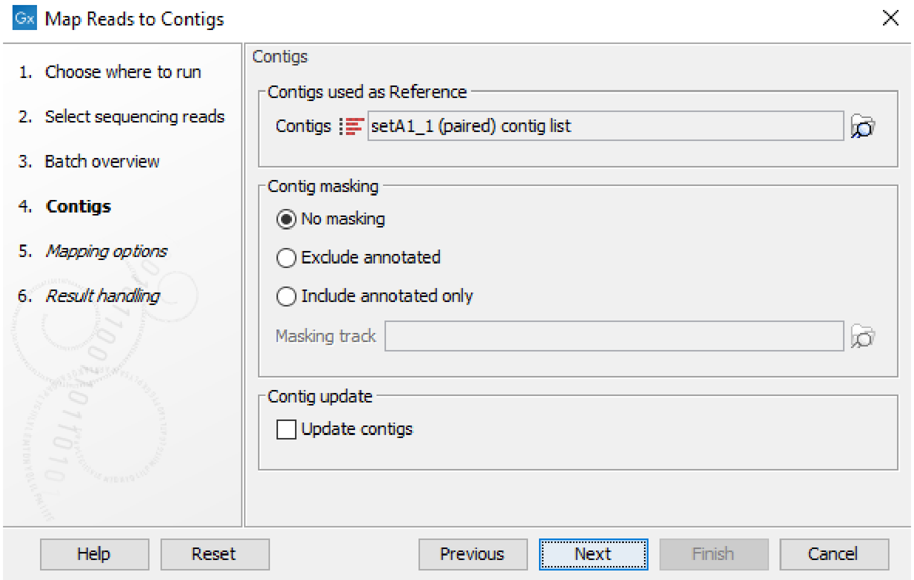
4. Check that everything looks as expected in "Batch overview". If you do not see this, you likely forgot to select **Batch**.

5. Click ( ) and locate the contigs from the "Assembly" folder (figure 8).

6. Leave the mapping options as default (figure 9) and click **Next**.

7. Select "Create stand-alone read mappings". Choose to save the result in the "Mapped reads 1" folder.

It is possible to run Bin Pangenome using contigs as input. However, running with a read mapping will generally produce more accurate results.



Contigs

Contigs used as Reference

Contigs: setA1_1 (paired) contig list

Contig masking

● No masking

○ Exclude annotated

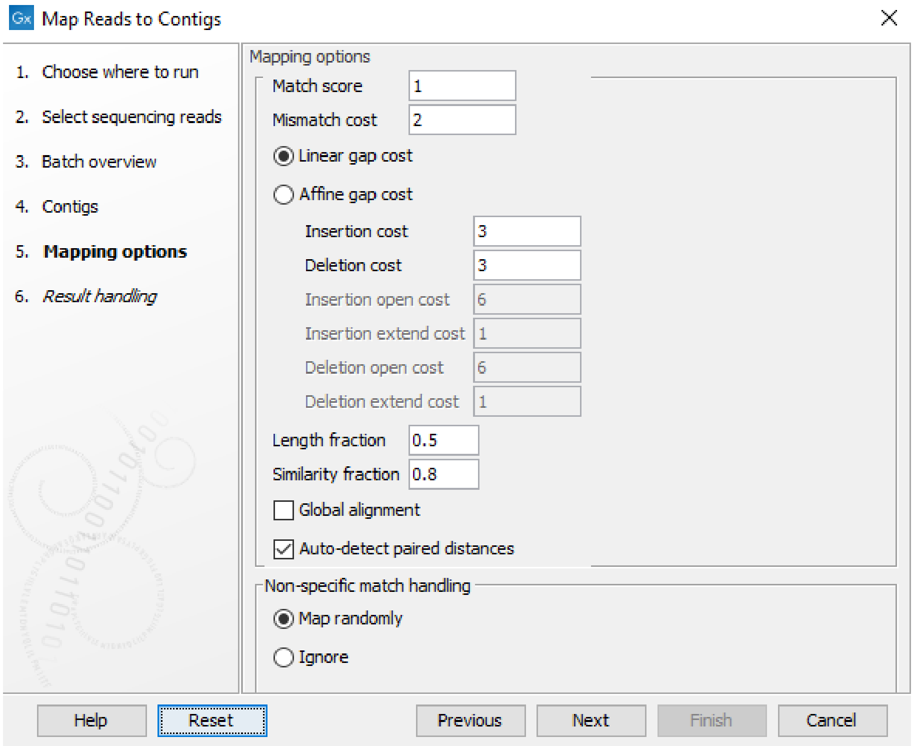
○ Include annotated only

Masking track

Contig update

□ Update contigs

Figure 8: *Locate the metagenome assembly.*



Mapping options

Match score: 1

Mismatch cost: 2

● Linear gap cost

○ Affine gap cost

Insertion cost: 3

Deletion cost: 3

Insertion open cost: 6

Insertion extend cost: 1

Deletion open cost: 6

Deletion extend cost: 1

Length fraction: 0.5

Similarity fraction: 0.8

□ Global alignment

☑ Auto-detect paired distances

Non-specific match handling

● Map randomly

○ Ignore

Figure 9: *Select the reads as input and select the Batch option.*

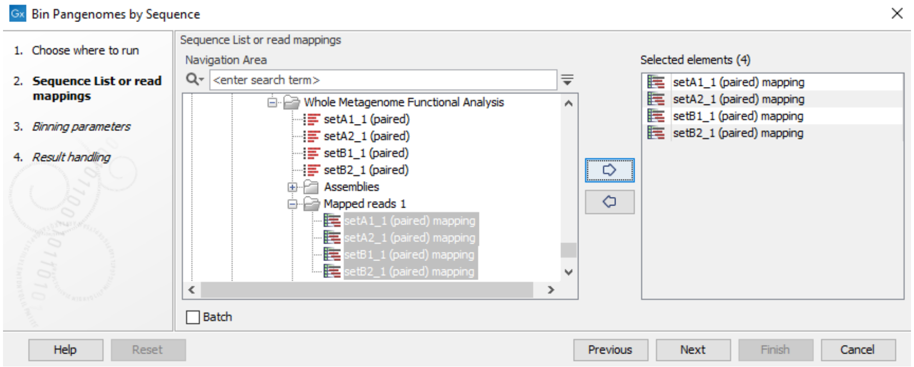
Bin Pangenomes by Sequence

We will now run the Bin Pangenomes by Sequence tool:

1. Go to:

   Metagenomics ( ) | **Taxonomic analysis** ( ) | **Bin Pangenomes by Sequence** ( )

2. Select the four read mappings generated in the previous step (figure 10).



Bin Pangenomes by Sequence

1. Choose where to run
2. **Sequence List or read mappings**
3. *Binning parameters*
4. *Result handling*

Sequence List or read mappings

Navigation Area

<enter search term>

- Whole Metagenome Functional Analysis
  - setA1_1 (paired)
  - setA2_1 (paired)
  - setB1_1 (paired)
  - setB2_1 (paired)
  - Assemblies
  - Mapped reads 1
    - setA1_1 (paired) mapping
    - setA2_1 (paired) mapping
    - setB1_1 (paired) mapping
    - setB2_1 (paired) mapping

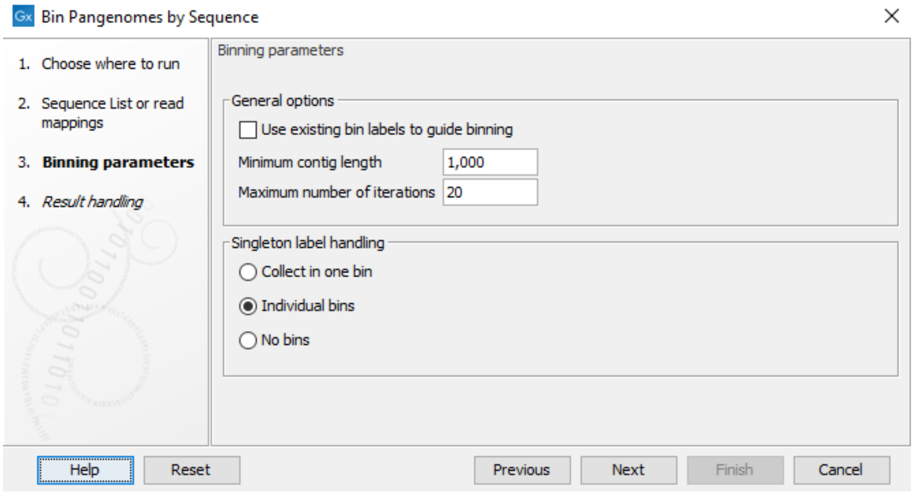
Selected elements (4)

- setA1_1 (paired) mapping
- setA2_1 (paired) mapping
- setB1_1 (paired) mapping
- setB2_1 (paired) mapping

□ Batch

Figure 10: *Right-click on the Mapped reads 1" folder and choose to "Add folder contents".*

3. Uncheck "Use existing bin labels to guide binning". Leave the other parameters as default figure 11.



Bin Pangenomes by Sequence

1. Choose where to run
2. Sequence List or read mappings
3. **Binning parameters**
4. *Result handling*

Binning parameters

General options

□ Use existing bin labels to guide binning

Minimum contig length: 1,000

Maximum number of iterations: 20

Singleton label handling

○ Collect in one bin

● Individual bins

○ No bins

Figure 11: *Binning parameters.*

4. Uncheck "Labelled reads" and **Save** the result into a separate folder, for example called "Bins by sequence".

Find Prokaryotic Genes

Once the reads have been binned, we need to functionally annotate the contigs. Before annotation with functional information, we need to identify coding regions in the contigs. Therefore we run the **Find Prokaryotic Genes** tool to identify genes and coding DNA sequences (CDS).

1. Go to:

   Functional analysis ( ) | **Find Prokaryotic Genes** ( )

2. Select the binned contigs generated in the previous step (figure 12).

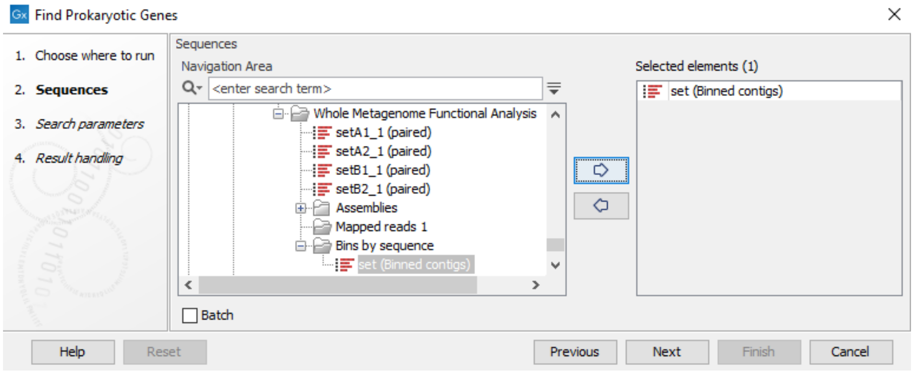
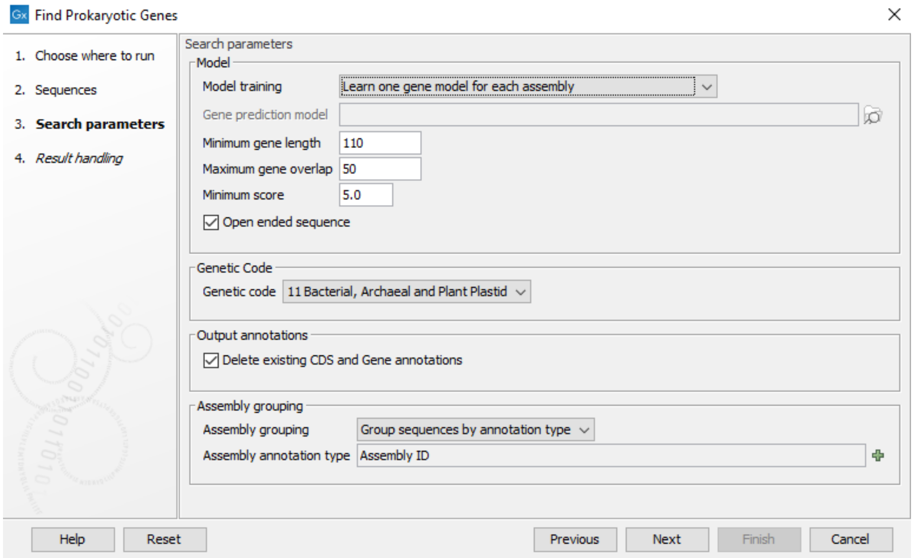


Figure 12: *Select the "Set (Binned contigs)" as input.*

3. In the next dialog, use the following parameters. **Model training** should be set to "Learn one gene model for each assembly" and the genetic code should be set to "11" (i.e. the genetic code used by bacteria, archaea and plant plastids) (figure 13).

   In order to annotate truncated genes at the beginning or end of the contigs, check the option "Open ended sequence". Finally, set **Assembly grouping** to "Group sequences by annotation type" and **Assembly annotation type** to "Assembly ID". Note that if working with Genomics Workbench 12, this option is hidden but grouping will treat each Assembly ID element as one assembly. Click **Next**.



Model training: Learn one gene model for each assembly
Gene prediction model:
Minimum gene length: 110
Maximum gene overlap: 50
Minimum score: 5.0
☑ Open ended sequence
Genetic code: 11 Bacterial, Archaeal and Plant Plastid
☑ Delete existing CDS and Gene annotations
Assembly grouping: Group sequences by annotation type
Assembly annotation type: Assembly ID

Figure 13: *Settings for the Find Prokaryotic Genes.*

4. Next, **Save** the results into a separate folder, for example called "Annotated Assembly".

Annotate CDS with Pfam domains and GO terms

In the next step, we will annotate the CDS with Pfam domains and GO terms by using the **Annotate CDS with Pfam Domains** tool.

1. Go to:

Metagenomics ( ) | **Functional Analysis** ( ) | **Annotate CDS with Pfam Domains** ( )

2. Select the annotated contig list generated in the previous step (figure 14).

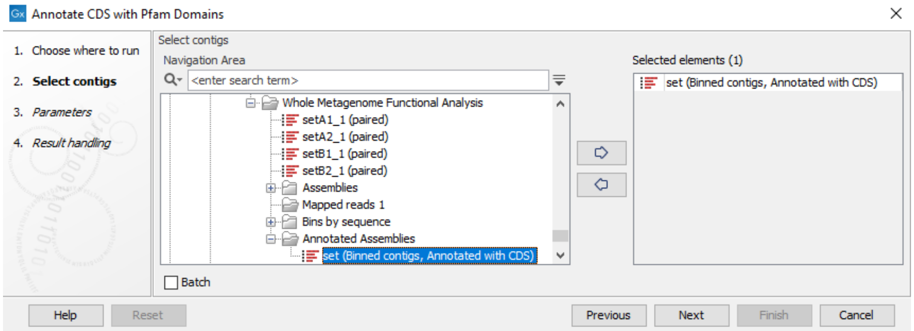
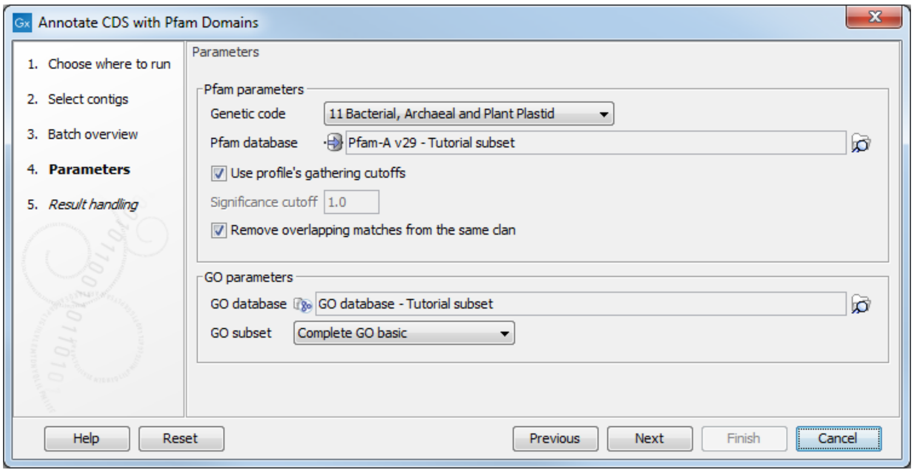


Figure 14: *Select the annotated contig list as input.*

3. Use the ''*Pfam-A v29 - Tutorial subset.clc* ( )'' as Pfam database and ''*GO database - Tutorial subset.clc* ( )'' as GO database, as shown in figure 15. Make sure the genetic code is set to "11 Bacterial, Archaeal and Plant Plastid" and that "Use profile's gathering cutoffs" and "Remove overlapping matches from the same clan" are checked. You can keep "Complete GO basic" as GO subset.



Genetic code: 11 Bacterial, Archaeal and Plant Plastid

Pfam database: Pfam-A v29 - Tutorial subset

☑ Use profile's gathering cutoffs

Significance cutoff: 1.0

☑ Remove overlapping matches from the same clan

GO database: GO database - Tutorial subset

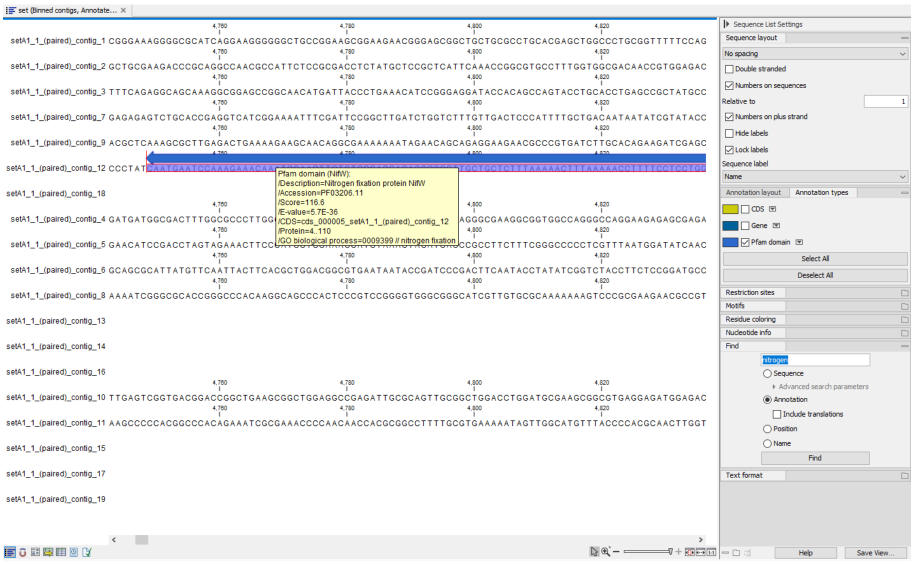
GO subset: Complete GO basic

Figure 15: *Parameters for Annotate CDS with Pfam Domains.*

4. Choose where you will save the output (the "Annotated assembly" folder for example) and click **Finish**.

Output from Annotate CDS with Pfam domains and GO terms

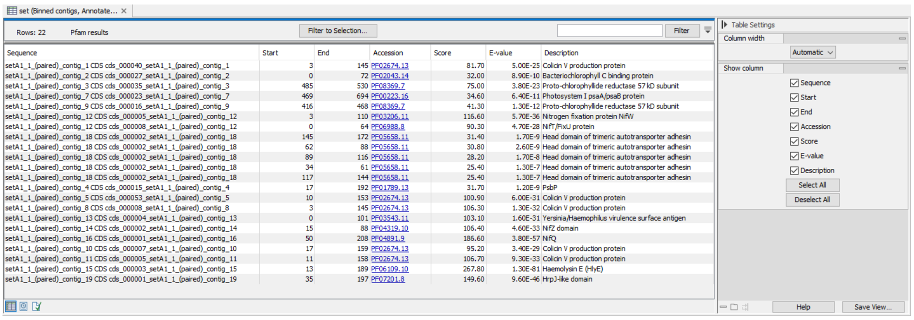
The tool will output contigs with Pfam annotations and keep existing annotations. It will also output a report. You can check that Pfam annotations have been added by opening ''*set (paired) contig list (Binned contigs, Annotated with CDS) (Pfam)* ( )''. To see Pfam annotations, open the **Annotation Type** tab on the right panel and click on **Pfam domain**. In the Find tab, type in "Pfam" in the top field, and select "Annotation" to find all Pfam annotations on each contig. If you hover over a Pfam annotation, you will be able to see the name of the Pfam domain, its description and the score of the match. When the Pfam domain can be matched to a GO term, a GO annotation will also be present, as shown in figure 16.



Pfam domain (NifW):
/Description=Nitrogen fixation protein NifW
/Accession=PF03206.11
/Score=116.6
/E-value=5.7E-36
/CDS=cds_000005_setA1_1_(paired)_contig_12
/Protein=4..110
/GO biological process=0009399 // nitrogen fixation

Figure 16: *Contig_12 contains a NifW protein domain, which is related to nitrogen fixation*

The tool also generates a table that recapitulates the found Pfam annotations (figure 17).



Rows: 22 Pfam results

Sequence	Start	End	Accession	Score	E-value	Description
setA1_1_(paired)_contig_1 CDS cds_000040_setA1_1_(paired)_contig_1	3	145	PF02674.13	81.70	5.00E-25	Colicin V production protein
setA1_1_(paired)_contig_2 CDS cds_000027_setA1_1_(paired)_contig_2	0	72	PF02043.14	32.00	8.90E-10	Bacteriochlorophyll C binding protein
setA1_1_(paired)_contig_3 CDS cds_000035_setA1_1_(paired)_contig_3	485	530	PF08369.7	75.00	3.80E-23	Proto-chlorophyllide reductase 57 kD subunit
setA1_1_(paired)_contig_7 CDS cds_000023_setA1_1_(paired)_contig_7	469	694	PF00223.16	34.60	6.40E-11	Photosystem I psaA/psaB protein
setA1_1_(paired)_contig_9 CDS cds_000016_setA1_1_(paired)_contig_9	416	468	PF08369.7	41.30	1.30E-12	Proto-chlorophyllide reductase 57 kD subunit
setA1_1_(paired)_contig_12 CDS cds_000005_setA1_1_(paired)_contig_12	3	110	PF03206.11	116.60	5.70E-36	Nitrogen fixation protein NifW
setA1_1_(paired)_contig_12 CDS cds_000008_setA1_1_(paired)_contig_12	0	64	PF06988.8	90.30	4.70E-28	NifT/FixU protein
setA1_1_(paired)_contig_18 CDS cds_000002_setA1_1_(paired)_contig_18	145	172	PF05658.11	31.40	1.70E-9	Head domain of trimeric autotransporter adhesin
setA1_1_(paired)_contig_18 CDS cds_000002_setA1_1_(paired)_contig_18	62	88	PF05658.11	30.80	2.60E-9	Head domain of trimeric autotransporter adhesin
setA1_1_(paired)_contig_18 CDS cds_000002_setA1_1_(paired)_contig_18	89	116	PF05658.11	28.20	1.70E-8	Head domain of trimeric autotransporter adhesin
setA1_1_(paired)_contig_18 CDS cds_000002_setA1_1_(paired)_contig_18	34	61	PF05658.11	25.40	1.30E-7	Head domain of trimeric autotransporter adhesin
setA1_1_(paired)_contig_18 CDS cds_000002_setA1_1_(paired)_contig_18	117	144	PF05658.11	25.40	1.30E-7	Head domain of trimeric autotransporter adhesin
setA1_1_(paired)_contig_4 CDS cds_000015_setA1_1_(paired)_contig_4	17	192	PF01789.13	31.70	1.20E-9	PsbP
setA1_1_(paired)_contig_5 CDS cds_000053_setA1_1_(paired)_contig_5	10	153	PF02674.13	100.90	6.00E-31	Colicin V production protein
setA1_1_(paired)_contig_8 CDS cds_000008_setA1_1_(paired)_contig_8	3	145	PF02674.13	106.30	1.30E-32	Colicin V production protein
setA1_1_(paired)_contig_13 CDS cds_000004_setA1_1_(paired)_contig_13	0	101	PF03543.11	103.10	1.60E-31	Yersinia/Haemophilus virulence surface antigen
setA1_1_(paired)_contig_14 CDS cds_000002_setA1_1_(paired)_contig_14	15	88	PF04319.10	106.40	4.60E-33	NifZ domain
setA1_1_(paired)_contig_16 CDS cds_000001_setA1_1_(paired)_contig_16	50	208	PF04891.9	186.60	3.80E-57	NifQ
setA1_1_(paired)_contig_10 CDS cds_000007_setA1_1_(paired)_contig_10	17	159	PF02674.13	95.20	3.40E-29	Colicin V production protein
setA1_1_(paired)_contig_11 CDS cds_000005_setA1_1_(paired)_contig_11	11	158	PF02674.13	106.70	9.30E-33	Colicin V production protein
setA1_1_(paired)_contig_15 CDS cds_000003_setA1_1_(paired)_contig_15	13	189	PF06109.10	267.80	1.30E-81	Haemolysin E (HlyE)
setA1_1_(paired)_contig_19 CDS cds_000001_setA1_1_(paired)_contig_19	35	197	PF07201.8	149.60	9.60E-46	HrpJ-like domain

Figure 17: *Table compiling the Pfam results.*

The metagenome assembly is now annotated.

Building functional profiles

We now want to re-map the original reads and estimate the abundance of functional categories in the samples.

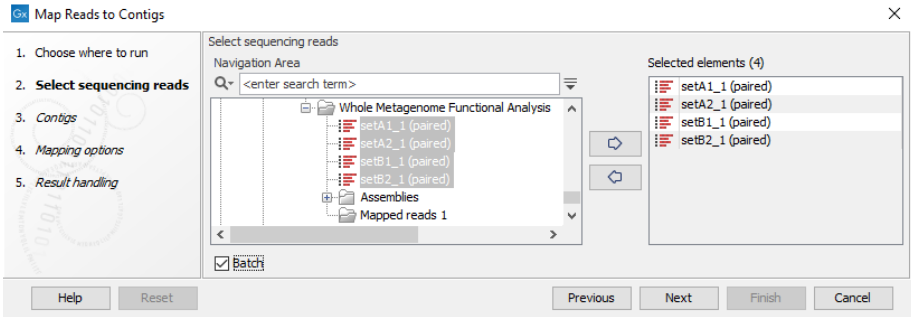
Map reads to estimate abundance

First, map the reads to the annotated metagenome assembly.

1. Create a folder called "Mapped reads 2" to store the results in.
2. From the Toolbox, choose:

   De Novo Sequencing ( ) | **Map Reads to Contigs** ( )

3. Select the reads as input and select the **Batch** option 18. Click **Next**.
4. Check that everything looks as expected in "Batch overview". If you do not see this, you likely forgot to select **Batch**.

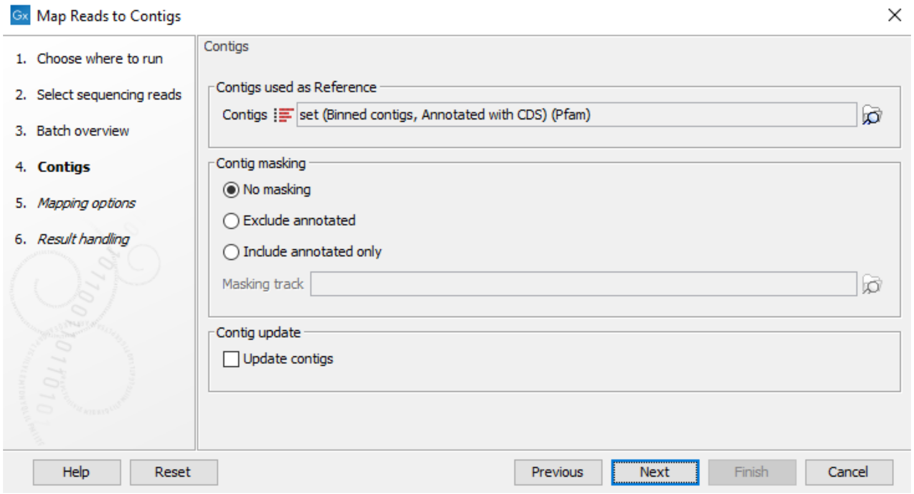


Selected elements (4)
setA1_1 (paired)
setA2_1 (paired)
setB1_1 (paired)
setB2_1 (paired)

☑ Batch

Figure 18: *Select the reads as input and select the Batch option.*

5. Select ''*set (Binned contigs, Annotated with CDS) (Pfam)* (≣)'' from the "Annotated Assembly" folder as reference as shown in figure 19). Keep "No masking" checked and click **Next**.



Contigs used as Reference
Contigs: set (Binned contigs, Annotated with CDS) (Pfam)

Contig masking
● No masking
○ Exclude annotated
○ Include annotated only
Masking track

Contig update
□ Update contigs

Figure 19: *Select the annotated reference.*

6. Keep the Mapping options at their default values.
7. In the result handling window, select the option to **Create stand-alone read mappings**. Stand-alone read mappings (≣) are preferable because they allow to run Build Functional Profile without having to specify a reference. Save the results in the new "Mapped reads 2" folder you created.

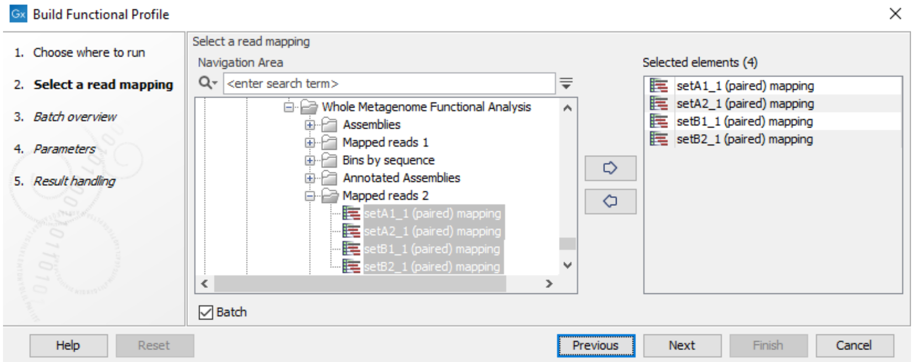
When the mapping is complete, the read mappings, for example ''*setA1_1 (paired) mapping* (≣)'', will be created.

Build GO functional profile

Next, we build the GO functional profile for each sample using the **Build Functional Profile** tool.

1. Create a folder called "Functional profiles" to store results.
2. From the Toolbox, choose:

   Functional Analysis () | **Build Functional Profile** ()
3. Enable the **Batch** option and select the four read mappings (figure 20).
4. Click **Next** twice to pass the Batch overview window.

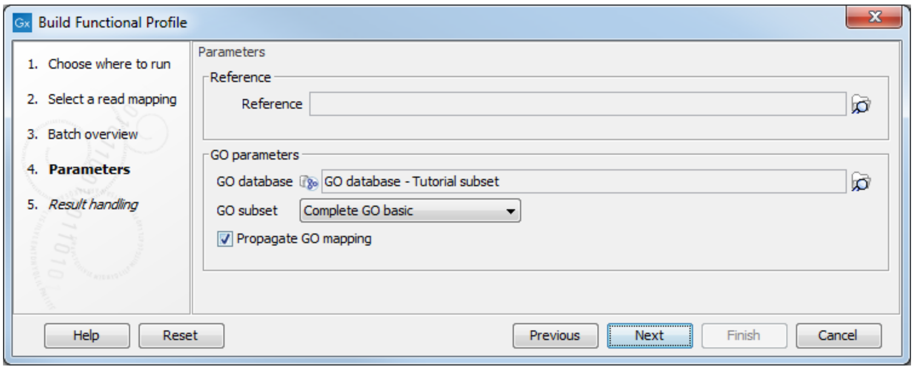


Selected elements (4)
- setA1_1 (paired) mapping
- setA2_1 (paired) mapping
- setB1_1 (paired) mapping
- setB2_1 (paired) mapping

☑ Batch

Figure 20: *Select the four read mappings and analyze them in batch.*

5. Use ''*GO database - Tutorial subset.clc* ()'' as GO database, as shown in figure 21.



Reference:

GO database: GO database - Tutorial subset

GO subset: Complete GO basic

☑ Propagate GO mapping

Figure 21: *Parameters for Build Functional Profile.*

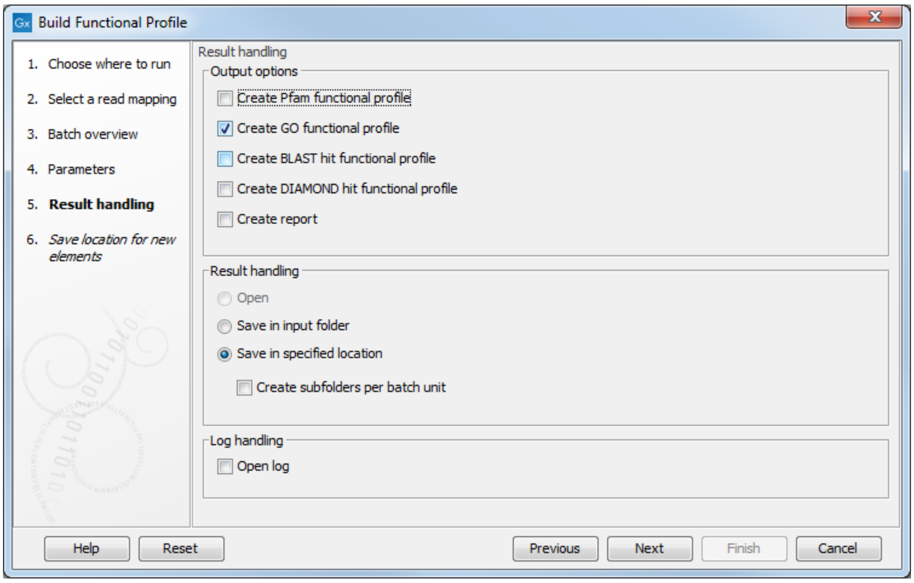
6. Finally, choose **Create GO functional profile** only and save to a new "Functional profiles" folder (figure 22).

Merge abundance

You have now built a functional profile for each sample. We now want to merge them using the **Merge Abundance Tables** tool.

1. Create a folder to store the results (such as "Statistical analyses") and go to:

   Metagenomics () | **Abundance Analysis** () | **Merge Abundance Tables** ()
2. Select the four GO profiles as input (figure 23).



Build Functional Profile

1. Choose where to run
2. Select a read mapping
3. Batch overview
4. Parameters
5. **Result handling**
6. *Save location for new elements*

Result handling

Output options

☐ Create Pfam functional profile

☑ Create GO functional profile

☐ Create BLAST hit functional profile

☐ Create DIAMOND hit functional profile

☐ Create report

Result handling

○ Open

○ Save in input folder

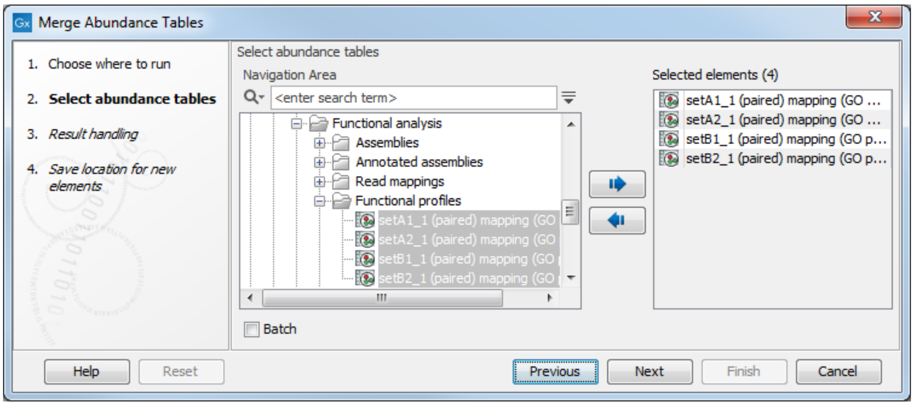
● Save in specified location

☐ Create subfolders per batch unit

Log handling

☐ Open log

Figure 22: *Create only a GO functional profile.*



Merge Abundance Tables

1. Choose where to run
2. **Select abundance tables**
3. *Result handling*
4. *Save location for new elements*

Select abundance tables

Navigation Area

<enter search term>

- Functional analysis
  - Assemblies
  - Annotated assemblies
  - Read mappings
  - Functional profiles
    - setA1_1 (paired) mapping (GO
    - setA2_1 (paired) mapping (GO
    - setB1_1 (paired) mapping (GO
    - setB2_1 (paired) mapping (GO

Selected elements (4)

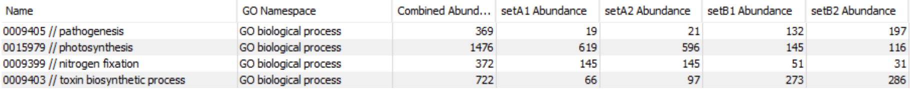
- setA1_1 (paired) mapping (GO ...
- setA2_1 (paired) mapping (GO ...
- setB1_1 (paired) mapping (GO p...
- setB2_1 (paired) mapping (GO p...

☐ Batch

Figure 23: *Merge the GO functional profiles.*

3. Save the merged profile in a new "Statistical analyses" folder.

The tool will create an abundance table called ‘‘*merged* ()’’ containing functional abundances for the four samples. You can now open the table to explore the results of the functional analysis (figure 24). Observe the functional abundance values for the four GO terms. As expected, abundance values for "pathogenesis" and "toxin biosynthetic process" are higher in group B, whereas group A is enriched in "photosynthesis" and "nitrogen fixation".



Name	GO Namespace	Combined Abund...	setA1 Abundance	setA2 Abundance	setB1 Abundance	setB2 Abundance
0009405 // pathogenesis	GO biological process	369	19	21	132	197
0015979 // photosynthesis	GO biological process	1476	619	596	145	116
0009399 // nitrogen fixation	GO biological process	372	145	145	51	31
0009403 // toxin biosynthetic process	GO biological process	722	66	97	273	286

Figure 24: *Result of the GO functional analysis.*

Performing statistical analyses

A heat map and dendrogram help assessing similarity between samples.

1. Open the **Metagenomics** () | **Abundance Analysis** () | **Create Heat Map for Abundance Table ()** and choose the "merged" table as input.
2. Leave the parameters as set by default, i.e., the distance to **Euclidean** and clusters to **Complete linkage**. Click **Next**.
3. In the next wizard window, do not set any particular filter by selecting the option "No filtering" and click **Next**.
4. Save the result in the "Statistical analyses" folder.

Display the heat map by double-clicking on it in the Navigation Area (figure 25).

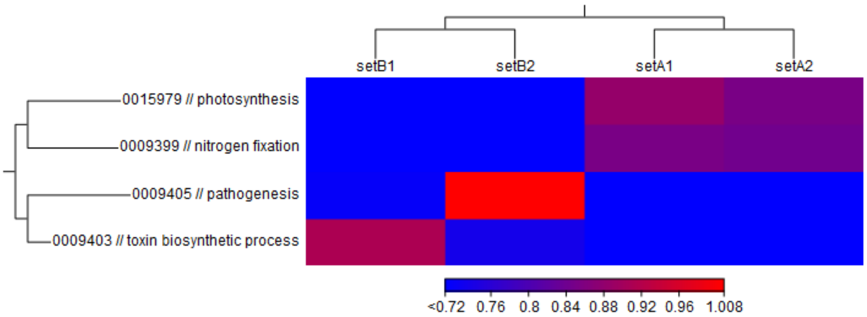


Figure 25: *Heat map from the abundance table.*

As we would have expected from the description of the data-set by Lindgreen et al., 2016 in the beginning of this tutorial, the normalized values for toxin biosynthesis and pathogenesis are over-expressed in group B, while the normalized values for photosynthesis and nitrogen fixation are enriched in group A. Furthermore, the samples from each group cluster together, as shown in the dendrogram at the top of the figure.

It is also possible to use as additional statistical analyses the **Differential Abundance Analysis** tool, although the interest of a Venn diagram is quite limited when the data set is only made of two distinctive groups as it is for this tutorial.

Although the results are hardly surprising, it is always re-assuring and good scientific practice to first apply a method to a problem with a known solution in order to verify everything works out exactly as expected before moving on to harder problems. Well done! At this point, we'd like to point out again that it is important to download the full versions of the Pfam and GO databases (by using the tools provided in the workbench) prior to using the functional annotation pipeline for a complete functional analysis of your own datasets.

Bibliography

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29.

[Bateman et al., 2004] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L. L., Studholme, D. J., Yeats, C., and Eddy, S. R. (2004). The Pfam protein families database. *Nucleic Acids Res*, 32(Database issue):D138–D141.

[Finn et al., 2016] Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., Salazar, G. A., Tate, J., and Bateman, A. (2016). The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 44(D1):D279–D285.

[Lindgreen et al., 2016] Lindgreen, S., Adair, K. L., and Gardner, P. P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6:19233–.

[The Gene Ontology Consortium, 2015] The Gene Ontology Consortium (2015). Gene ontology consortium: going forward. *Nucleic Acids Research*, 43(D1):D1049–D1056.