



## BIOL 470- Special Topics in Bioinformatics Version 2

Forked from [BIOL 354W - Research Methods in Advance Microbiology](#)

Rosa Leon<sup>1</sup>

<sup>1</sup>Willamette University

[dx.doi.org/10.17504/protocols.io.utjewkn](https://doi.org/10.17504/protocols.io.utjewkn)

Version 2

Oct 18, 2018

In devel.

Rosa Leon  
Willamette University

### ABSTRACT

This protocol series will guide students through the experience of analyzing metagenomic data.

### PROTOCOL STATUS

#### In development

We are still developing and optimizing this protocol

### DNA quality assessment and assurance

- 1 The first step in analyzing the sequencing data set is to assess the quality of the sequence, and then to edit the dataset in order to retain only the highest quality sequences for the following analysis.

To this end we will use: FastQC - A high throughput sequence QC analysis tool

Familiarize your self with the software by looking at their [web page](#) - check out the video tutorial!

#### COMMAND

```
alias fastqc=/storage/BioInfo_tools/FastQC/fastqc
```

Create an alias for the command

#### COMMAND

```
fastqc seqfile1.fastq
```

Now that the computer knows where to find the software, you can use a the

#### NOTE

You can perform the fastqc file on .fastq files and also in .fastq.gz files or compressed files

#### COMMAND

```
scp -r username@bio-server-2.willamette.edu:/home/username/folder_with_fastqc_file ~/Desktop/
```

Now that the software has run and you have folders and files with date, you should look at the data to assess the quality and make decision about the quality control step that we will work on next. For this you can unzip you folder where there will be detail information about the results, as well as a summary of the run. You can also download the .html file to look at the graphic representation of the run, the same format you experienced on the fastqc web and tutorial

#### NOTE

This step must be done from a Terminal window that is looking at your own computer and not connected to the sever

#### NOTE

### Assuring DNA sequencing quality using Trimmomatic

- 2 Trimmomatic: A flexible read trimming tool for Illumina NGS data ([Website](#))

#### Description

Trimmomatic performs a variety of useful trimming tasks for illumina paired-end and single ended data. The selection of trimming steps and their associated parameters are supplied on the command line.

The current trimming steps are:

- ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality
- TRAILING: Cut bases off the end of a read, if below a threshold quality
- CROP: Cut the read to a specified length
- HEADCROP: Cut the specified number of bases from the start of the read
- MINLEN: Drop the read if it is below a specified length
- TOPHRED33: Convert quality scores to Phred-33
- TOPHRED64: Convert quality scores to Phred-64

#### COMMAND

```
java -jar /storage/BioInfo_tools/Trimmomatic-0.36/trimmomatic-0.36.jar PE -threads 5 -phred33 input_forward.fq.gz input_reverse.fq.gz output_forward_paired.fq.gz output_forward_unpaired.fq.gz output_reverse_paired.fq.gz output_revers
input_forward.fq.gz = L6_R1_subset.fastq
input_reverse.fq.gz = L6_R2_subset.fastq
```

### Metagenomic assembly

- 3 To assemble our metagenomes we will try two different assemblies and compare them. We will try IDBA\_UD and Megahit assemblies. These are going to be one of the most time intensive process that we will do in the class.

Megahit github - <https://github.com/voutcn/megahit/>

Megahit article - <https://academic.oup.com/bioinformatics/article/31/10/1674/177884>

**COMMAND**

```
/storage/BioInfo_tools/megahit/megahit -1 file_R1.fq.gz -2 file_R2.fq.gz -o megahit_out -t 5
```

file\_R1.fq.gz = your trimmed forward or R1 reads  
 file\_R2.fq.gz = your trimmed reversed or R2 reads  
 megahit\_out = output folder - you can call it what ever you want and you don't need to make it before your run the software

**Assessing the quality of the assemblies**

- 4 We can investigate assembly statistics to compare which assembly is best between the two assemblies utilized. For this we can use a software called Quast.

Metrics based only on contigs:

- Number of large contigs (i.e., longer than 500 bp) and total length of them.
- Length of the largest contig.
- N50 (length of a contig, such that all the contigs of at least the same length together cover at least 50% of the assembly).
- Number of predicted genes, discovered either by GeneMark.hmm (for prokaryotes), GeneMark-ES or GlimmerHMM (for eukaryotes), or MetaGeneMark (for metagenomes).

**COMMAND**

```
/storage/BioInfo_tools/quast/metaquast.py contig.fa --gene-finding -t 5
```

QUAST evaluates genome assemblies by computing various metrics.

**Binning assembled metagenomes with MaxBin**

- 5 MaxBin is a software for binning assembled metagenomic sequences based on an Expectation-Maximization algorithm.

Users provide the assembled metagenomic sequences and the reads coverage information or sequencing reads. MaxBin will report genome-related statistics, including estimated completeness, GC content and genome size in the binning summary page.

MaxBin article - <https://academic.oup.com/bioinformatics/article/32/4/605/1744462>

**COMMAND**

```
perl /storage/BioInfo_tools/MaxBin-2.2.4/run_MaxBin.pl -contig "assembled_contigs.fa" -reads "reads file" -reads2 "readsfile" -out "out directory" -thread 5
```

MaxBin requires the assembled contains file and also the file that contains the sequence reads  
 assembled\_contigs.fa = your contigs file (remember to add the full path if you are in a different directory)  
 reads = the path to your reads (or paired R1 followed by R2 reads).  
 out directory = a directory that you create to save your bins

**Assessing the quality of your bins via CheckM**

- 6 Checkm article - <http://genome.cshlp.org/content/25/7/1043>

Also check out the website for information on CheckM - [CheckM website](#)

Before running Checkm the software pplacer must be included in the PATH by addind  
 export PATH="/storage/anaconda3/bin:\$PATH" to the .bashrc file in your home directory under the  
 # User specific aliases and functions section.

**COMMAND**

```
nano .bashrc
```

##copy and paste

User specific aliases and functions  
 export PATH="/storage/anaconda3/bin:\$PATH"

## Save file changes by "control + O" and then "control + X", then close the window and log in again to the server

**COMMAND**

```
/usr/bin/checkm lineage_wf -x fasta ./bins_folder ./checkm_out_folder -t 5
```

CheckM will assess the quality of each of your bins. All bins must be in the same directory/folder. All bins must have a .fasta ending  
 bins\_folder = the path to the folder where your bins are located

**COMMAND**

```
/usr/bin/checkm qa lineage.ms -o 2
```

This command will help you generate an expanded information table about each of your bins. Run this command from within the directory where your checkm data is located  
 copy the table that this command generated onto an excel sheet and analyze to then run VizBin

**Use VizBin to further curate your bins**

- 7 VizBin is a java software that calculates kmer composition and creates a pictographical output that shows the similarity between contigs related to how close they are position to each other. We will use VizBin to help us de-contaminate out bins

VizBin will generate a visualization window. Each point represents a genomic fragment (by default of length >= 1,000nt). VizBin is designed with the user in mind. All that is needed is a fasta file containing the sequences of interest. A step-by-step guide on using VizBin - including a description of loading the data, selecting points, and exporting the sequences represented by the selected points - is provided on the tutorial page of [VizBin's github wiki](#)

In order to run VizBin with you data you must download you bins fasta files onto your desktop.

To download go to the [VizBin page](#)

### Perform taxonomic identification using Phylosift

- 8 Phylosift software searches for single copy marker genes and finds their taxonomic classification

Before running this command take a moment to learn about the software at the [Phylosift webpage](#)

#### COMMAND

```
/usr/local/phylosift_v1.0.1/bin/phylosift all your_bin.fasta --threads 3
```

To run Phylosift you only need to have change your\_bin.fasta for the files (and path if required) for each of your individual bins

### Prokka - software for annotations

- 9 Learn about how to set up a prokka run and what the outputs are by looking at the git hub [prokka webpage](#)

#### COMMAND

First export the executable files in the bin directory :

```
export PATH=$PATH:/opt/BioInfo_tools/prokka-1.11/bin/
```

Then run PROKKA:

```
/opt/BioInfo_tools/prokka-1.11/bin/prokka contigs.fasta
```

We will annotated our curated bins using PROKKA

### Compare genomes to various databases

- 10 In order to assess the metabolic potential of you Metagenome Assembled Genomes (MAGs) we will compare their predicted proteins against a few different databases. These databases will provide information about what pathways or protein groups your annotated proteins belong to. This will help you assess what kind of metabolic potential your organisms possess.

We will start by taking our annotated proteins and running it in the BlastKoala web platform. <http://www.kegg.jp/blastkoala/>  
Use your PROKKA.faa file to copy the protein annotations and past on the box label Enter FAST A sequences or upload the PROKKA.faa file. Add you email so they can keep you update on the progress of your analysis.

Once you submit your PROKKA.faa and receive an email that your results are ready to view. Go to the View tab on top of the pie chart and press download details to get information about what metabolic pathway your proteins are associated with. After doing this, go back to the pie chart webpage and click on the Reconstruct Modules link at the bottom of the page. This will show metabolic pathways and in the detailed tab will show you which of your proteins fall within each pathways. Copy this and use as a text or save as PDF (by using Safari web browser)

- 11 In order to run the next few steps we need to add another set of software to our path

#### COMMAND

```
nano .bashrc
```

```
##copy and paste
```

User specific aliases and functions

```
export PATH=$PATH:/opt/ncbi-blast-2.7.1+/bin/
```

```
## Save file changes by "control + O" and then "control + X", then close the window and log in again to the server
```

This step is crucial to successfully run the next few steps

- 12 Compare annotated proteins to the Cluster of Orthologous Genes (COG)

#### COMMAND

```
rpblast -query PROKKA.faa_file (include the path if necessary) -db /opt/BioInfo_tools/Cog -out output_file.out -evalue 0.00001 -outfmt "6 qseqid sseqid eval evalue pident score qstart qend sstart send length slen" -max_target_seqs 1
```

#### COMMAND

```
perl /opt/BioInfo_tools/cdd2cog.pl -r output_file.out -c /opt/BioInfo_tools/COG/cddid.tbl -f /opt/BioInfo_tools/COG/fun.txt -w /opt/BioInfo_tools/COG/whog -a
```

Once we have generated a blast output, which provides a comparison of our annotated bins to the COG database we can use the cdd2cog perl script to count and parse that information for us

### Running Phylosift on metagenomic reads using tmux

- 13 In order to assess the community composition of the whole metagenome we can use phylosift to find short pieces of markers in our reads. While running Phylosift with out bins/genomes takes maybe one to a few hours running Phylosift with millions of reads will take multiple hours to days. For this reason we need to use a a window manager software call tmux. tmux will allow us to set up a process/job to run in a parallel window and exit the window while the process keeps running in the background.


In order to run Phylosift using tmux :

- Type = tmux new -s session-name example of a session-name - phylosft\_3B
- On the new window write your script command for Phylosift  
/usr/local/phylosift\_v1.0.1/bin/phylosift all --paired R1.fastq R2.fastq
- Verify that is running by typing = tmux ls
- Press together the keys control+b+z in your keyboard to disconnect from the parallel window
- To go back to that window type = tmux a -t session-name for example tmux a -t phylosft\_3B
- If something went terrible wrong you can kill your parallel window by typing = tmux kill-session -t phylosft\_3B

### Running PROKKA and COG on your metagenomic contigs

14 Given that we decided that we will be working with megahit assemblies - we will use the final\_contigs.fa file to run both PROKKA on the full metagenome (as opposed to bins recovered from the metagenome) and COG on the full metagenome.

To do this use the same commands as above for both PROKKA and COG, but change the fasta file to the final\_contigs.fa from your whole metagenome

 This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited