Andrew Lutsky
October 16, 2022
MCB432
Group Members: Andrew Huang, Yishuo Jiang

MA2 Group Assignment

I have read the instructions and follow the rules of this major assignment.

This major assignment has a couple of main goals. That is to concatenate and compile several datasets together to produce a database that has been curated. That is to say the concatenated database has no duplicates. After the database has been created using several datasets, we would like to create a PCoA plot of the database, create a Maximum Likelihood tree, and create a metadata table using the clustering centers from the clustering package cd-hit.

The datasets used to compile the database or long fasta file are datasets 1,3,and 13 as well as SeqDatabase_2, instead of SeqDatabase_4. SeqDatabase_2 was used instead of 4 because 4 already has duplicates removed and is already curated. If we use an earlier version of the database, that is to say SeqDatabase_2, we can remove duplicates in one step and simultaneously generate the metadata required. The concatenated dataset still has duplicates and its metadata is required as well. Both of these steps can be performed by using a different shell script that will remove duplicates, generate metadata, format the header information for the SeqDatabase file, and perform a muscle alignment.The shell script required to merge these databases,remove duplicates, and generate metadata is seen below:

```
#!/bin/sh
#Def2Meta3.sh
#activates conda environment
conda activate myenv_x86
#merge SeqDatabase_2, MAData_1, MAData_3, MAData_4
cat MA2_data.1.fasta SeqDatabase_2.fasta >SeqDatabase_6.1.fasta
cat MA2_data.3.fasta SeqDatabase_6.1.fasta > SeqDatabase_6.2.fasta
cat MA2_data.13.fasta SeqDatabase_6.2.fasta > SeqDatabase_6.0.fasta




#step 1 - Remove duplicates from the SeqDatabase_6.0.fasta
awk '{gsub(/\r/,"")} />/&&list~substr($1,2){flag=0} />/&&list!~substr($1,2){n++;list=list",
"substr($1,2);flag=1}flag==1{print}'<SeqDatabase_6.0.fasta >SeqDatabase_6.1.fasta


#step2 - generate metadata for the SeqDatabase_6.1 fasta file
awk '/>/{print substr($0,2)}'<SeqDatabase_6.1.fasta| awk 'BEGIN{print
"ACC,GenSp,SubSp,Strain1,Strain2,Gene,comments"}$4=="16S"{$4=",,"$4}$4=="strain"
```

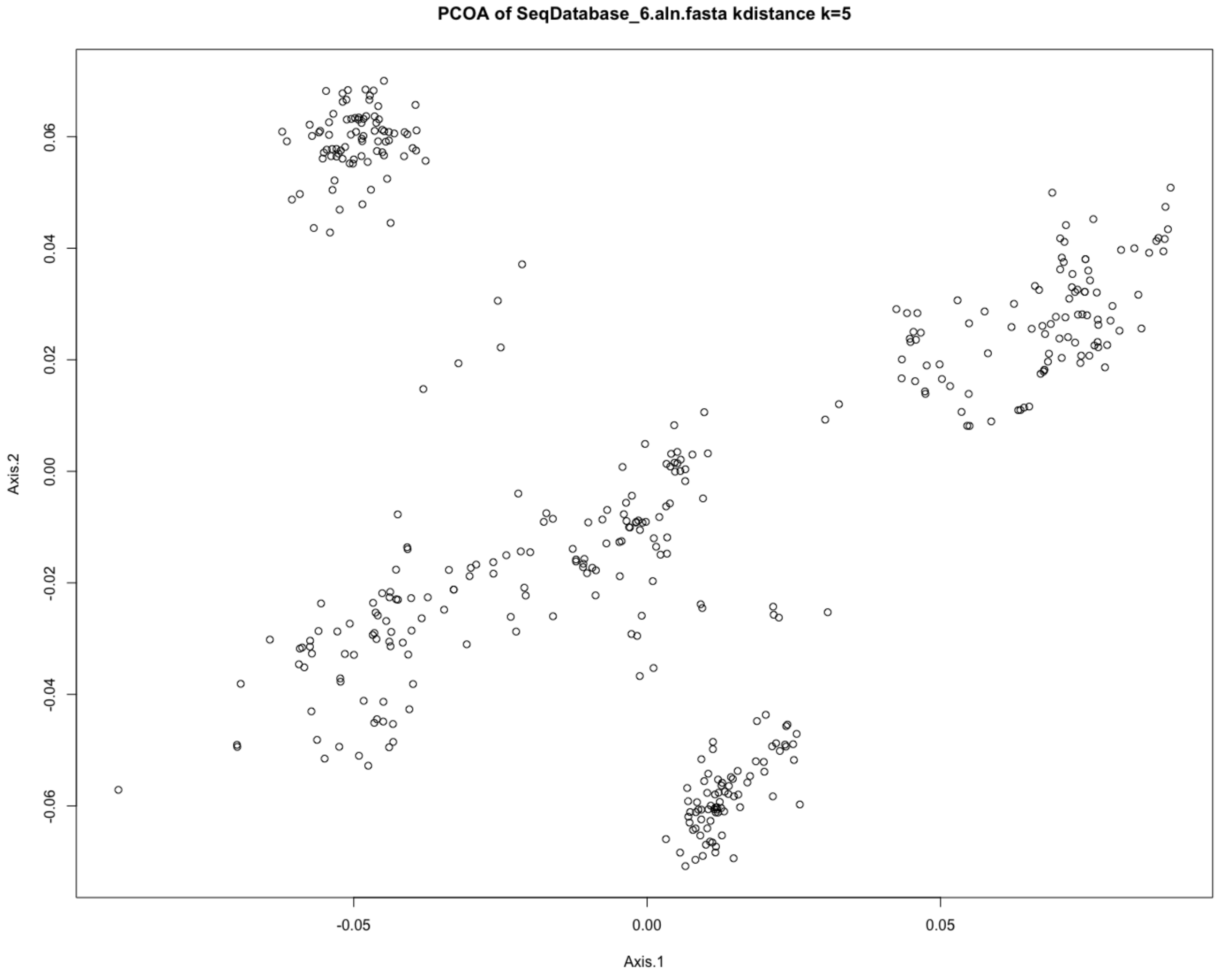
```
{$4=","$4}{$2=","$2; $3=substr($3,2)","}/strain/&&!/subsp/{$4=","$4;gsub(/
strain/,",strain")}!/strain/&&!/subsp/{$4=","$4;gsub(/ 16S ribosomal/,", 16S
ribosomal")}/strain/&&/subsp/{$5=$5","} !/strain/&&/subsp/{$6=","$6; gsub(/ 16S
ribosomal/,", 16S ribosomal")}{gsub(/ 16S ribosomal/,",16S ribosomal");gsub(/
strain/,",strain");gsub(/ partial/,"partial");gsub(/ complete/,"complete")}{print}'
>SeqData6_meta.csv

#step 3 - remove the excess information from the header of the SeqDatabase file
awk '/>/{print $1} !/>/&&$1!=""{print}'<SeqDatabase_6.1.fasta >SeqDatabase_6.fasta

#step 4 - do a muscle alignment
muscle -super5 SeqDatabase_6.fasta -output SeqDatabase_6.aln.fasta
```

After executing this shell script, using the following command in terminal:

```
% source Def2Meta.sh
```

Now that we have generated the necessary files for data analysis, we can move onto creating a PCoA analysis of the database and create an ML tree. PCoA plots can be created using the ape, phangorn, and kmer packages. The following code will use the aligned SeqDatabase_6 to create a PCoA plot of the Database. The following code can be used in an R script to generate a PCoA plot. We use a distance of 5 for this particular portion of the clustering analysis because we have previously determined that for k>5 there are diminishing returns on the clustering of the data points.

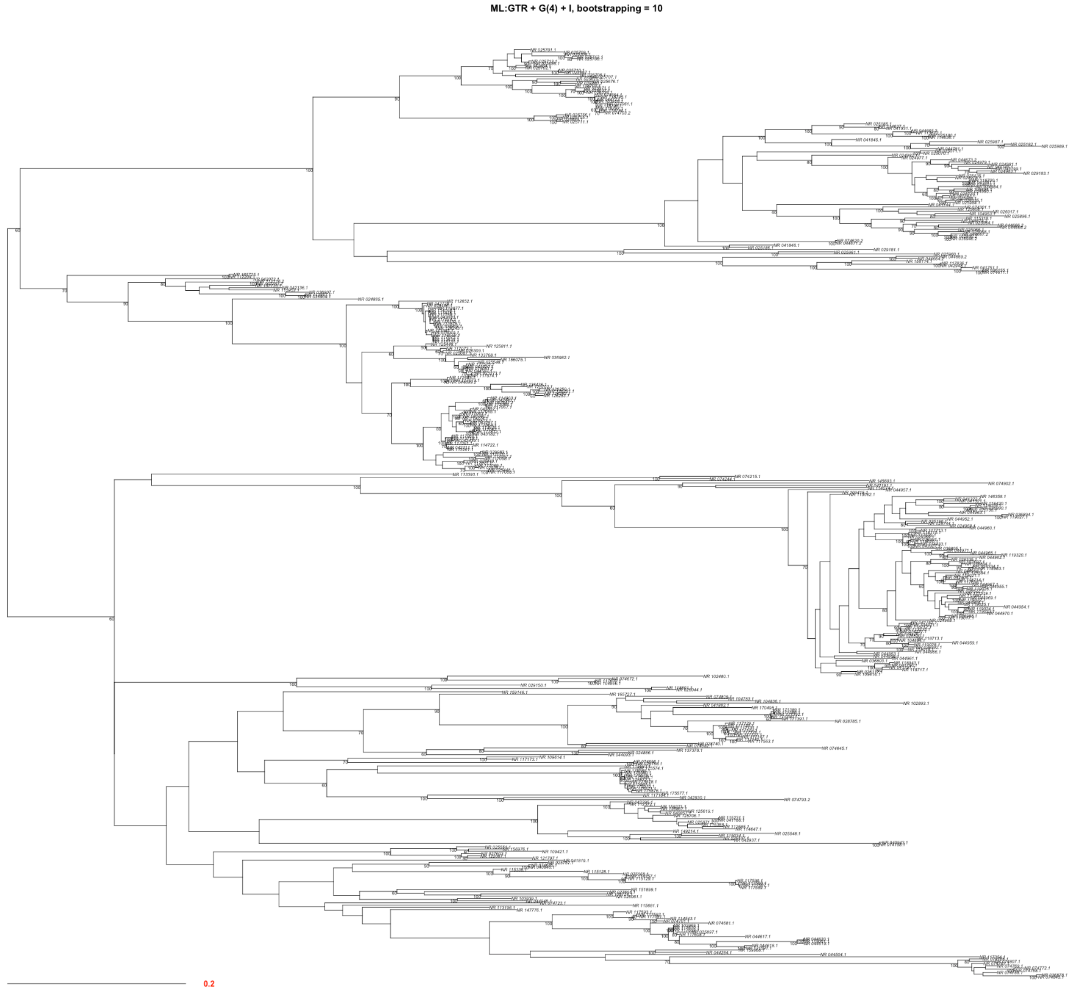
```
#imports necessary libraries
library(ape)
library(phangorn)
library(ape)
#creates a panel of 1
layout(matrix(c(1))); layout.show(1);
db6 = read.FASTA("SeqDatabase_6.aln.fasta")
#creates the kdistance matrix for k=5
kd4.5 <- kdistance(db6, 5)
#plots the pcoa graph
plot(pcoa(kd4.5)$vectors[,1:2],main="PCOA of SeqDatabase_6.aln.fasta kdistance
k=5");
```



PCoA of Aligned SeqDatabase_6 using kmer package(k=5)

It is obvious from the above plot that we appear to have several clusters. This agrees with the hypothesis that concatenating several different 16S databases together would create the same number of clusters on the PCoA.

Now that we have created a PCoA plot of the aligned sequence database we would like to create a ML tree for our dataset. From the SeqDatabase_5.aln.fasta file we can create an ML tree or a Maximum Likelihood. Maximum Likelihood methods, at least in terms of generating a tree, can best be described as “maximizing” the likelihood function and finding those specific parameters for creating the maximum likelihood. After generating a ML tree we now have to optimize these ML trees. This can be done using various methods/models. In our case we will use the GTR +G(4) + I model to generate a ML tree, using 10 bootstraps and midpoint rooting. Bootstrapping is a statistical tool/process that involves using resampling. Bootstrapping takes samples and assigns measures of accuracy to those samples. Midpoint rooting calculates tip to tip root distances from the two longest “tips” and calculates the halfway point and “roots” there. The Rscript code for doing so is below:



Optimized ML trees for SeqDatabase_6, bs=10

Now that we have generated both an ML tree and done clustering analysis using PCoA plots, we would like to add clustering centers to the metadata table. This can be done in various ways, but we will use the clustering centers generated from cd-hit using a threshold of 99, which was determined in the previous report to return the highest number of clustering centers. Therefore we need to run the following command to generate a cluster file using cd-hit in the terminal:

```
#activates the conda environment
```

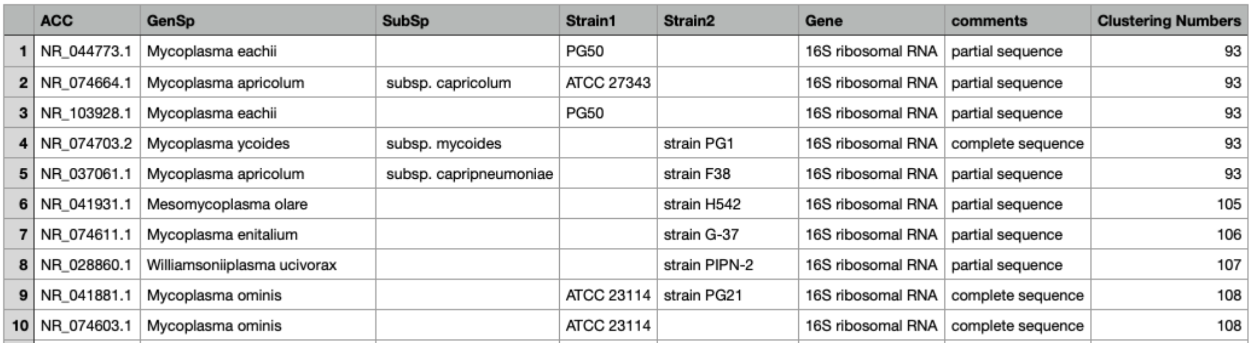
```
%conda activate my_envx86
#does the cd-hit clustering
%cd-hit -i SeqDatabase_6.fasta -o tmp.99
```

After generating the clustering information we would like to add those cluster numbers to the metadata file. This can be done in a couple different ways, but using R we can add this information using a relatively short R script:

```
#####################################################################################################
#sets the working directory to the MA2 Group Folder
setwd("~/Desktop/School/MCB432_folder/MA2Group")
#Reads in the metadata table from producing SeqDatabase_6.fasta
meta = read.csv("SeqData6_meta.csv")
#reads in the clustering data from cd-hit, using line breaks a as a delimiter
clstr = read.table("tmp.99.clstr",sep='\n',header=F)
#converts the table file to a dataframe
clstr = data.frame(clstr)

#sets the column name to clustering center text
colnames(clstr) = "Clustering Center Text"

#initializes a counting variable and an empty list
c = 0;
listAccCent = (NULL);
#loops through all row numbers
for(k in 1:nrow(clstr)){
  #looks for rows that start with a >, this looks for the clustering center #
  if(substr(clstr[k,1],0,1) == ">") {
          #adds one to the counter variable
          c = c+1
  }
  #looks to see if this line conttains a substring ">NR"
  if(grepl(">NR",clstr[k,1],fixed = TRUE)){
          #initializes an empty integer variable
          ind = 0
          #searches through the line looking for substring of size 3 that is >NR
          for(s in 1:nchar(clstr[k,1])-3){
          #prints the substring
          print(substr(clstr[k,1],s,s+2))

          if( substr(clstr[k,1],s,s+2) == ">NR"){
          #sets the index variable equal to the index of wher the >NR starts
```

```
        ind = s
        }
        }
        #prints the index
        print(ind)
        #adds the substring of the line, the accession number to a list with the
        #clustering center number
        a=list(c, substr(clstr[k,1],ind+1,ind+11))
        listAccCent = c(listAccCent,list(a))
  }

}
#initializes an empty column in the metadata table
meta[,8] = 0
#sets column name equal to "Clustering Numbers"
names(meta)[names(meta)=="V8"] <- "Clustering Numbers"


#loops through each object in the listAccCent
for(j in listAccCent){
  #loops through all rows in meta
  for(i in 1:nrow(meta)){
        print(substr(meta$ACC[[i]],1,11))
        print(j[[2]])
        #checks if substrings match
        if(substr(meta$ACC[[i]],1,11)==j[[2]]){
        #writes the clustering number to its appropriate row
        meta$`Clustering Numbers`[[i]] = j[[1]]
        }
  }

}

write.csv(meta,"metadata_Seq6.csv")
```

A quick snapshot of the csv with clustering information can be seen below:



	ACC	GenSp	SubSp	Strain1	Strain2	Gene	comments	Clustering Numbers
1	NR_044773.1	Mycoplasma eachii		PG50		16S ribosomal RNA	partial sequence	93
2	NR_074664.1	Mycoplasma apricolum	subsp. capricolum	ATCC 27343		16S ribosomal RNA	partial sequence	93
3	NR_103928.1	Mycoplasma eachii		PG50		16S ribosomal RNA	partial sequence	93
4	NR_074703.2	Mycoplasma ycoides	subsp. mycoides		strain PG1	16S ribosomal RNA	complete sequence	93
5	NR_037061.1	Mycoplasma apricolum	subsp. capripneumoniae		strain F38	16S ribosomal RNA	partial sequence	93
6	NR_041931.1	Mesomycoplasma olare			strain H542	16S ribosomal RNA	partial sequence	105
7	NR_074611.1	Mycoplasma enitalium			strain G-37	16S ribosomal RNA	partial sequence	106
8	NR_028860.1	Williamsoniiplasma ucivorax			strain PIPN-2	16S ribosomal RNA	partial sequence	107
9	NR_041881.1	Mycoplasma ominis		ATCC 23114	strain PG21	16S ribosomal RNA	complete sequence	108
10	NR_074603.1	Mycoplasma ominis		ATCC 23114		16S ribosomal RNA	complete sequence	108

Now that we have compiled the metadata table for the whole database, completed both PCoA clustering analysis, and created an ML tree we would like to summarize the database. The database itself is approximately 414 sequences long, with several main clusters as illustrated by the PCoA. This makes sense as there should be several clusters close together due to the concatenation of the datasets. Additionally, we can see from the ML tree that there appear to be about 4-5 long branches that most likely occur due to the same reason. We are essentially taking different 16S RNA databases and concatenating them together, so we would expect to see about 4-5 larger clusters of data.