8. Worksheet: Phylogenetic Diversity - Traits

Atalanta Ritter; 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.pdf**).

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/Atalanta/GitHub/QB2023_Ritter/2.Worksheets/8.PhyloTraits"
```

```
setwd("~/GitHub/QB2023_Ritter/2.Worksheets/8.PhyloTraits")
package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger',
'picante', 'stats', 'RColorBrewer', 'caper', 'phylolm', 'pmc',
'ggplot2', 'tidyr', 'dplyr', 'phangorn', 'pander', 'phytools', 'vegan',
'cluster', 'dendextend', 'phylogram', 'bios2mds')
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

##
## 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
```

```
# Some bioinformatics packages come through BioConductor
# Requires own installation method
if (!require("BiocManager", quietly = TRUE)) {
install.packages("BiocManager")
}
```

```
## Bioconductor version '3.14' is out-of-date; the current release version '3.16'
##   is available with R version '4.2'; see https://bioconductor.org/install
```

```
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, 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

## Warning: package 'S4Vectors' was built under R version 4.1.3

##
## 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

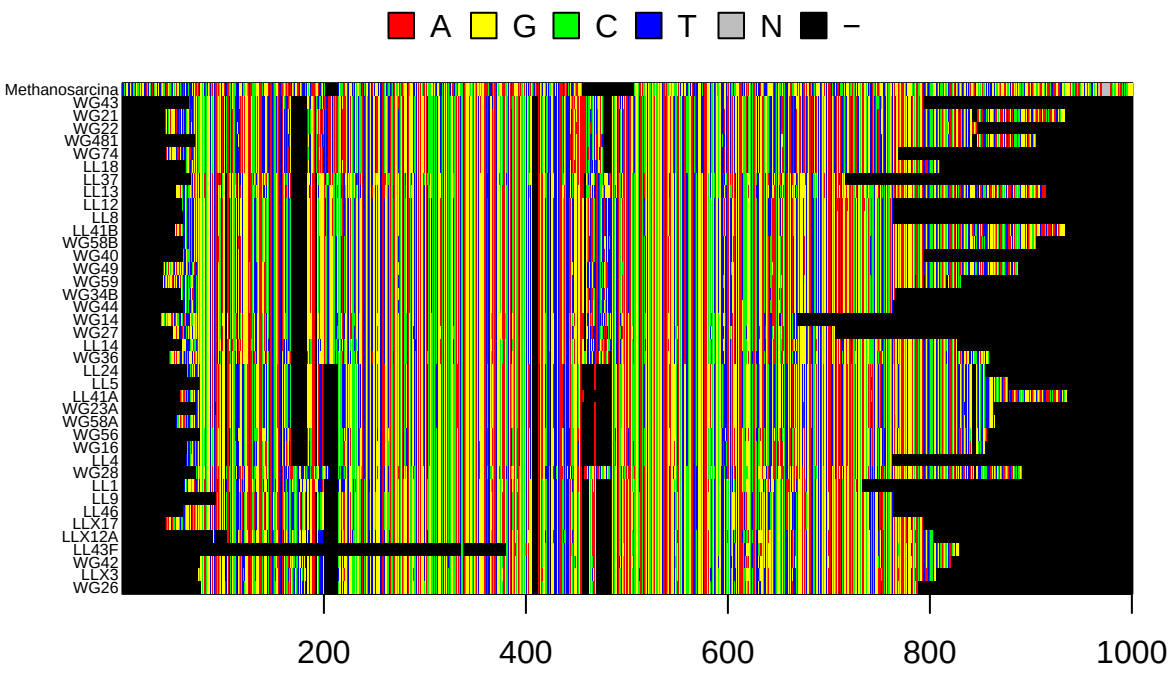
Question 1: Using your favorite text editor, compare the `p.isolates.fasta` file and the `p.isolates.afa` file. Describe the differences that you observe between the two files.

Answer 1:

In the 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.

```
# read alignment file
seqs <- readDNAStringSet("data/p.isolates.fasta", format = "fasta")
read.aln <- msaMuscle(seqs)
# convert alignment to DNAbin object
```

```
p.DNAbin <- as.DNAbin(read.aln)
# identify base pair region of 16S rRNA gene to visualize
window <- p.DNAbin[, 0:1000]
# visualzie sequence alignment
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: About 700 bps. Around 700, some samples no longer have data and by 800, most do not have data. ***Answer 2b***: Anywhere from 100-700 would be appropriate, since almost all samples have data in this window. Also there are lots of vertical lines of the same color, meaning samples have the same nucleotide across sites, but there are also chunks within those vertical lines that are of a different color, which suggests shared divergence among some sites.

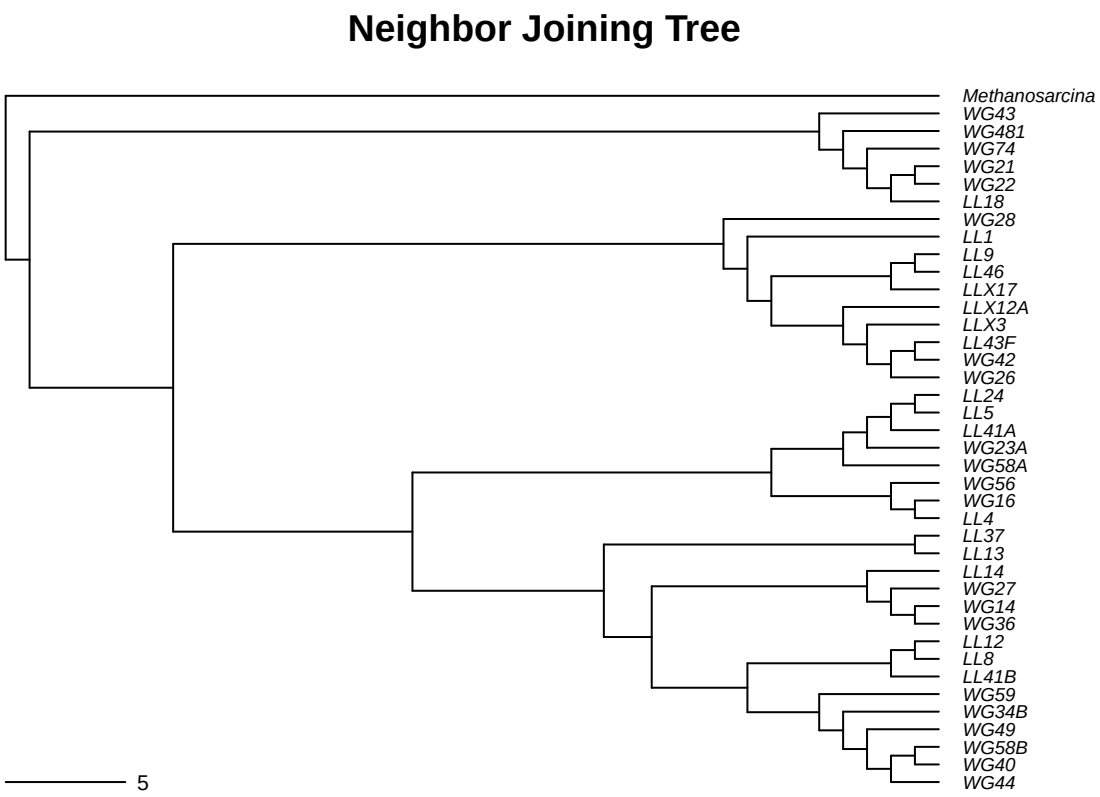
4) MAKING A PHYLOGENETIC TREE

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 with "raw" model
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)
# neighbor joining tree based on distances
nj.tree <- bionj(seq.dist.raw)
# define methanosarcina as outgroup
outgroup <- match("Methanosarcina", nj.tree$tip.label)
#root the tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
# plot rooted tree
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(nj.rooted, main = "Neighbor Joining Tree", "phylogram",
use.edge.length = FALSE, direction = "right", cex = 0.6,
label.offset = 1)
add.scale.bar(cex = 0.7)
```



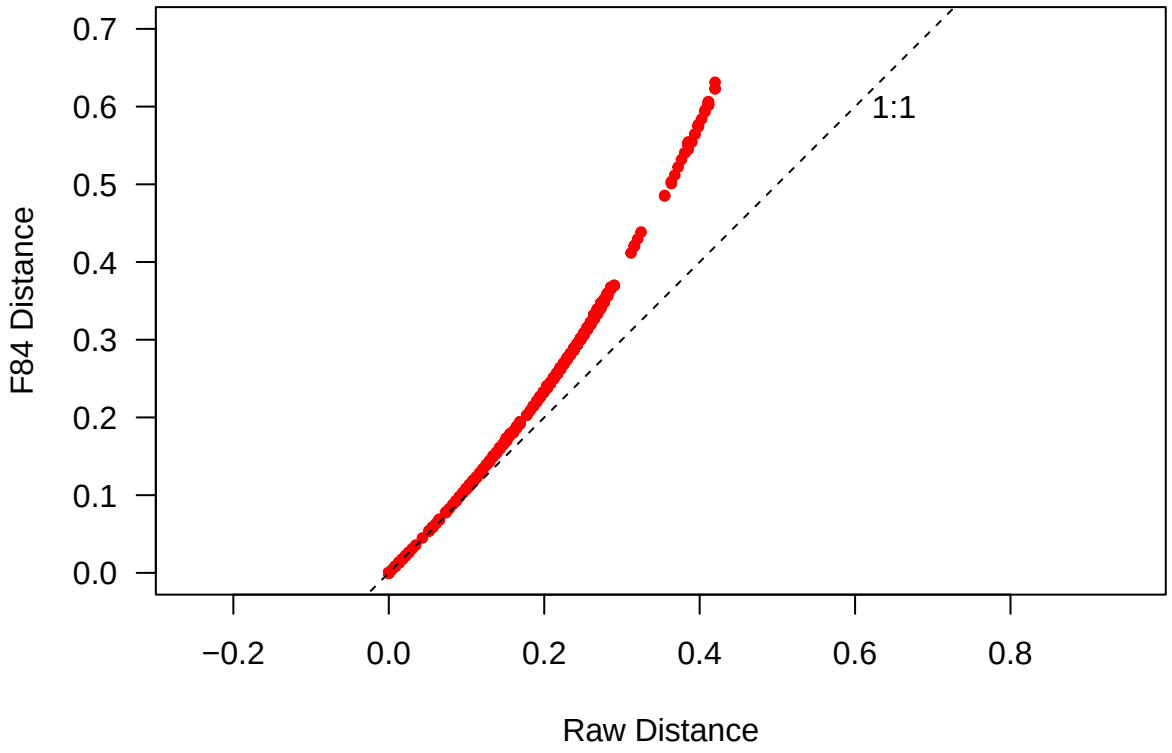
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: Neighbor joining trees can be a good starting point when making a phylogenetic tree and can give you a relatively quick, easy visualization of the evolutionary relationships between your data because the algorithm just uses a distance matrix. However, they do not take into account multiple substitutions at the same locus over time (i.e. it will underestimate the number of differences between taxa) and it does not take into account nucleotide substitution biases.

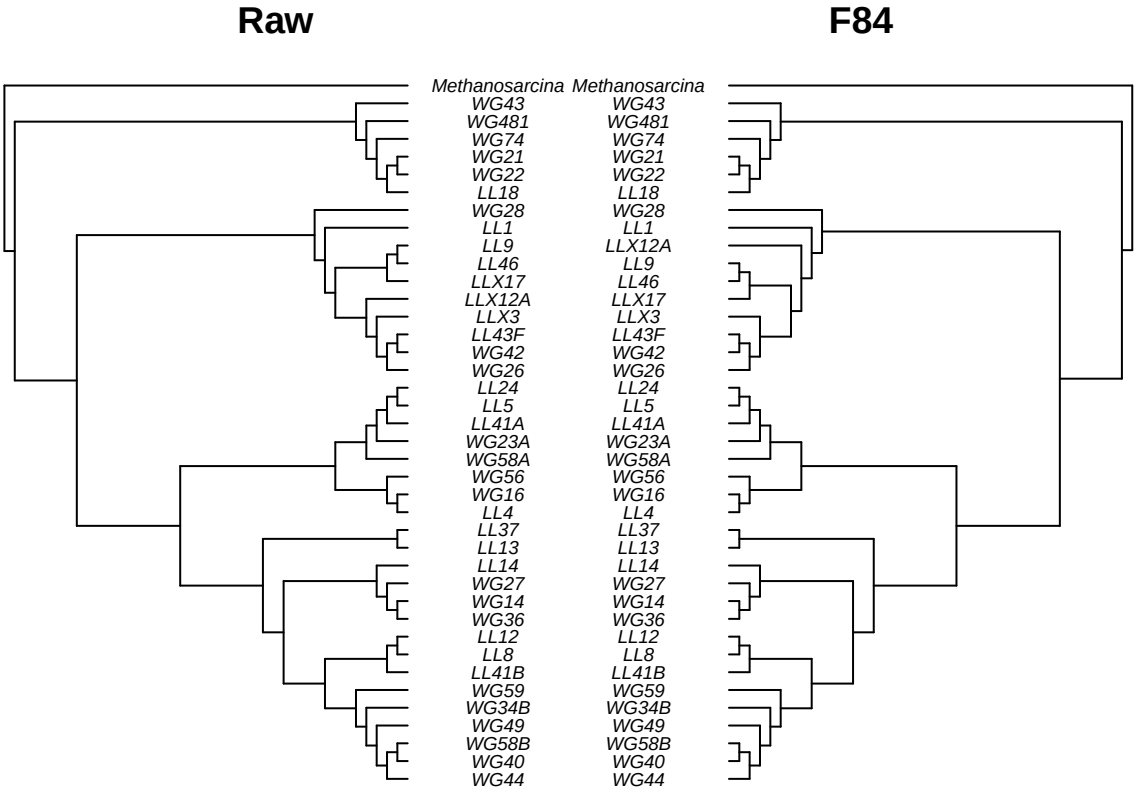
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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.

```
# distance matrix based on F84 model
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise.deletion = FALSE)
# Plot Distances from raw and F84 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")
```



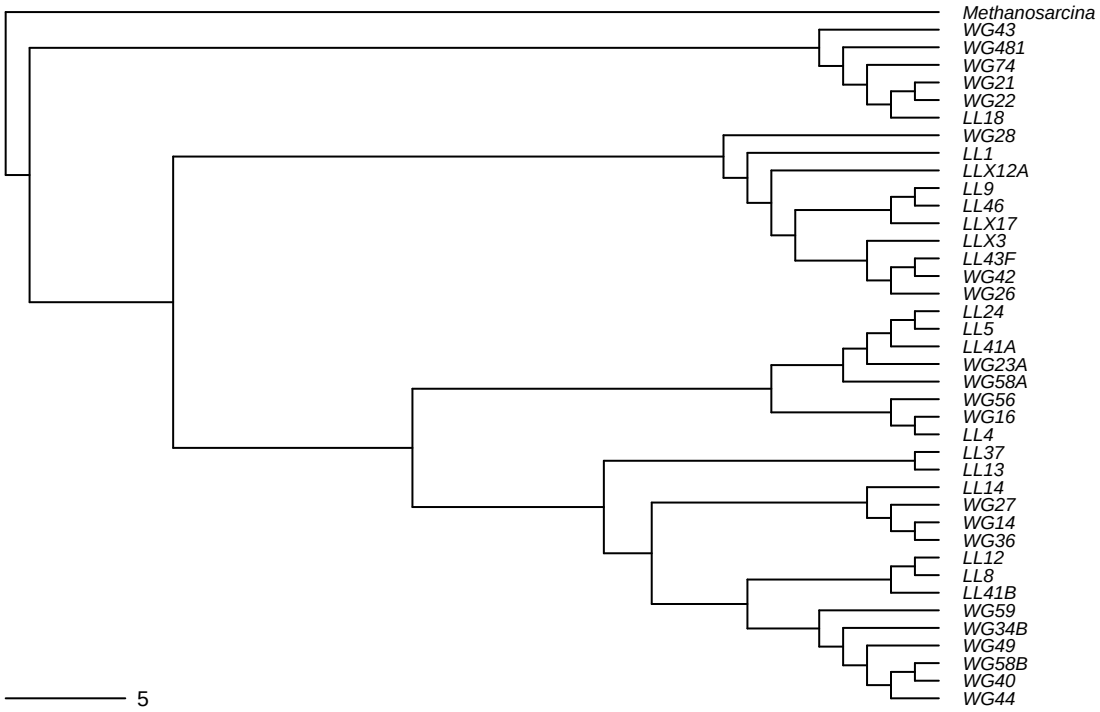

```
# Make Neighbor Joining Trees Using Different DNA Substitution Models {ape}
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)
# Define Outgroups
raw.outgroup <- match("Methanosarcina", raw.tree$tip.label)
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)
# Root the Trees {ape}
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)
# Make Cophylogenetic Plot {ape}
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")
```



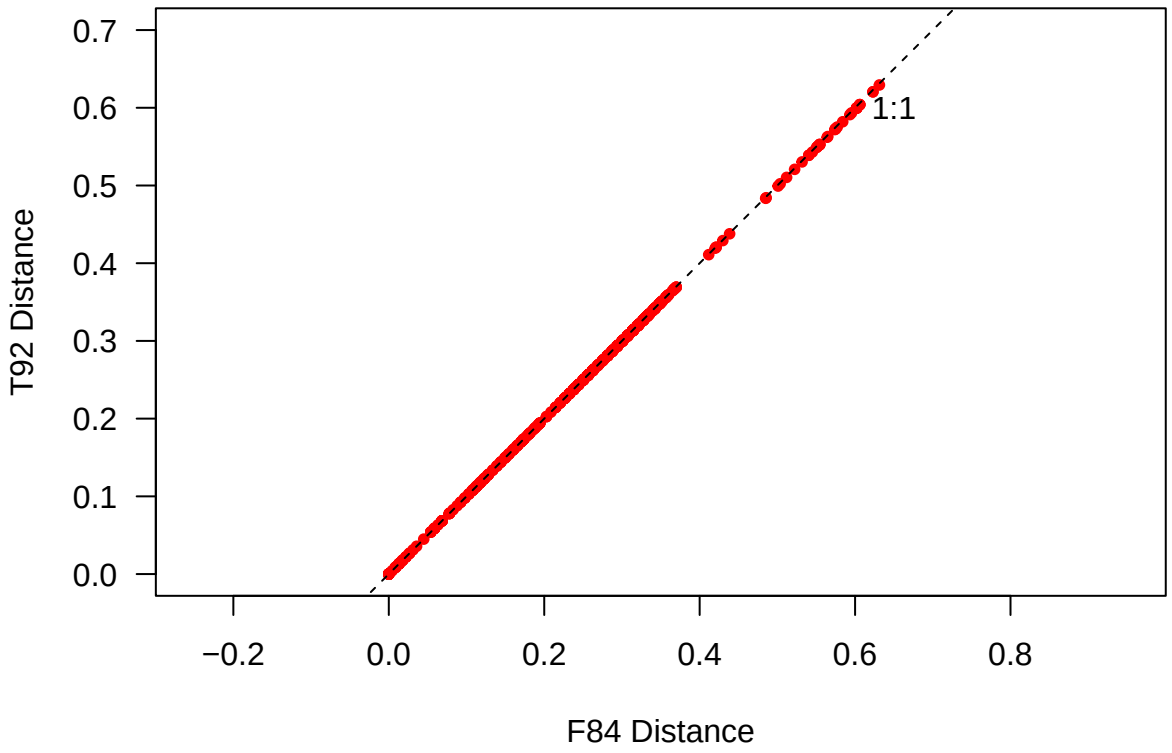
In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model, 4. make a cophylogenetic plot that compares the topologies of both models, and
5.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

```
# distance matrix based on T92 model
seq.dist.T92 <- dist.dna(p.DNAbin, model = "T92", pairwise.deletion = FALSE)
# make tree for T92 model
T92.tree <- bionj(seq.dist.T92)
T92.outgroup <- match("Methanosarcina", T92.tree$tip.label)
T92.rooted <- root(T92.tree, outgroup, resolve.root = TRUE)
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(T92.rooted, main = "T92 Model Tree", "phylogram",
use.edge.length = FALSE, direction = "right", cex = 0.6,
label.offset = 1)
add.scale.bar(cex = 0.7)
```

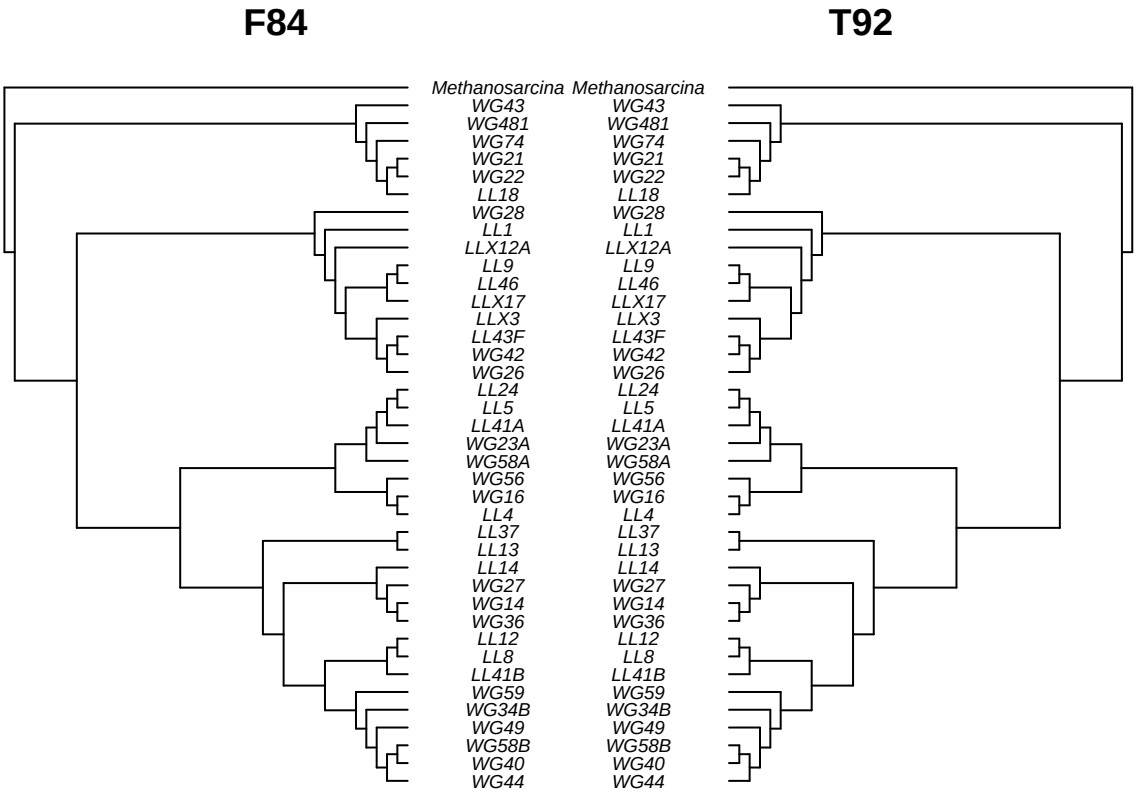
T92 Model Tree




```
# saturation plot comparing F84 and T92
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.F84, seq.dist.T92,
pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7),
ylim = c(0, 0.7), xlab = "F84 Distance", ylab = "T92 Distance")
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1")
```




```
T92.rooted <- root(T92.tree, T92.outgroup, resolve.root = TRUE)
# Make Cophylogenetic Plot {ape}
layout(matrix(c(1, 2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(F84.rooted, type = "phylogram", 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(T92.rooted, type = "phylogram", direction = "left",
           show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
           cex = 0.6, label.offset = 2, main = "T92")
```



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 Tamura model (T92), which accounts for G + C content and recognizes that transition mutations occur with higher probability than transversion mutations. It differs from the F84 model in that F84 allows for differences in base frequencies but does not specify G + C content. ***Answer 4b***: For the saturation plot, the points for the T92 model fall exactly on the line of the F84 model. The F84 and T92 models also give the exact same phylogenetic plot. ***Answer 4c***: These two models give the exact same output. Both of these models account for differences in substitution rate of nucleotide transitions. Compared with the earlier raw model, this means that accounting for substitution rate makes a huge difference on your results, but any further corrections (e.g. T92's inclusion of G+C content) will probably show diminishing returns.

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

```
# 1. Read in the maximum likelihood phylogenetic tree used in the handout.
# Requires alignment to be read in with as phyDat object
phyDat.aln <- msaConvert(read.aln, type = "phangorn::phyDat")
# Make the NJ tree for the maximum likelihood method.
# {Phangorn} requires a specific attribute (attr) class.
```

```
# So we need to remake our trees with the following code:
aln.dist <- dist.ml(phyDat.aln)
aln.NJ <- NJ(aln.dist)
fit <- pml(tree = aln.NJ, data = phyDat.aln)
# Fit tree using a 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
```

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

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

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

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

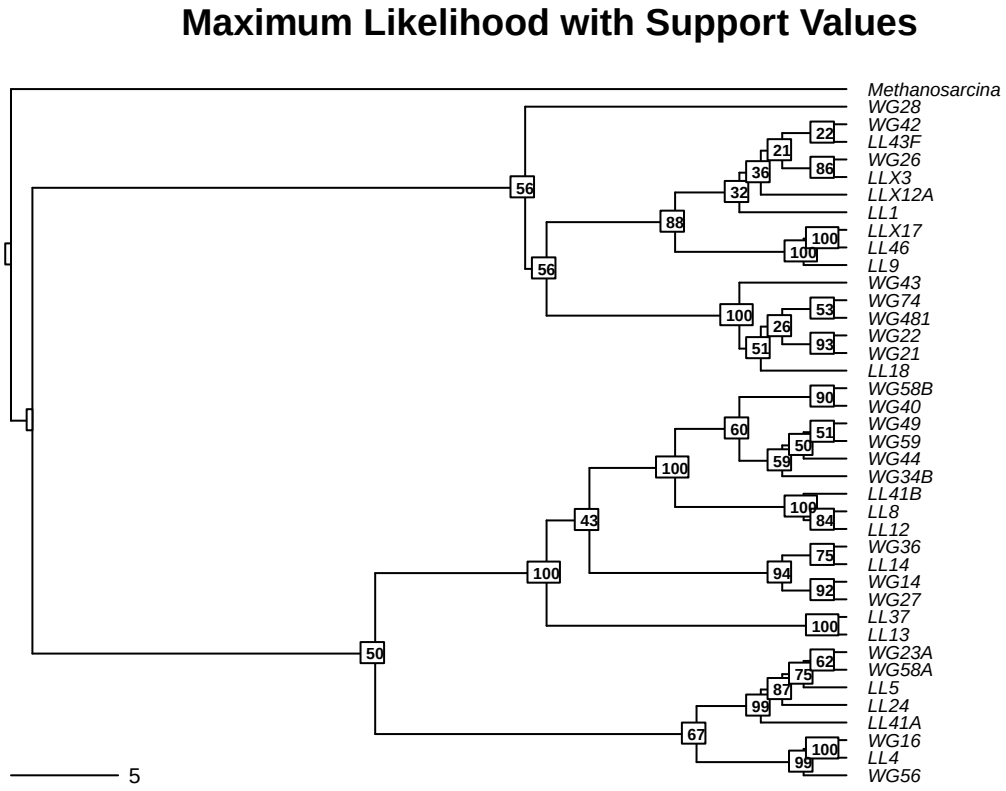
```
AIC(fitJC)
```

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

```
AIC(fitGTR)
```

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

```
# 2. Plot bootstrap support values onto the tree
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 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 and neighbor-joining tree are similar in overall shape and grouping, but the further you go into the clades, the more inconsistent the sample groupings are. This is probably because maximum likelihood trees use more robust methods than a simple distance matrix, so it is able to resolve some species relationships with more certainty. ***Answer 5b***: Bootstrapping allows us to test how reliable a phylogeny is. ***Answer 5c***: The value in a node represents the percentage of samples that placed that node where it is. Higher values mean that a high percentage of samples agree, suggesting that the node is reliable and probably true. ***Answer 5d***: The branches for WG42 and LL43F have very low values (22%). ***Answer 5e***: No, we shouldn't trust those branches.

5) INTEGRATING TRAITS AND PHYLOGENY

A. Loading Trait Database

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

```
# Import Growth Rate Data
p.growth <- read.table("./data/p.isolates.raw.growth.txt", sep = "\t",
```

```
                        header = TRUE, row.names = 1)
# Standardize Growth Rates Across Strains
p.growth.std <- p.growth / (apply(p.growth, 1, sum))
```

B. Trait Manipulations

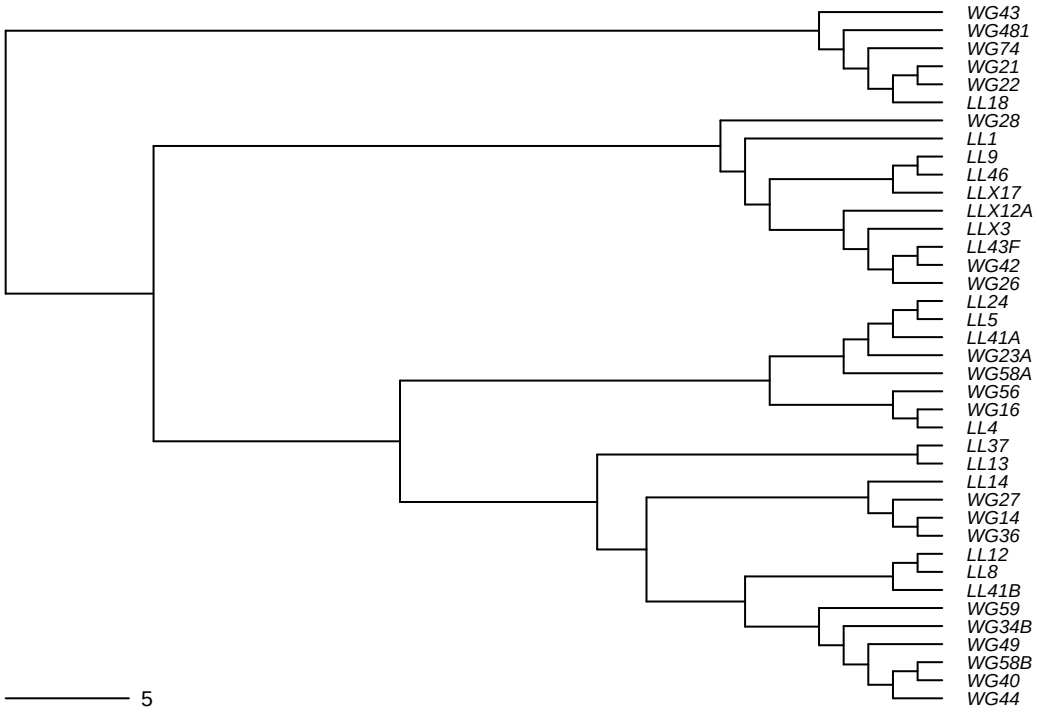
In the R code chunk below, do the following:
1. calculate the maximum growth rate (μ_{max}) of each isolate across all phosphorus types,
2. create a function that calculates niche breadth (nb), and
3. use this function to calculate nb for each isolate.

```
# Calculate Max Growth Rate
umax <- (apply(p.growth, 1, max))
# 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)
}
# Calculate Niche Breadth for Each Isolate
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

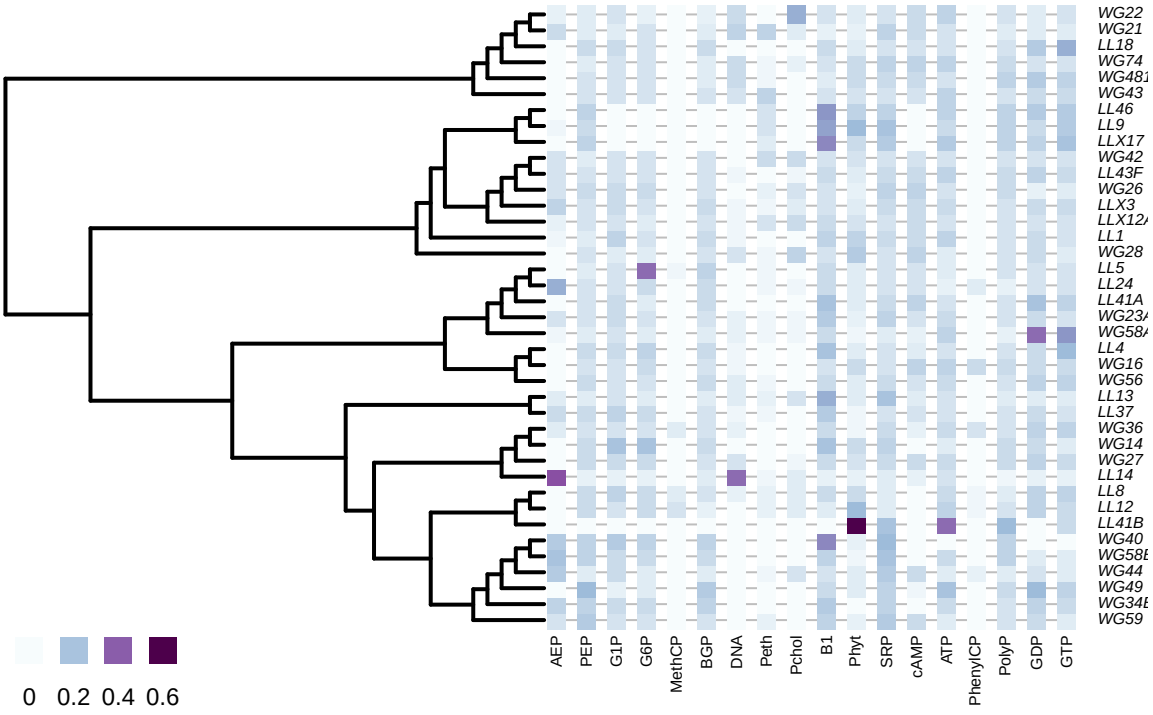
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```
# 1. Make a neighbor joining tree with T92 model
T92.tree <- bionj(seq.dist.T92)
# 2. define your outgroup and root the tree
T92.outgroup <- match("Methanosarcina", T92.tree$tip.label)
T92.rooted <- root(T92.tree, outgroup, resolve.root = TRUE)
T92.rooted <- drop.tip(nj.rooted, "Methanosarcina" ) # Keep Rooted but Drop Outgroup Branch
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(T92.rooted, main = "T92 Model Tree", "phylogram",
use.edge.length = FALSE, direction = "right", cex = 0.6,
label.offset = 1)
add.scale.bar(cex = 0.7)
```

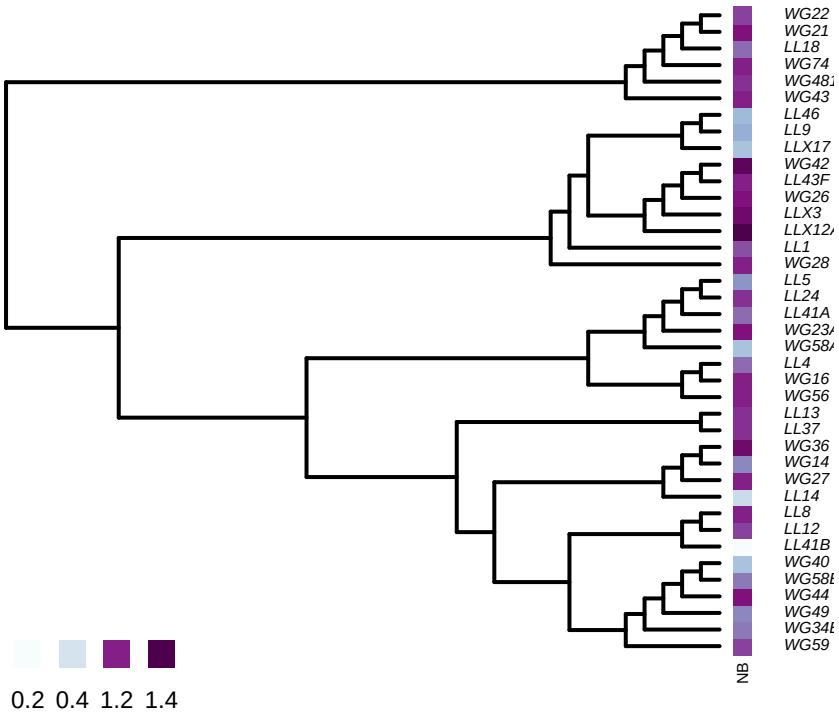


In the R code chunk below, do the following:
1. define a color palette (use something other than "YlOrRd"),
2. map the phosphorus traits onto your phylogeny,
3. map the *nb* trait on to your phylogeny, and
4. customize the plots as desired (use `help(table.phylo4d)` to learn about the options).

```
# 1. Define Color Palette
mypalette <- colorRampPalette(brewer.pal(9, "BuPu")) #bupu!
# First, Correct for Zero Branch-Lengths on Our Tree
T92.plot <- T92.rooted
T92.plot$edge.length <- T92.plot$edge.length + 10^-1
# 2. Map Phosphorus Traits {adephylo}
par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(T92.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)
```




```
# 3. Map the *nb* trait on to your phylogeny
par(mar = c(1, 5, 1, 5) + 0.1)
x.nb <- phylo4d(T92.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?

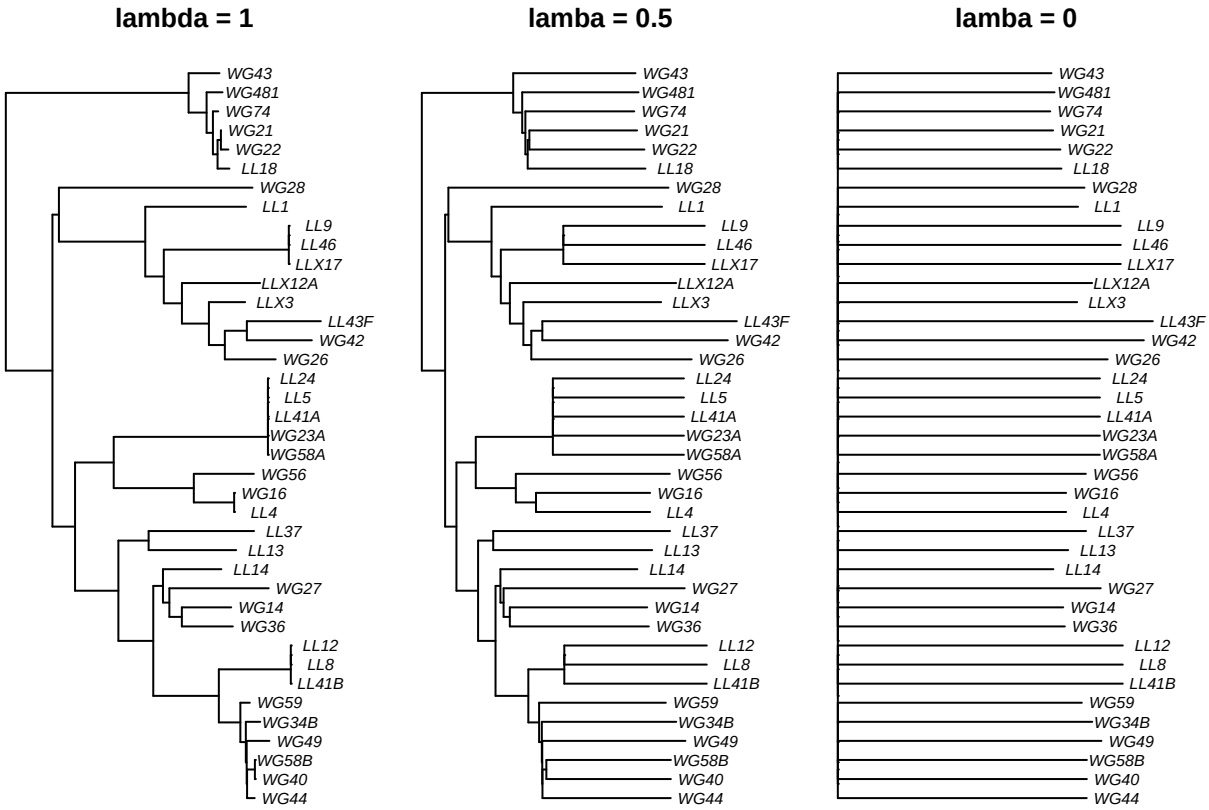
Answer 6a: I hypothesize that, for a given species, it will either be a specialist (low niche breadth but high growth rates) or a generalist (high niche breadth, lower growth rates). There will not be any species that have both high niche breadth and high growth rate. ***Answer 6b***: Specialist species will have very high growth rates on only 1 or 2 sources of phosphorus (low niche breadth) while generalist species will have relatively lower growth rates across many sources of phosphorus (high niche breadth).

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.

```
# 1. create two rescaled phylogenetic trees using lambda values of 0.5 and 0
T92.lambda.5 <- geiger::rescale(T92.rooted, "lambda", 0.5)
T92.lambda.0 <- geiger::rescale(T92.rooted, "lambda", 0)
layout(matrix(c(1, 2, 3), 1, 3), width = c(1, 1, 1))
# 2. plot original tree and scaled trees
par(mar=c(1, 0.5, 2, 0.5) + 0.1)
plot(T92.rooted, main = "lambda = 1", cex = 0.7, adj = 0.5)
plot(T92.lambda.5, main = "lamba = 0.5", cex = 0.7, adj = 0.5)
plot(T92.lambda.0, main = "lamba = 0", cex = 0.7, adj = 0.5)
```



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

```
# Generate Test Statistics for Comparing Phylogenetic Signal {geiger}
fitContinuous(T92.rooted, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.041353
##  sigsq = 0.140025
##  z0 = 0.660660
##
##  model summary:
##  log-likelihood = 21.360155
##  AIC = -36.720309
##  AICc = -36.034595
##  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(T92.lambda.0, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.139673
##  z0 = 0.655220
##
##  model summary:
##  log-likelihood = 21.332374
##  AIC = -36.664747
##  AICc = -35.979033
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 81
##  frequency of best fit = 0.81
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

> ***Answer 7a***: Untransformed lambda = 0.041356. Transformed lambda = 0. ***Answer 7b***: Untransformed AIC = -36.720312. Transformed AIC = -36.664747. Since these values do not have a difference greater than 2, these models are the same. ***Answer 7c***: No, these results suggest that there is no phylogenetic signal in this data.

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.

```
# First, Correct for Zero Branch-Lengths on Our Tree
T92.rooted$edge.length <- T92.rooted$edge.length + 10^-7
# Calculate Phylogenetic Signal for Growth on All Phosphorus Resources
# First, Create a Blank Output Matrix
p.phylosignal <- matrix(NA, 6, 18)
colnames(p.phylosignal) <- colnames(p.growth.std)
rownames(p.phylosignal) <- c("K", "PIC.var.obs", "PIC.var.mean",
"PIC.var.P", "PIC.var.z", "PIC.P.BH")
# Use a For Loop to Calculate Blomberg's K for Each Resource
for (i in 1:18){
  x <- setNames(as.vector(p.growth.std[,i]),row.names(p.growth))
```

```
  out <- phylosignal(x, T92.rooted)
  p.phylosignal[1:5, i] <- round(t(out), 6)
}
# Use the BH Correction on P-values:
p.phylosignal[6, ] <- round(p.adjust(p.phylosignal[4, ], method = "BH"), 3)
# Check out the results
print(p.phylosignal)
```

```
##                      AEP         PEP         G1P         G6P     MethCP
## K               0.000008    0.000010    0.000008    0.000002   0.000006
## PIC.var.obs  4050.665715  659.174149  926.260466 5887.270186 350.858926
## PIC.var.mean 7497.835766 1362.725228 1635.055141 3320.859355 457.174964
## PIC.var.P       0.303000    0.113000    0.170000    0.806000   0.399000
## PIC.var.z      -0.771721   -1.125462   -0.963351    1.058542  -0.322846
## PIC.P.BH        0.606000    0.339000    0.437000    0.843000   0.653000
##                      BGP         DNA        Peth       Pchol          B1
## K               0.000013    0.000101    0.000041    0.000032    0.000006
## PIC.var.obs   510.562292  237.149224  192.479320  397.379600 3357.125632
## PIC.var.mean 1580.071888 4679.066025 1604.508843 2898.152324 4757.226703
## PIC.var.P       0.047000    0.003000    0.007000    0.010000    0.294000
## PIC.var.z      -1.591565   -1.207749   -1.867006   -1.505589   -0.654058
## PIC.P.BH        0.169000    0.045000    0.045000    0.045000    0.606000
##                     Phyt         SRP        cAMP         ATP    PhenylCP
## K               0.000004    0.000006    0.000021    0.000003    0.000002
## PIC.var.obs  9230.269410 1166.314065  678.817905 3942.591177 1224.017615
## PIC.var.mean 8433.826409 1425.478206 2679.956959 2690.626216  699.373291
## PIC.var.P       0.603000    0.362000    0.009000    0.670000    0.843000
## PIC.var.z       0.102142   -0.465761   -2.290702    0.582766    1.105672
## PIC.P.BH        0.775000    0.652000    0.045000    0.804000    0.843000
##                    PolyP         GDP         GTP
## K               0.000004    0.000003    0.000004
## PIC.var.obs  1081.899858 4469.581630 2714.556642
## PIC.var.mean 1086.731875 3128.564016 2606.106331
## PIC.var.P       0.554000    0.726000    0.583000
## PIC.var.z      -0.009012    0.639005    0.080721
## PIC.P.BH        0.775000    0.817000    0.775000
```

```
# 3. Calculate Blomberg's K for niche breadth
signal.nb <- phylosignal(nb, T92.rooted)
signal.nb
```

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 4.302103e-06         48546.44               44246.2           0.62
##   PIC.variance.Z
## 1      0.2219118
```

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

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

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

Answer 8a: There is significant phylogenetic signal for standardized growth on DNA (P = 0.024), Peth (P = 0.018), Pchol (P = 0.024), and cAMP (0.040). There is no significant phylogenetic

signal for niche breadth (P = 0.616). ***Answer 8b***: For standardized growth, the K values are all very close to 0, which means that the traits are overdispersed – closely related species are less similar than expected by chance.

C. Calculate Dispersion of a Trait

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

```
# 1. Turn Continuous Data into Categorical Data
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)
# Look at Phosphorus 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
```

```
# 2. Add Names Column to Data
p.growth.pa$name <- rownames(p.growth.pa)
# 3. Merge Trait and Phylogenetic Data
p.traits <- comparative.data(T92.rooted, p.growth.pa, "name")
# 4. use `phylo.d()` to calculate *D* on at least three phosphorus traits
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 :  T92.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.9482193
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.4083
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0107
```

```
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 :  T92.rooted
##   Number of permutations :  10000
##
```

```
## Estimated D :  0.5612
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0219
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0097
```

```
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 :  T92.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.07430007
## Probability of E(D) resulting from no (random) phylogenetic structure :  2e-04
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.4029
```

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?

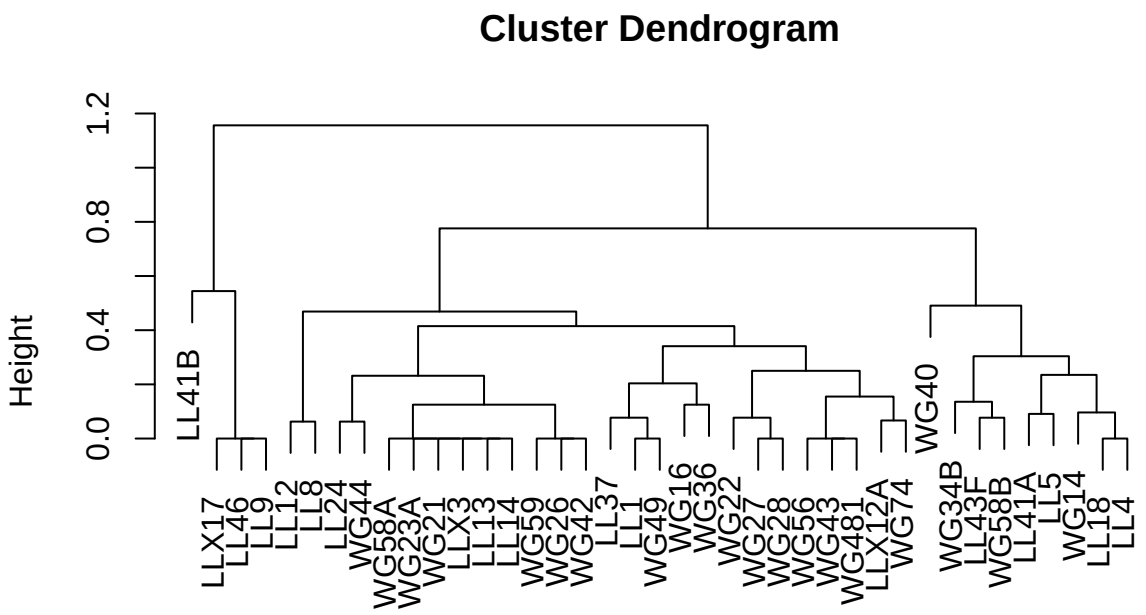
c. Discuss what factors might give rise to differences between the metrics.

Answer 9a: For phenylCP, D = 0.9490325,meaning this trait is randomly dispersed. For DNA, D = 0.5671066, so this trait is probably randomly dispersed. For cAMP, D = 0.07750072, so the trait is as clumped as if it had evolved under Brownian motion. ***Answer 9b***: These results do not align with the Blomberg's K analysis, which suggested that the traits were overdispersed. ***Answer 9c***: The two metrics interpret traits differently. Dispersion is based on categorical data (growth vs no growth) whereas Blomberg's K takes into account the value of the observed trait (growth rate).

D. Correspondence between trait clusters and phylogeny

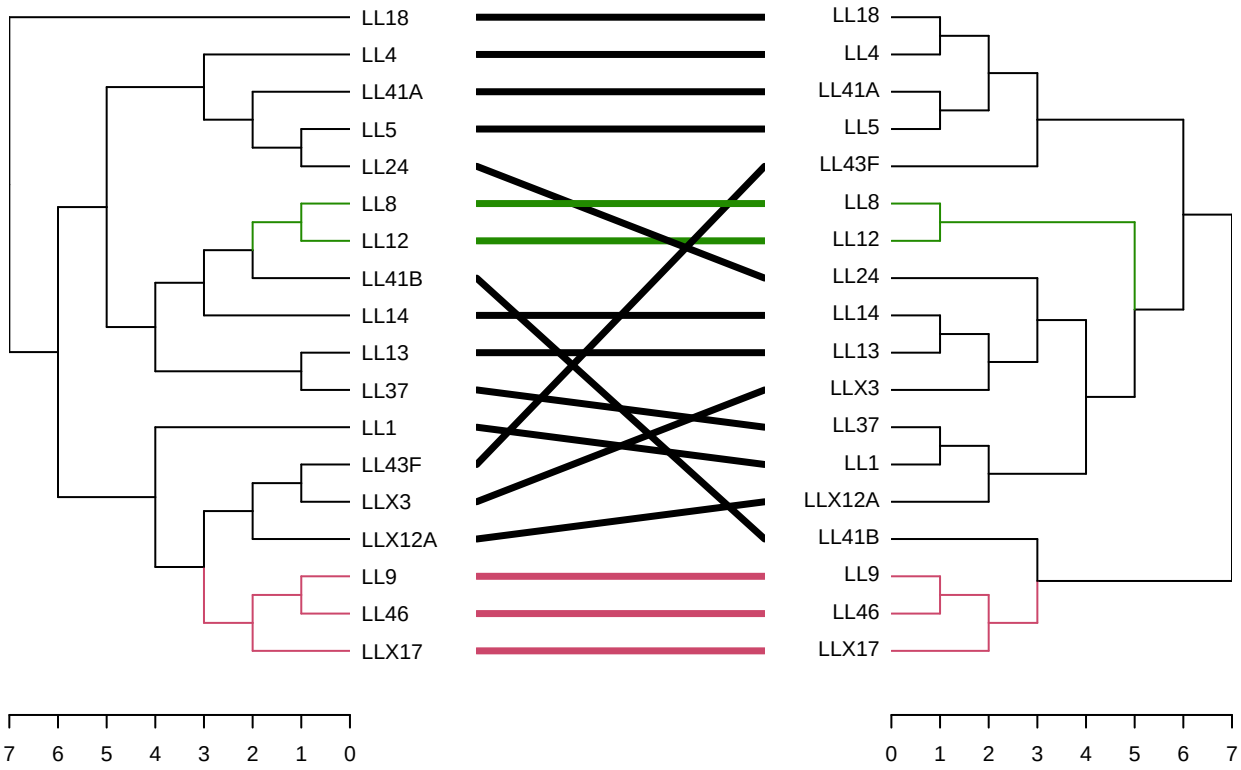
In the R code chunk below, do the following: 1. calculate Jaccard Index on resource use incidence matrix 2. create a hierarchical cluster of resource use 3. map the resource use cluster onto the phylogeny for each environment, and 4. use RF.dist and mantel to measure the degree of correspondence between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

```
# 1. Calculate Jaccard Index on resource use incidence matrix
no <- vegdist(p.growth.pa[,1:18], method = "jaccard", binary = TRUE)
# 2. Generate hierarchical cluster of resource use
no.tree <- hclust(no, method = "ward.D2")
plot(no.tree)
```



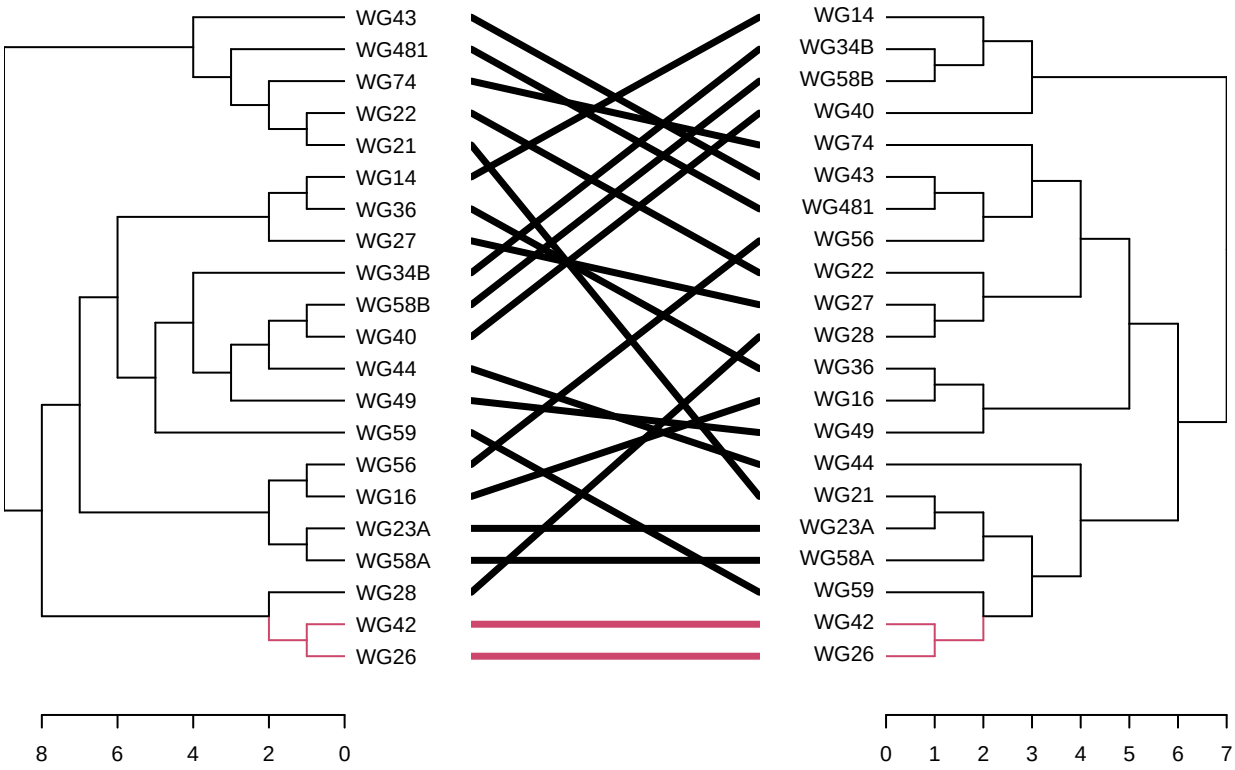
no
hclust (*, "ward.D2")

```
# 3. map the resource use cluster onto the phylogeny for each environment
# dendrogram for LL
LL.tree <- drop.tip(T92.rooted,c(T92.rooted$tip.label[grepl("WG",
T92.rooted$tip.label)]))
LL.function <- drop.tip(as.phylo(no.tree),
                c(no.tree$labels[grepl("WG", no.tree$labels)]))
# dendrogram for WG
WG.tree <- drop.tip(T92.rooted, c(T92.rooted$tip.label[grepl("LL", T92.rooted$tip.label)]))
WG.function <- drop.tip(as.phylo(no.tree),
c(no.tree$labels[grepl("LL",no.tree$labels)]))
# plotting dendrogram for LL
par(mar = c(1, 5, 1, 5) + 0.1)
dendlist(as.cladogram(as.dendrogram.phylo(LL.tree)),
         as.cladogram(as.dendrogram(LL.function))) %>%
         untangle(method = "step2side") %>% # Find the best alignment layout
         tanglegram(common_subtrees_color_branches = TRUE,
        highlight_distinct_edges = FALSE, highlight_branches_lwd = FALSE,
        margin_inner = 5) %>%# Draw the two dendrograms
        entanglement() #score between 0 and 1, closer to 0 is a better alignment
```




```
## [1] 0.09936619
```

```
# plotting dendrogram for WG
par(mar = c(1, 5, 1, 5) + 0.1)
dendlist(as.cladogram(as.dendrogram.phylo(WG.tree)),
        as.cladogram(as.dendrogram(WG.function))) %>%
        untangle(method = "step2side") %>% # Find the best alignment layout
        tanglegram(common_subtrees_color_branches = TRUE,
        highlight_distinct_edges = FALSE, highlight_branches_lwd = FALSE,
        margin_inner = 5) %>%# Draw the two dendrograms
        entanglement() #score between 0 and 1, closer to 0 is a better alignment
```




```
## [1] 0.2692713
```

```
# 4. use RF.dist and mantel to measure the degree of correspondence between each dendrogram
# Measure the degree of correspondence between each dendrogram
# 0 = complete congruence, 1 = complete incongruence
RF.dist(LL.tree,as.phylo(as.dendrogram(LL.function)), normalize = TRUE,
check.labels = TRUE, rooted = FALSE)
```

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

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

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

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

```
##
## Mantel statistic based on Spearman's rank correlation rho
##
## Call:
## mantel(xdis = cophenetic.phylo(LL.tree), ydis = cophenetic.phylo(LL.function),      method = "spearma
##
## Mantel statistic r: 0.07611
##       Significance: 0.221
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.131 0.170 0.202 0.232
```

```
## Permutation: free
## Number of permutations: 999
```

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

```
##
## Mantel statistic based on Spearman's rank correlation rho
##
## Call:
## mantel(xdis = cophenetic.phylo(WG.tree), ydis = cophenetic.phylo(WG.function),      method = "spearma
##
## Mantel statistic r: -0.08469
##       Significance: 0.761
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.135 0.209 0.260 0.313
## Permutation: free
## Number of permutations: 999
```

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

> ***Answer 10a***: In LL there is relatively more overlap between the phylogeny and hierarchical cluster, which means that closely related taxa tend to overlap niches. In WG there is hardly any overlap between phylogeny and resource use, meaning closely related taxa tend to have very different niches. ***Answer 10b***: The results of the R-F index suggest high incongrueunce between phylogeny and resource use. The Mantel correlation was also not significant in either lake. This analysis tells us more definitively that there is no relationship between phylogeny and resource use, which wasn't entirely certain after just looking at the tanglegrams.

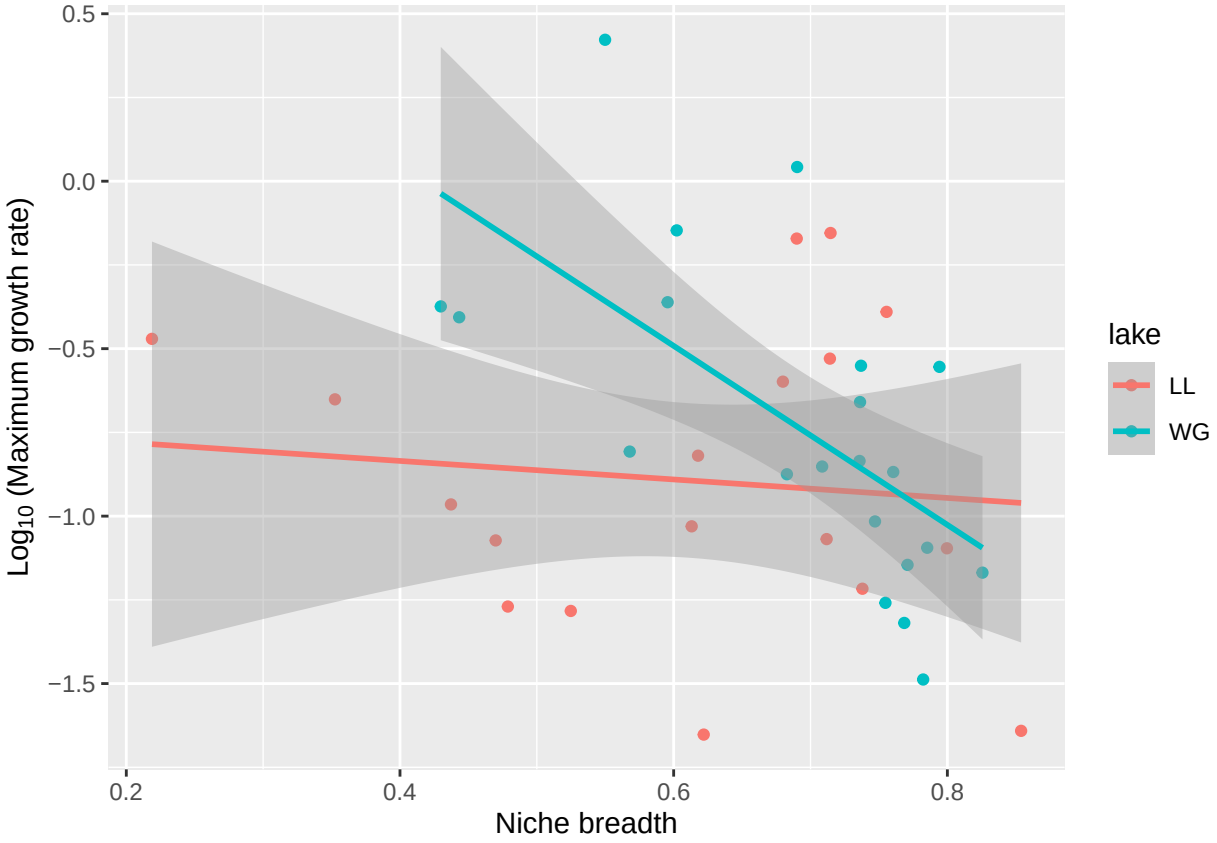
7) PHYLOGENETIC REGRESSION

In the R code chunk below, do the following:
1. Clean the resource use dataset to perform a linear regression to test for differences in maximum growth rate by niche breadth and lake environment. 2. Fit a linear model to the trait dataset, examining the relationship between maximum growth rate by niche breadth and lake environment, 3. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny.

```
# Using the niche breadth data, create a 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 meaningful column name to the 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, T92.rooted,
model = "lambda", boot = 0)
summary(fit.plm)
```

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = T92.rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  41.12 -14.56
##
## Raw residuals:
##      Min       1Q   Median       3Q      Max
## -0.75573 -0.18983 -0.07978  0.32375  0.95388
##
## Mean tip height: 0.1411154
## Parameter estimate(s) using ML:
## lambda : 0.4838773
## sigma2: 1.152634
##
## Coefficients:
##             Estimate    StdErr t.value p.value
## (Intercept) -0.908378  0.367115 -2.4744 0.01834 *
## NB           0.018988  0.523770  0.0363 0.97129
## lakeWG       1.464617  0.576672  2.5398 0.01569 *
## NB:lakeWG   -1.997263  0.846829 -2.3585 0.02406 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1968    Adjusted R-squared: 0.1279
##
## Note: p-values and R-squared are conditional on lambda=0.4838773.
```

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

Answer 11a: Sometimes patterns in data can be explained by shared evolutionary history as opposed to whatever we are hypothesizing, so correcting for it ensures we are not violating the assumption that our data is independent of one another. ***Answer 11b***: In phylogenetic regressions, the variance of the residuals covary based on the branch lengths of the underlying phylogeny, whereas in a standard regression, the residuals are independent of each other and are normally distributed. ***Answer 11c***: The slope of the standard linear regression is and the correlation coefficient (R) = 0.2595. The slope of the phylogenetic regression is and the correlation coefficient = 0.1935. Accounting for shared evolutionary history slightly worsened the fit. ***Answer 11d***: If you were measuring a quantitative trait in some closely related group (e.g. brain size

vs. body size in primates), accounting for phylogeny could reduce the degree to which you see a pattern.

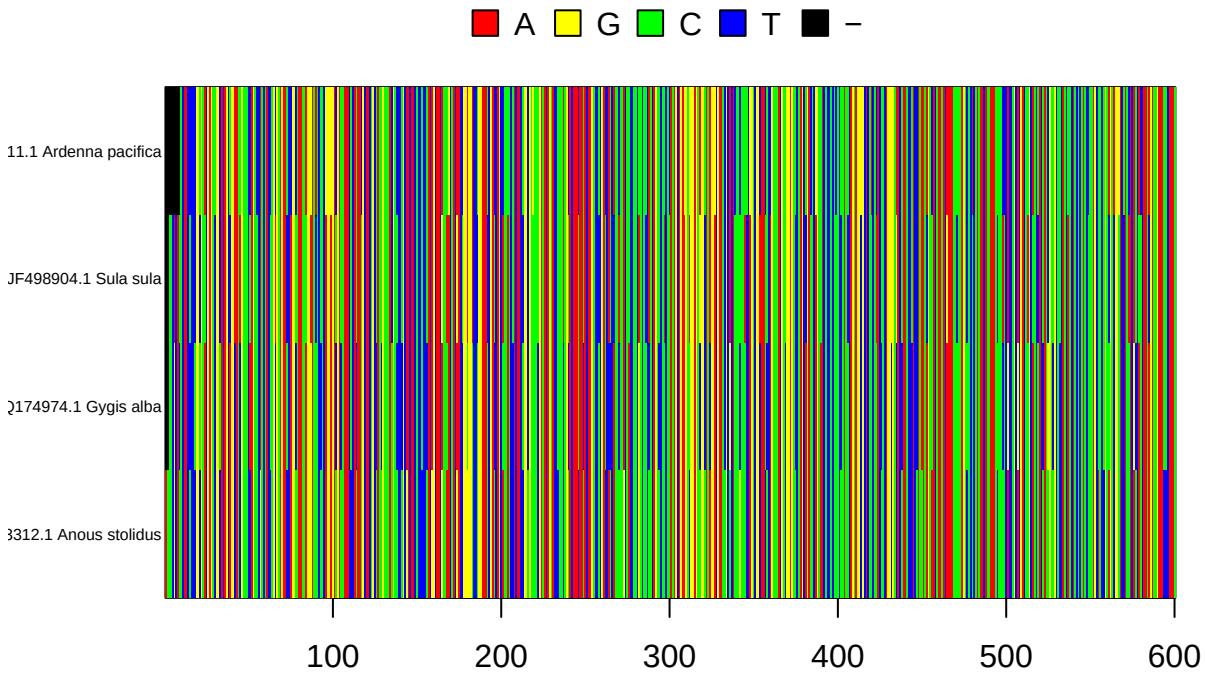
7) SYNTHESIS

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.

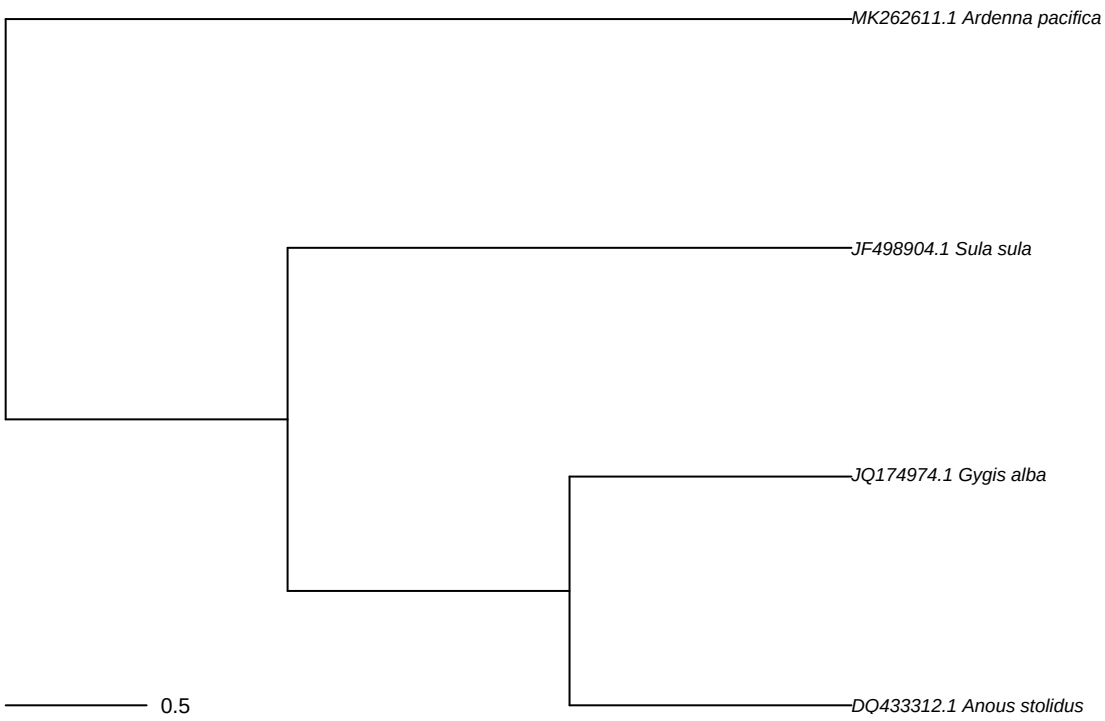
After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. Based on the decisions above and the output, does your tree jibe with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

```
seabird.seqs <- readDNAStringSet("seabird.fasta", format = "fasta")
seabird.read.aln <- msaMuscle(seabird.seqs)
# convert alignment to DNAbin object
s.DNAbin <- as.DNAbin(seabird.read.aln)
# identify base pair region of COI to visualize
s.window <- s.DNAbin[, 0:600]
# visualize sequence alignment
image.DNAbin(s.window, cex.lab = 0.50)
```



Looks like it's pretty well-aligned?

```
# distance matrix based on T92 model
seq.dist.sea <- dist.dna(s.DNAbin, model = "T92", pairwise.deletion = FALSE)
# make tree for T92 model
sea.tree <- bionj(seq.dist.sea)
sea.outgroup <- match("Chicken -- Gallus gallus", sea.tree$tip.label)
sea.rooted <- root(sea.tree, outgroup, resolve.root = TRUE)
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(sea.rooted, main = "Seabird Tree", "phylogram",
use.edge.length = FALSE, direction = "right", cex = 0.6,
label.offset = 0)
add.scale.bar(cex = 0.7)
```



This tree does align with what we know about these species. Anous stolidus and Gygis alba are both in the family Laridae, whereas the other two species are different orders.