8. Worksheet: Phylogenetic Diversity - Traits

Bryan Guevara; Z620: Quantitative Biodiversity, Indiana University

28 February, 2025

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, **push** this file to your GitHub repo.
6. For the assignment portion of the worksheet, follow the directions at the bottom of this file.
7. When you are done, **Knit** the text and code into a PDF file.
8. After Knitting, submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `PhyloTraits_Worskheet.Rmd` and the PDF output of `Knitr` (`PhyloTraits_Worskheet.pdf`).

The completed exercise is due on **Wednesday, February 26th, 2025 before 12:00 PM (noon)**.

1) SETUP

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 `Week6-PhyloTraits/` folder, and
4. load all of the required R packages (be sure to install if needed).

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

```
## [1] "/cloud/project/QB2025_Guevara/Week6-PhyloTraits"
```

```
setwd("/cloud/project/QB2025_Guevara/Week6-PhyloTraits")

? msaMuscle
```

```
## No documentation for 'msaMuscle' in specified packages and libraries:
## you could try '??msaMuscle'
```

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***: From what I can see, prior to performing an alignment between all of our given sequences, we can see the sequences are relatively smaller than they are in '.afa' as the '.afa' file contains the alignment amongst all of the given sequences. We can see where gaps show up, where nucleotides have a weak signal, as well as see (if we look hard enough) regions of the sequence that are relatively strongly conserved across most of the sequences.

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.

```
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: 'phytools'
```

```
## The following object is masked from 'package:phylobase':
##
##     readNexus
```

```
##
## Attaching package: 'permute'

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

##
## Attaching package: 'vegan'

## The following object is masked from 'package:phytools':
##
##     scores

##
## 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
```

```
###To properly install Biostrings for our current version of R
#if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
#BiocManager::install("Biostrings")
library(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, saveRDS, setdiff,
##     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 object is masked from 'package:utils':
##
##     findMatches
```

```
## 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:seqinr':
##
##     translate

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

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

```
### to properly install msa for our current version of R since it is outdated.
#if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
#BiocManager::install("msa")
library(msa)


install.packages("bios2mds")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(bios2mds)
```

```
## Loading required package: amap
##
## Attaching package: 'amap'
## The following object is masked from 'package:vegan':
##
##     pca
## Loading required package: e1071
## Loading required package: scales
##
## Attaching package: 'scales'
## The following object is masked from 'package:phytools':
##
##     rescale
## Loading required package: cluster
##
## Attaching package: 'cluster'
## The following object is masked from 'package:maps':
##
##     votes.repub
## Loading required package: rgl
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
## Warning: 'rgl.init' failed, will use the null device.
## See '?rgl.useNULL' for ways to avoid this warning.
```

```
library(msa)
#Import and view unaligned sequences {Biostrings}
seqs <- readDNAStringSet("data/p.isolates.fasta", format = 'fasta')
seqs
```

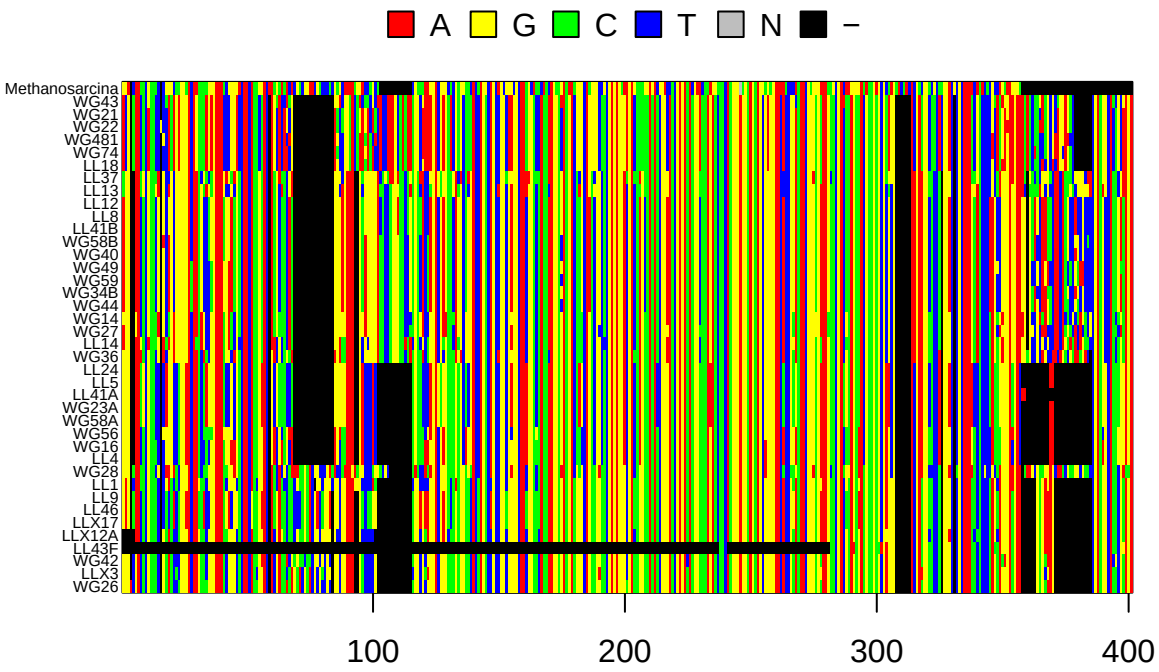
```
## DNAStringSet object of length 40:
##      width seq                                               names
##  [1]   619 ACACGTGAGCAATCTGCCCTTCT...TTCTCTGGGAATACCTGACGCT LL9
##  [2]   597 CGGCAGCGGGAAGTAGCTTGCTA...AACTGTTCAGCTAGAGTCTTGT WG14
##  [3]   794 CAGCGGCGGACGGGTGAGTAACA...GCTAACGCATTAAGCACTCCGC WG28
##  [4]   716 CTTCAGAGTTAGTGGCGGACGGG...TGCTAGTTGTCGGGATGCATGC LL24
##  [5]   803 ACGAACTCTTCGGAGTTAGTGGC...TAAAACTCAAAGGAATTGACGG LL41A
##  ...   ... ...
## [36]   652 TTCGGGAGTACACGAGCGGCGAA...TTCTCTGGGAATACCTGACGCT LL46
## [37]   661 GCGAACGGGTGAGTAACACGTGG...GAGCGAAAGCGTGGGTAGCGAA WG26
## [38]   694 GGCGAACGGGTGAGTAACACGTG...ACCCTGGTAGTCCACGCCGTAA WG42
## [39]   699 TACAGGTACCAGGCTCCTTCGGG...AAAGCATGGGTAGCGAACAGGA LLX17
## [40]  1426 TTCTGGTTGATCCTGCCAGAGGT...AACCTNAATTTTGCAAGGGGGG Methanosarcina
```

```
## Align sequences using default MUSCLE parameters {msa}
read.aln <- msaMuscle(seqs)

## Save and export the alignment to use later
save.aln <- msaConvert(read.aln, type = "bios2mds::align")
export.fasta(save.aln, "data/p.isolates.afa")
```

```
## Convert Alignment to DNAbin Object {ape}
install.packages("ape")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(ape)
p.DNAbin <- as.DNAbin(read.aln)

## Identify base pair region of 16S rRNA gene to visualize
window <- p.DNAbin[,100:500]

## Command to visualize sequence alignment {ape}
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:It seems that our sequence reads are roughly just over 400 nucleotides/bases long
Answer 2b:Probably somewhere between bases ~120 and ~305 as they seem to have the least gaps among the reads compared to other regions of the alignment.

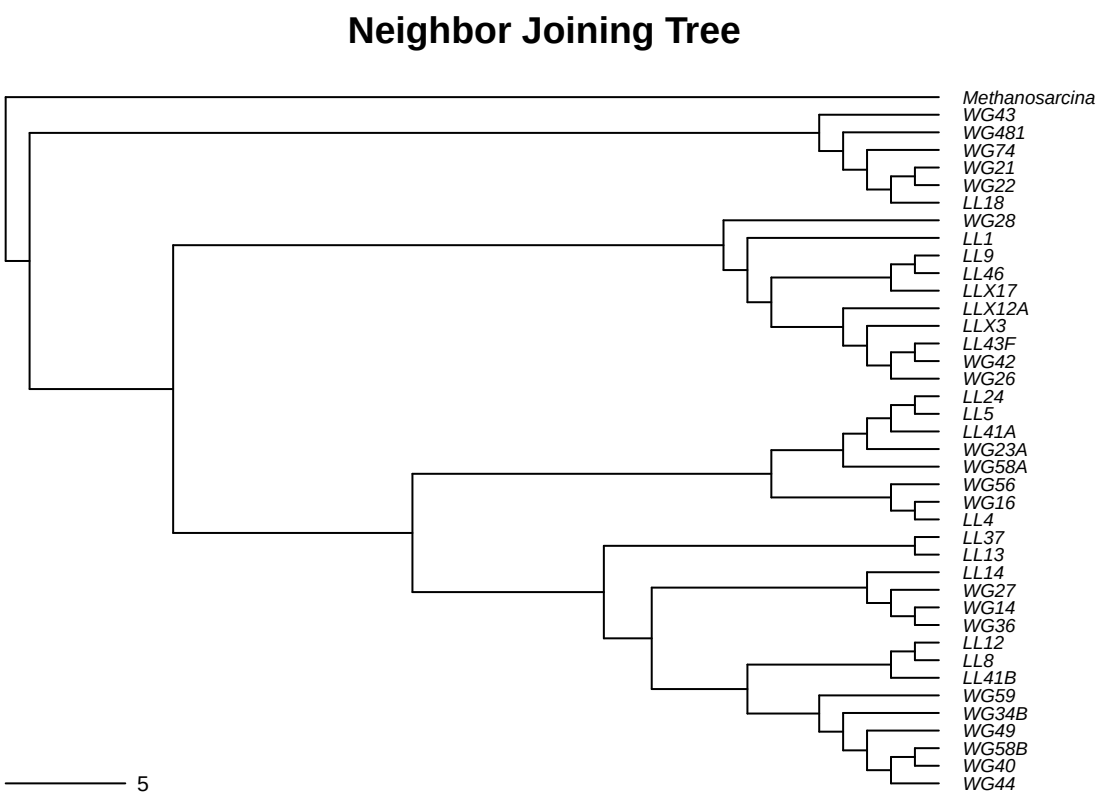
4) MAKING A PHYLOGENETIC TREE

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.

```
#Creating a distance matrix with "raw" model
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)

nj.tree <- bionj(seq.dist.raw)
##Identify outgroup sequence
outgroup <- match("Methanosarcina", nj.tree$tip.label)

##Rooting the tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

#Plot the rooted trees
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(nj.rooted, main = "Neighbor Joining Tree", "phylogram",
           use.edge.length = FALSE, direction = "right", cex = 0.6, label.offset = 1)
add.scale.bar(cex = 0.7)
```



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

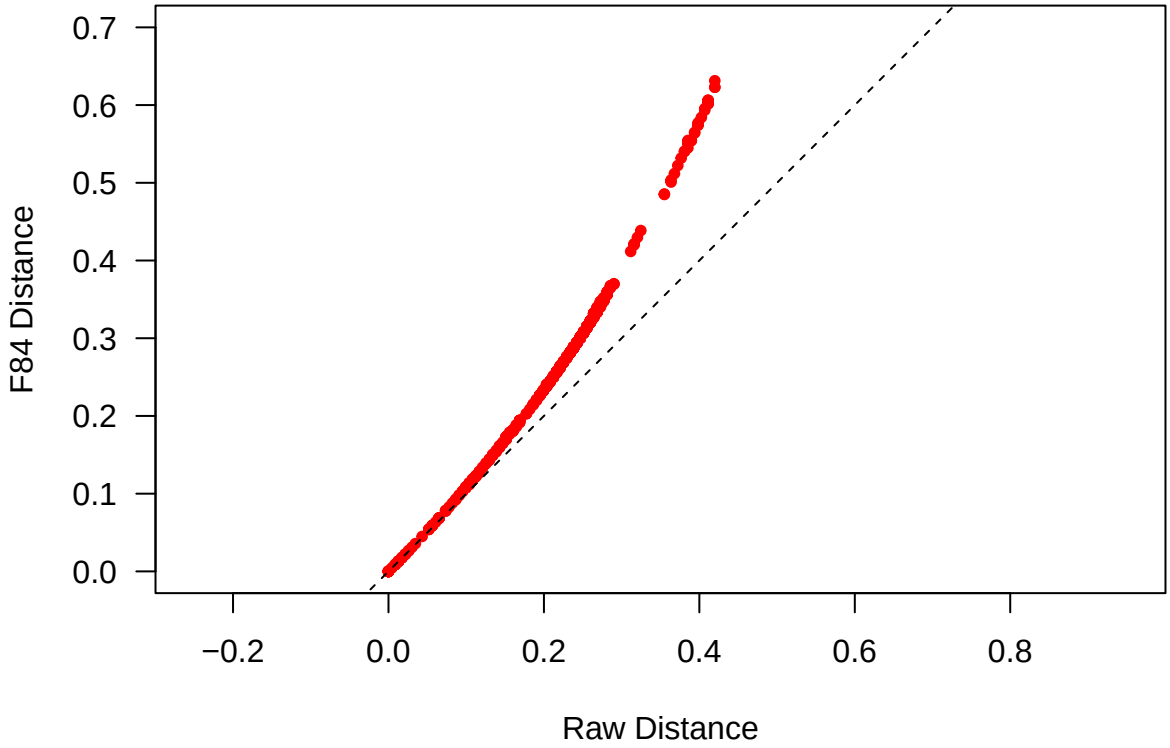
> ***Answer 3***:
> In our case, it didn't seem that the nj method was too computationally intensive. It's nice in that it produces a single tree based on distance, making it easy to understand. Some disadvantages are that it does not incorporate mutation rate or other evolutionary models like some other methods. This makes it more inaccurate compared to other models and understanding the true evolutionary

relationships between taxa.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

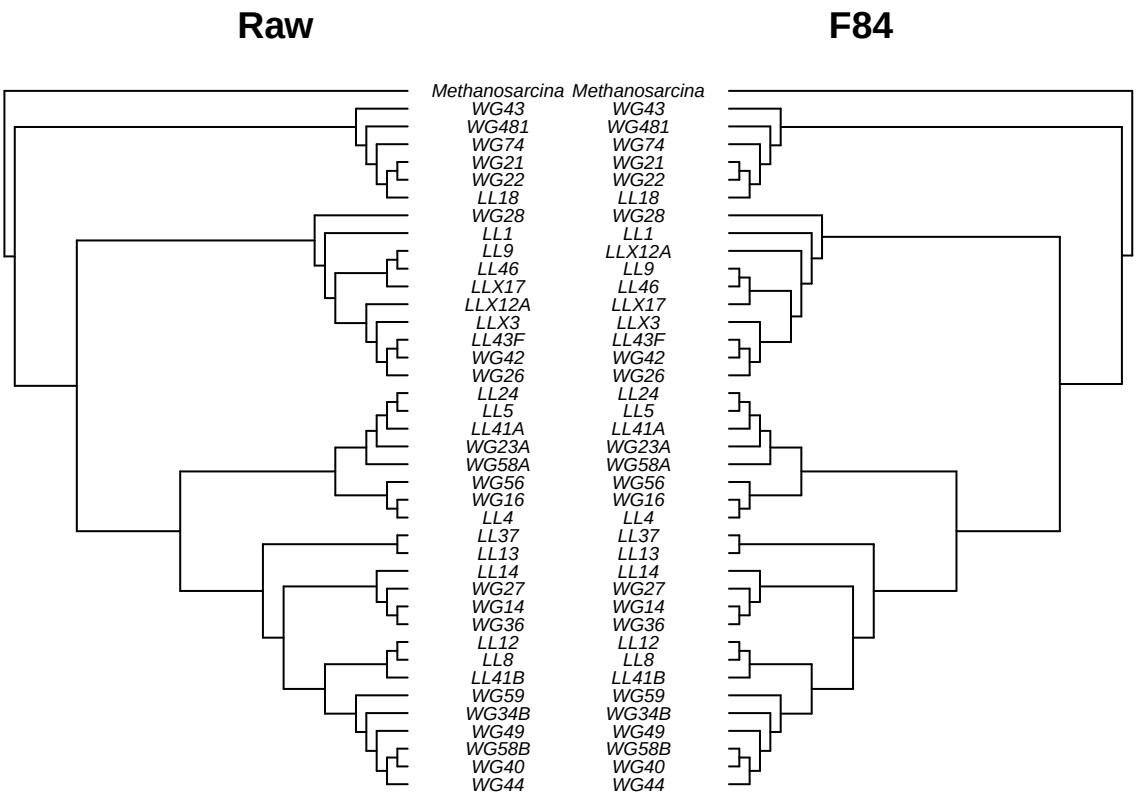
In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,
3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare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




```
##Make NJ trees using different DNA substitution models
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)

#Define outgroups
raw.outgroup <- match("Methanosarcina", raw.tree$tip.label)
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)

#Root the trees
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)

#Make Cophylogenetic plot
```

```
layout(matrix(c(1,2),1,2), width = c(1,1))
par(mar = c(1,1,2,0))
plot.phylo(raw.rooted, type = "phylogram", direction = "right",
           show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
           cex = 0.6, label.offset = 2, main = "Raw")
par(mar = c(1,0,2,1))
plot.phylo(F84.rooted, type = "phylogram", direction = "left",
           show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
           cex = 0.6, label.offset = 2, main = "F84")
```




```
dist.topo(raw.rooted, F84.rooted, method = "score")
```

```
##            tree1
## tree2 0.04219896
```

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

```
##Requires alignment to be read in with as phyDat object
phyDat.aln <- msaConvert(read.aln, type = "phangorn::phyDat")

aln.dist <- dist.ml(phyDat.aln)
aln.NJ <- NJ(aln.dist)

fit <- pml(tree = aln.NJ, data = phyDat.aln)

##Fit tree using JC69 subsitution model
```

```
fitJC <- optim.pml(fit, TRUE)
```

```
## optimize edge weights:  -10571.04 --> -10396.64
## optimize edge weights:  -10396.64 --> -10396.64
## optimize topology:  -10396.64 --> -10341.45  NNI moves:  10
## optimize edge weights:  -10341.45 --> -10341.45
## optimize topology:  -10341.45 --> -10341.45  NNI moves:  0
```

```
##Fit tree using a GTR model with gamma distributed rates
fitGTR <- optim.pml(fit, model = "GTR", optInv = TRUE, optGamma = TRUE,
                    rearrangement = "NNI", control = pml.control(trace = 0))
```

```
## only one rate class, ignored optGamma
```

```
##perform model selection with either an ANOVA test or with AIC
anova(fitJC, fitGTR)
```

```
## Likelihood Ratio Test Table
##   Log lik. Df Df change Diff log lik. Pr(>|Chi|)
## 1 -10341.5 77
## 2  -9790.4 86         9        1102.1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

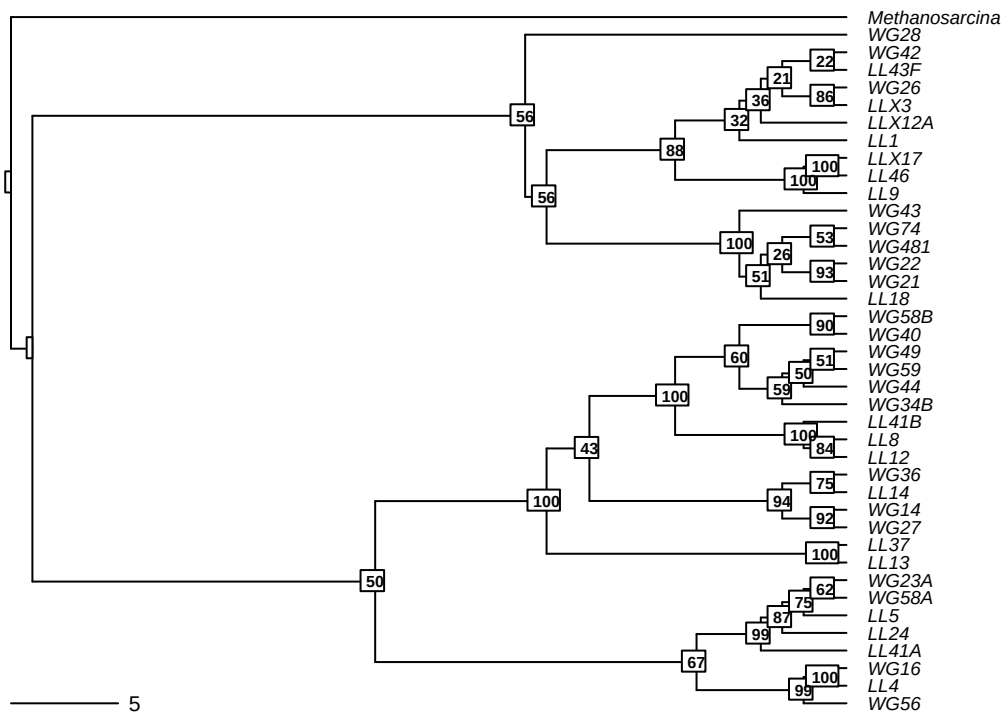
```
AIC(fitJC)
```

```
## [1] 20836.9
```

```
AIC(fitGTR)
```

```
## [1] 19752.84
```

```
##BOOTSTRAPPING
ml.bootstrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")
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 = 2, bg = "white", frame = "r", cex = 0.5)
```



Question 4:

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? Why or why not?

Answer 4a:It seems that there are some slight differences in which taxa are more closely related to one another. It is also clear that the branch lengths differ compared to nj tree, as we are not considering the substitution rate as an estimate of evolultionary change. ***Answer 4b***:We bootstrap to determine the reliability or confidence of the tree. We can see the confidence or bootstrap support at each of the nodes where each value represents how much that node is supported after resampling. ***Answer 4c***: Bootstrap values are essentially statistical support for each branch and ultimately the tree. They essentially tell us how much a clade or grouping shows when the data is resampled. Values of 95 (95%) or greater are 'operationally correct'. Values greater than 70 have moderate support while nodes at 50 or less or unresolved or not as confident. ***Answer 4d***: Branches with bootstrapping values of 50 or less have very low support. ***Answer 4e***: We should trust the branches with bootstrapping values of 95 or more as these groupings showed up in 95% of the trees generated from resamplings of the data.

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.

```
#import growth rate data
p.growth <- read.table("./data/p.isolates.raw.growth.txt", sep = "\t",
                       header = TRUE, row.names = 1)
#Standarize growth rates across strains
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.

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

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

```
#Generate nj tree using F84 DNA substitution model
nj.tree <- bionj(seq.dist.F84)

#Define outgruop
outgroup <- match("Methanosarcina", nj.tree$tip.label)
#Create rooted tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

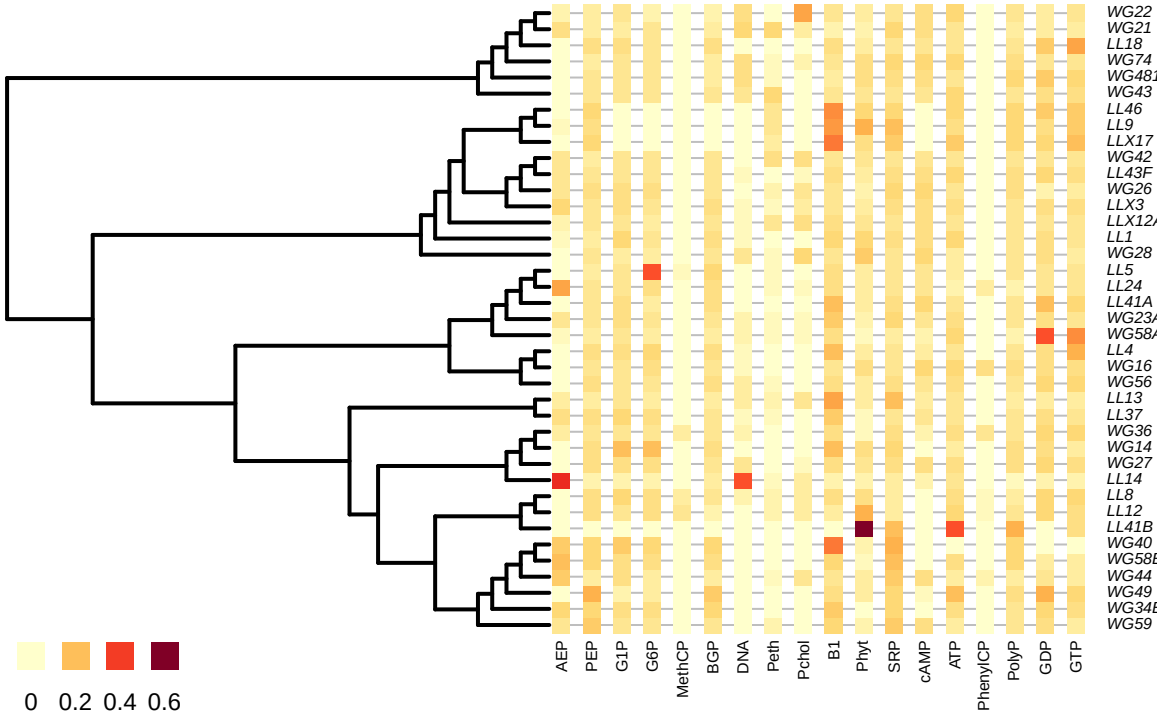
In the R code chunk below, do the following:
1. define a color palette (use something other than "YlOrRd"),
2. map the phosphorus traits onto your phylogeny,
3. map the nb trait on to your phylogeny, and
4. customize the plots as desired (use `help(table.phylo4d)` to learn about the options).

```
##Defining color palettte
library(RColorBrewer)

mypalette <- colorRampPalette(brewer.pal(9, "YlOrRd"))

nj.plot <- nj.rooted
```

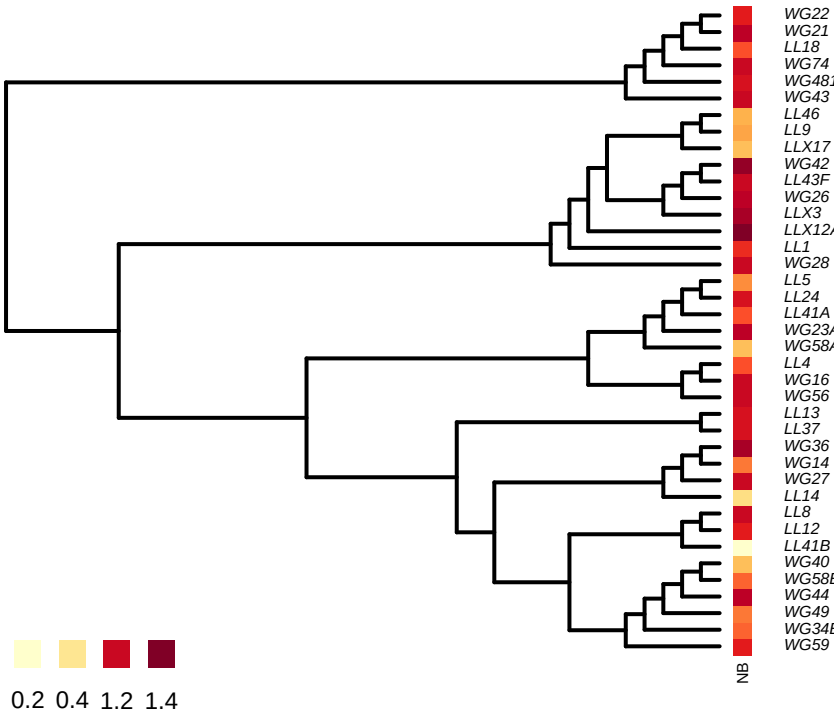
```
nj.plot$edge.length <- nj.plot$edge.length + 10^-1

par(mar = c(1,1,1,1), +0.1)
```

```
## Warning in par(mar = c(1, 1, 1, 1), +0.1): argument 2 does not name a graphical
## parameter
```

```
x <- phylo4d(nj.plot, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FALSE
              edge.width = 2, box = FALSE, col = mypalette(25), pch = 15, cex.symbol = 1.25, ratio.tree
```




```
par(mar = c(1,5,1,5) +0.1)
x.nb <- phylo4d(nj.plot, nb)
table.phylo4d(x.nb, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FA
              edge.width = 2, box = FALSE, col = mypalette(25), pch = 15, cex.symbol = 1.25, var.label =
```



Question 5:

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

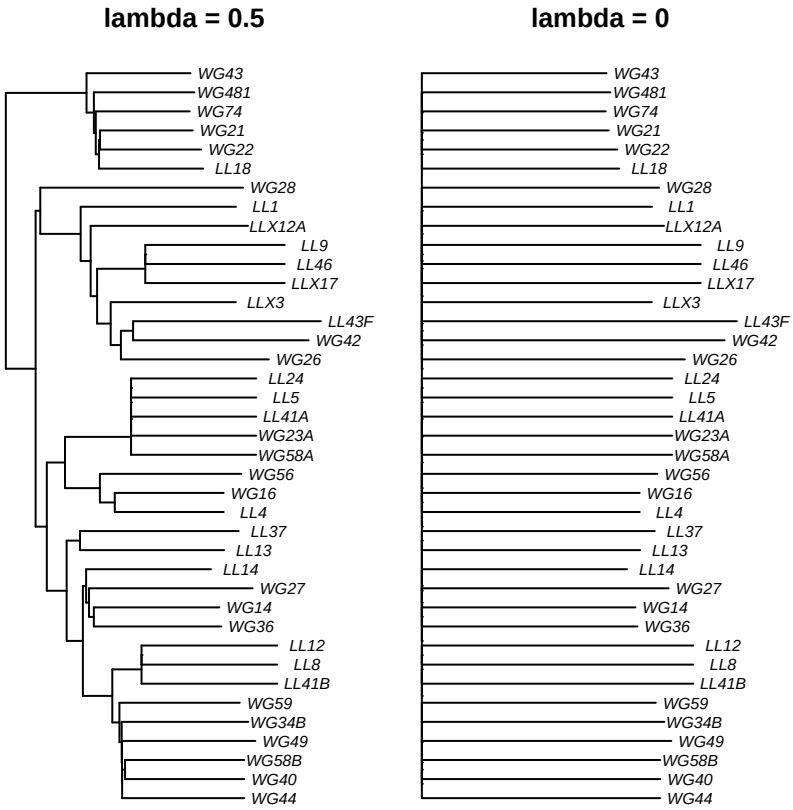
b) What kind of patterns would you expect to see from growth rate and niche breadth values that would support this hypothesis?

Answer 5a:I hypothesize that generalists species benefit from multiple different phosphorous sources, but each being less than a specialists ideal phosphorous sources' maximal growth rate.
Answer 5b: I would expect to see multiple relatively higher maximal growth rates across multiple phosphorous sources for a generalist species compared to a source in which a specialist species is not specialized for. The niche breath value would be greater as generalists are capable of inhabiting a larger fundamental niche compared to specialists. ## 6) HYPOTHESIS TESTING

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.

```
library(geiger)
nj.lambda.5 <- geiger:::rescale.phylo(nj.rooted, model = "lambda", lambda = 0.5)
nj.lambda.0 <- geiger:::rescale.phylo(nj.rooted, model = "lambda", 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(nj.lambda.5, main = "lambda = 0.5", cex = 0.7, adj = 0.5)
plot(nj.lambda.0, main = "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(nj.rooted, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.006976
##  sigsq = 0.108060
##  z0 = 0.657697
##
##  model summary:
##  log-likelihood = 21.503414
##  AIC = -37.006827
##  AICc = -36.321113
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 42
##  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.lambda.0, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.108048
##  z0 = 0.656477
##
##  model summary:
##  log-likelihood = 21.502505
##  AIC = -37.005010
##  AICc = -36.319295
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 77
##  frequency of best fit = 0.770
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
# Compare Pagel's lambda score with likelihood ratio test
# Lambda = 0, no phylogenetic signal
phylosig(nj.rooted, nb, method = "lambda", test = TRUE)
```

```
##
## Phylogenetic signal lambda : 0.00699105
## logL(lambda) : 21.5034
## LR(lambda=0) : 0.00181763
## P-value (based on LR test) : 0.965994
```

```
phylosig(nj.lambda.0, nb, method = "lambda", test = TRUE)
```

```
##
## Phylogenetic signal lambda : 0.0999267
## logL(lambda) : 21.5025
## LR(lambda=0) : 0
## P-value (based on LR test) : 1
```

Question 6: 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 6a***: in our transformed tree with a lambda set to zero, our lambda ends up at 0.006975, which is still small and indicates low coordination of phylogenetic history. It also just means that this tree has a very weak phylogenetic signal compared to the untransformed tree with a lambda value much closer to one which means that phylogeny explains variation. ***Answer 6b***: AIC is a metric of how good our model fits the data. In this case, the AICs are quite similar indicating that accounting for phylogeny does not really improve the fit of our model. I would probably

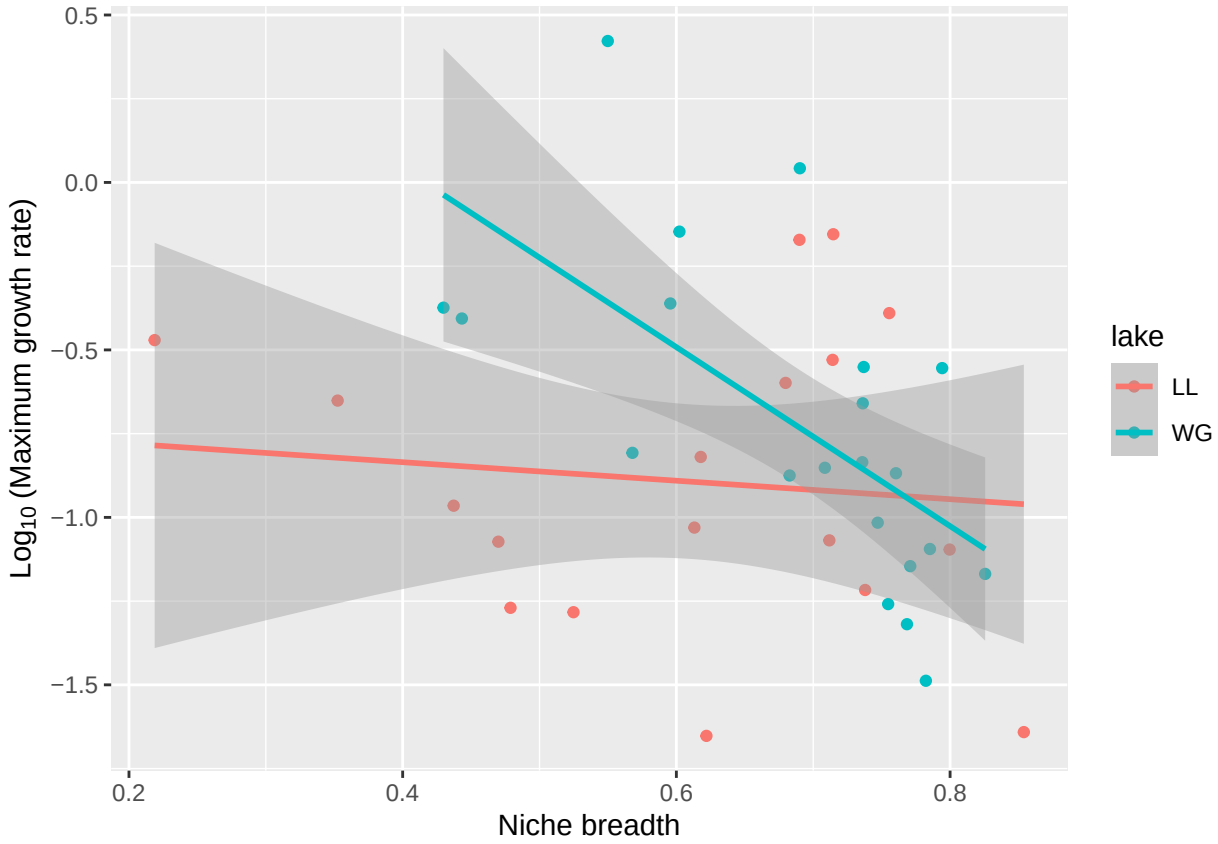
choose the model with lambda set to zero since that model provides a slightly more negative AIC value. ***Answer 6c***: The result of our transformed tree indicates very low phylogenetic signal.

7) PHYLOGENETIC REGRESSION

Question 7: 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, 2. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny

```
###Using niche breadth data, creating column that indicates the lake origin of each strain
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")
}


##Adding names to columns
colnames(nb.lake)[1] <- "NB"

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

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'
```




```
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
```

```
fit.plm <- phylolm(log10(umax) ~ NB * lake, data = nb.lake, nj.rooted,
                   model = "lambda", boot = 0)
summary(fit.plm)
```

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = nj.rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  41.08 -14.54
##
## Raw residuals:
##      Min       1Q   Median       3Q      Max
## -0.75804 -0.18999 -0.07425  0.32496  0.95857
##
## Mean tip height: 0.1814501
## Parameter estimate(s) using ML:
## lambda : 0.4861372
## sigma2: 0.9184437
##
## Coefficients:
##              Estimate    StdErr t.value p.value
## (Intercept) -0.891268  0.370036 -2.4086 0.02142 *
## NB          -0.004805  0.521303 -0.0092 0.99270
## lakeWG       1.438930  0.577231  2.4928 0.01755 *
## NB:lakeWG   -1.966388  0.848702 -2.3169 0.02648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1935    Adjusted R-squared: 0.1243
##
## Note: p-values and R-squared are conditional on lambda=0.4861372.
```

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

```
## [1] 41.07574
```

```
? AIC
```

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 7a: When considering shared evolutionary history, we need to correct because our observations are no longer independent and failing to do so could lead to false rejections of the null hypothesis. ***Answer 7b***: The residual errors are desfcribed by a covariance matrix in a phylogenetic regression, taking into account the branch lengths of underlying phylogeny. Standard linear regression's residual errors are independent and follow a normal distribution. ***Answer 7c***: It would seem, based on the AIC that the phylogenetic linear model is the better fitted model. The slopes for both models appear as negative. However, it looks like the fit.lm slope is much steeper with a more negative value. ***Answer 7d***: It would probably disappear when looking at the relationship between any given trait and a specific consumable resource. We would see a

positive coorelation between these two meaning that greater abundance of given resource leads to increases in that phenotype, however, increases in that phenotype are likley ancestral, meaning that phylogeny would cause the relationship to disappear. Perhaps that resource selected for such a trait at one point, however, that doesn't necessarily mean that this is why the trait persists to present day.

7) SYNTHESIS

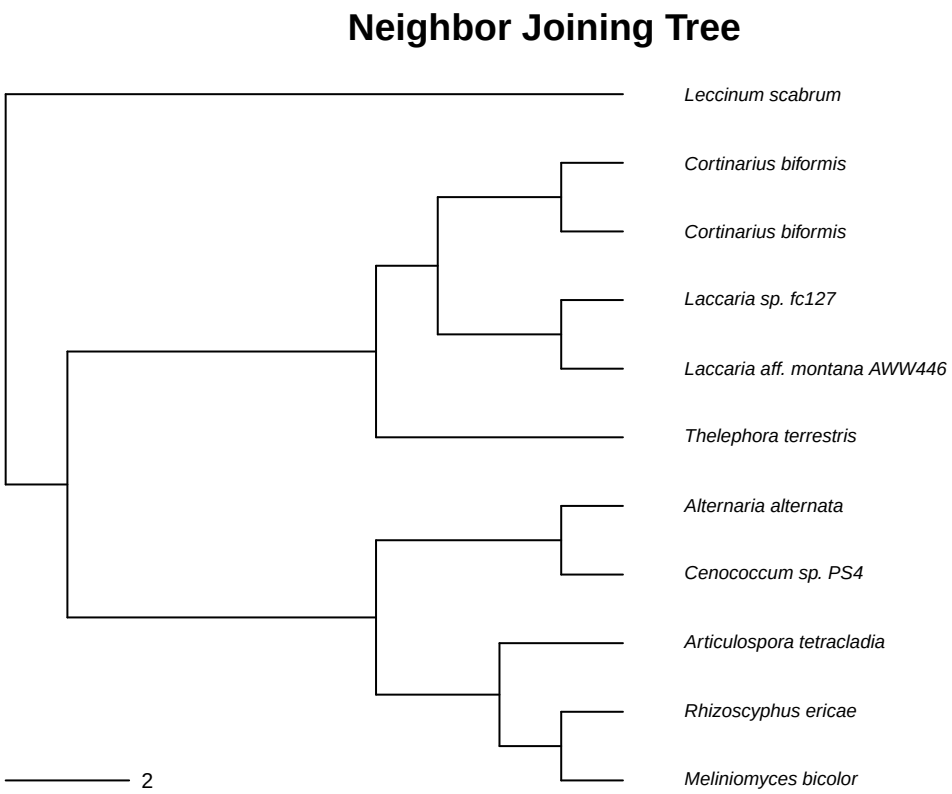
Work with members of your Team Project to obtain reference sequences for 10 or more taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. 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**. For example, here an accession number for a fungal isolate that our lab has worked with: JQ797657. 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. This is pretty cool. Give it a try.

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 course 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.

After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. 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?

```
Fungi.seqs <- readDNAStringSet("data/MAT_Fungi.fasta", format = 'fasta')
Fungi_read.aln <- msaMuscle(Fungi.seqs)
Fungi.DNAbin <- as.DNAbin(Fungi_read.aln)

##Requires alignment to be read in with as phyDat object
FungDat.aln <- msaConvert(Fungi_read.aln, type = "phangorn::phyDat")

Fungi.aln.dist <- dist.ml(FungDat.aln)
Fung.aln.NJ <- NJ(Fungi.aln.dist)

#Creating a distance matrix with "raw" model
Fungi.seq.dist.raw <- dist.dna(Fungi.DNAbin, model = "raw", pairwise.deletion = FALSE)

Fungi.nj.tree <- bionj(Fungi.seq.dist.raw)
##Identify outgroup sequence
Fungi.outgroup <- match("Leccinum scabrum", Fungi.nj.tree$tip.label)

##Rooting the tree
```

```
Fungi.nj.rooted <- root(Fungi.nj.tree, outgroup, resolve.root = TRUE)

#Plot the rooted trees
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(Fungi.nj.rooted, main = "Neighbor Joining Tree", "phylogram",
           use.edge.length = FALSE, direction = "right", cex = 0.6, label.offset = 1)
add.scale.bar(cex = 0.7)
```




```
##It seems that the output does indeed jibe with the evolutionary history about our organisms, keeping
##the fungi are almost all ECM or 'maybe 'ECM', there does not seem to be any differentiation between t
###In future analyses, it would be good to incorporate bootstrapping values to get a better idea of wha
###Then, it would be good perform a tree using Maximum Likelihood in order to account for nucleotide su
```

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed `8.PhyloTraits_Worksheet.Rmd` document, push it to GitHub, and create a pull request. Please make sure your updated repo include both the pdf and RMarkdown files. Unless otherwise noted, this assignment is due on **Wednesday, February 26th, 2025 at 12:00 PM (noon)**.