M14: Final Project

Xulin Ge[1]

[1] AS.410.734.82.FA22 Practical Introduction to Metagenomics.

Received on 12/19/2022

Instructor: Joshua Orvis

ABSTRACT

Motivation: Many important problems in cell biology require the dense nonlinear interactions between functional modules to be considered. The importance of computer simulation in understanding cellular processes is now widely accepted, and a variety of simulations algorithms useful for studying certain subsystems have been designed.

1 INTRODUCTION

Metagenomic data from microbial community can be analyzed to show the taxonomic composition of the microorganisms present in a diverse sample. However, the amount of computing resource by the entire metagenome analysis required may be overwhelming to one trying to run the pipeline.

In this final project, I'm going to use Google Cloud Platform (GCP) to present and validate the metagenomic pipeline authored by Garfias-Gallegos, D. et al. (2022) [1]. I want to analyze the reads file available from NCBI, explore the microbial community of the microorganisms from the sample, and visualize the taxonomic composition results in plots.

A typical metagenomic analysis includes steps of filtering low-quality reads, metagenome assembly, binning, and taxonomic assignment (**Fig. 1**) [1].

For most sequencing reads data, the quality of the reads needs to be examined by “FastQC” [3], and then low-quality reads can be filtered-out using “Trimmomatic” [4].

The assembly step that creates large sequences from short reads is often time-consuming but necessary. “metaSPAdes” is a good software designed to meet difficulties during assembly, like the differences in coverage [5].

To study a specific taxon in the community, individual genomes of the community need to be separated based on criteria like sequencing depth. The binning process would return MAGs (metagenome-assembled genomes). “MaxBin” can separate the contigs or scaffolds of a metagenomic assembly into bins based on differences in coverage and tetranucleotide composition [6]. And “CheckM” can analyze the quality of the resulting MAGs by calculating MAGs completeness and contamination [7].

Finally, to reveal the community composition and the taxon of the obtained MAGs, we need to perform the taxonomic assignment. This can be performed by using Kraken2 [8]. The R package Phyloseq can be used to visualize the resulting taxonomic composition [9].

The authors extracted the total DNA of the microorganisms inhabiting the roots of maize plant and analyzed the taxonomic composition of the sample. Through applying a metagenomic analysis workflow on sequencing reads, they visualized the relative abundance at the phylum level of the samples. The results help them reveal the nature of the microorganisms community inhabiting the roots of cultured plants.

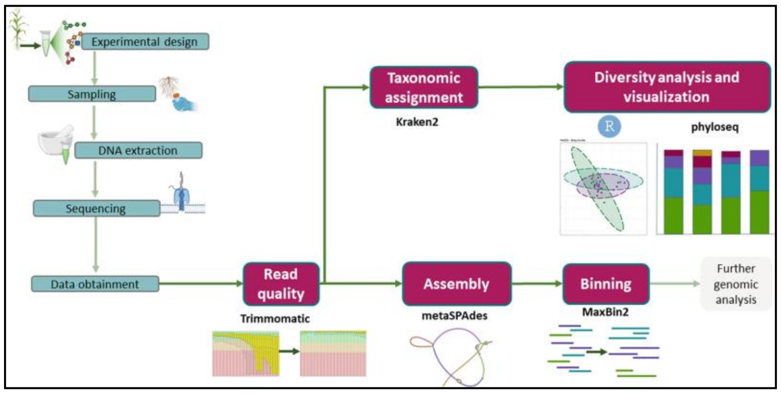


Fig. 1. A brief Metagenomics Bioinformatic Pipeline from Garfias-Gallegos, D. et al. (2022) paper [1]. This final project is going to run the five analysis steps shown in red squares

2 METHODS

System

The pipeline is run in the command line of a 64-bit Linux system (Ubuntu). The virtual machine (VM) instance created in Google Cloud Platform (GCP) is set as 16 threads CPU, 64GB RAM, and 500GB disk space.

Tools for Bash

Tools in Bash:

SRA toolkit v2.11.0 (https://github.com/ncbi/sra-tools)
FastQC v0.11.7 (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
Trimmomatic v0.38 (http://www.usadellab.org/cms/?=trimmomatic)
Kraken2 v2.1.1(http://ccb.jhu.edu/software/kraken2/)
MaxBin2 v2.2.7 (https://anaconda.org/bioconda/maxbin2)
metaSPAdes v3.14.1 (https://cab.spbu.ru/software/spades /)
kraken-biom v1.0.1 (https://github.com/smdabdoub/kraken-biom)
CheckM v1.1.3 (https://ecogenomics.github.io/CheckM/)

Tools for R

Phyloseq v 1.36.0 (https://github.com/joey711/phyloseq)
ggplot2 v 3.3.3 (https://ggplot2.tidyverse.org/).

Data used

The data used in the project are Endophyte from maize root. (Endophyte from maize root).
Analysis data:
SRR11131035 https://www.ncbi.nlm.nih.gov/sra/SRX7767547[accn]

I also downloaded the three data files used in Garfias-Gallegos, D. et al. (2022) paper. However, the analysis of them would cost me too much time. In fact, my GCP VM instance crashed during the progress because of running out of disk space. So, just list their accessions here:
SRR11131028 https://www.ncbi.nlm.nih.gov/sra/SRX7767554[accn]
SRR11131029 https://www.ncbi.nlm.nih.gov/sra/SRX7767553[accn]
SRR11131030 https://www.ncbi.nlm.nih.gov/sra/SRX7767552[accn]

Estimated running time and cost
Installation of all tools takes about 1 hours. Building Kraken2 database costs 12 hours.
Analysis steps before assembly (including downloading data) takes about 1 hours. Metagenome assembly using "metaSPAdes" costs 8 hours on SRR11131035 dataset. The subsequent analysis takes about 2 hours.
Google Cloud shows a billing credit cost of $ 50.14

Code Availability
The code of the entire process, including the installation of all tools, is contained in the "Supplementary material" file.
My GCP project has added professor J.Orvis (jorvis@gmail.com) as the project owner who has the full access to VM instances and editing.

3 RESULTS

3.1 Project Organization and Data Obtainment

Set up Google Cloud Platform (GCP) and virtual machine (VM) instance. Enter the created VM instance, install necessary tools (See Supplementary material **Installing tools**), and organize the project directory.

```
├── r-analysis
├── raw-reads
└── results
    ├── assemblies
    ├── fastqc
    ├── taxonomy
    ├── trimmed-reads
    └── untrimmed-reads
```

Download SRR11131035 data using command `fastq-dump` from "SRA-toolkit":
SRR11131035_1.fastq 5.4G
SRR11131035_2.fastq 5.4G

3.2 Assessing Read Quality'

Before any further metagenomic analysis, any downloaded reads file is suggested to assess the quality of the reads. 'FastQC' is a good tool that can be used to visualize the read quality of each sample [3].

```
$ time fastqc SRR11131035*
Start-
ed analysis of SRR11131035_1.fastq
...
...
Analy-
sis complete for SRR11131035_2.fastq

real    4m13.409s
user    4m10.137s
sys     0m8.599s
```

'FastQC' returns a ".html" report (**Fig. 2. top**) and saves output in a ".zip" file for each sample.

```
741K SRR11131035_1_fastqc.html
459K SRR11131035_1_fastqc.zip
747K SRR11131035_2_fastqc.html
479K SRR11131035_2_fastqc.zip
```

The ".zip" output can be decompressed to an output folder, which contains files concerning the process ("summary" and "_data.txt"), a copy of the ".html" report, and files used to generate the output report [1].

```
$ ls
Icons  Images  fastqc.fo  fastqc_dat
a.txt  fastqc_report.html  sum-
mary.txt
```

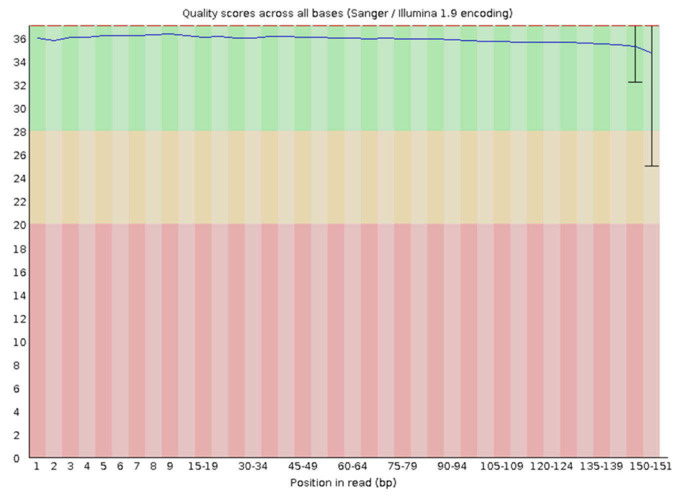
3.3 Trimming and Filtering

'FastQC' makes an assessment of reads quality and helps us establish quality thresholds. Next, we will remove low-quality sequences to reduce the false-positive rates that come from sequencing error.
As shown in the 'FastQC' report (**Fig. 2. top**), the minimum expected length for this data will be set at 35 bases. Thus, we can use "Trimmomatic" to remove the undesired low-quality data below the quality threshold.

```
$ time for file in *_1.fastq; do bas
e=$(basename ${file} _1.fastq)
 java -
jar ~/Trimmomatic/dist/jar/trimmomat
ic-0.40-
rc1.jar PE ${file} ${base}_2.fastq \
 ${base}_1.trim.fastq ${base}_1.unpa
ir.fastq \
 ${base}_2.trim.fastq ${base}_2.unpa
ir.fastq \
 SLIDINGWINDOW:4:20 MINLEN:35
 done

Trimmo-
maticPE: Started with arguments:
 SRR11131028_1.fastq SRR11131028_2.f
astq SRR11131028_1.trim.fastq SRR111
31028_1.unpair.fastq SRR11131028_2.t
rim.fastq SRR11131028_2.unpair.fastq
 SLIDINGWINDOW:4:20 MINLEN:35
Quality encoding detected as phred33
...
...
Trimmo-
maticPE: Completed successfully

real    18m44.606s
user    12m12.193s
sys     5m35.007s
```



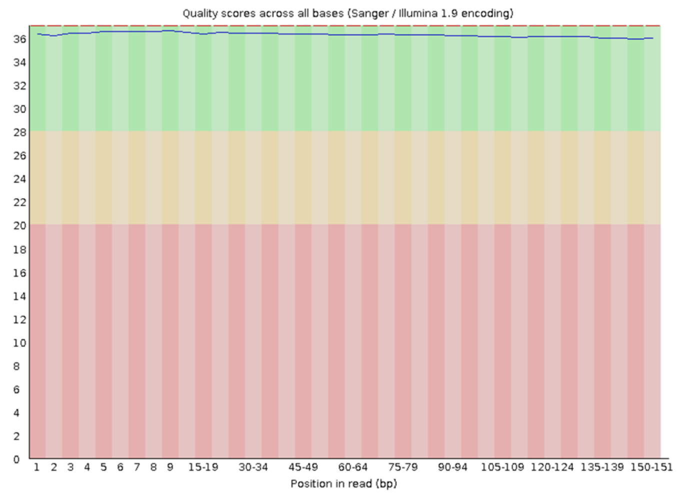


Fig. 2. Per base sequence quality of SRR11131035_1.fastq file from FastQC before (**top**) and after (**bottom**) trimming

3.4 Metagenome Assembly

It is a time-consuming process of reads assembly where reads will be assembled into contigs and these contigs into scaffolds. Various algorithms have been developed to meet the difficulties of the process, like the differences in coverage.

"metaSPAdes" has been demonstrated to be a suitable tool that has a good performance in assembling metagenomes from different types of samples [5].

Run "metaSPAdes" to perform the assembly with each pair of files from each sample. This can take quite some time.

```
$ time for file in *_1.trim.fastq;
>  do base=$(basename ${file} _1.tri
m.fastq)
>  metaspades.py -1 ${file} \
>  -2 ${base}_2.trim.fastq \
>  -o ~/final-
project/results/assemblies/assembly-
${base}
>  done
......
......

===== Assembling finished. Used k-
mer sizes: 21, 33, 55
......
......

======= SPAdes pipeline finished.
......
......

real    392m27.662s
user    5349m46.826s
sys     19m58.439s
```

"metaSPAdes" will generate an output folder of each assembly and "scaffolds.fasta" is where the assembly of each sample is located. Rename the assembled sequence file and move to another folder.

```
$ ls
SRR11131035-
scaffolds.fasta  assembly-
SRR11131035
```

3.5 Metagenome Binning

In order to separate the taxon in the community, the binning process is needed. The binning of the assembled scaffolds will return metagenome-assembled genomes (MAGs). We will use "MaxBin" for binning metagenomic assembly into bins, and "CheckM" for analyzing the quality of the resulting MAGs.

```
$ run_MaxBin.pl -thread 16 -
contig SRR11131035-scaffolds.fasta \
>  -reads ~/final-
project/results/trimmed-
reads/SRR11131035_1.trim.fastq \
>  -reads2 ~/final-
project/results/trimmed-
reads/SRR11131035_2.trim.fastq \
>  -out ./SRR11131035-
maxbin/SRR11131035
```

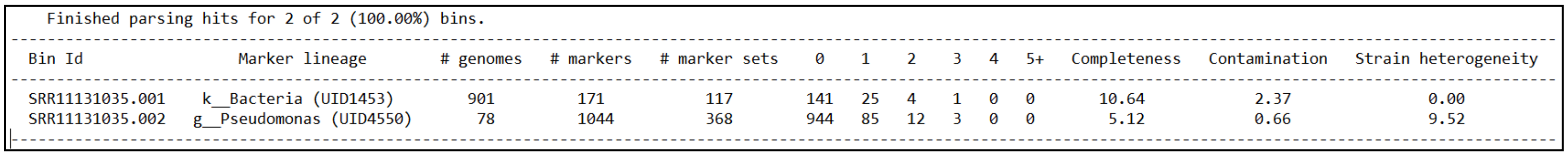
```
MaxBin 2.2.4
Thread: 16
Input contig: SRR11131035-
scaffolds.fasta......
......
========== Job finished ==========
Yield-
ed 2 bins for contig (scaffold) file
 SRR11131035-scaffolds.fasta
......
......
========== Elapsed Time ==========
0 hours 9 minutes and 10 seconds.
```

The binning results of "MaxBin" are saved in the ".summary" file.

```
$ cat SRR11131035-maxbin/*.summary
Bin name          Completeness    Geno
me size      GC content
SRR11131035.001.fasta    15.9%    1054
955 42.5
SRR11131035.002.fasta    30.8%    1343
859 50.8
```

Now, let's run to "CheckM" to access the quality of the resulting MAGs (**Table 1**).

```
$ time checkm lineage_wf -
x fasta SRR11131035-
maxbin/ SRR11131035-checkm -t 16
......
    Fin-
ished parsing hits for 2 of 2 (100.0
0%) bins.
......
```



```
  Finished parsing hits for 2 of 2 (100.00%) bins.
----------------------------------------------------------------------------------------------------------------------------------------------------
 Bin Id                Marker lineage          # genomes   # markers   # marker sets    0     1    2    3    4    5+   Completeness   Contamination   Strain heterogeneity
----------------------------------------------------------------------------------------------------------------------------------------------------
 SRR11131035.001     k__Bacteria (UID1453)        901         171          117         141    25   4    1    0    0       10.64            2.37                0.00
 SRR11131035.002    g__Pseudomonas (UID4550)       78         1044         368         944    85   12   3    0    0        5.12            0.66                9.52
----------------------------------------------------------------------------------------------------------------------------------------------------
```

Table 1. Quality of the MAGs assessed by "CheckM"

Typically, we want the genomes of high completeness [>90%] and low contamination [<5%] for further studies. Our results here have a low contamination, but not much completeness. Thus, the subsequent taxonomic analysis is used for reference.

3.6 Taxonomic Assignment

It's easy to use "Kraken2" for taxonomic assignment. But it's time-consuming to download and build a "Kraken2" database that is necessary for the process. Build a "Kraken2" standard database could cost a long time and consume a large disk space. Here, we will only build the important "Kraken2" bacteria database which is the main taxonomy in our study. Still, it takes 12 hours and 147GB.

```
$ time kraken2-build --download-
library bacteria  --threads 16 --
db kraken_bac
......
Masking low-
complexi-
ty regions of downloaded library...

 done.

real    700m13.248s
user    585m33.619s
sys     36m24.477s
```

Performing taxonomic assignment using "Kraken2" is much more straightforward.

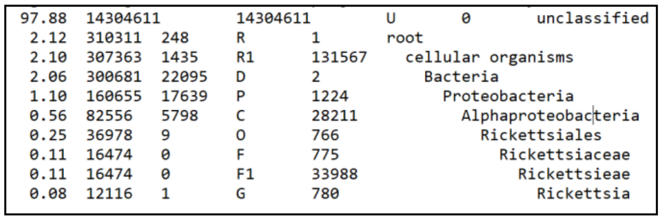
```
$ time for file in *_1.trim.fastq; d
o base=$(basename ${file} _1.trim.fa
stq)
>  mkdir ~/final-
project/results/taxonomy/$base
>  kraken2 --db ~/final-
pro-
ject/results/taxonomy/kraken_bac \
>  --paired --fastq-
input $file ${base}_2.trim.fastq \
>  --output ~/final-
pro-
ject/results/taxonomy/$base/$base.kr
aken \
>  --report ~/final-
pro-
ject/results/taxonomy/$base/$base.re
port
>  done
```

```
......
......
14614922 sequences (3975.65 Mbp) pro
cessed in 540.916s (1621.1 Kseq/m, 4
40.99 Mbp/m).
  310311 sequences classified (2.12%)
  14304611 sequences unclassified (9
7.88%)

real    10m56.853s
user    8m24.225s
sys     0m59.789s
```

The process will return the ".kraken" and ".report" files. And there is a readable presentation of the taxonomic assignment inside the ".report" files (**Table 2**).

```
$ head SRR11131035/SRR11131035.repor
t
```



```
97.88  14304611         14304611        U       0       unclassified
 2.12  310311  248      R       1       root
 2.10  307363  1435     R1      131567    cellular organisms
 2.06  300681  22095    D       2           Bacteria
 1.10  160655  17639    P       1224          Proteobacteria
 0.56  82556   5798     C       28211           Alphaproteobacteria
 0.25  36978   9        O       766               Rickettsiales
 0.11  16474   0        F       775                 Rickettsiaceae
 0.11  16474   0        F1      33988                 Rickettsieae
 0.08  12116   1        G       780                     Rickettsia
```

Table 2. The first few lines of `SRR11131035.report`

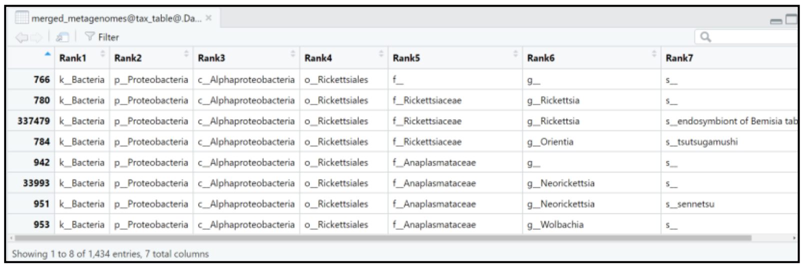
3.7 Diversity Tackled with R

Next, we will use R to analyze the taxonomic composition of the community in the sample. But before that, we need to generate a ".biom" object with "kraken-biom" algorithm that hoards the taxonomic assignment data generated with "kraken2".

```
$ kraken-
biom SRR11131035/SRR11131035.report
--fmt json -o taxonomy.biom
$ ls -lh
......
 268K taxonomy.biom
```

Now, we can work with R and import the "taxonomy.biom" with "phyloseq" package to load the taxonomic information (**Table 3**).

```
> library("phyloseq")
> merged_metagenomes <-
 im-
port_biom("E:\\data\\taxonomy.biom")
......
> View(merged_metagenomes@tax_table@
.Data)
```



merged_metagenomes@tax_table@.Da...

	Rank1	Rank2	Rank3	Rank4	Rank5	Rank6	Rank7
766	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__	g__	s__
780	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Rickettsiaceae	g__Rickettsia	s__
337479	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Rickettsiaceae	g__Rickettsia	s__endosymbiont of Bemisia tab
784	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Rickettsiaceae	g__Orientia	s__tsutsugamushi
942	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Anaplasmataceae	g__	s__
33993	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Anaplasmataceae	g__Neorickettsia	s__
951	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Anaplasmataceae	g__Neorickettsia	s__sennetsu
953	k__Bacteria	p__Proteobacteria	c__Alphaproteobacteria	o__Rickettsiales	f__Anaplasmataceae	g__Wolbachia	s__

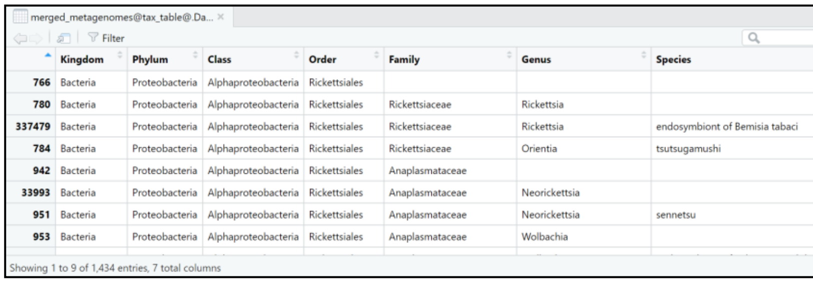
Showing 1 to 8 of 1,434 entries, 7 total columns

Table 3. The raw "tax_table" containing needed and unneeded taxonomy table information. The table requires trimming before further analysis.

However, the raw data from the ".biom" object contains many unnecessary information. For example, each taxonomic rank name has four unnecessary characters at the beginning, etc.

Thus, we need to trim the table for a better upcoming analysis (**Table 4**).

```
> # trim the data
> merged_metagenomes@tax_table@.Data
 <-
 sub-
string(merged_metagenomes@tax_table@
.Data, 4)
> colnames(merged_metagenomes@tax_ta
ble@.Data) <-
 c("Kingdom", "Phylum", "Class", "Or
der", "Family", "Genus", "Species")
> View(merged_metagenomes@tax_table@
.Data)
```



merged_metagenomes@tax_table@.Da...

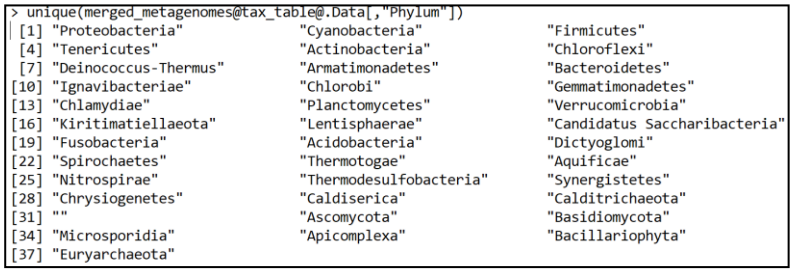
	Kingdom	Phylum	Class	Order	Family	Genus	Species
766	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales			
780	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Rickettsiaceae	Rickettsia	
337479	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Rickettsiaceae	Rickettsia	endosymbiont of Bemisia tabaci
784	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Rickettsiaceae	Orientia	tsutsugamushi
942	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Anaplasmataceae		
33993	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Anaplasmataceae	Neorickettsia	
951	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Anaplasmataceae	Neorickettsia	sennetsu
953	Bacteria	Proteobacteria	Alphaproteobacteria	Rickettsiales	Anaplasmataceae	Wolbachia	

Showing 1 to 9 of 1,434 entries, 7 total columns

Table 4. The trimmed "tax_table" now contains only taxonomic information.

Use "unique()" to list the different phyla identified by "Kraken2" (**Table 5**). We will plot the phyla using bar plots in the next section.

```
> unique(merged_metagenomes@tax_tabl
e@.Data[,"Phylum"])
```



```
> unique(merged_metagenomes@tax_table@.Data[,"Phylum"])
 [1] "Proteobacteria"          "Cyanobacteria"            "Firmicutes"
 [4] "Tenericutes"             "Actinobacteria"           "Chloroflexi"
 [7] "Deinococcus-Thermus"     "Armatimonadetes"          "Bacteroidetes"
[10] "Ignavibacteriae"         "Chlorobi"                 "Gemmatimonadetes"
[13] "Chlamydiae"              "Planctomycetes"           "Verrucomicrobia"
[16] "Kiritimatiellaeota"      "Lentisphaerae"            "Candidatus Saccharibacteria"
[19] "Fusobacteria"            "Acidobacteria"            "Dictyoglomi"
[22] "Spirochaetes"            "Thermotogae"              "Aquificae"
[25] "Nitrospirae"             "Thermodesulfobacteria"    "Synergistetes"
[28] "Chrysiogenetes"          "Caldiserica"              "Calditrichaeota"
[31] ""                        "Ascomycota"               "Basidiomycota"
[34] "Microsporidia"           "Apicomplexa"              "Bacillariophyta"
[37] "Euryarchaeota"
```

Table 5. Display the phyla identified by "Kraken2" with "unique()".

Before moving to the step of plotting the phyla, we can explore some metric of the taxonomic information in the data.
For example, we can use "plot_richness()" from the "phyloseq" package to plot the alpha-diversity indexes (**Fig. 3**).

```
> plot_richness(physeq=merged_metage
nomes, measures=c("Observed","Chao1"
,"Shannon"))
```

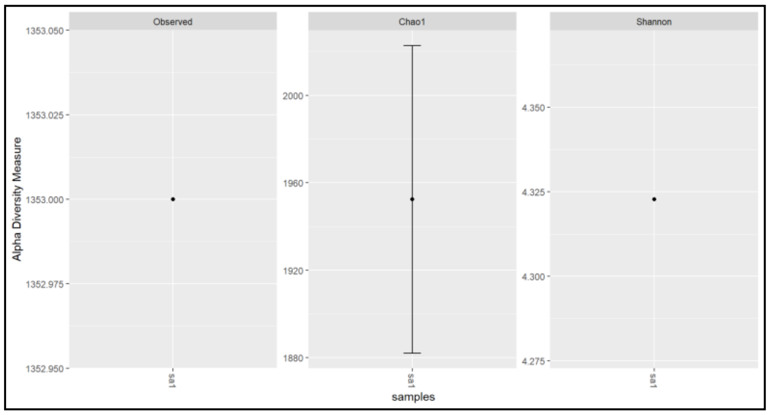
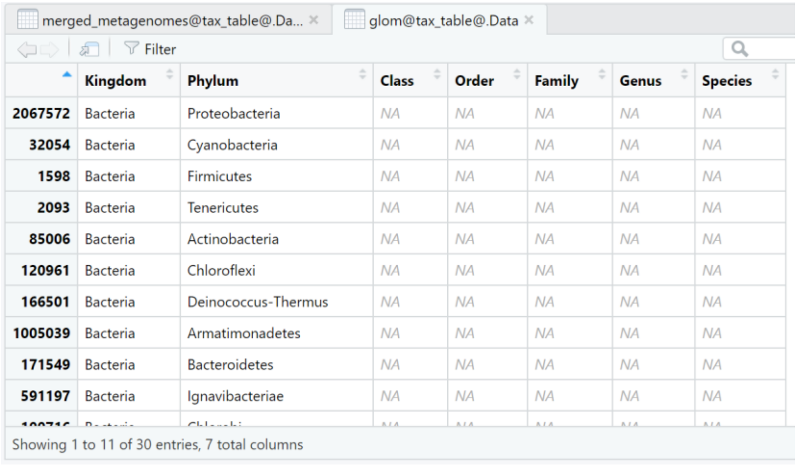


Fig. 3. Alpha diversity of the sample SRR11131035 with three different metrics (from left to right): Observed, Chao1, and Shannon.

3.8 A Focused Taxonomic Exploration

Finally, we are going to explore the taxonomic information inside the sample SRR11131035.
After transforming the data to relative abundances, the agglomeration of the abundance information at the desired taxonomic level can be made by "tax_glom()" (**Table 6**).

```
> percent-
ages = transform_sample_counts(
+      merged_metagenomes, function(x
) x*100 / sum(x) )
> glom <-
 tax_glom(percentages, taxrank='Phyl
um')
> View(glom@tax_table@.Data)
```



	Kingdom	Phylum	Class	Order	Family	Genus	Species
2067572	Bacteria	Proteobacteria	NA	NA	NA	NA	NA
32054	Bacteria	Cyanobacteria	NA	NA	NA	NA	NA
1598	Bacteria	Firmicutes	NA	NA	NA	NA	NA
2093	Bacteria	Tenericutes	NA	NA	NA	NA	NA
85006	Bacteria	Actinobacteria	NA	NA	NA	NA	NA
120961	Bacteria	Chloroflexi	NA	NA	NA	NA	NA
166501	Bacteria	Deinococcus-Thermus	NA	NA	NA	NA	NA
1005039	Bacteria	Armatimonadetes	NA	NA	NA	NA	NA
171549	Bacteria	Bacteroidetes	NA	NA	NA	NA	NA
591197	Bacteria	Ignavibacteriae	NA	NA	NA	NA	NA

Showing 1 to 11 of 30 entries, 7 total columns

Table 6. Agglomeration of the abundance at the phylum level

In order to show the abundance of the different phyla, we need to create a data frame with"psmelt()" and then make the bar plots. All the phyla will be displayed in the bar plot (**Fig. 4**).

```
> percentages <- psmelt(glom)
> # dis-
plat-
ing the abundance of the different p
hyla of each sample
> rel.plot <- ggplot(
+      data=percentages, aes(x=Sample
, y=Abundance, fill=Phylum))+geom_ba
r(
+           aes(), stat="identity", po
sition="stack")
> rel.plot
```

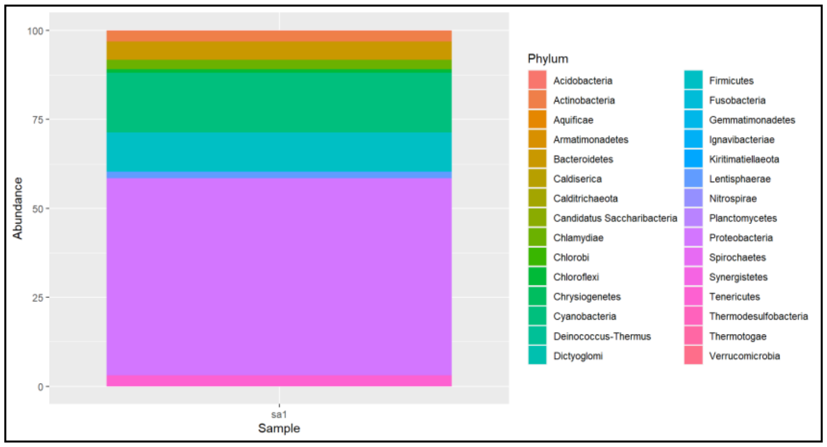


Fig. 4. Bar plots of the relative abundance at the phylum level. Each color marks a different phylum.

Because metagenomic sequencing often results in a lot of information, the phylum that are shown directly in different colors in the bar plot are not user-friendly for reading. So, we can simplify the bar plot by wrapping all the minority phyla in a new category containing all the OTUs with less than 0.5%.

```
> percent-
ag-
es$Phylum[percentages$Abundance < 0.
5] <- "Phyla < 0.5% abund."
> unique(percentages$Phylum)
 [1] "Proteobacteria"      "Cyanobac
teria"       "Firmicutes"
 [4] "Bacteroidetes"       "Tenericu
tes"         "Actinobacteria"
 [7] "Chlamydiae"          "Lentisph
aerae"       "Chloroflexi"
[10] "Phyla < 0.5% abund."
```

In this way, we make the bar plot much clearer for showing the taxonomic information (**Fig. 5**).

As shown from the new bar plot, it is easy to find the top 3 abundant phylum inside the sample SRR11131035 are:"Proteobacteria", "Cyanobacteria", and "Firmicutes" (**Fig. 5** purple, green, and cyan bar).
This result agrees with the agglomeration result listed in **Table 6**.

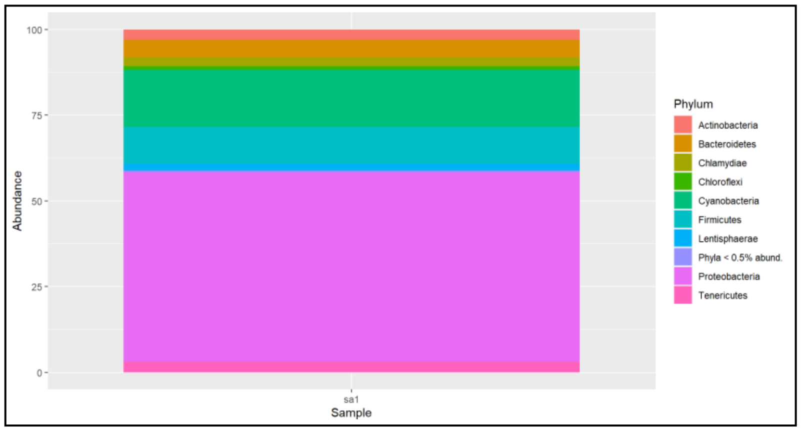


Fig. 5. Bar plots is much easier to visualize after rearranging the phyla categories using threshold of abundances >0.5%.

4 DISCUSSION

In this final project, I preprocess (assessing quality, trimming and filtering) the metagenomic data from the samples collected from the roots of maize plants.

After assembling the reads into scaffolds, I performed the taxonomic assignment using “Kraken2” database.

Then the “Kraken2” result is converted to a “.biom” object that can be used to analyze the taxonomic information inside the sample using R.

The R package “phyloseq” is useful for exploring the taxonomic information from the data, and “ggplot2” is also useful for visualizing the abundance with bar plots.

In summary, the analysis of metagenomic sequencing reads often takes long time and costs large disk space and eats memory. With the help of the powerful computing engines of Google Cloud Platform (GCP), I downloaded and processed the reads file SRR11131035. It is one of the reads files generated in the project of the paper Garfias-Gallegos, D. et al. (2022) [1], which was conducted to study the taxonomic composition of microorganisms inhabiting the roots of maize plant.

By using various open-sourced tools, including “FastQC”, “Trimmomatic”, “metaSPAdes”, “MaxBin”, “CheckM”, and “Kraken2”, as well as R packages, the taxonomic composition of the community at the phylum level were explored and visualized.

The top abundance result in SRR11131035 agrees with the conclusion of the paper Garfias-Gallegos, D. et al. (2022) [1].

Therefore, our conclusion validates the results of that paper.

Nevertheless, this project has some drawbacks. The authors used three reads file to show their results. SRR11131028, SRR11131029 , and SRR11131030. Because of the large size of these files, I was unable to process all of them. So, I chose to analyze another data SRR11131035 which was also generated by the research but not presented in the paper.

The future part of this final project should include the analysis of those three reads files, too.

REFERENCES

1. Garfias-Gallegos, D. et al. (2022). Metagenomics Bioinformatic Pipeline. In: Pereira-Santana, A., Gamboa-Tuz, S.D., Rodríguez-Zapata, L.C. (eds) Plant Comparative Genomics. Methods in Molecular Biology, vol 2512. Humana, New York, NY. https://doi.org/10.1007/978-1-0716-2429-6_10
2. Fadiji, A.E., Ayangbenro, A.S. & Babalola, O.O. Organic Farming Enhances the Diversity and Community Structure of Endophytic Archaea and Fungi in Maize Plant: a Shotgun Approach. J Soil Sci Plant Nutr 20, 2587–2599 (2020). https://doi.org/10.1007/s42729-020-00324-9
3. Andrews S (2010) FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc
4. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30(15):2114–2120. https://doi.org/10.1093/bioinformatics/btu170
5. Nurk S, Meleshko D, Korobeynikov A et al (2017) MetaSPAdes: a new versatile metagenomic assembler. Genome Res 27(5):824–834. https://doi.org/10.1101/gr.213959.116
6. Wu YW, Tang YH, Tringe SG et al (2014) MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. Microbiome 2(1):26. https://doi.org/10.1186/2049-2618-2-26
7. Parks DH, Imelfort M, Skennerton CT et al (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res 25(7):1043–1055. https://doi.org/10.1101/gr.186072.114
8. Wood DE, Salzberg SL (2014) Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol 15(3):R46. https://doi.org/10.1186/gb-2014-15-3-r46
9. McMurdie PJ, Holmes S (2013) Phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. PLoS One 8(4):e61217. https://doi.org/10.1371/journal.pone.0061217