8. Worksheet: Phylogenetic Diversity - Traits

Erica Nadolski; Z620: Quantitative Biodiversity, Indiana University

22 February, 2023

OVERVIEW

Up to this point, we have been focusing on patterns taxonomic diversity in Quantitative Biodiversity. Although taxonomic diversity is an important dimension of biodiversity, it is often necessary to consider the evolutionary history or relatedness of species. The goal of this exercise is to introduce basic concepts of phylogenetic diversity.

After completing this exercise you will be able to:

1. create phylogenetic trees to view evolutionary relationships from sequence data
2. map functional traits onto phylogenetic trees to visualize the distribution of traits with respect to evolutionary history
3. test for phylogenetic signal within trait distributions and trait-based patterns of biodiversity

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo, at whatever stage you are. This will enable you to pull your work onto your own computer.
6. When you have completed the worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '8.BetaDiversity' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**11.PhyloTraits_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**11.PhyloTraits_Worksheet.pd**

The completed exercise is due on **Wednesday, February 22nd, 2023 before 12:00 PM (noon)**.

1) SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:
1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/8.PhyloTraits*" folder, and
4. load all of the required R packages (be sure to install if needed).

```
rm(list = ls())
getwd()
```

```
## [1] "/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits"
```

```
setwd("/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits")

package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger', 'picante', 'stats', 'RColorBrewer
for (package in package.list) {
  if (!require(package, character.only=TRUE, quietly=TRUE)) {
    install.packages(package)
    library(package, character.only=TRUE)
  }
}
```

```
##
## Attaching package: 'seqinr'

## The following objects are masked from 'package:ape':
##
##     as.alignment, consensus

##
## Attaching package: 'phylobase'

## The following object is masked from 'package:ape':
##
##     edges

##
## Attaching package: 'permute'

## The following object is masked from 'package:seqinr':
##
##     getType

## This is vegan 2.6-4

##
## Attaching package: 'nlme'

## The following object is masked from 'package:seqinr':
##
##     gls
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following object is masked from 'package:nlme':
##
##     collapse

## The following object is masked from 'package:seqinr':
##
##     count

## The following object is masked from 'package:ape':
##
##     where

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'phangorn'

## The following objects are masked from 'package:vegan':
##
##     diversity, treedist

##
## Attaching package: 'phylogram'

## The following object is masked from 'package:phylobase':
##
##     prune

## Registered S3 methods overwritten by 'dendextend':
##   method              from
##   as.dendrogram.phylo phylogram
##   rev.hclust          vegan

##
## ---------------------
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
```

```
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ----------------------

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:phylogram':
##
##     prune

## The following object is masked from 'package:permute':
##
##     shuffle

## The following object is masked from 'package:geiger':
##
##     is.phylo

## The following objects are masked from 'package:phylobase':
##
##     labels<-, prune

## The following objects are masked from 'package:ape':
##
##     ladderize, rotate

## The following object is masked from 'package:stats':
##
##     cutree
```

2) DESCRIPTION OF DATA

The maintenance of biodiversity is thought to be influenced by **trade-offs** among species in certain functional traits. One such trade-off involves the ability of a highly specialized species to perform exceptionally well on a particular resource compared to the performance of a generalist. In this exercise, we will take a phylogenetic approach to mapping phosphorus resource use onto a phylogenetic tree while testing for specialist-generalist trade-offs.

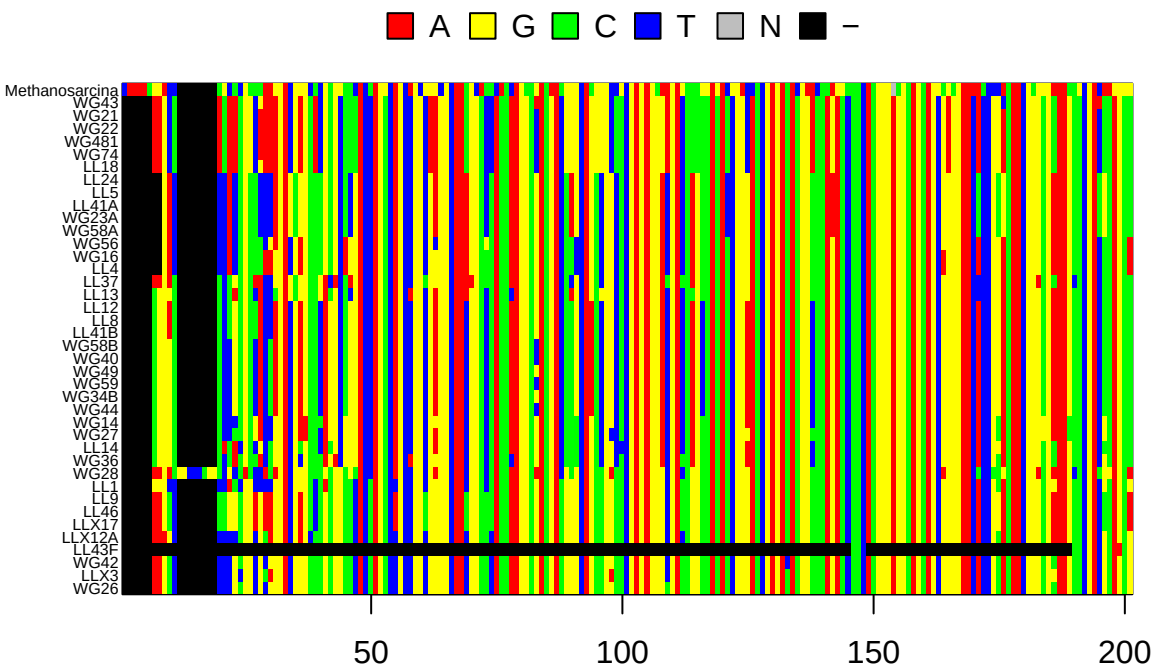
3) SEQUENCE ALIGNMENT

Question 1: Using your favorite text editor, compare the `p.isolates.fasta` file and the `p.isolates.afa` file. Describe the differences that you observe between the two files.

Answer 1: Both appear to be formatted in basic FASTA file formatting, but the .FASTA is the raw sequences for each isolate, and the .afa is the alignment sequences with alignment gaps added in so the entire set of sequences aligns best end to end.

In the R code chunk below, do the following: 1. read your alignment file, 2. convert the alignment to a DNAbin object, 3. select a region of the gene to visualize (try various regions), and 4. plot the alignment using a grid to visualize rows of sequences.

```
# import alignment file
p.align <- read.alignment("./data/p.isolates.afa", format="fasta")
# convert to DNAbin
p.DNAbin <- as.DNAbin(p.align)
# identify region of 16S rRNA gene to visualize
window <- p.DNAbin[,200:400]
# visualize alignment in rows of sequences
image.DNAbin(window, cex.lab=0.50)
```



Question 2: Make some observations about the `muscle` alignment of the 16S rRNA gene sequences for our bacterial isolates and the outgroup, *Methanosarcina*, a member of the domain Archaea. Move along the alignment by changing the values in the `window` object.

a. Approximately how long are our sequence reads?

b. What regions do you think would are appropriate for phylogenetic inference and why?

Answer 2a: The Methanosarcina sequence in >1400bp, but the isolate sequences range from approximately 600-800bp. ***Answer 2b***: The maximum region I would choose would be from bp 100-800, but it may improve the inference to use bp ~230-700 because this region is represented in nearly every isolate.

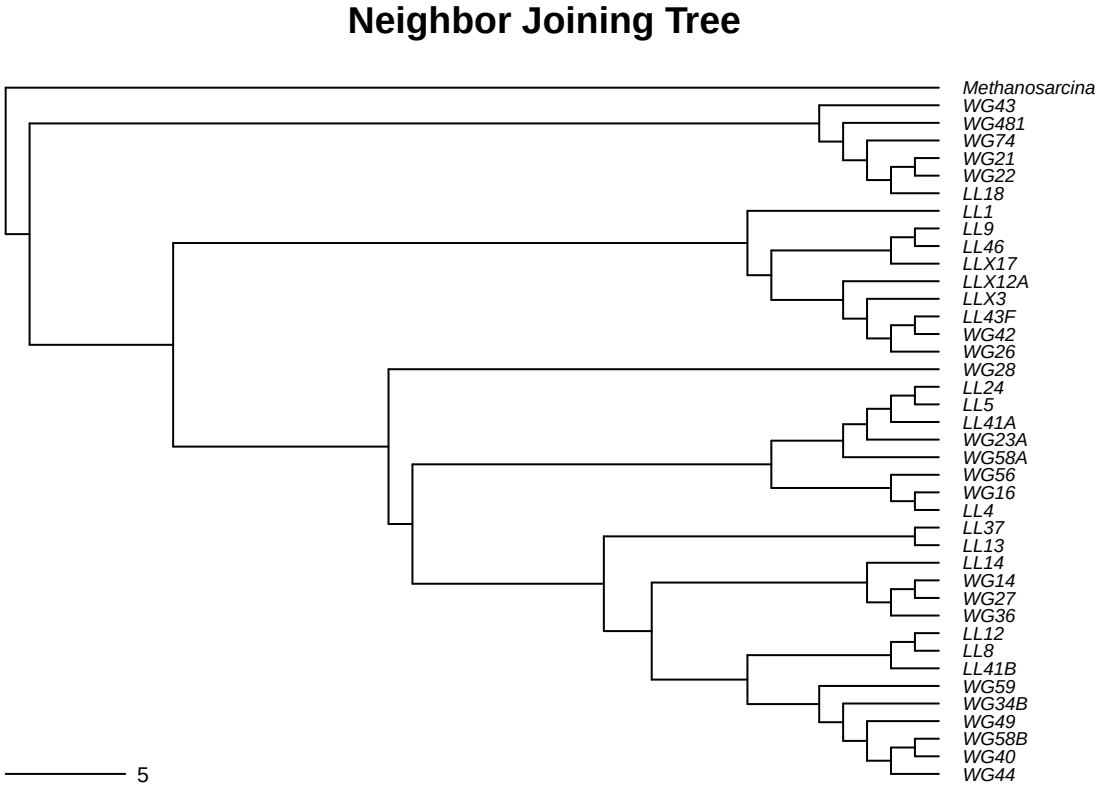
4) MAKING A PHYLOGENETIC TREE

Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

A. Neighbor Joining Trees

In the R code chunk below, do the following:
1. calculate the distance matrix using `model = "raw"`,
2. create a Neighbor Joining tree based on these distances,
3. define "Methanosarcina" as the outgroup and root the tree, and
4. plot the rooted tree.

```
# distance matrix with "raw" model
seq.dist.raw <- dist.dna(p.DNAbin, model="raw",pairwise.deletion=FALSE)
# neighbor joining algorithm using ape
nj.tree <- bionj(seq.dist.raw)
# identify outgroup sequence
outgroup <- match("Methanosarcina", nj.tree$tip.label)
# root the tree with outgroup
nj.rooted <- root(nj.tree,outgroup,resolve.root=TRUE)
# plot rooted tree
par(mar= c(1,1,2,1)+0.1)
plot.phylo(nj.rooted, main="Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, direction="ri
add.scale.bar(cex=0.7)
```



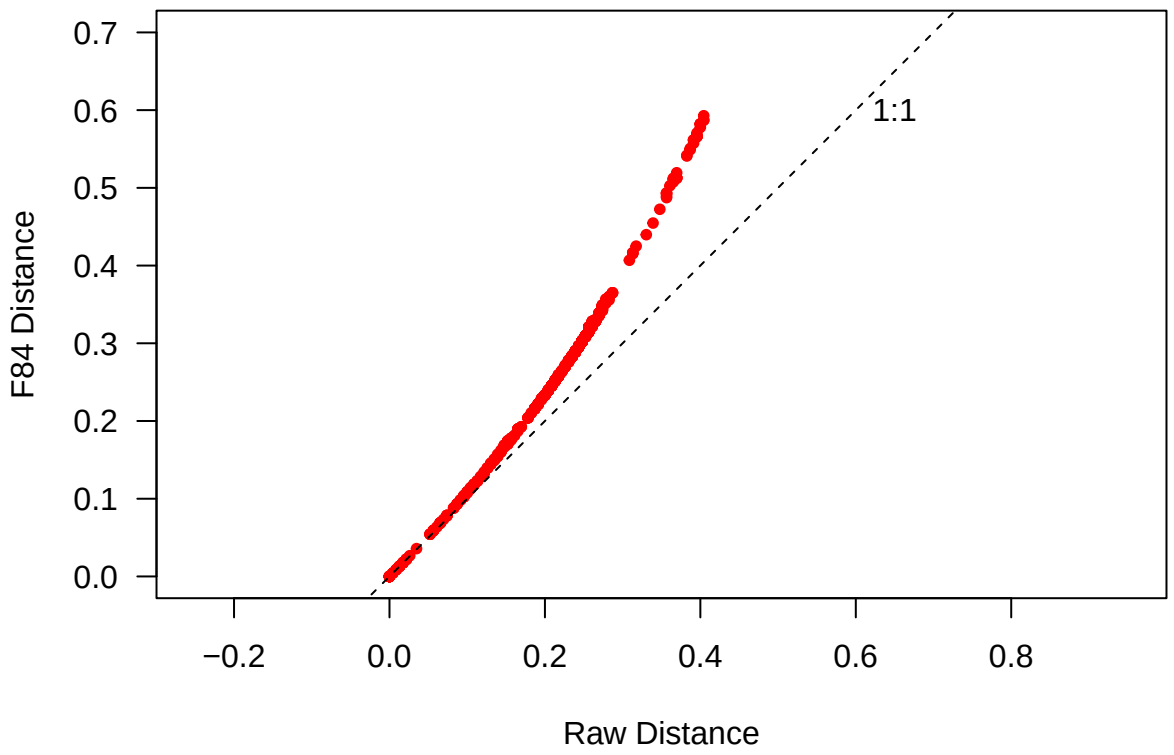
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

Answer 3: This method of phylogenetic tree creation is computationally easy and straightforward, but the assumption it makes or 'model of substitution' that it uses is overly simplistic i.e. it assumes all nucleotides and substitutions occur equally often.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,
3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare the topologies of the trees.

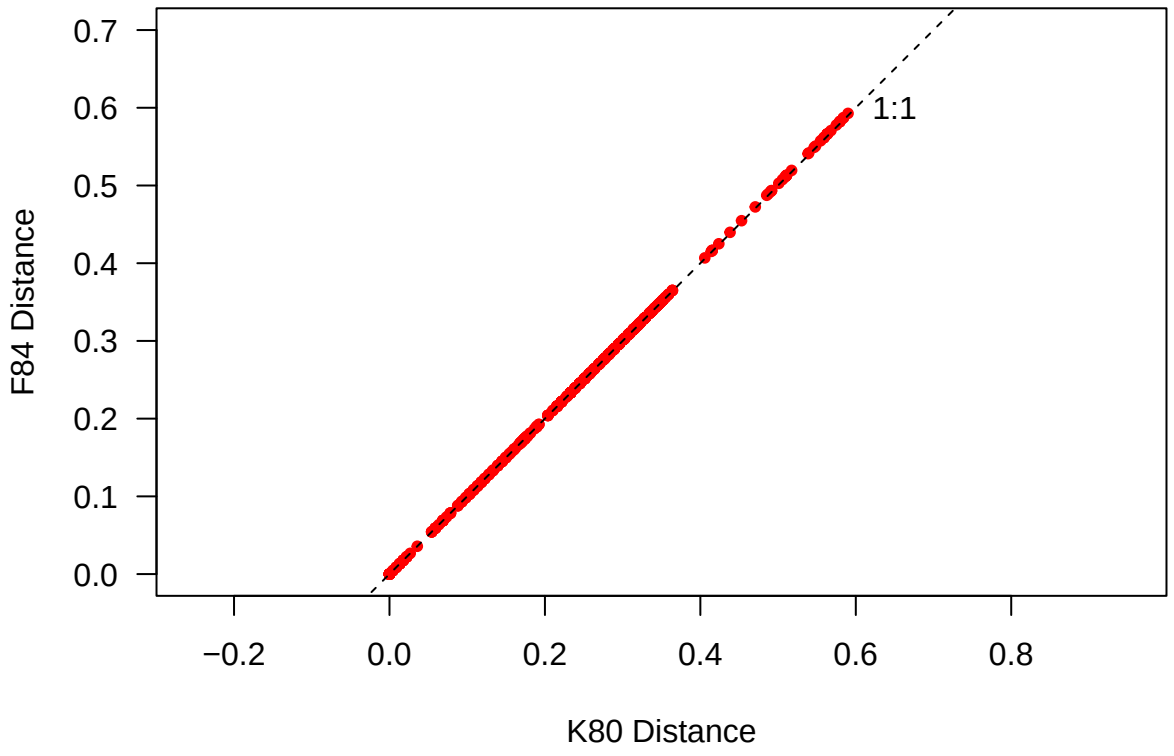
```
seq.dist.F84 <- dist.dna(p.DNAbin, model="F84", pairwise.deletion=FALSE)
par(mar= c(5,5,2,1)+0.1)
plot(seq.dist.raw, seq.dist.F84, pch=20, col="red", las=1, asp=1, xlim= c(0, 0.7),
     ylim = c(0, 0.7), xlab="Raw Distance", ylab="F84 Distance");
abline(b=1,a=0,lty=2);
text(0.65,0.6,"1:1")
```



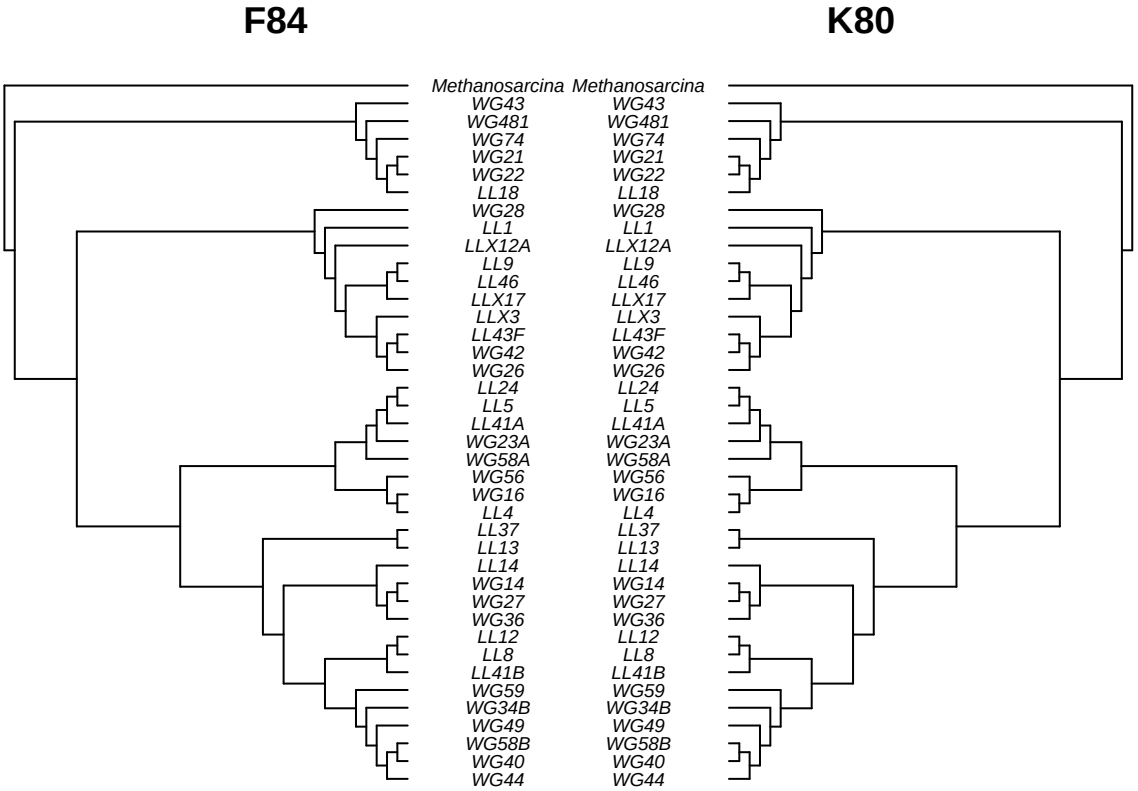

```
# F84 distance is greater than raw so it is correcting for multiple substitutions
```

In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5. be sure to format, add appropriate labels, and customize each plot.

```
# distance matrix for K80 model
seq.dist.K80 <- dist.dna(p.DNAbin, model="K80", pairwise.deletion=FALSE)
# trees for models
K80.tree <- bionj(seq.dist.K80)
F84.tree <- bionj(seq.dist.F84)
# root trees
K80.outgroup <- match("Methanosarcina", K80.tree$tip.label)
K80.rooted <- root(K80.tree, K80.outgroup, resolve.root=TRUE)

F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root=TRUE)

# make saturation plot
par(mar= c(5,5,2,1)+0.1)
plot(seq.dist.K80, seq.dist.F84, pch=20, col="red", las=1, asp=1, xlim= c(0, 0.7),
     ylim = c(0, 0.7), xlab="K80 Distance", ylab="F84 Distance");
abline(b=1,a=0,lty=2);
text(0.65,0.6,"1:1")
```




```
# make co-phylogenetic plot using ape
layout(matrix(c(1,2), 1, 2), width=c(1,1))
par(mar= c(1,1,2,0))
plot.phylo(F84.rooted, type="phylogram", show.tip.label=TRUE, use.edge.length = FALSE,
           direction="right", adj=0.5, cex=0.6, label.offset=2, main="F84")
par(mar=c(1,0,2,1))
plot.phylo(K80.rooted, type="phylogram", show.tip.label=TRUE, use.edge.length = FALSE,
           direction="left", adj=0.5, cex=0.6, label.offset=2, main="K80")
```



Question 4:

a. Describe the substitution model that you chose. What assumptions does it make and how does it compare to the F84 model?
b. Using the saturation plot and cophylogenetic plots from above, describe how your choice of substitution model affects your phylogenetic reconstruction. If the plots are inconsistent with one another, explain why.
c. How does your model compare to the *F84* model and what does this tell you about the substitution rates of nucleotide transitions?

Answer 4a: The Felsenstein model assumes that rates of base transition and transversion are different and also allows for different base frequencies. The Kimura model also assumes that rates of base transition and transversion are different (transitions are more probable) but assumes nucleotide frequencies are equal. ***Answer 4b***: For these data, the models generate highly concordant results. ***Answer 4c***: Although the Kimura model assumes equal frequency of nucleotides while the Felsenstein model does not, they both assume different rates of transitions vs. transversions which appears important for the phylogenetic analysis of these data.

C) ANALYZING A MAXIMUM LIKELIHOOD TREE

In the R code chunk below, do the following:
1. Read in the maximum likelihood phylogenetic tree used in the handout. 2. Plot bootstrap support values onto the tree

```
ml.bootstrap <- read.tree("/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits/data/r
dev.off()
```

```
## null device
##           1
```

```
par(mar=c(1,1,2,1)+0.1)
plot.phylo(ml.bootstrap, type="phylogram", direction="right",
           show.tip.label = TRUE, use.edge.length = FALSE, cex=0.6,
           label.offset = 1, main = "Maximum Likelihood with Support Values");
add.scale.bar(cex=0.7);
nodelabels(ml.bootstrap$node.label, font=1, bg="white",frame="r",cex=0.5)
```

Question 5:

a) How does the maximum likelihood tree compare the to the neighbor-joining tree in the handout? If the plots seem to be inconsistent with one another, explain what gives rise to the differences.

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches?

Answer 5a: The plots display largely different phylogenetic relationships. Notably, the ML tree shows at the first node a bifurcation of the isolate taxa into two large branches, while the NJ tree shows more iterative branching of smaller groups of isolates. The differences are because 1) neighbor joining is highly sensitive to the distance matrix and we do not know the real correct distance matrix here and 2) neighbor joining is also sensitive to the model of nucleotide substitution while ML methods are not. ***Answer 5b***: Bootstrapping allows a statistical test of the resulting ML tree because it involves re-sampling the data to see how robust the nodes of the tree are. ***Answer 5c***: Bootstrap values indicate how robust each node was to re-sampling (whether and how often the resampling changed the node) which is used to infer correctness of the relationship at that node. ***Answer 5d***: There are nodes with very low support close to the tree tips for WG42 and LL43F, as well as some close to the base of the tree. ***Answer 5e***: These branches should not be assumed to be correct because these data could not resolve those nodes through bootstrapping.

5) INTEGRATING TRAITS AND PHYLOGENY

A. Loading Trait Database

In the R code chunk below, do the following:
1. import the raw phosphorus growth data, and
2. standardize the data for each strain by the sum of growth rates.

```
p.growth <- read.table("/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits/data/p.is
p.growth.std <- p.growth/ (apply(p.growth, 1, sum))
```

B. Trait Manipulations

In the R code chunk below, do the following:
1. calculate the maximum growth rate (μ_{max}) of each isolate across all phosphorus types,
2. create a function that calculates niche breadth (nb), and
3. use this function to calculate nb for each isolate.

```r
# calculate max growth rate
umax <- (apply(p.growth, 1, max))

# function to calculate nb
niche.breadth <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
  }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}
# calculate nb
nb <- as.matrix(niche.breadth(p.growth.std))

# add isolate names to nb matrix
nb <- setNames(as.vector(nb), as.matrix(row.names(p.growth)))
```

C. Visualizing Traits on Trees

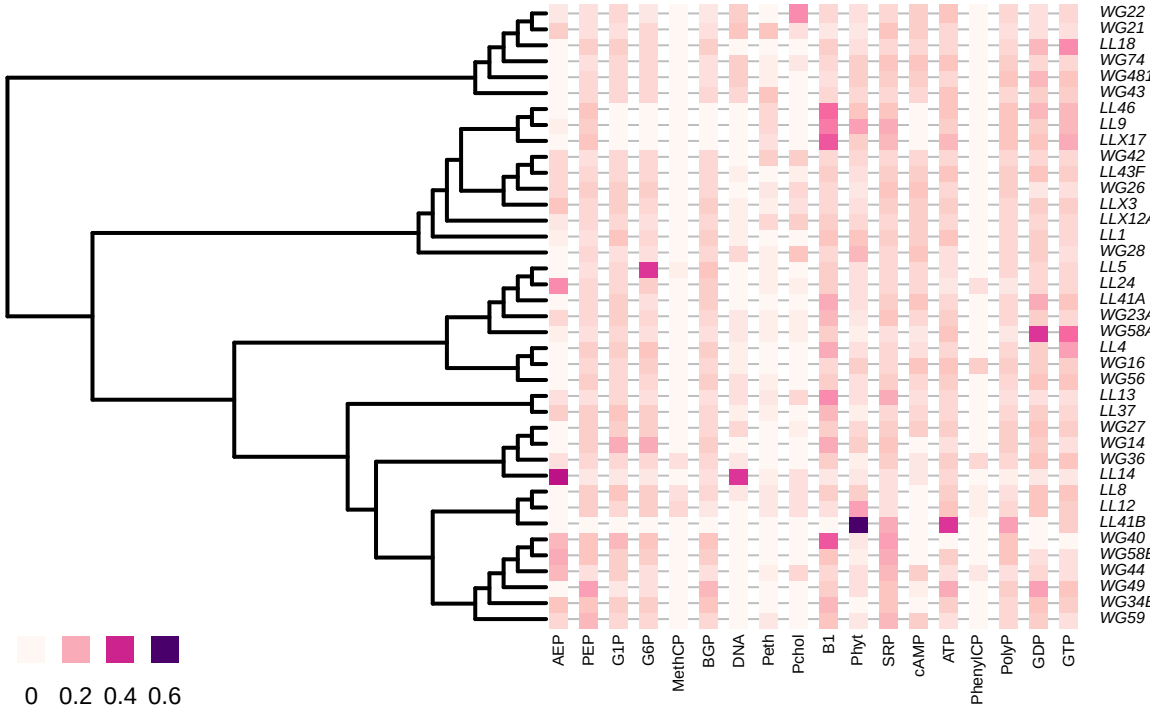
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```r
# generate NJ tree
F84tree <- bionj(seq.dist.F84)
# define outgroup
F84outgroup <- match("Methanosarcina", F84tree$tip.label)
# root tree
F84rooted <- root(F84tree, F84outgroup, resolve.root=TRUE)
# keep rooted but drop outgroup branch
F84rooted <- drop.tip(F84rooted, "Methanosarcina")
# plot(F84rooted) # plot to look at tree
```

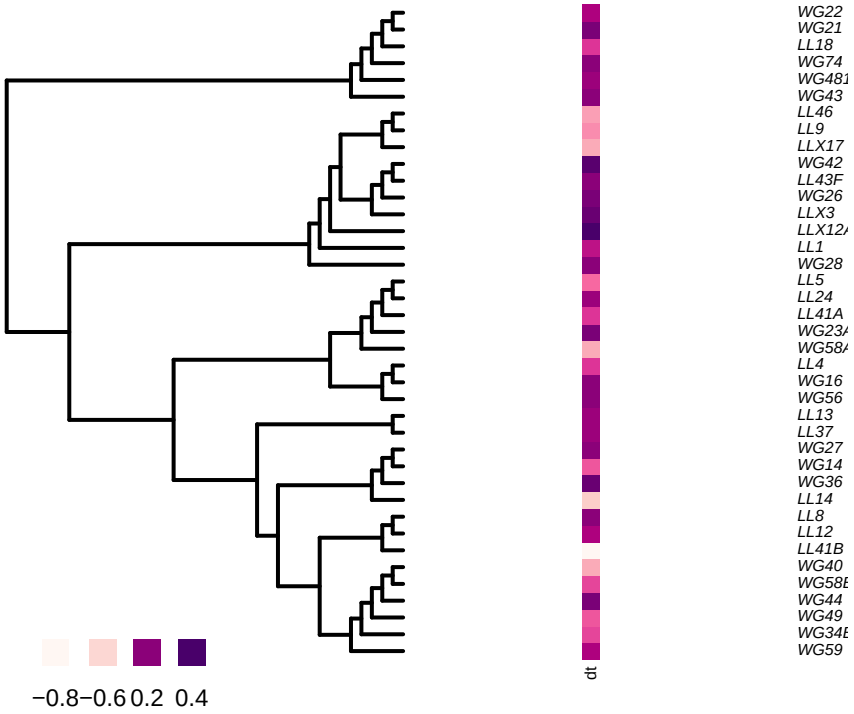
In the R code chunk below, do the following:
1. define a color palette (use something other than "YlOrRd"),
2. map the phosphorus traits onto your phylogeny,
3. map the *nb* trait on to your phylogeny, and
4. customize the plots as desired (use `help(table.phylo4d)` to learn about the options).

```r
# define color palette
mypalette <- colorRampPalette(brewer.pal(9, "RdPu"))
# correct for zero length branches on tree
njplot <- F84rooted
njplot$edge.length <- njplot$edge.length + 10^-1

# map phosphorus traits
par(mar=c(1,1,1,1)+0.1)
x <- phylo4d(njplot, p.growth.std)
table.phylo4d(x, treetype="phylo", symbol="colors", show.node=TRUE,
              cex.label=0.5, scale=FALSE, use.edge.length=FALSE,
```

```
              edge.color="black", edge.width=2, box=FALSE,
              col=mypalette(25), pch=15, cex.symbol=1.25,
              ratio.tree=0.5, cex.legend=1.5, center=FALSE)
```




```
# niche breadth
par(mar=c(1,5,1,5)+0.1)
x.nb <- phylo4d(njplot,nb)
table.phylo4d(x.nb, treetype="phylo", symbol="colors", show.node=TRUE,
              cex.label=0.5, scale=FALSE, use.edge.length=FALSE,
              edge.color="black", edge.width=2, box=FALSE,
              col=mypalette(25), pch=15, cex.symbol=1.25,
              ratio.tree=0.5, cex.legend=1.5, center=TRUE)
```



Question 6:

a) Make a hypothesis that would support a generalist-specialist trade-off.

b) What kind of patterns would you expect to see from growth rate and niche breadth values that would support this hypothesis?

Answer 6a: I hypothesize that these lake communities are highly competitive over phosphorus resources, which has led some isolates to evolve to specialize on one or a few resources and benefit from high relative growth rates only on those resources, while other isolates have evolved as generalists which have moderate growth rates on a variety of resources. ***Answer 6b***: I would expect to see high variance in growth rates and niche breadth across the phylogeny, with the species exhibiting the highest growth rates overall only in the presence of one or a few resources, with other species exhibiting growth rates closer to the mean for all resources.

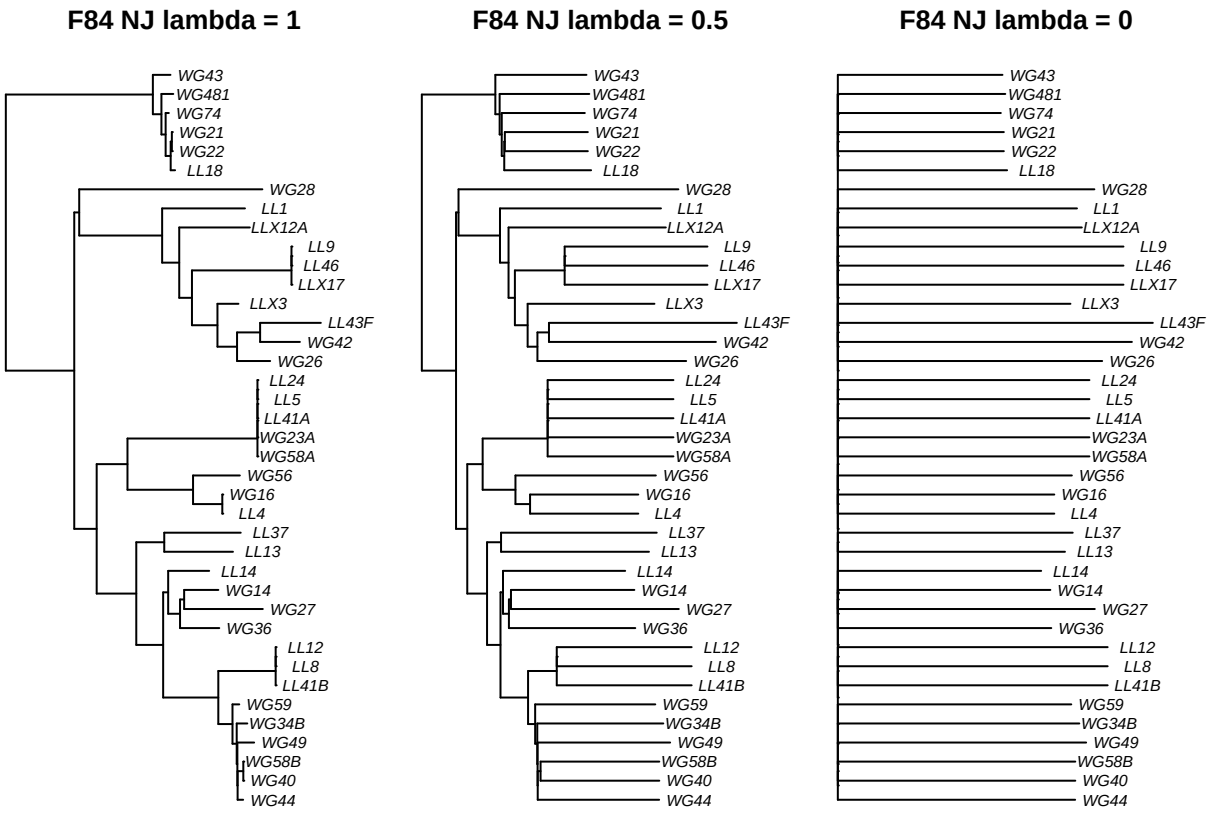
6) HYPOTHESIS TESTING

A) Phylogenetic Signal: Pagel's Lambda

In the R code chunk below, do the following:
1. create two rescaled phylogenetic trees using lambda values of 0.5 and 0,
2. plot your original tree and the two scaled trees, and
3. label and customize the trees as desired.

```
# visualize trees with different level of phylogenetic signal {geiger}
nj.lambda.5 <- geiger::rescale(F84rooted, "lambda", 0.5)
nj.lambda0 <- geiger::rescale(F84rooted, "lambda", 0)
layout(matrix(c(1,2,3),1,3), width=c(1,1,1))
par(mar=c(1,0.5,2,0.5) +0.1)
plot(F84rooted, main="F84 NJ lambda = 1", cex=0.7, adj =0.5)
```

```
plot(nj.lambda.5, main="F84 NJ lambda = 0.5", cex=0.7, adj =0.5)
plot(nj.lambda0, main="F84 NJ lambda = 0", cex=0.7, adj =0.5)
```



In the R code chunk below, do the following:
1. use the `fitContinuous()` function to compare your original tree to the transformed trees.

```
# generate test statistics for comparing phylogenetic signal {geiger}
fitContinuous(F84rooted, nb, model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.020847
##  sigsq = 0.106492
##  z0 = 0.661368
##
##  model summary:
##  log-likelihood = 21.661104
##  AIC = -37.322208
##  AICc = -36.636494
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 58
##  number of iterations with same best fit = NA
##  frequency of best fit = NA
##
```

```
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
fitContinuous(nj.lambda0, nb, model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.106395
##  z0 = 0.657777
##
##  model summary:
##  log-likelihood = 21.652293
##  AIC = -37.304587
##  AICc = -36.618872
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 86
##  frequency of best fit = 0.86
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

Question 7: There are two important outputs from the `fitContinuous()` function that can help you interpret the phylogenetic signal in trait data sets. a. Compare the lambda values of the untransformed tree to the transformed (lambda = 0). b. Compare the Akaike information criterion (AIC) scores of the two models. Which model would you choose based off of AIC score (remember the criteria that the difference in AIC values has to be at least 2)? c. Does this result suggest that there's phylogenetic signal?

> ***Answer 7a***: lambda for the untransformed tree was 0.020847, very close to zero. ***Answer 7b***: The AICs were -37.322208 for the transformed model and -37.304587 for the untransformed model; since they are different by <2 the models are equivalent, so I would choose the untransformed model. ***Answer 7c***: There is minimal phylogenetic signal in this trait-tree dataset, meaning that the similarity we see in traits is not due exclusively to phylogenetic relatedness.

B) Phylogenetic Signal: Blomberg's K

In the R code chunk below, do the following:
1. correct tree branch-lengths to fix any zeros,
2. calculate Blomberg's K for each phosphorus resource using the `phylosignal()` function,
3. use the Benjamini-Hochberg method to correct for false discovery rate, and
4. calculate Blomberg's K for niche breadth using the `phylosignal()` function.

```
# correct for zero length branches
F84rooted$edge.length <- F84rooted$edge.length + 10^-7

### calculate Blomberg's K for each phos. resource
# first create a blank output matrix
p.phylosignal <- matrix(NA,6,18)
colnames(p.phylosignal) <- colnames(p.growth.std)
rownames(p.phylosignal) <- c("K","PIC.var.obs","PIC.var.mean","PIC.var.P","PIC.var.z",
                             "PIC.P.BH")
# for loop to calculate Blombergs K for each resource
for (i in 1:18){
  x <- setNames(as.vector(p.growth.std[,i]), row.names(p.growth))
  out <- phylosignal(x, F84rooted)
  p.phylosignal[1:5,i] <- round(t(out),3)
}

# BH correction on p values
p.phylosignal[6,] <- round(p.adjust(p.phylosignal[4,], method="BH"), 3)
```

Question 8: Using the K-values and associated p-values (i.e., "PIC.var.P"") from the `phylosignal` output, answer the following questions:

a. Is there significant phylogenetic signal for niche breadth or standardized growth on any of the phosphorus resources?

b. If there is significant phylogenetic signal, are the results suggestive of clustering or overdispersion?

Answer 8a: There is significant phylogenetic signal (corrected P < 0.05) for growth on two of the phosphorus resources: DNA and cAMP. ***Answer 8b***: I am still not sure how to interpret this table. Based on my reading of the handout, K can be interpreted by its value relative to 1, but all of the K values generated by this code are zeroes. So, perhaps this means that all of the traits are overdispersed and less similar than expected by chance?

C. Calculate Dispersion of a Trait

In the R code chunk below, do the following:
1. turn the continuous growth data into categorical data,
2. add a column to the data with the isolate name,
3. combine the tree and trait data using the `comparative.data()` function in `caper`, and
4. use `phylo.d()` to calculate D on at least three phosphorus traits.

```
# generate categorical data from continuous
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)
# look at phos. use for each resource
apply(p.growth.pa, 2, sum)
```

```
##      AEP      PEP      G1P      G6P   MethCP      BGP      DNA     Peth
##       20       38       35       34        3       35       19       21
##    Pchol       B1     Phyt      SRP     cAMP      ATP PhenylCP    PolyP
##       18       38       36       39       29       38        6       39
##      GDP      GTP
##       37       38
```

```r
# add names column
p.growth.pa$name <- rownames(p.growth.pa)
# merge trait and phylo sata, run phylo.d
p.traits <- comparative.data(F84rooted, p.growth.pa, "name")
phylo.d(p.traits, binvar= MethCP, permut=10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  MethCP
##   Counts of states:  0 = 36
##                      1 = 3
##   Phylogeny :  F84rooted
##   Number of permutations :  10000
##
## Estimated D :  -0.2381232
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0113
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.6557
```

```r
phylo.d(p.traits, binvar= Phyt, permut=10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  Phyt
##   Counts of states:  0 = 3
##                      1 = 36
##   Phylogeny :  F84rooted
##   Number of permutations :  10000
##
## Estimated D :  0.3022086
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0477
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.4084
```

```r
phylo.d(p.traits, binvar= GTP, permut=10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  GTP
##   Counts of states:  0 = 1
##                      1 = 38
##   Phylogeny :  F84rooted
##   Number of permutations :  10000
##
## Estimated D :  4.199123
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.8449
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.072
```

Question 9: Using the estimates for D and the probabilities of each phylogenetic model, answer the following questions:

a. Choose three phosphorus growth traits and test whether they are significantly clustered or overdispersed?

b. How do these results compare the results from the Blomberg's K analysis?

c. Discuss what factors might give rise to differences between the metrics.

> ***Answer 9a***: D is -0.22 for growth on MethCP, 0.30 for growth on Phyt, and 4.40 for growth on GTP. The GTP growth trait is highly overdispersed, the MethCP growth trait is slightly clustered, and the Phyt growth trait is slightly overdispersed. ***Answer 9b***: These results match the overall pattern of Blomberg's K indicating overdispersion. ***Answer 9c***: I am still confused on Blomberg's K so I am not sure!

D. Correspondence between trait clusters and phylogeny

In the R code chunk below, do the following: 1. calculate Jaccard Index on resource use incidence matrix 2. create a hierarchical cluster of resource use 3. map the resource use cluster onto the phylogeny for each environment, and 4. use RF.dist and mantel to measure the degree of correspondence between each dendrogram.

```
# calculate Jaccard dissimilarity
no <- vegdist(p.growth.pa[,1:18], method='jaccard', binary=TRUE)
# test clustering that best fits data
# define linkage methods
m <- c("average","single","complete","ward")
names(m) <- c("average","single","complete","ward")

# function to compute agglomerative coefficient
ac <- function(x){
  cluster::agnes(no, method = x)$ac
}

# calculate agglomerative coefficient for each clustering linkage method
# use the method with the highest coefficient, indicates better fit
sapply(m, ac)
```

```
##   average    single  complete      ward
## 0.9064731 0.8881997 0.9207206 0.9470011
```

```
# generate hierarchical cluster
no.tree <- hclust(no, method="ward.D2")
dev.off()
```

```
## null device
##           1
```

```
plot(no.tree)

# compare topology between phylogeny and niche overlap with a tanglegram
# requires ultrametirc tree or dendrogram
# branch lengths are equidistant from the root node
is.ultrametric(F84.rooted)
```

```
## [1] FALSE
```

```
# visualize differences between each lake
# drop tips from each dendrogram to plot tips that come from a single lake
LL.tree <- drop.tip(F84rooted, c(F84rooted$tip.label[grepl("WG",F84rooted$tip.label)]))
LL.function <- drop.tip(as.phylo(no.tree), c(no.tree$labels[grepl("WG", no.tree$labels)]))
WG.tree <- drop.tip(F84rooted, c(F84rooted$tip.label[grepl("LL",F84rooted$tip.label)]))
WG.function <- drop.tip(as.phylo(no.tree), c(no.tree$labels[grepl("LL", no.tree$labels)]))

# plot each dendrogram and link their tips
# an untangling algorithm is used to maximize alignment
# for tanglegram visualization, highlight matches between two dendrograms
# resource use similarity on right; phylogeny on left

par(mar = c(1,5,1,4)+0.1)
dendlist(as.cladogram(as.dendrogram.phylo(LL.tree)),
         as.cladogram(as.dendrogram.phylo(LL.function))) %>%
  untangle(method="step2side") %>% # find best alignment layout
  tanglegram(common_subtrees_color_branches=TRUE,
             highlight_distinct_edges=FALSE, highlight_branches_lwd=FALSE,
             margin_inner=5) %>% # draw the two dendrograms
  entanglement() # score 0-1, closer to 0 is better alignment
```

```
## [1] 0.1124409
```

```
par(mar = c(1,5,1,4)+0.1)
dendlist(as.cladogram(as.dendrogram.phylo(WG.tree)),
         as.cladogram(as.dendrogram.phylo(WG.function))) %>%
  untangle(method="step2side") %>% # find best alignment layout
  tanglegram(common_subtrees_color_branches=TRUE,
             highlight_distinct_edges=FALSE, highlight_branches_lwd=FALSE,
             margin_inner=5) %>% # draw the two dendrograms
  entanglement()
```

```
## [1] 0.2682644
```

```
# measure degree of correspondence between each dendrogram
# 0 = complete congruence, 1 = no congruence
RF.dist(LL.tree, as.phylo(as.dendrogram(LL.function)), normalize=TRUE,
        check.labels=TRUE, rooted=FALSE)
```

```
## [1] 0.8
```

```
RF.dist(WG.tree, as.phylo(as.dendrogram(WG.function)), normalize=TRUE,
        check.labels=TRUE, rooted=FALSE)
```

```
## [1] 0.9444444
```

```
# mantel test to correlate patristic distance (pairwise sum of branch lengths)
# in phylogeny and jaccard index in the resource use hierarchical cluster
mantel(cophenetic.phylo(LL.tree), cophenetic.phylo(LL.function), method="spearman", permutations = 999)
```

```
##
## Mantel statistic based on Spearman's rank correlation rho
##
## Call:
## mantel(xdis = cophenetic.phylo(LL.tree), ydis = cophenetic.phylo(LL.function),      method = "spearma
##
## Mantel statistic r: 0.09132
##       Significance: 0.188
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.133 0.171 0.208 0.272
## Permutation: free
## Number of permutations: 999
```

```
mantel(cophenetic.phylo(WG.tree), cophenetic.phylo(WG.function), method="spearman", permutations = 999)
```

```
##
## Mantel statistic based on Spearman's rank correlation rho
##
## Call:
## mantel(xdis = cophenetic.phylo(WG.tree), ydis = cophenetic.phylo(WG.function),      method = "spearma
##
## Mantel statistic r: -0.05907
##       Significance: 0.681
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.146 0.185 0.223 0.282
## Permutation: free
## Number of permutations: 999
```

Question 10: Using a hierarchical clustering algorithm, map similarity in resource use map onto the phylogeny and answer the following questions: a. Compare the patterns between resource use and phylogeny between each lake. How do the two sets of tanglegrams differ between the taxa isolated from each lake? b. Interpret the Robinson-Fould index and Mantel correlation test results. How does each analysis differ and shape our interpretation of correlating niche overlap with phylogeny.

> ***Answer 10a***: Neither lake's tanglegram shows much congruence between resource use and phylogeny, although LL is slightly more congruent, with three more corresponding untangled lines.
> ***Answer 10b***: The RF index is also known as a symmetric difference metric, which calculates the distance between phylogenetic trees as the sum number of partitions of data implied by the

first tree but not the second tree and the number of partitions of data implied by the second tree but not the first tree, scaled from 0 to 1. In this analysis, numbers closer to 1 indicate higher incongruence; so both lakes have generally high incongruence (LL = 0.8, WG = 0.9444) between niche overlap and phylogenetic relatedness. The Mantel test is a correlation analysis that looks for correlations between elements of the data, so both lakes are generally uncorrelated (LL = 0.091, WG = -0.06), although WG shows slight anti-correlation that is not statistically significant.

7) PHYLOGENETIC REGRESSION

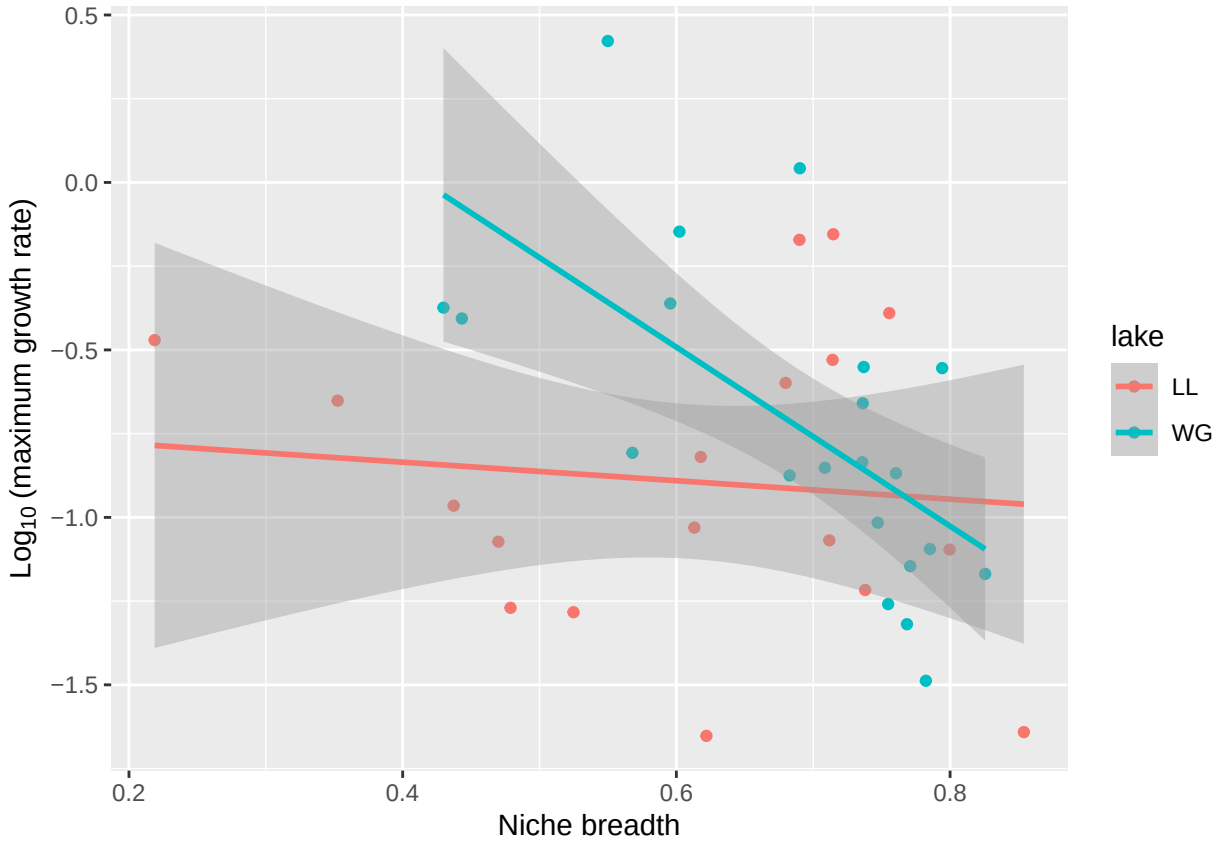
In the R code chunk below, do the following:
1. Clean the resource use dataset to perform a linear regression to test for differences in maximum growth rate by niche breadth and lake environment.

2. Fit a linear model to the trait dataset, examining the relationship between maximum growth rate by niche breadth and lake environment,

3. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny

```
# using niche breadth data create a lake origin column
nb.lake <- as.data.frame(as.matrix(nb))
nb.lake$lake =rep('A')

for(i in 1:nrow(nb.lake)){
  ifelse(grepl("WG",row.names(nb.lake)[i]), nb.lake[i,2] <- "WG",
                nb.lake[i,2] <- "LL")
}

# add niche breadth column name
colnames(nb.lake)[1] <- "NB"

# calculate max growth rate
umax <- as.matrix(apply(p.growth,1,max))
nb.lake = cbind(nb.lake,umax)

# plot maximum growth by niche breadth
ggplot(data = nb.lake, aes(x = NB, y = log10(umax), color=lake)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Niche breadth") +
  ylab(expression(Log[10]~"(maximum growth rate)"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
### simple linear regression
fit.lm <- lm(log10(umax) ~ NB*lake, data=nb.lake)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = log10(umax) ~ NB * lake, data = nb.lake)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7557 -0.3108 -0.1077  0.3102  0.7800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7247     0.3852  -1.882   0.0682 .
## NB           -0.2763     0.6097  -0.453   0.6533
## lakeWG        1.8364     0.6909   2.658   0.0118 *
## NB:lakeWG    -2.3958     1.0234  -2.341   0.0251 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.418 on 35 degrees of freedom
## Multiple R-squared:  0.2595, Adjusted R-squared:  0.196
## F-statistic: 4.089 on 3 and 35 DF,  p-value: 0.01371
```

```
AIC(fit.lm)
```

```
## [1] 48.413
```

```
### phylogeny-corrected regression with no bootstrap replicates
fit.plm <- phylolm(log10(umax) ~ NB*lake, data=nb.lake, F84rooted, model="lambda", boot=0)
summary(fit.plm)
```

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = F84rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  40.41 -14.21
##
## Raw residuals:
##     Min      1Q  Median      3Q     Max
## -0.7540 -0.1859 -0.0711  0.3285  0.9615
##
## Mean tip height: 0.1838433
## Parameter estimate(s) using ML:
## lambda : 0.4992841
## sigma2: 0.9153845
##
## Coefficients:
##              Estimate    StdErr t.value p.value
## (Intercept) -0.891228  0.371149 -2.4013 0.02179 *
## NB          -0.011367  0.520672 -0.0218 0.98271
## lakeWG       1.435117  0.574450  2.4982 0.01733 *
## NB:lakeWG   -1.958375  0.844383 -2.3193 0.02634 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1953    Adjusted R-squared: 0.1264
##
## Note: p-values and R-squared are conditional on lambda=0.4992841.
```

```
AIC(fit.plm)
```

```
## [1] 40.41178
```

Question 11 a. Why do we need to correct for shared evolutionary history? b. How does a phylogenetic regression differ from a standard linear regression? c. Interpret the slope and fit of each model. Did accounting for shared evolutionary history improve or worsen the fit? d. Try to come up with a scenario where the relationship between two variables would completely disappear when the underlying phylogeny is accounted for.

> ***Answer 11a***: Shared evolutionary history means that our data violate the assumption of independence that is required to perform a linear regression analysis, so if we do not correct for this, then the ability to interpret the regression is compromised by the fact that the regression could

just be picking up on this shared evolutionary history. ***Answer 11b***: In a phylogenetic regression, the variance of the residuals is described by a covariance matrix that takes the phylogeny (branch lengths) into account, rather than assuming the residual errors are due to independent randomly distributed variables. ***Answer 11c***: The adjusted R squared value actually decreased from 0.19 to 0.12 when phylogenetic relationships were taken into account meaning that niche breadth and max growth rate show less correlation when their shared evolutionary history is explicit in the model, but the AIC value decreased meaning that the phylogenetic linear regression better fits the data. ***Answer 11d***: We might be interesting in studying insect diets. We might measure growth rate of different species when offered different food sources. We could perform a linear regression of growth rate over food particle size, but any correlated relationship would likely disappear when evolutionary history is taken into account because insect mouthpart morphology is generally highly conserved across species and this could explain correlation in the ease of eating various food sources.

7) SYNTHESIS

Work with members of your Team Project to obtain reference sequences for taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. In almost all cases, researchers must deposit their sequences in places like NCBI before a paper is published. Those sequences are checked by NCBI employees for aspects of quality and given an **accession number**. You can use the NCBI program nucleotide **BLAST** to find out more about information associated with the isolate, in addition to getting its DNA sequence: https://blast.ncbi.nlm.nih.gov/. Alternatively, you can use the `read.GenBank()` function in the `ape` package to connect to NCBI and directly get the sequence. But before your team proceeds, you need to give some thought to which gene you want to focus on.For microorganisms like the bacteria we worked with above, many people use the ribosomal gene (i.e., 16S rRNA). This has many desirable features, including it is relatively long, highly conserved, and identifies taxa with reasonable resolution. In eukaryotes, ribosomal genes (i.e., 18S) are good for distinguishing coarse taxonomic resolution (i.e. class level), but it is not so good at resolving genera or species.Therefore, you may need to find another gene to work with, which might include protein-coding gene like cytochrome oxidase (COI) which is on mitochondria and is commonly used in molecular systematics. In plants, the ribulose-bisphosphate carboxylase gene (*rbcL*), which on the chloroplast, is commonly used. Also, non-protein-encoding sequences like those found in **Internal Transcribed Spacer (ITS)** regions between the small and large subunits of of the ribosomal RNA are good for molecular phylogenies. With your team members, do some research and identify a good candidate gene.

1. Download sequences and create a properly formatted fasta file.
2. Align the sequences and confirm that you have a good alignment.
3. Choose a substitution model and make a tree of your choice.
4. Based on the decisions above and the output, does your tree jibe with what is known about the evolutionary history of your organisms? If not, why? Is there anything you could do differently that would improve your tree, especially with regard to future analyses done by your team?

```
# set up packages
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("msa")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.19), R 4.2.2 (2022-10-31)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'msa'
```

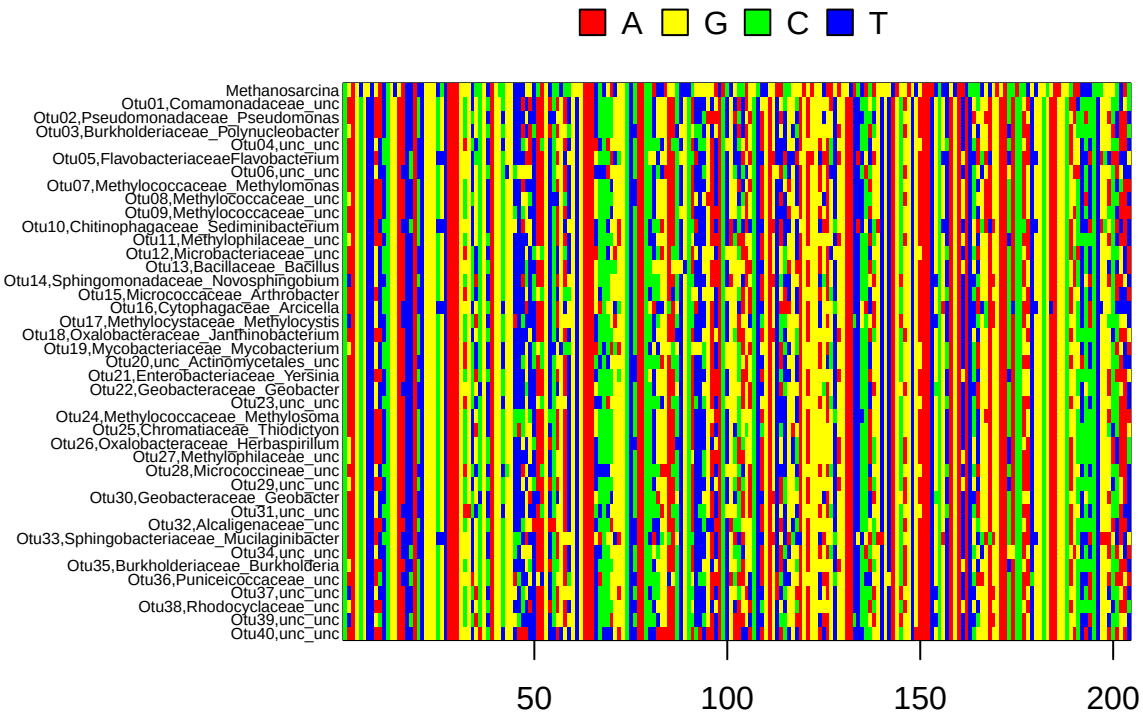
```
library(msa)
```

```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following object is masked from 'package:ade4':
##
##     score

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:tidyr':
##
##     expand

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:nlme':
##
##     collapse

## Loading required package: XVector

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'

## Also defined by 'S4Vectors'

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:dendextend':
##
##     nnodes

## The following object is masked from 'package:seqinr':
##
##     translate
```

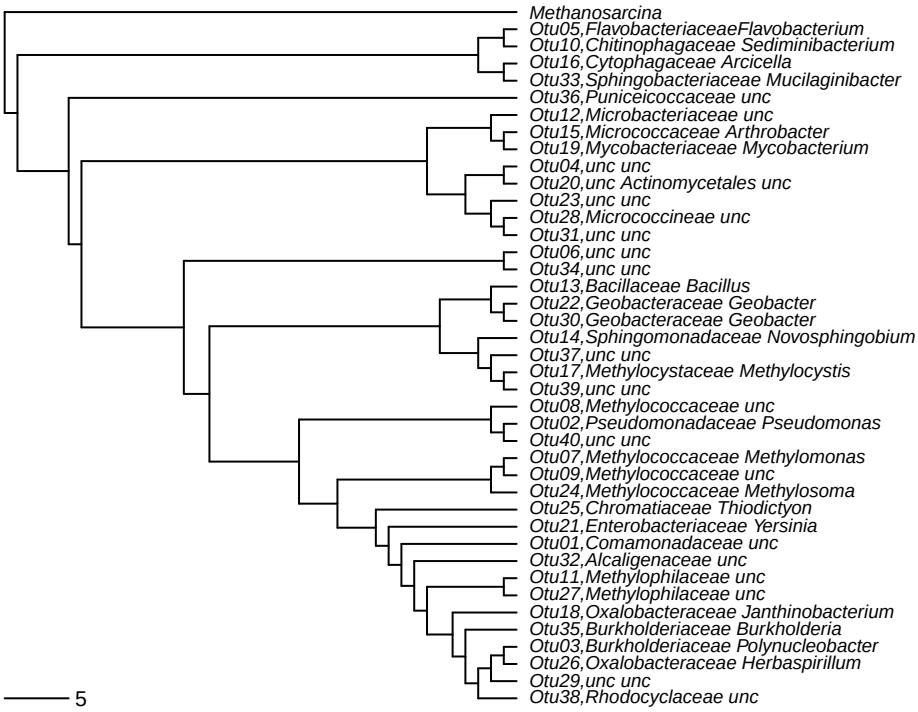
```
## The following object is masked from 'package:ape':
##
##     complement

## The following object is masked from 'package:base':
##
##     strsplit

##
## Attaching package: 'msa'

## The following object is masked from 'package:BiocManager':
##
##     version
```

```
library(Biostrings)

# import taxonomy info for OTUs
taxa <- as.matrix(read.csv("/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits/40ta

# data wrangling to get Otu names
#taxa[,"genus"]
#taxa[,"family"]

# import fasta with outgroup
bac <- readDNAStringSet("/Users/ericanadolski/GitHub/QB2023_Nadolski/2.Worksheets/8.PhyloTraits/40bac.fa

# align sequences using default MUSCLE parameters
# read.aln <- msaMuscle(bac)

# convert alignment to DNAbin object {ape}
bac.DNAbin <- as.DNAbin(bac)

# visualize alignment {ape}
par(mar = c(4, 10, 4, 2))
image.DNAbin(bac.DNAbin, cex.lab=0.5)
```




```
# distance matrix for F84 model tree
bac.dist.F84 <- dist.dna(bac.DNAbin, model="F84", pairwise.deletion=FALSE)
# tree for model
bac.F84.tree <- bionj(bac.dist.F84)
# root tree
bac.F84.outgroup <- match("Methanosarcina", bac.F84.tree$tip.label)
bac.F84.rooted <- root(bac.F84.tree, bac.F84.outgroup, resolve.root=TRUE)
# plot F84 neighbor joining tree
par(mar= c(1,1,2,1)+0.1)
plot.phylo(bac.F84.rooted, main="Pond Bacteria Neighbor Joining Tree", "phylogram", use.edge.length = F
add.scale.bar(cex=0.7)
```

Pond Bacteria Neighbor Joining Tree



Synthesis Answer: For this week, we decided it would be easiest to work with a subset of the ~35000 OTUs from the pond dataset, so we took a subset of the 40 OTUs with the highest overall read counts across the pond samples (coincidentally this metric was how the OTUs were named in the original project so we have a list of OTU 01-40). Our tree does fit with the known evolutionary relationships of the bacteria, we have the tips labeled by OTU with taxonomic information that we added up to the family level, and OTUs of the same families and genera are clustering together in our tree which confirms the accuracy of our tree compared to known relationships.