8. Worksheet: Phylogenetic Diversity - Traits

Joy O'Brien; 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 22$^{\text{nd}}$, 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/joyobrien/GitHub/QB2023_OBrien/2.Worksheets/8.PhyloTraits"
```

```
setwd("~/GitHub/QB2023_OBrien/2.Worksheets/8.PhyloTraits")
```

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***: There are many differences between these two file types. To start, the .fasta file uses lowercase bases while the .afa file uses uppercase bases. In the .afa file I notice a lot of "N"s which represents uncertainty in the sequence. However, I do not see any "N"s. Lastly in the .afa file, there are a bunch dashes while in the .fasta file there are no dashes and therefore no missing information.

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: '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: 'phytools'

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

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

## The following object is masked from 'package:ape':
##
##     drop.tip.multiPhylo

##
## Attaching package: 'cluster'

## The following object is masked from 'package:maps':
##
##     votes.repub

## Registered S3 method overwritten by 'dendextend':
##   method     from
##   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:phytools':
##
##     untangle

## 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

##
## Attaching package: 'phylogram'

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

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

##
## Attaching package: 'scales'

## The following object is masked from 'package:geiger':
##
##     rescale
```

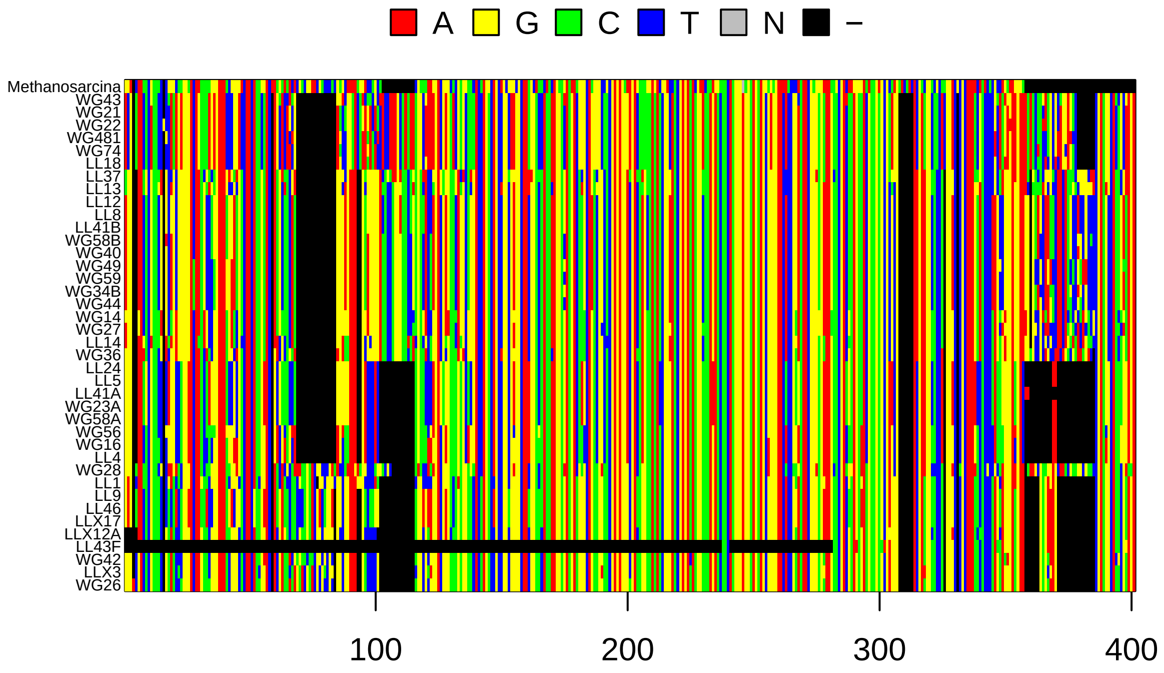
```r
# Loading/installing bioconductor
if (!require("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
}
if(!require("msa", quietly = TRUE)) {
  BiocManager::install("msa")
}
```

```
##
## 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

##
## 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

##
## Attaching package: 'IRanges'

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

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

## 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'

##
## 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
```

```r
library(msa)
# Read the alignment file
seqs <- readDNAStringSet("data/p.isolates.fasta", format = 'fasta')
# View sequences
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 MUSCLE parameters (msa)
read.aln <- msaMuscle(seqs)
# Save and export
save.aln <- msaConvert(read.aln, type = "bios2mds::align")
export.fasta(save.aln, "./data/p.isolates.afa")

# Visualize the alignment
# Convert alignment to DNAbin object
p.DNAbin <- as.DNAbin(read.aln)

# Identify base pair region
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***: Approximately 400 nucleotides. ***Answer 2b***: The regions that have vertical bars aligned with others of the same color are good for phylogenetic inference because this indicates a shared nucleotide. We would not want to use the alignment gaps (the black sections) because these are sections used by the program to increase nucelotide matching and represent either insertions or deletions.

4) MAKING A PHYLOGENETIC TREE

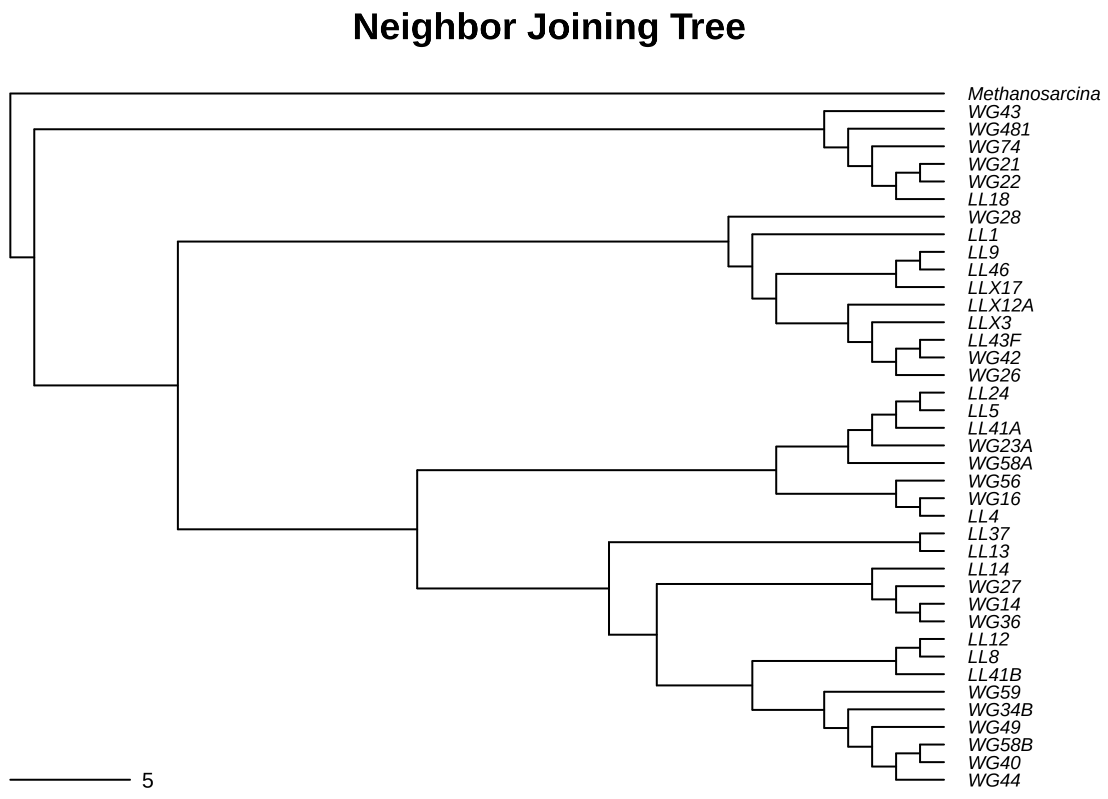
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

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.

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

# Create neighbor joining tree
nj.tree <- bionj(seq.dist.raw)
# Defining outgroup and root of tree
outgroup <- match("Methanosarcina", nj.tree$tip.label)
# Root tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
# Plotting 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 = "right", cex = 0.6,
           label.offset = 1)
add.scale.bar(cex = 0.7)
```



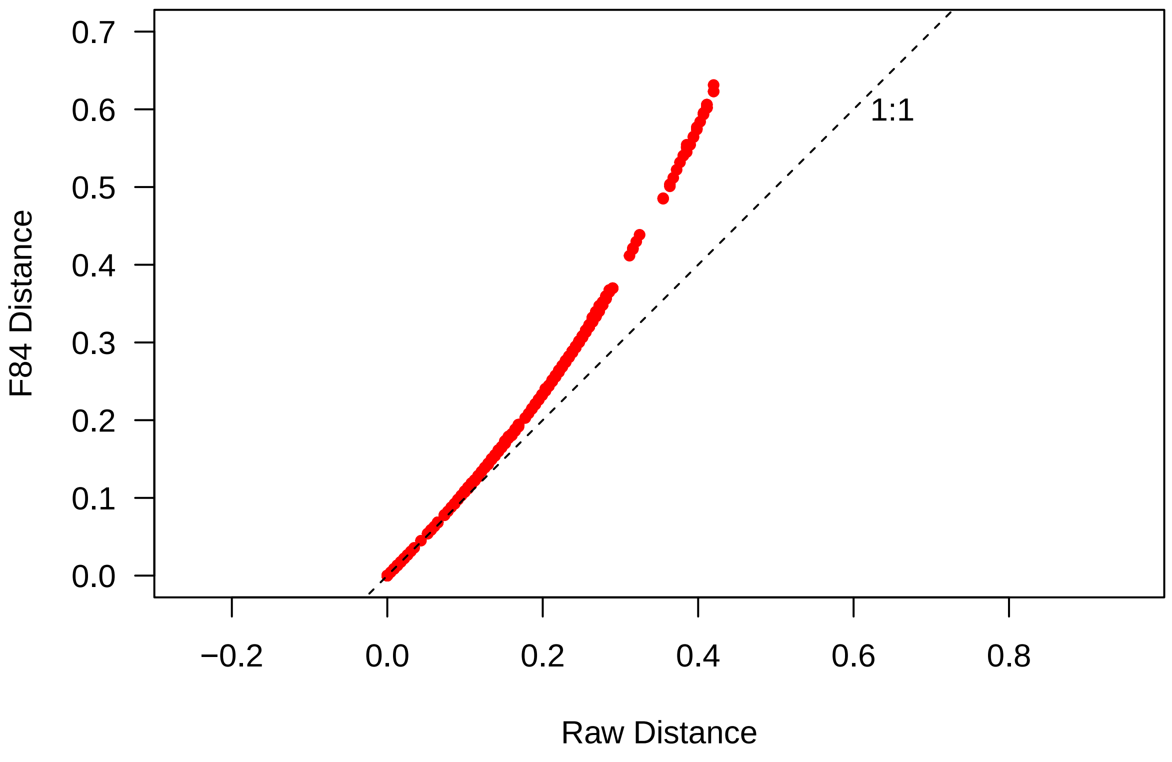
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: The advantages of making a neighbor joining tree is that you can use it as a guide tree which allows for further evolution modeling. Another advantage of making a neighbor joining tree is that it is computationally inexpensive. One of the disadavantages of neighbor-joining trees is that because this model serves as a starting point, it generally is not considered good enough to use in a publication.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

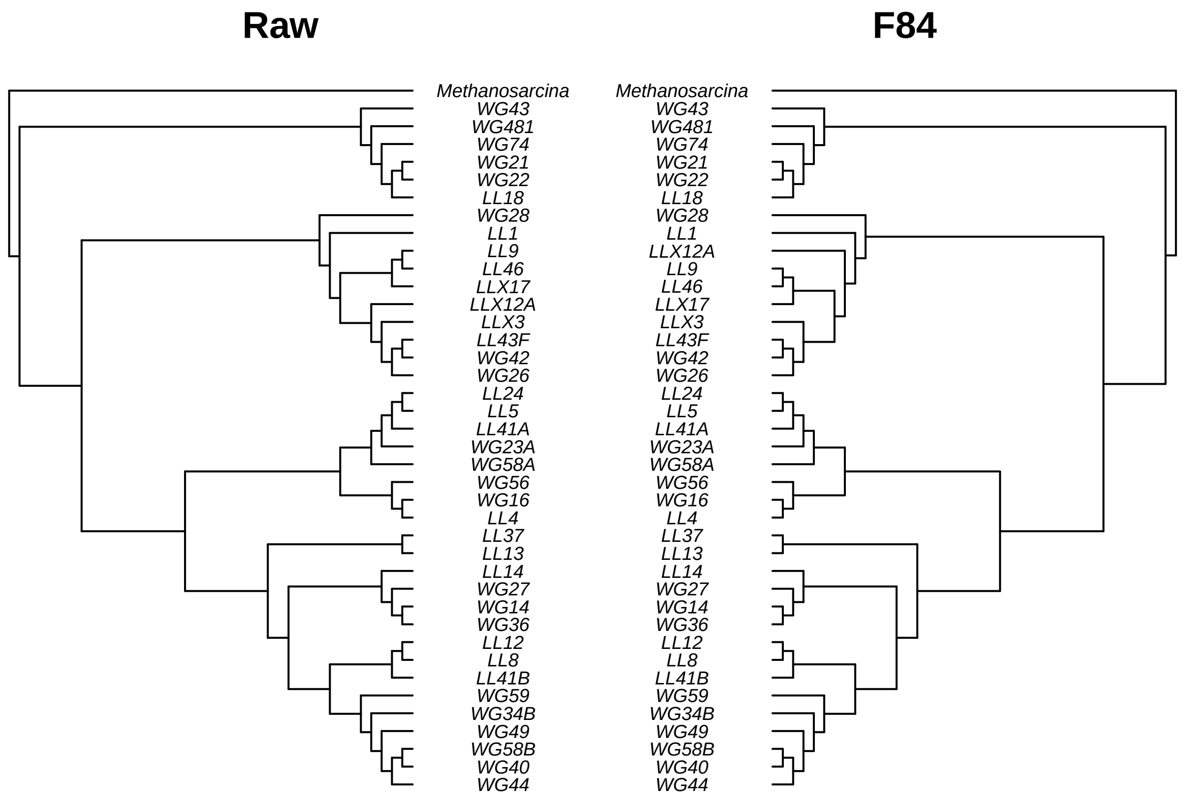
In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,
3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare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

```r
# Create distance matrix with F84 model
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise = FALSE)

# Plot distances from different DNA substitution models
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")
```



```r
# Make NJ trees using DNA substituation 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)
```

```r
# 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, 0, 2, 1))
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")
```



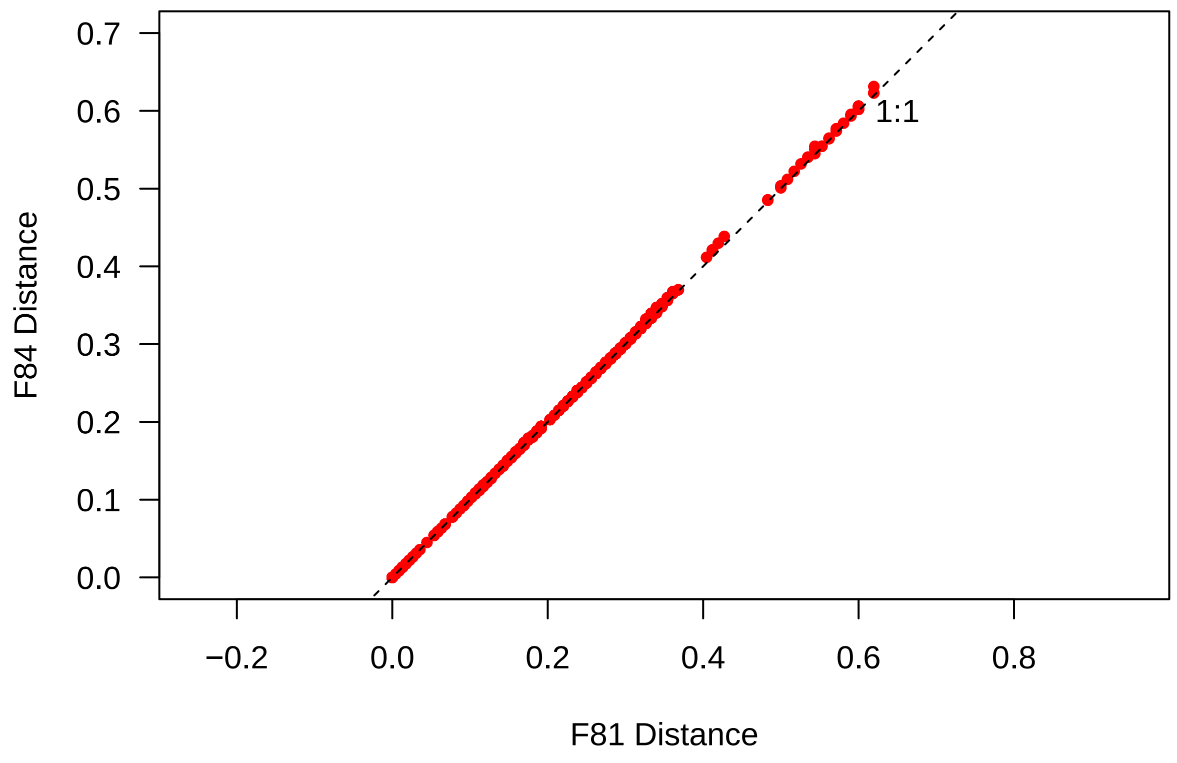
```r
# Calculate branch length score
dist.topo(raw.rooted, F84.rooted, method = "score")
```

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

In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,

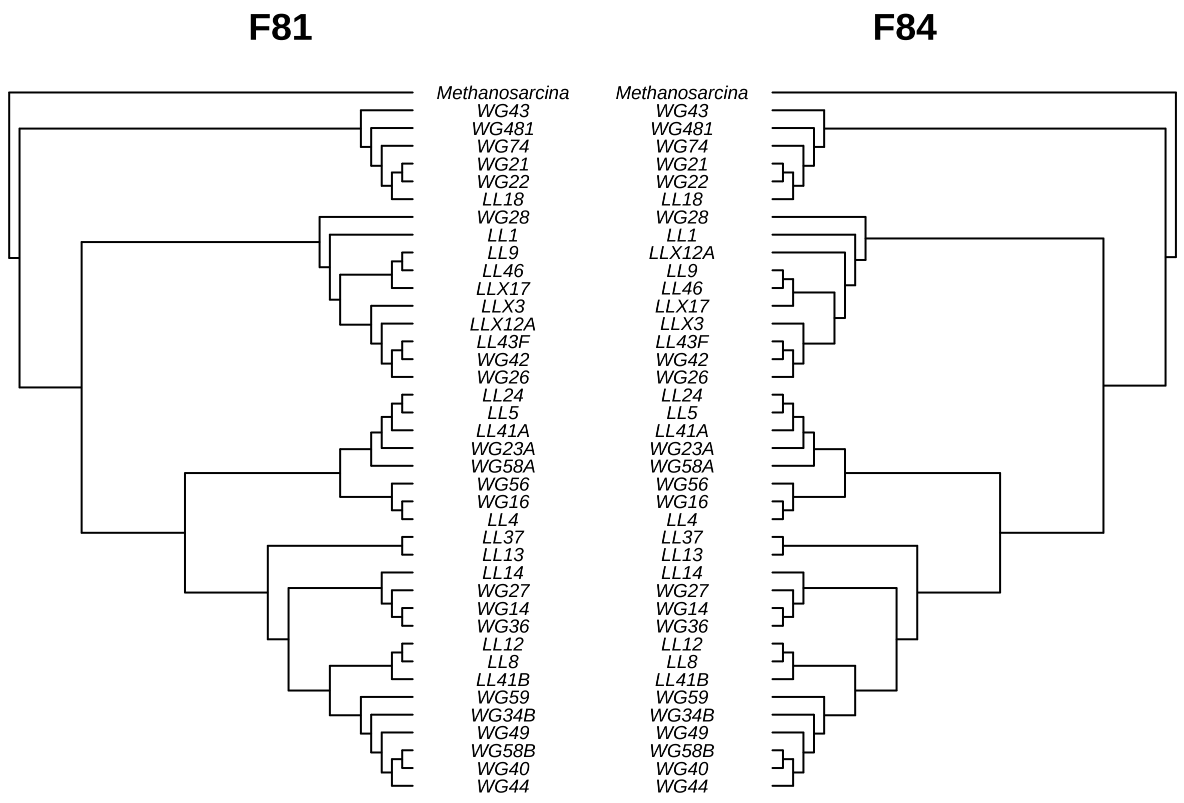
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5. be sure to format, add appropriate labels, and customize each plot.

```r
# Going to use F81 model
# Create distance matrix with F81 model
seq.dist.F81 <- dist.dna(p.DNAbin, model = "F81", pairwise = FALSE)

# Create saturation plot that compares to F84 model
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.F81, seq.dist.F84,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7),
     ylim = c(0, 0.7), xlab = "F81 Distance", ylab = "F84 Distance")
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1")
```



```r
# Cophylogenetic plot to compare topologies of both models

# Make NJ trees using DNA substituation model F81
F81.tree <- bionj(seq.dist.F81)

# Define outgroup
F81.outgroup <- match("Methanosarcina", raw.tree$tip.label)

# Root the trees
F81.rooted <- root(F81.tree, F81.outgroup, resolve.root = TRUE)
```

```r
# Make the figure

layout(matrix(c(1, 2), 1, 2), width = c(1,1))
par(mar = c(1, 0, 2, 1))
plot.phylo(F81.rooted, type = "phylogram", direction = "right",
           show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
           cex = 0.6, label.offset = 2, main = "F81")
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")
```



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 substitution model that I chose was Felenstein model (F81). This model is based on JC69 but assumes that nucleotide frequiencies can vary. Additionally, like JC69, it assumes that the probability of nucleotide mutation is equal across nucleotides. F84 is different from F81 because F84 assumes different rates of base transitions/transversions and accounts for differences in base frequencies. ***Answer 4b***: The choice of using the F81 substitution model compared to the F84 model does not really affect the phylogenetic reconstruction according to the saturation plot above, which details that there is little to no difference in the distances between the two models. By looking at the cophylogenetic plot, there are only three differences in branch tips which is likely due to base transitions or transversions. ***Answer 4c***: As mentioned above, there are only 3 branch tips that are disagreement between the F81 and F84 model. This indicates that there is a low substitution rate of nucleotide transitions in the F84 model and overall indicates that regardless of assumptions, there is little to no difference between the substitution rates within models.

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

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

# Make NJ tree for the maximum likelihood method
# Phangorn requires attr class, remake trees
aln.dist <- dist.ml(phyDat.aln)
aln.NJ <- NJ(aln.dist)

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

# Fit tree using JC69 substitution 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
```

```r
# 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
```

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

```
## Likelihood Ratio Test Table
##   Log lik. Df Df change Diff log lik. Pr(>|Chi|)
## 1 -10341.5 77
## 2  -9786.1 86         9        1110.6  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

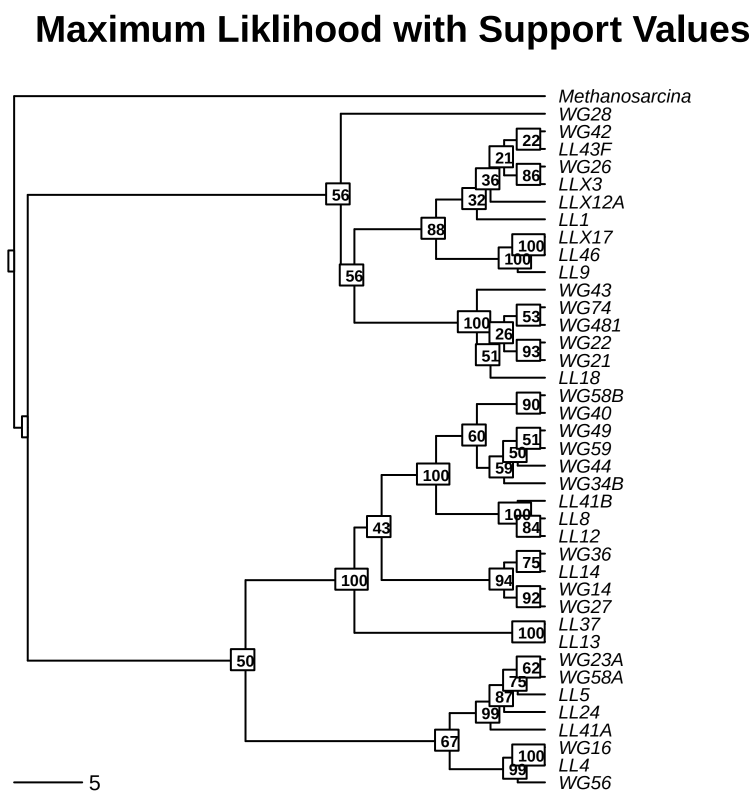
```r
AIC(fitJC)
```

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

```r
AIC(fitGTR)
```

```
## [1] 19744.27
```

```r
# Bootstrap support values
ml.bootstrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")
par(mar = c(1,10, 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 Liklihood with Support Values")
add.scale.bar(cex = 0.7)
nodelabels(ml.bootstrap$node.label, font = 2, 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 maximum liklihood tree looks like it forms two main branches while the NJ tree has multiple branches and this may be because of the way the phylogenetic distances are calculated. These differences include that the NJ tree does not take into account specific nucleotide states nor does it act as a statistical method. The ML tree is based on that maximum liklihood statistical procedure and is least affected by sampling error. ***Answer 5b***: We bootstrap to re-sample the data in order to understand how reliable the results of the tree are. ***Answer 5c***: Bootstrap values tell you how ***Answer 5d***: Branches that have a bootstrap value of 50% or less are considered insufficiently resolved and have low support. ***Answer 5e***: No we should not trust these branches because they are insufficiently resolved.

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.

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

# Standardize 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.

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

# Create a function that calculates niche breadth
levins <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
  }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}

# Using function to calculate nb for isolates
```

```r
nb <- as.matrix(levins(p.growth.std))

# Add row names to niche breadth matrix
nb <- setNames(as.vector(nb), as.matrix(row.names(p.growth)))
```

C. Visualizing Traits on Trees

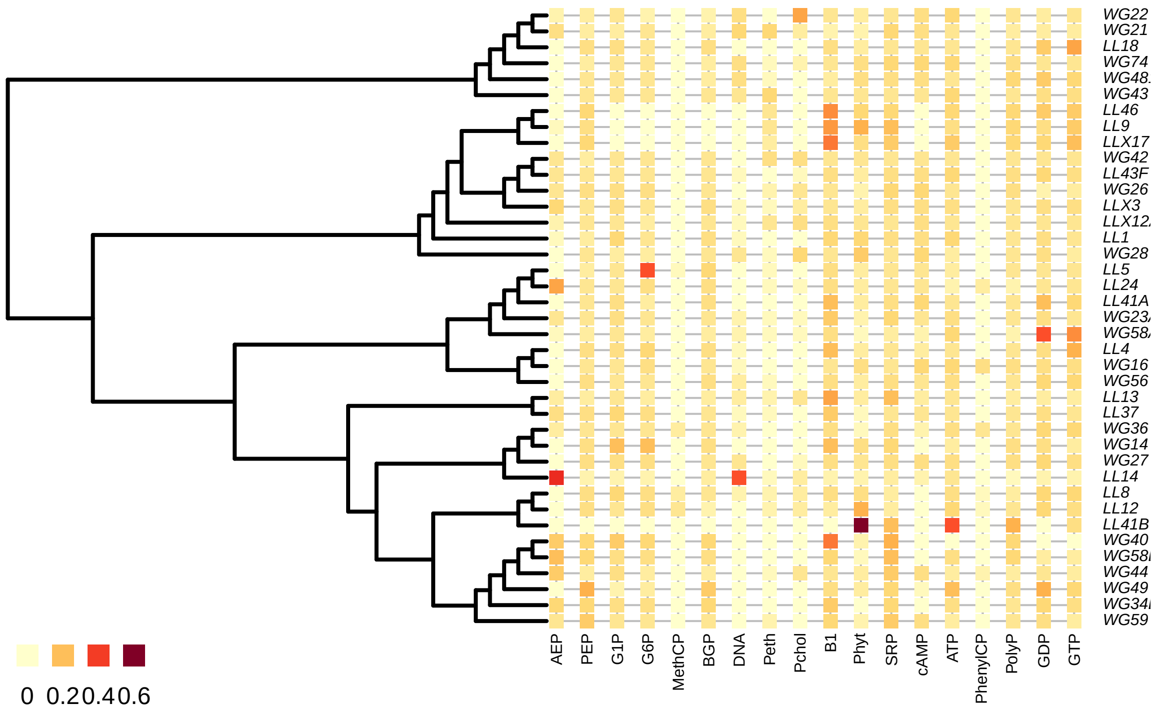
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```r
# Make a NJ tree using F84 DNA substitution model
nj.tree <- bionj(seq.dist.F84)

# Define outgroup
outgrop <- match("Methanosarcina", nj.tree$tip.label)

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

# Keep rooted but drop outgroup branch
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).

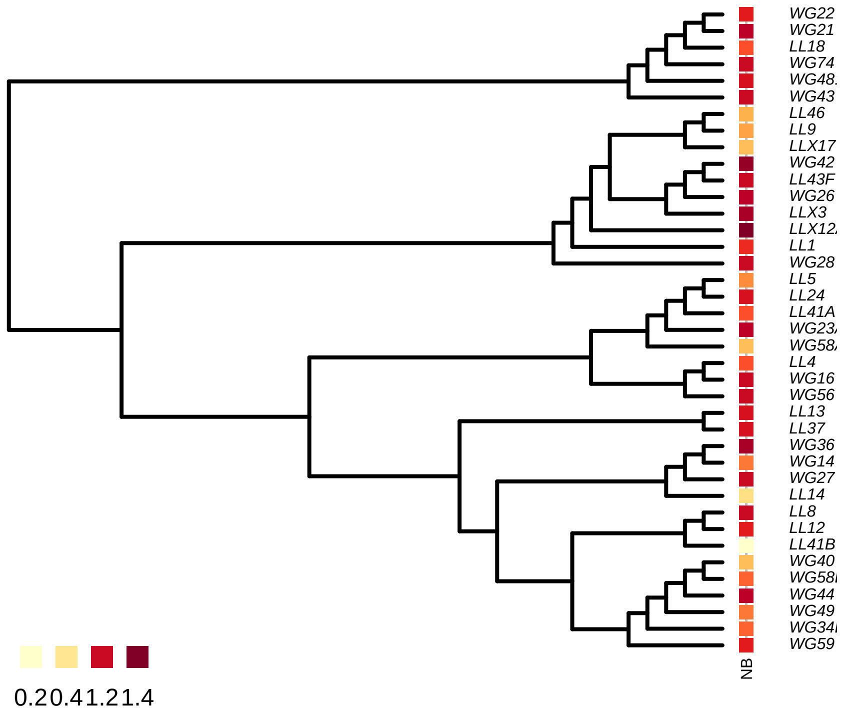
```r
# Define color palette
mypalette <- colorRampPalette(brewer.pal(9, "YlOrRd"))

# First correct for 0 branch lengths
nj.plot <- nj.rooted
nj.plot$edge.length <- nj.plot$edge.length + 10^-1

# Map P traits
par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(nj.plot, 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, cer.symbol = 1.25,
              ratio.tree = 0.5, cex.legend = 1.5, center = FALSE)
```

```
## Warning in plot.window(...): "cer.symbol" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "cer.symbol" is not a graphical parameter

## Warning in title(...): "cer.symbol" is not a graphical parameter
```



```r
# Niche breadth
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 = FALSE, use.edge.length = FALSE,
              edge.color = "black", edge.width = 2, box = FALSE,
              col=mypalette(25), pch = 15, cer.symbol = 1.25, var.label = ("NB"), ratio.tree = 0.90,
              cex.legend = 1.5, center = FALSE)
```

```
## Warning in plot.window(...): "cer.symbol" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "cer.symbol" is not a graphical parameter

## Warning in title(...): "cer.symbol" is not a graphical parameter
```



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?

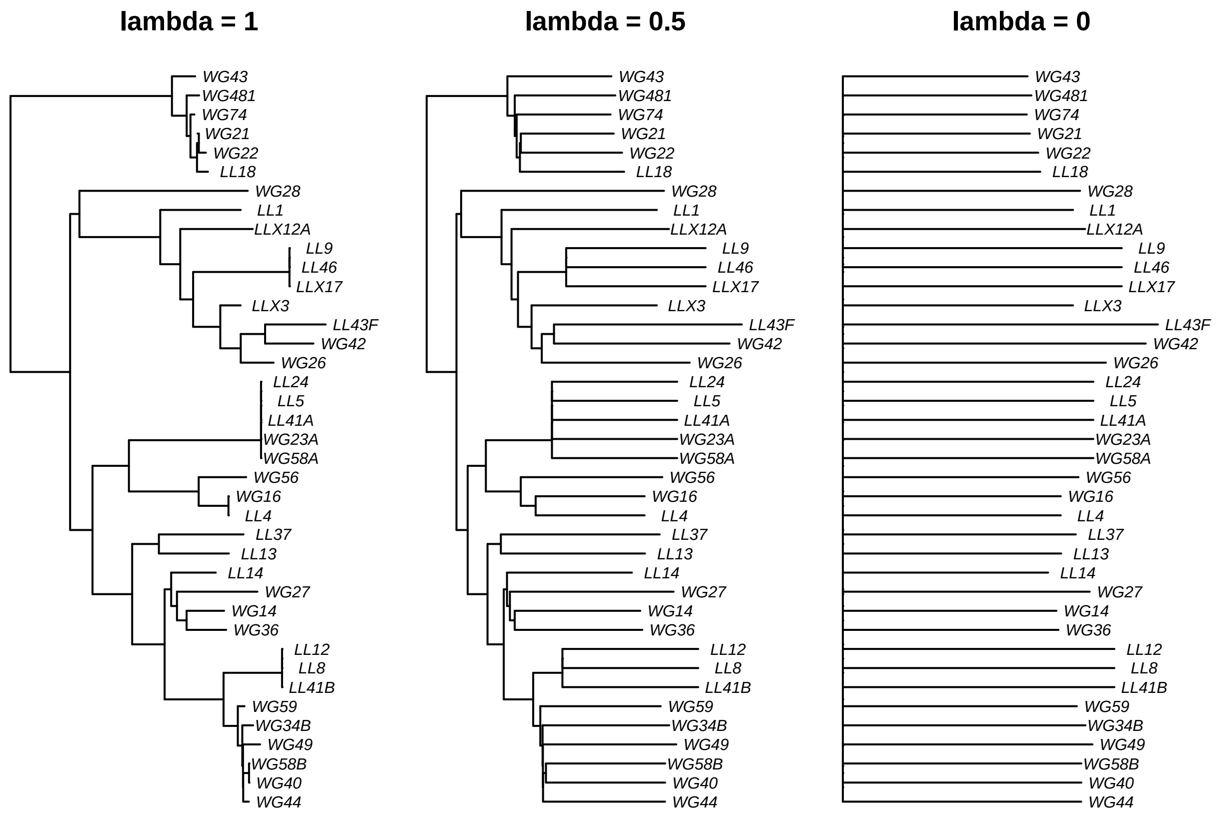
> ***Answer 6a***: Specialists will have a higher maximum growth rate on very few phosphorous resources while generalists will have a lower maximum growth rate but will be able to grow on multiple phosophorous resources.
> ***Answer 6b***: For growth rate for specialists, we would expect a higher maximum growth rate since they will be able to use their phosphorous resources more efficiently than a generalist. Niche breadth values for specialists will be lower (closer to 0) because specialists favor specific environments while the generalist would have a niche breadth value closer to 1 because it may be abundant in multiple environments.

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

```r
# Rescaling phylogenetic trees
nj.lambda.5 <- geiger::rescale(nj.rooted, "lambda", 0.5)
nj.lambda.0 <- geiger::rescale(nj.rooted, "lambda", 0)
layout(matrix(c(1, 2, 3), 1, 3), width = c(1, 1, 1))
# Plotting original tree and two scaled trees
par(mar=c(1, 0.5, 2, 0.5) + 0.1)
plot(nj.rooted, main = "lambda = 1", cex = 0.7, adj = 0.5)
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)
```



```r
# Generate test stats for comparing phylogenetic signal
fitContinuous(nj.rooted, 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 = 47
##  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 = 83
##  frequency of best fit = 0.83
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
# Compare AIC scores
# Lambda = 0, no phylogenetic signal
phylosig(nj.rooted, nb, method = "lambda", test = TRUE)
```

```
##
## Phylogenetic signal lambda : 0.00698413
## logL(lambda) : 21.5034
## LR(lambda=0) : 0.00181764
## P-value (based on LR test) : 0.965993
```

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: The lambda value for the untransformed tree is 0, while the lambda value for the transformed tree is 0.006975. ***Answer 7b***: The untransformed model has an AIC score of -37.006827 while the transformed model has an AIC value of -37.00510. Since models with lower AIC scores are better, I would choose the untransformed model (but really both models could be used). ***Answer 7c***: Since the models AIC values do not differ in a value greater than or equal to 2, the models are considered equivalent. This is also indicated by a p-value of 0.96 which indicates that these models are not statistically different.

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 branch lengths for zeros
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7
# Calculate Blomberg's K for resources
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 for Blomberg's K
for (i in 1:18){
  x <- setNames(as.vector(p.growth.std[,i]), row.names(p.growth))
  out <- phylosignal(x, nj.rooted)
  p.phylosignal[1:5, i] <- round(t(out), 3)
}

# Use BH method to correct for FDR
p.phylosignal[6,] <- round(p.adjust(p.phylosignal[4, ], method = "BH"), 3)
print(p.phylosignal)
```

```
##                   AEP      PEP      G1P      G6P  MethCP      BGP      DNA
## K               0.000    0.000    0.000    0.000   0.000    0.000    0.000
## PIC.var.obs  4050.685  659.175  926.262 5887.270 350.859  510.561  237.150
## PIC.var.mean 7211.145 1374.523 1654.230 3305.577 433.660 1555.610 4438.741
## PIC.var.P       0.301    0.107    0.175    0.796   0.431    0.054    0.001
## PIC.var.z      -0.709   -1.149   -0.965    1.046  -0.251   -1.490   -1.144
## PIC.P.BH        0.602    0.321    0.450    0.843   0.705    0.194    0.018
##                  Peth    Pchol       B1     Phyt      SRP     cAMP      ATP
## K               0.000    0.000    0.000    0.000    0.000    0.000    0.000
## PIC.var.obs   192.520  397.370 3357.131 9230.269 1166.316  678.817 3942.591
## PIC.var.mean 1666.100 2935.811 4808.485 8041.456 1459.277 2729.591 2728.489
## PIC.var.P       0.004    0.009    0.274    0.632    0.340    0.007    0.662
## PIC.var.z      -1.816   -1.522   -0.665    0.154   -0.515   -2.343    0.536
## PIC.P.BH        0.036    0.040    0.602    0.794    0.612    0.040    0.794
##               PhenylCP    PolyP      GDP      GTP
## K                0.000    0.000    0.000    0.000
## PIC.var.obs   1224.017 1081.902 4469.581 2714.560
## PIC.var.mean   678.432 1107.452 3116.350 2545.163
## PIC.var.P       0.849    0.542    0.732    0.606
## PIC.var.z       1.167   -0.046    0.665    0.129
## PIC.P.BH        0.849    0.794    0.824    0.794
```

```
# Calculate Blomberg's K for nb
signal.nb <- phylosignal(nb, nj.rooted)
signal.nb
```

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 3.608803e-06         48546.48              44187.34          0.612
##   PIC.variance.Z
## 1      0.2348881
```

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***: Niche breadth does not have a significant phylogenetic signals but for the growth on phosphorous resources, DNA, Peth, Pchol there is a significant p-value with a K value of 0.
> ***Answer 8b***: Because the K value is less than 1, we would conslude that the phylogenetic signal results are suggestive of overdispersion.

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.

```
# Continuous to categorical
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)

# P 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
```

```
# Add column names to data
p.growth.pa$name <- rownames(p.growth.pa)

# Merge trait and phylogenetic data
p.traits <- comparative.data(nj.rooted, p.growth.pa, "name")
phylo.d(p.traits, binvar = AEP, permut = 10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  AEP
##   Counts of states:  0 = 19
##                      1 = 20
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.5323292
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0168
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0111
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  PhenylCP
##   Counts of states:  0 = 33
##                      1 = 6
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.8885263
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.3577
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0137
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  DNA
##   Counts of states:  0 = 20
##                      1 = 19
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.5205028
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0146
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0139
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  cAMP
##   Counts of states:  0 = 10
##                      1 = 29
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.1055551
## Probability of E(D) resulting from no (random) phylogenetic structure :  7e-04
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.3527
```

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***: AMP has an esimated D of 0.53 indicating that it is significantly overdispersed. DNA is 0.52 which is also overdispersed and cAMP is 0.09 which indicates that it is overdispersed. ***Answer 9b***: These results agree with with the results from the Blomberg's K analysis because the associated values all indicate overdispersion. ***Answer 9c***: The way in which these metrics are calculated may give rise to differences between their resulting values. For example, Blomberg's K uses values for traits from the phylogenetic-corrected mean based on variance and covariance within the matrix while the calculation for the dispersion of a trait relies on categorical data followed by a permutation test.

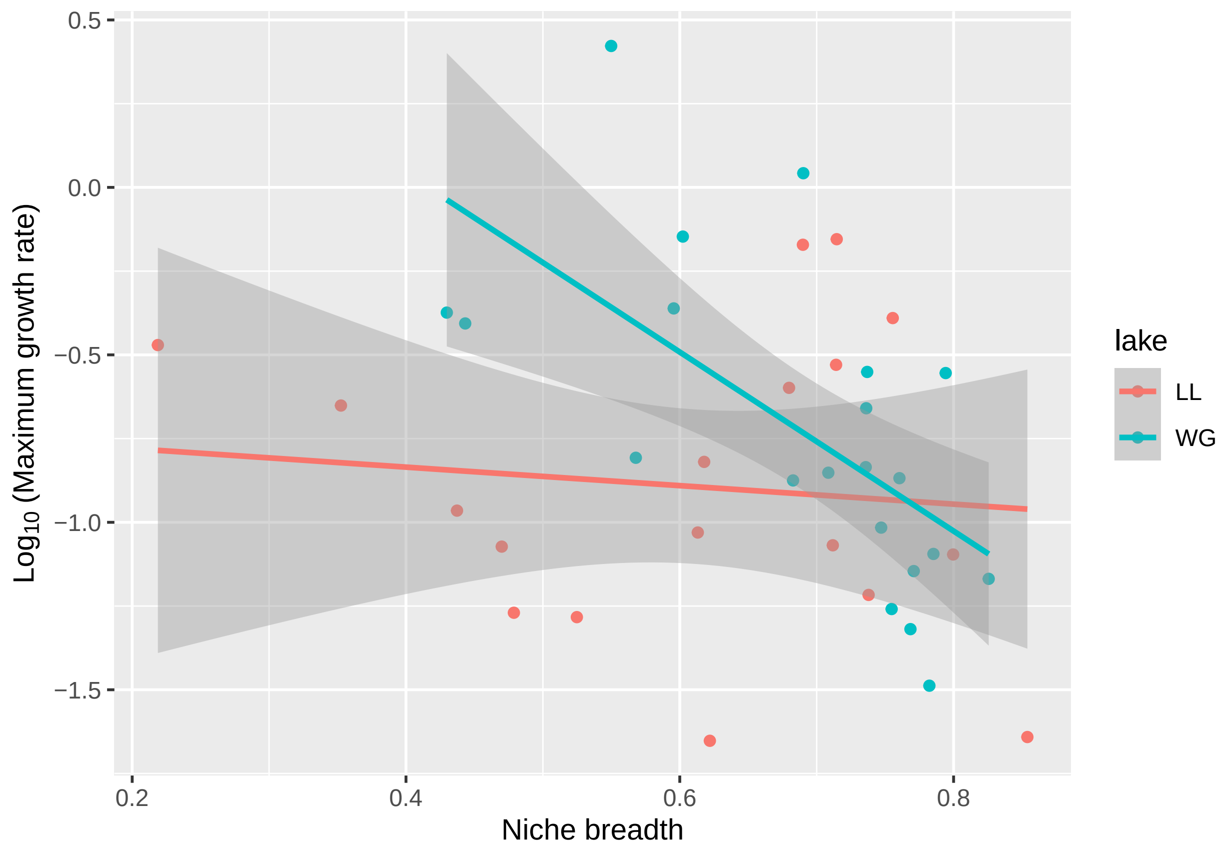
7) PHYLOGENETIC REGRESSION

In the R code chunk below, do the following:
1. Load and clean the resource use dataset and perform a linear regression to test for differences in maximum growth rate by nb and environment, 2. Fit a linear model to the trait dataset, examining the relationship between maximum growth rate by nb and lake environment, 2. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny

```r
# create 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")
}
# Add a column name to niche breadth values
colnames(nb.lake)[1] <- "NB"

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

```
# Plot maximum growth rate 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
```

```
# Run a phylogeny-corrected regression with no bootstrap replicates
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: 0.1814508
## Parameter estimate(s) using ML:
## lambda : 0.4861386
## sigma2: 0.9184409
##
## Coefficients:
##               Estimate     StdErr t.value p.value
## (Intercept) -0.8912676  0.3700360 -2.4086 0.02142 *
## NB          -0.0048049  0.5213029 -0.0092 0.99270
## lakeWG       1.4389308  0.5772311  2.4928 0.01755 *
## NB:lakeWG   -1.9663889  0.8487018 -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=0.4861386.
```

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 10a: We need to correct for shared evolutionary history because it violates the assumption of independence which is needed in order to perform a linear regression. The way to correct

for it is to perform a phylogenetic regression.
Answer 10b: Phylogenetic regression differs from a standard linear regression because the variance of the residual errors are described by a covariance matrix which accounts for branch lengths of the phylogeny. The standard linear regression the residuals are assumed to be independent and normally distributed. ***Answer 10c***: Accounting for shared evolutionary history is a better fit model because there is phylogenetic signal in the residuals. ***Answer 10d***: * Come back to this ***** More closely related bacteria can persist in harsh environments.

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**. 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? > Answer: Based on the F84 tree and the organism names on the tips of the tree branches, the tree does support the evolutionary history of the organisms because there are relationships within the branching of the tree and the genuses listed.

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? > Answer: Since we only used the top 40 taxa that were most abundant throughout the samples, it may be advantageous to look at more taxa to distinguish and characterize evolutionary history of the taxa within the lake.

```r
# The gene we are using is the 16S rRNA gene, since the data that we are working with has sequenced thi

# Align sequences
# We couldn't find the raw sequence data so we downloaded the aligned fasta file, deleted the alignment
# as the raw seq file, "40bac.fasta"
# Load packages needed for exercise
# package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger', 'picante', 'stats', 'RColorBrew
# for (package in package.list) {
#   if (!require(package, character.only=TRUE, quietly=TRUE)) {
#     install.packages(package)
#     library(package, character.only=TRUE)
#   }
# }
# # Loading/installing bioconductor
# if (!require("BiocManager", quietly = TRUE)) {
#   install.packages("BiocManager")
# }
# if(!require("msa", quietly = TRUE)) {
#   BiocManager::install("msa")
# }
#
# library(msa)
# # Read the alignment file
# pondseqs <- readDNAStringSet("data/40bac.fasta", format = 'fasta')
# # View sequences
# pondseqs
#
# # Align sequences using MUSCLE parameters (msa)
#
# library(msa)
# #ead.aln.pond <- msaMuscle(pondseqs)
# # Save and export
# save.aln.pond <- msaConvert(read.aln.pond, type = "bios2mds::align")
# export.fasta(save.aln.pond, "./data/40bac.afa")
#
# # Visualize the alignment
# # Convert alignment to DNAbin object
# p.DNAbin.pond <- as.DNAbin(read.aln.pond)
# # Command to visualize sequence alignment (ape)
# image.DNAbin(p.DNAbin.pond, cex.lab = 0.50)
#
# # Substitution model and make tree
# # Create distance matrix
# seq.dist.raw.pond <- dist.dna(p.DNAbin.pond, model = "raw", pairwise.deletion = FALSE)
#
# # Create neighbor joining tree
# njpond.tree <- bionj(seq.dist.raw.pond)
# # Defining outgroup and root of tree
# outgroup.pond <- match("Methanosarcina", njpond.tree$tip.label)
# # Root tree
# njpond.rooted <- root(njpond.tree, outgroup.pond, resolve.root = TRUE)
# # Plotting rooted tree
# par(mar = c(1,1,2,1) + 0.1)
# plot.phylo(njpond.rooted, main = "Neighbor Joining Tree", "phylogram",
#            use.edge.length = FALSE, direction = "right", cex = 0.6,
#            label.offset = 1)
# add.scale.bar(cex = 0.7)
#
# # Using a substitution model
# # Create distance matrix with F84 model
# seq.dist.F84.pond <- dist.dna(p.DNAbin.pond, model = "F84", pairwise = FALSE)
#
# # Plot distances from different DNA substitution models
# par(mar = c(5, 5, 2, 1) + 0.1)
# plot(seq.dist.raw.pond, seq.dist.F84.pond,
#      pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7),
#      ylim = c(0, 0.7), xlab = "Raw Distance", ylab = "F84 Distance")
# abline(b = 1, a = 0, lty = 2)
# text(0.65, 0.6, "1:1")
#
# # Make NJ trees using DNA substituation models
# raw.tree.pond <- bionj(seq.dist.raw.pond)
# F84.tree.pond <- bionj(seq.dist.F84.pond)
#
# # Define outgroups
# raw.outgroup.pond <- match("Methanosarcina", raw.tree.pond$tip.label)
# F84.outgroup.pond <- match("Methanosarcina", F84.tree.pond$tip.label)
#
# # Root the trees
# raw.rooted.pond <- root(raw.tree.pond, raw.outgroup.pond, resolve.root = TRUE)
# F84.rooted.pond <- root(F84.tree.pond, F84.outgroup.pond, resolve.root = TRUE)
#
# # Make cophylogenetic plot
# layout(matrix(c(1, 2), 1, 2), width = c(1,1))
# par(mar = c(1, 0, 2, 1))
# plot.phylo(raw.rooted.pond, 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.pond, type = "phylogram", direction = "left",
#            show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
#            cex = 0.6, label.offset = 2, main = "F84")
#
# # Calculate branch length score
# dist.topo(raw.rooted.pond, F84.rooted.pond, method = "score")
```

‘‘‘