8. Worksheet: Phylogenetic Diversity - Traits

Madison Stoltz; 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/madisonstoltz/GitHub/QB2023_Stoltz/2.Worksheets/8.PhyloTraits"
```

```
setwd("~/GitHub/QB2023_Stoltz/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: 'phytools'

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

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

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

##
## 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:phytools':
##
##     rescale

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

```
#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
```

```
library(msa)
```

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.

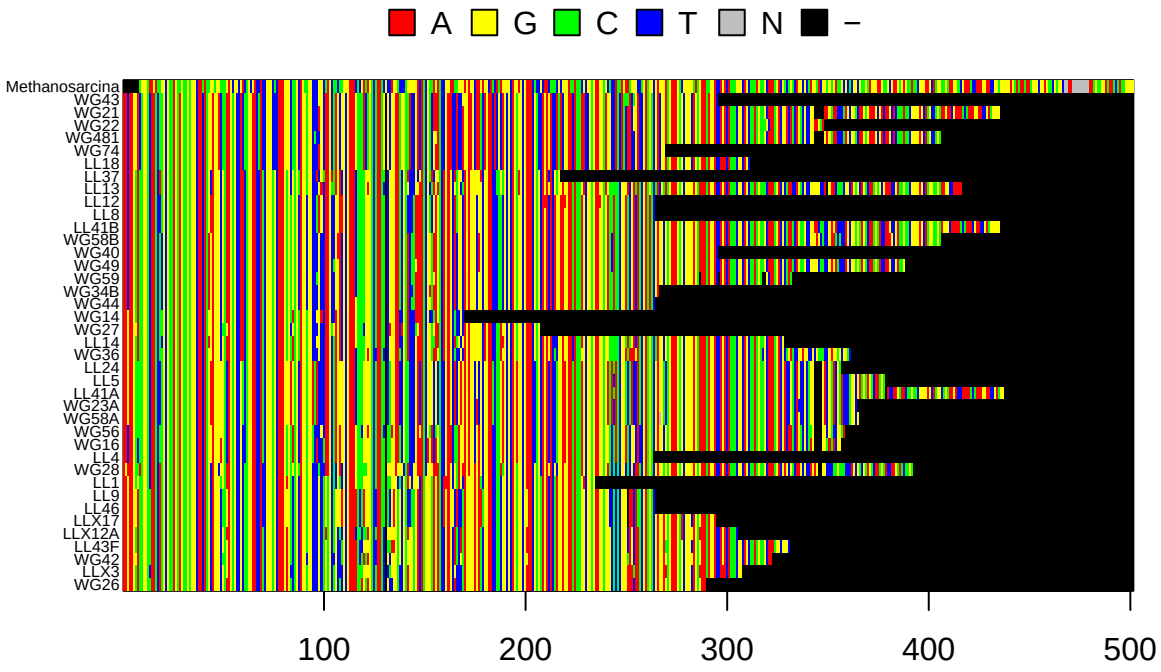
> ***Answer 1***: In the 'afa' file there appears to be many gaps, varying in sizes, within its sequences. Contrastingly, the 'fasta' file does not appear to have any gaps in its sequences.

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.

```
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
```

```
read.aln <- msaMuscle(seqs)
save.aln <- msaConvert(read.aln, type="bios2mds::align")
export.fasta(save.aln, "./data/p.isolates.afa")

p.DNAbin <- as.DNAbin(read.aln)
window <- p.DNAbin[, 500:1000]
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***: After moving the window to 500:1000, I can visually see that most sequences end around 350. ***Answer 2b***: The black coloration shows alignment gaps. I would choose regions that do not have these alignment gaps to make the best phylogenetic inference possible.

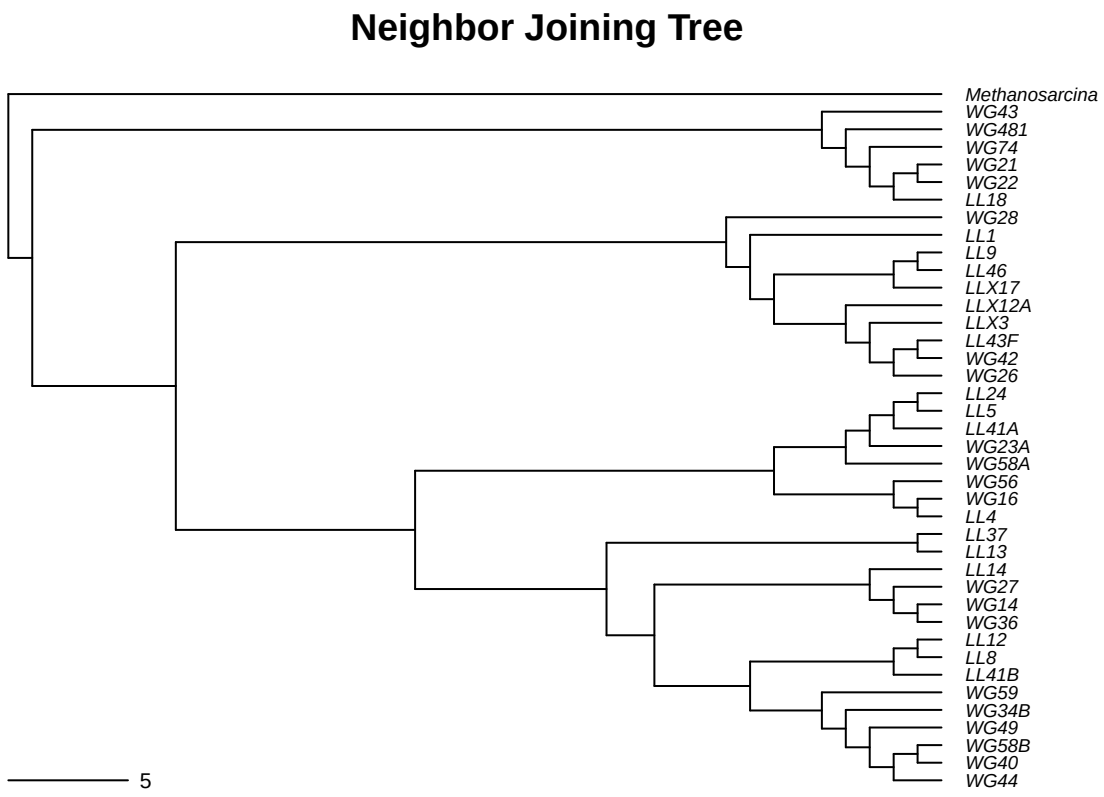
4) MAKING A PHYLOGENETIC TREE

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.

```
seq.dist.raw <- dist.dna(p.DNAbin, model="raw", pairwise.deletion = FALSE)
nj.tree <- bionj(seq.dist.raw)
outgroup <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root=TRUE)
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)
```



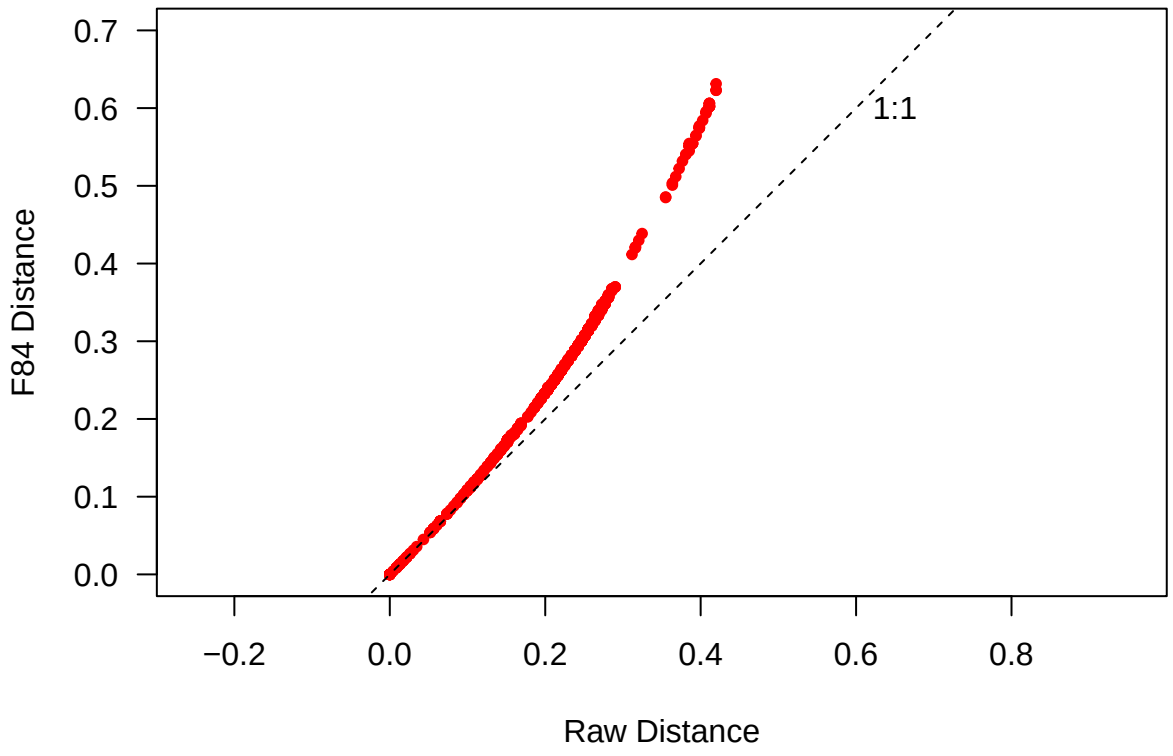
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: A neighbor joining tree has raw estimates of phylogenetic distance and does not account for the fact that multiple substitutions may have occured at the same site over time.

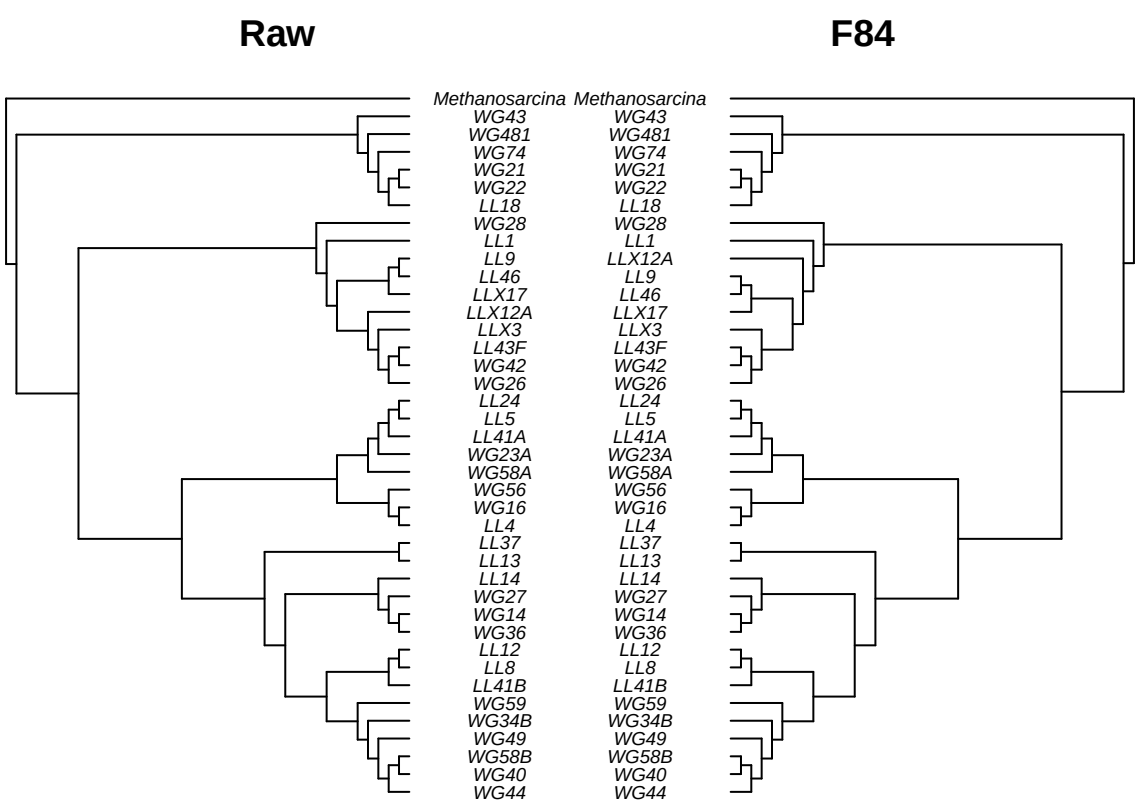
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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")
```

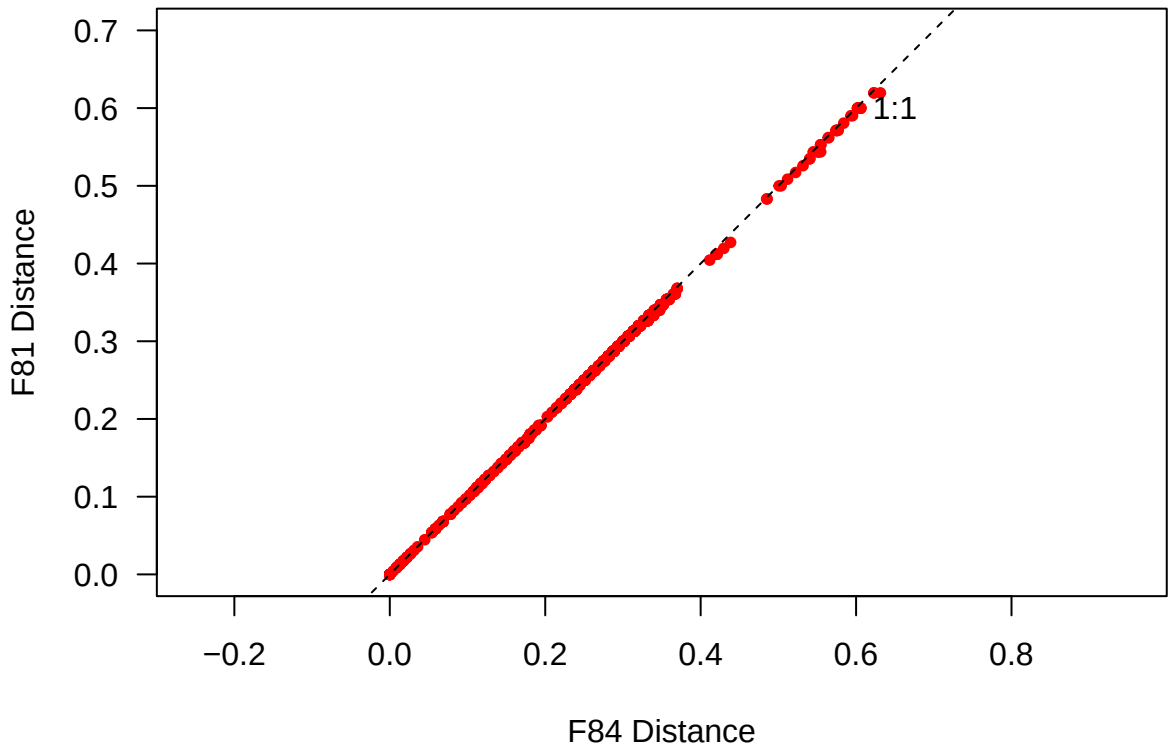



```
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)

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

raw.rooted <- root(raw.tree, raw.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)

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")
```

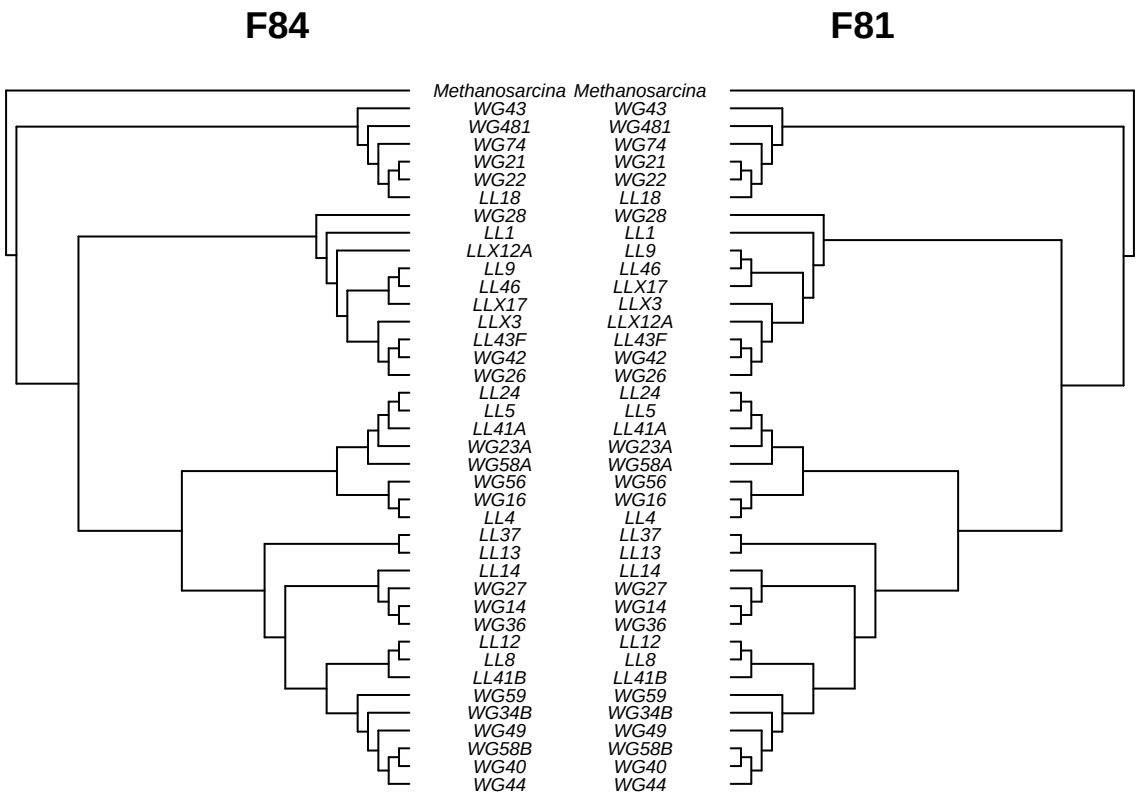


In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5.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

```
#Use F81
seq.dist.F84 <- dist.dna(p.DNAbin, model="F84", pairwise.deletion = FALSE)
seq.dist.F81 <- dist.dna(p.DNAbin, model="F81", pairwise.deletion = FALSE)

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




```
F81.tree <- bionj(seq.dist.F81)
F84.tree <- bionj(seq.dist.F84)

F81.outgroup <- match("Methanosarcina", F81.tree$tip.label)
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)

F81.rooted <- root(F81.tree, F81.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)

layout(matrix(c(1,2), 1, 2), width=c(1, 1))
par(mar=c(1, 1, 2, 0))
plot.phylo(F84.rooted, type="phylogram", direction="right",
           show.tip.label=TRUE, use.edge.length = FALSE, adj=0.5,
           cex=0.6, label.offset=2, main="F84")
par(mar=c(1,0,2,1))
plot.phylo(F81.rooted, type="phylogram", direction="left",
           show.tip.label=TRUE, use.edge.length = FALSE, adj=0.5,
           cex=0.6, label.offset=2, main="F81")
```



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: I chose the Felsenstein model F81. This model builds from JC69 by allowing nucleotide frequencies to vary. The Felsenstein model F84 assumes different rates of base transitions and transversion while allowing for differences in base frequencies. ***Answer 4b***: The two plots are extremlely alike, except for one or two changes. This could be due to a small difference in nucleotide frequencies. ***Answer 4c***: As mentioned above, there is likely a difference in the substitution rates of nucleotide transitions.

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

```
#page 10
dist.topo(raw.rooted, F84.rooted, method = "score")
```

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

```
#page 11
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)

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
```

```
fitGTR <- optim.pml(fit, model="GTR", optInv=TRUE, optGamma=TRUE,
          rearrangement = "NNI", control = pml.control(trace=0))
```

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

```
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
```

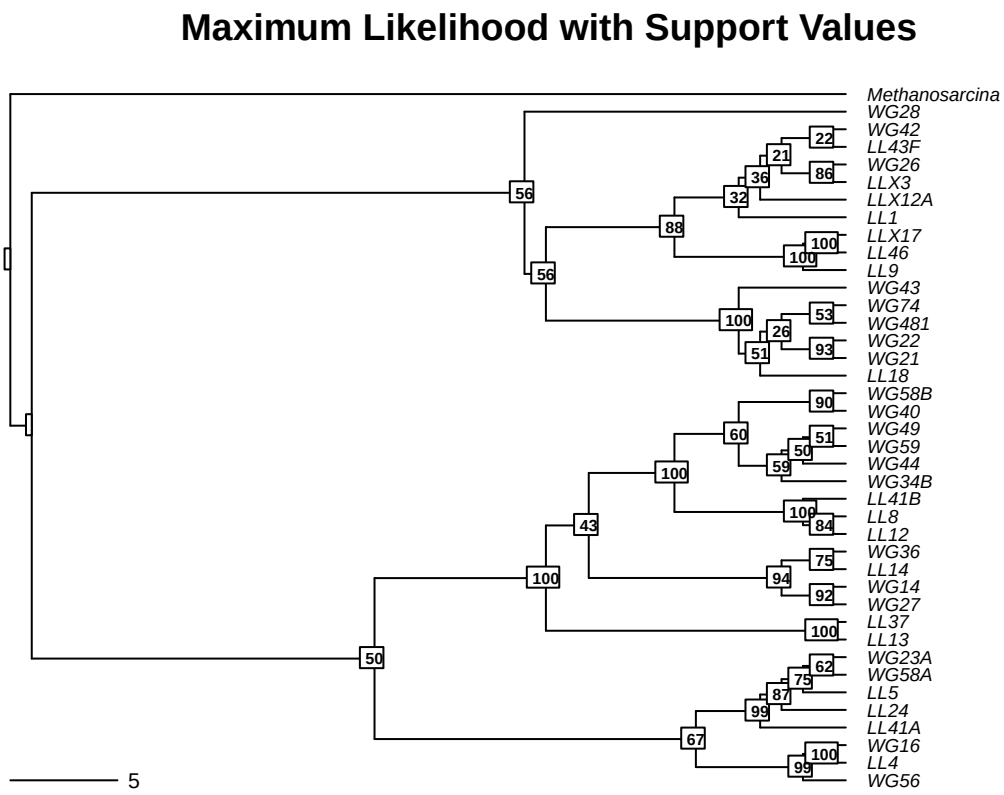
```
AIC(fitJC)
```

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

```
AIC(fitGTR)
```

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

```
#page 12
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 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 likelihood tree is much more reliable than the neighbor-joining tree, because the maximum likelihood tree is built on the robust statistical procedue of ML. ***Answer 5b***: We bootstrap to see how confident we can be in our tree. ***Answer 5c***: The bootstrap values will tell us how close two trees are to eachother. We will want a high value for similar trees. ***Answer 5d***: Original tree branches with a score of 0 show they are different from the bootstrap tree, which show low support. ***Answer 5e***: We should not trust these brances, because they are different. We want branches with a score of 1, which show similarity between the original and the bootstrap.

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("./data/p.isolates.raw.growth.txt", sep="\t",
            header = TRUE, row.names = 1)
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.

```
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

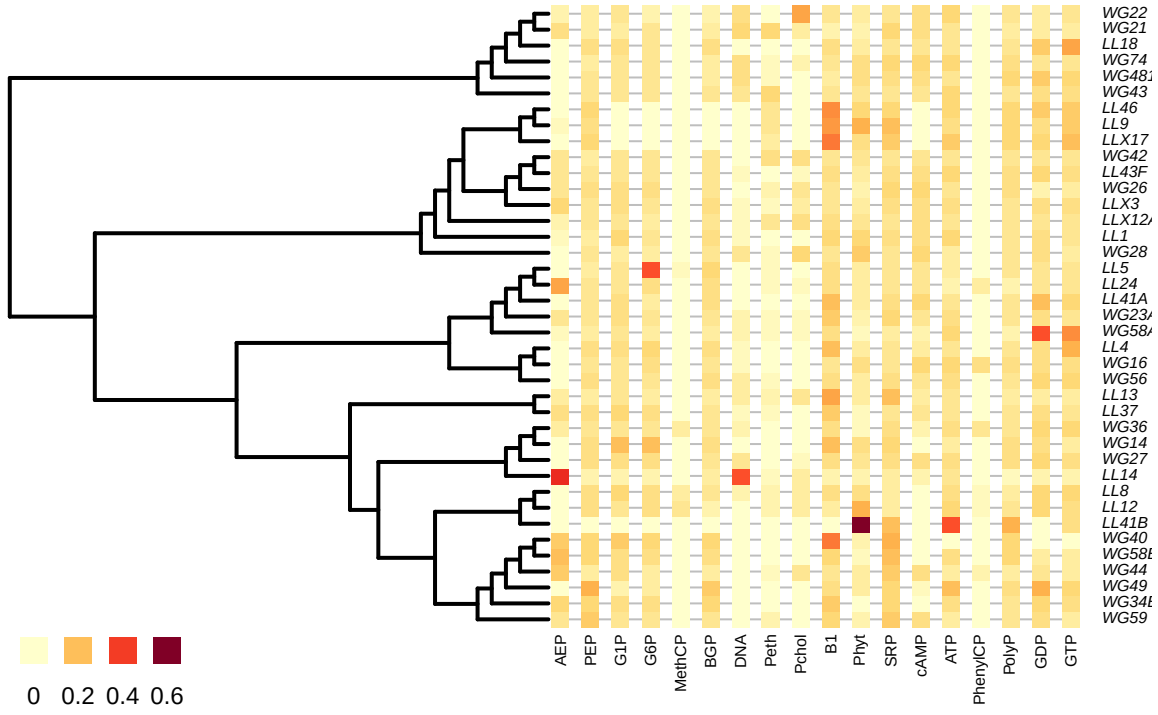
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```
nj.tree <- bionj(seq.dist.F84)
outgroup <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

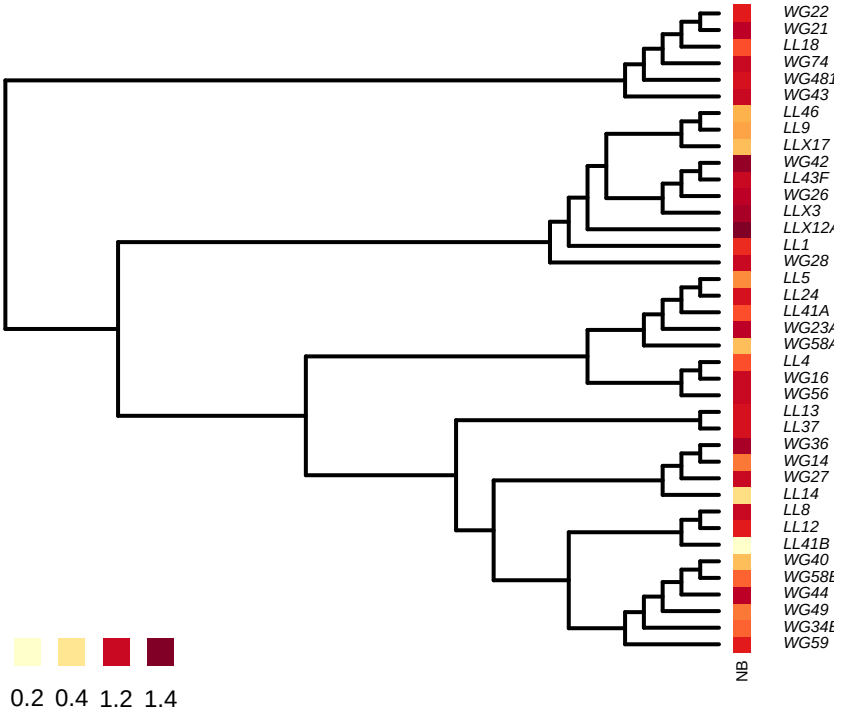
In the R code chunk below, do the following:
1. define a color palette (use something other than "YlOrRd"),
2. map the phosphorus traits onto your phylogeny,
3. map the nb trait on to your phylogeny, and
4. customize the plots as desired (use `help(table.phylo4d)` to learn about the options).

```
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)
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, cex.symbol=1.25,
              ratio.tree=0.5, cex.legend=1.5, center = FALSE)
```




```
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, cex.symbol = 1.25, var.label = ("NB"), ratio.tree = 0.90,
      cex.legend = 1.5, center = FALSE)
```



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: A generalist can survive under a wider range of conditions in comparison to a specialist. Due to this, it is possible that increased branching off depicts a specialist species. ***Answer 6b***: Having a high growth rate and larger niche breadth could mean that the species is a specialist and not a generalist.

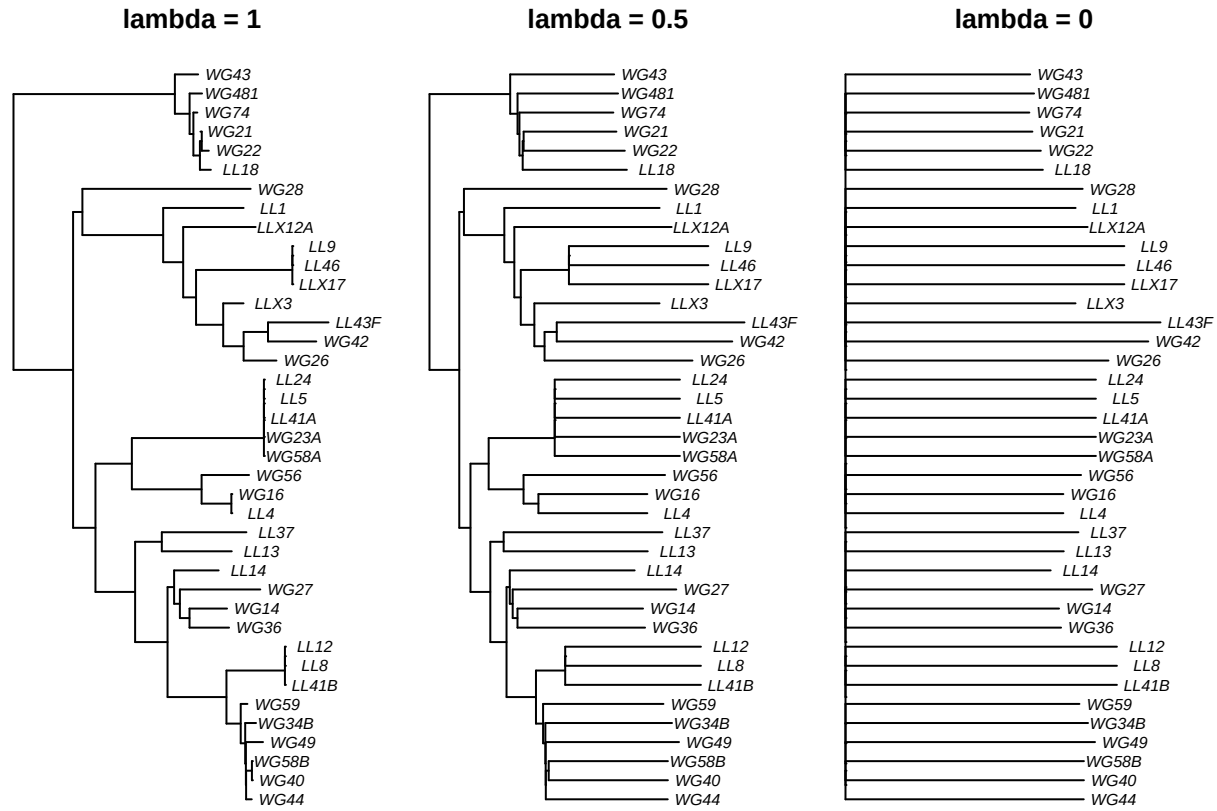
6) HYPOTHESIS TESTING

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.

```
nj.lambda.5 <- geiger::rescale(nj.rooted, "lambda", 0.5)

#top page 17
nj.lambda.0 <- geiger::rescale(nj.rooted, "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.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)
```



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

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.006975
##  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 = 60
##  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 = 85
##  frequency of best fit = 0.85
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
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***: Lambda is 0 and 0.006975. ***Answer 7b***: The AIC scores are very close and both negative values. Because the models' AIC values are not greater than 2, the two models are considered equivalent. ***Answer 7c***: No, there is not a phylogenetic signal (p-value = 0.965993).

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.

```
#page 18
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7

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 (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)
 }

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 7568.466 1395.832 1696.552 3243.841 443.889 1511.066 4518.958
## PIC.var.P       0.276    0.130    0.155    0.813   0.442    0.037    0.001
## PIC.var.z      -0.767   -1.084   -1.076    1.064  -0.268   -1.554   -1.176
## PIC.P.BH        0.552    0.390    0.399    0.861   0.723    0.133    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 1678.116 2979.249 4792.023 7889.392 1432.053 2726.048 2634.962
## PIC.var.P       0.008    0.009    0.256    0.633    0.331    0.003    0.698
## PIC.var.z      -1.843   -1.512   -0.683    0.177   -0.489   -2.279    0.596
## PIC.P.BH        0.040    0.040    0.552    0.807    0.596    0.027    0.807
##               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  691.181 1075.432 3183.467 2551.515
## PIC.var.P       0.862    0.575    0.717    0.593
## PIC.var.z       1.200    0.012    0.619    0.128
## PIC.P.BH        0.862    0.807    0.807    0.807
```

```
#page 19
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              43844.72          0.638
```

```
##   PIC.variance.Z
## 1      0.2420708
```

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: No, there is not a significant phylogenetic signal for niche breadth. However, there is a sig phylogenetic signal for DNA, Peth, Pchol, cAMP. ***Answer 8b***: All the K values are 0 or less (less than 1), meaning they are overdispersed.

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.

```
#top of page 20
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)
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
```

```
p.growth.pa$name <- rownames(p.growth.pa)

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 :  0.5356366
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.017
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.013
```

```
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 :  0.8946609
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.3637
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0124
```

```
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 :  0.5173439
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0134
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0164
```

```
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 :  0.0961074
## Probability of E(D) resulting from no (random) phylogenetic structure :  1e-04
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.3686
```

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.

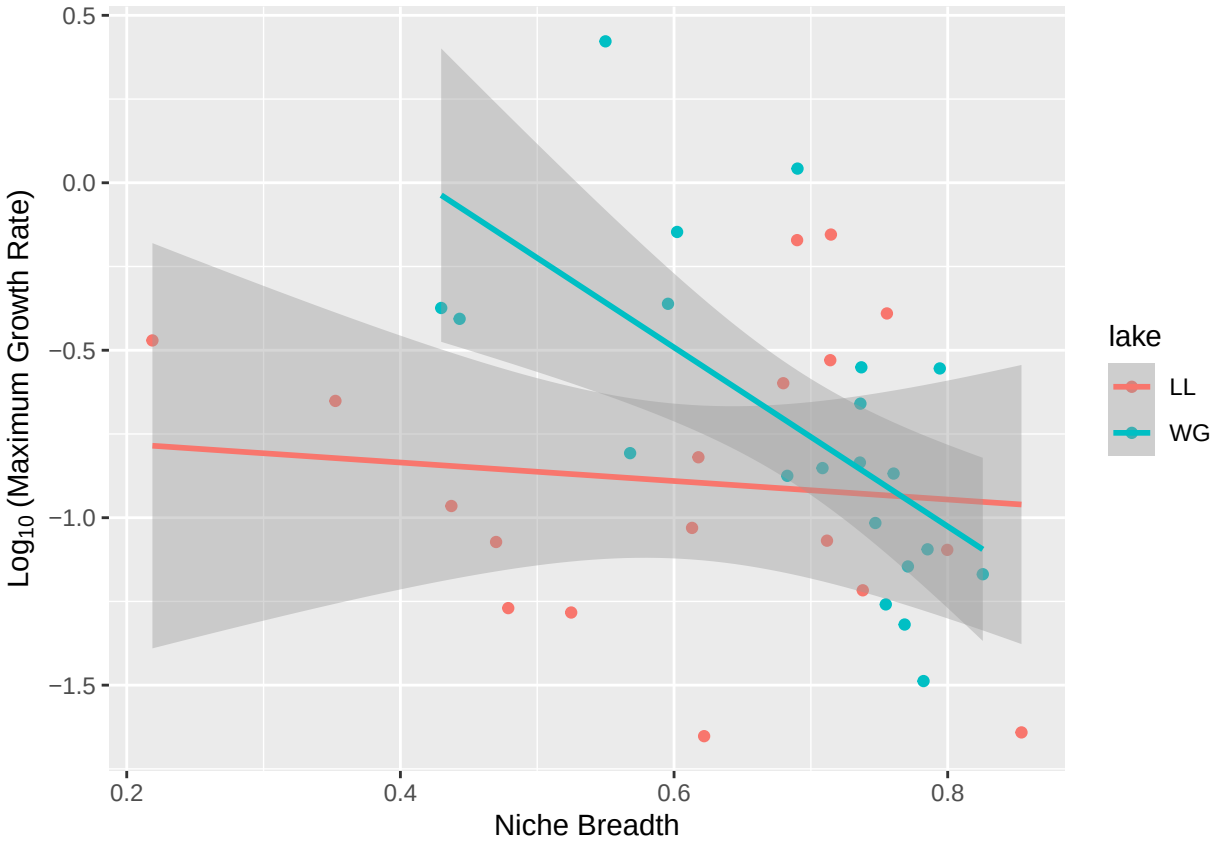
Answer 9a: AEP 0.53 -> overdispered, DNA 0.52 -> overdispered, cAMP 0.09 -> overdispered ***Answer 9b***: These results agree with the results from the Blomberg's K analysis. ***Answer 9c***: The difference in metrics is likely due to the difference in trait dispersion (categorial vs non-categorial).

7) PHYLOGENETIC REGRESSION

In the R code chunk below, do the following:
1. EDITED VIA SLACK

```
#EDITED - page 25 until end
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")
}

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'
```




```
#page 26
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: 0.1814508
## 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.
```

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

```
## [1] 41.07572
```

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: Shared evolutionary history violates the assumption of independence for regression analysis. ***Answer 10b***: A SLR assumes independent residual errors and a PR the variance of the residual errors are described by a covariance matrix. ***Answer 10c***: Accounting for shared evolutionary history improves the fit. ***Answer 10d***: A scenario like this could be convergent evolution.

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.

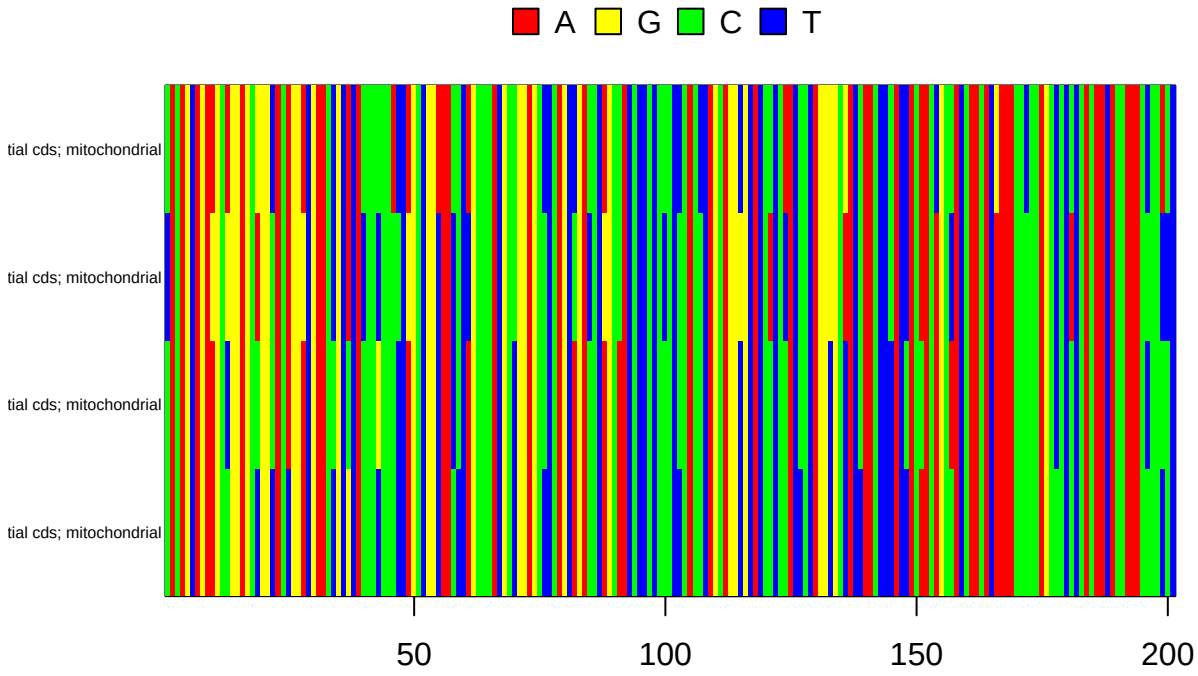
But before your team proceeds, you need to give some thought to which gene you want to focus on. For microorganisms like the bacteria we worked with above, many people use the ribosomal gene (i.e., 16S rRNA). This has many desirable features, including it is relatively long, highly conserved, and identifies 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? 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?

```
#We have 4 seabird species: Brown noddy, Red footed booby, Wedge tailed shearwater, White tern
#Upload fasfa seabird data set

#align
seqs2 <- readDNAStringSet("data/QB Data Seabirds.fasta", format='fasta')
seqs2
```

```
## DNAStringSet object of length 4:
##     width seq                                               names
## [1]   744 NNCCTATACCTAATCTTTGGTGC...ANANANANANANANANANANANA gb|JF498904.1|:1-...
## [2]   744 NNNNNNNNNCTAATTTTTGGCGC...ANANANANANANANANANANANA gb|MK262611.1|:1-...
## [3]   744 ACCCTGTACCTAATCTTTGGTGC...ANANANANANANANANANANANA gb|DQ433312.1|:1-...
## [4]   744 NNCCTGTATCTAATTTTCGGCGC...ANANANANANANANANANANANA gb|JQ174974.1|:1-...
```

```
read.aln <- msaMuscle(seqs2)
save.aln <- msaConvert(read.aln, type="bios2mds::align")
export.fasta(save.aln, "./data/QB Data Seabirds.fasta")

p.DNAbin <- as.DNAbin(read.aln)
window <- p.DNAbin[, 300:500]
image.DNAbin(window, cex.lab=0.50)
```




```
#Unable to create tree, need to consider an outgroup going forward
```

Answer: In the future we could include sequences from similar seabird species found in the region to expand our tree beyond four species.