8. Worksheet: Phylogenetic Diversity - Traits

Anna Lennon

26 February, 2025

OVERVIEW

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

After completing this exercise you will be able to:

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

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. For the assignment portion of the worksheet, follow the directions at the bottom of this file.
7. When you are done, **Knit** the text and code into a PDF file.
8. After Knitting, submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `PhyloTraits_Worskheet.Rmd` and the PDF output of `Knitr` (`PhyloTraits_Worskheet.pdf`).

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

1) SETUP

In the R code chunk below, provide the code to:
1. clear your R environment,
2. print your current working directory,
3. set your working directory to your `Week6-PhyloTraits/` folder, and
4. load all of the required R packages (be sure to install if needed).

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

```
## [1] "/cloud/project/QB2025_ALennon/Week6-PhyloTraits"
```

```
setwd("/cloud/project/QB2025_ALennon/Week6-PhyloTraits")
# package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger', 'picante', 'stats', 'RColorBrew
# for (package in package.list) {
#   if (!require(package, character.only=TRUE, quietly=TRUE)) {
#     install.packages(package)
#     library(package, character.only=TRUE)
#   }
# }
# install.packages("pak")
# #comment out after first run to prevent continuous reinstall/update
# pak::pkg_install("msa")
#
# if (!require("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
#
# BiocManager::install("Biostrings")
library(bios2mds)
```

```
## Loading required package: amap
## Loading required package: e1071
## Loading required package: scales
## Loading required package: cluster
## Loading required package: rgl
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
## Warning: 'rgl.init' failed, will use the null device.
## See '?rgl.useNULL' for ways to avoid this warning.
```

```
library(Biostrings)
```

```
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
## Loading required package: S4Vectors
## Loading required package: stats4
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Loading required package: IRanges
## Loading required package: XVector
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
library(ape)
##
## Attaching package: 'ape'
## The following object is masked from 'package:Biostrings':
##
##     complement
library(seqinr)
##
## Attaching package: 'seqinr'
## The following objects are masked from 'package:ape':
##
##     as.alignment, consensus
## The following object is masked from 'package:Biostrings':
##
##     translate
library(phylobase)
##
## Attaching package: 'phylobase'
## The following object is masked from 'package:ape':
##
##     edges
library(adephylo)
## Loading required package: ade4
##
## Attaching package: 'ade4'
## The following object is masked from 'package:BiocGenerics':
##
##     score
```

```
library(geiger)
```

```
## Loading required package: phytools
```

```
## Loading required package: maps
```

```
##
## Attaching package: 'maps'
```

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

```
##
## Attaching package: 'phytools'
```

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

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

```
library(picante)
```

```
## Loading required package: vegan
```

```
## Loading required package: permute
```

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

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

```
## Loading required package: lattice
```

```
##
## Attaching package: 'vegan'
```

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

```
## The following object is masked from 'package:amap':
##
##     pca
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

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

```
## The following object is masked from 'package:Biostrings':
##
##     collapse
```

```
## The following object is masked from 'package:IRanges':
##
##     collapse
```

```
library(stats)
library(RColorBrewer)
library(phylolm)
library(caper)
```

```
## Loading required package: MASS
```

```
## Loading required package: mvtnorm
```

```
library(pmc)
library(ggplot2)
library(dplyr)
```

```
##
## 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:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
```

```
## The following object is masked from 'package:XVector':
##
##     slice
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

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

```
## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag
```

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

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
library(phangorn)
```

```
##
## Attaching package: 'phangorn'
```

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

```
library(pander)
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

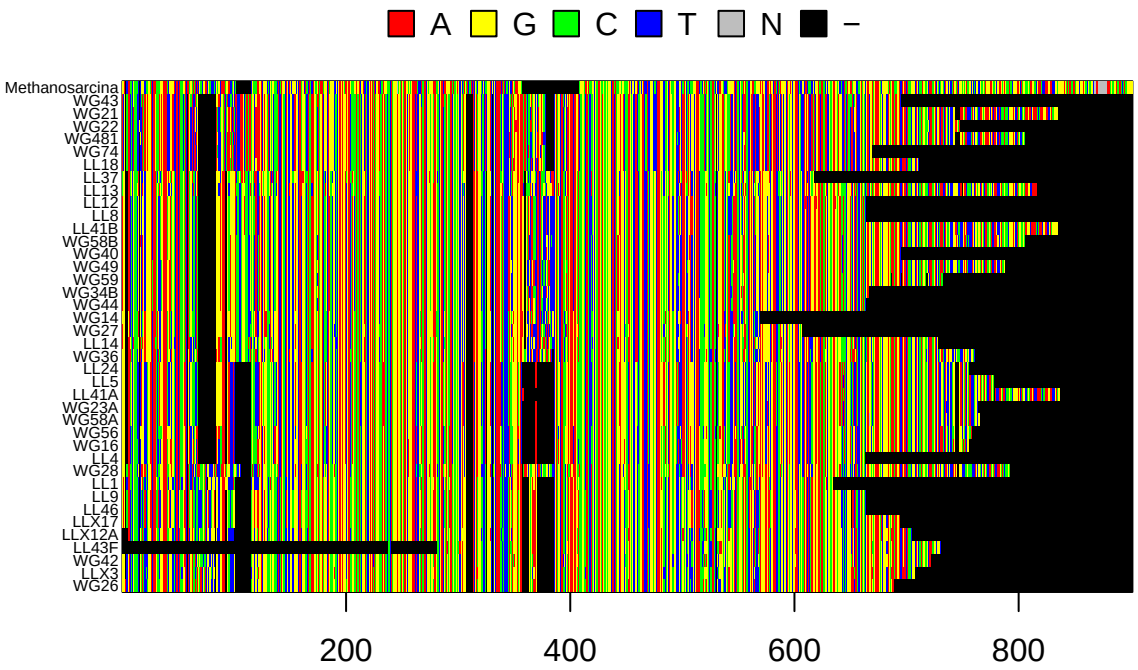
Question 1: Using your favorite text editor, compare the `p.isolates.fasta` file and the `p.isolates.afa` file. Describe the differences that you observe between the two files.

> ***Answer 1***: Both the fasta and afa file contain a series of nucleotides (ACGT) color coded. However, the fasta file does not contain any gaps and the output is much more extensive compared the afa file that is cmpressed to view the last line of nucleotides (1500) for the samples and contains the gaps present in the genome.

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.

```
seq.fasta <-readDNAStringSet("data/p.isolates.fasta", format = "fasta")
seq.afa <-readDNAStringSet("data/p.isolates.afa", format = "fasta")

# seq.fasta
# seq.afa

read.aln.fasta <-msaMuscle(seq.fasta)
save.aln.fasta <-msaConvert(read.aln.fasta, type = "bios2mds::align")
export.fasta(save.aln.fasta, "./data/p.isolates.2.afa")
```

```
p.DNAbin.fasta <-as.DNAbin(read.aln.fasta)
window.L <-p.DNAbin.fasta[,100:1000]
image.DNAbin(window.L, cex.lab = .5)
```



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:By changing the window amount, the length of the sequence reads are around 800 base pairs. However, some of the sequences are smaller (~700 bp). ***Answer 2b***: Regions with similar base pairs reveals conserved regions where as high amounts of color diversity could insiuate regions under selective pressures. At about 460 bp, there is a region of highly conserved genes with some variation (green) that could be interesting to make phylogenetic inferences. Comparatively, at about 580 bp there is a region of alternating blue-green that could insight a region under selective pressures that would be fascinating to study.

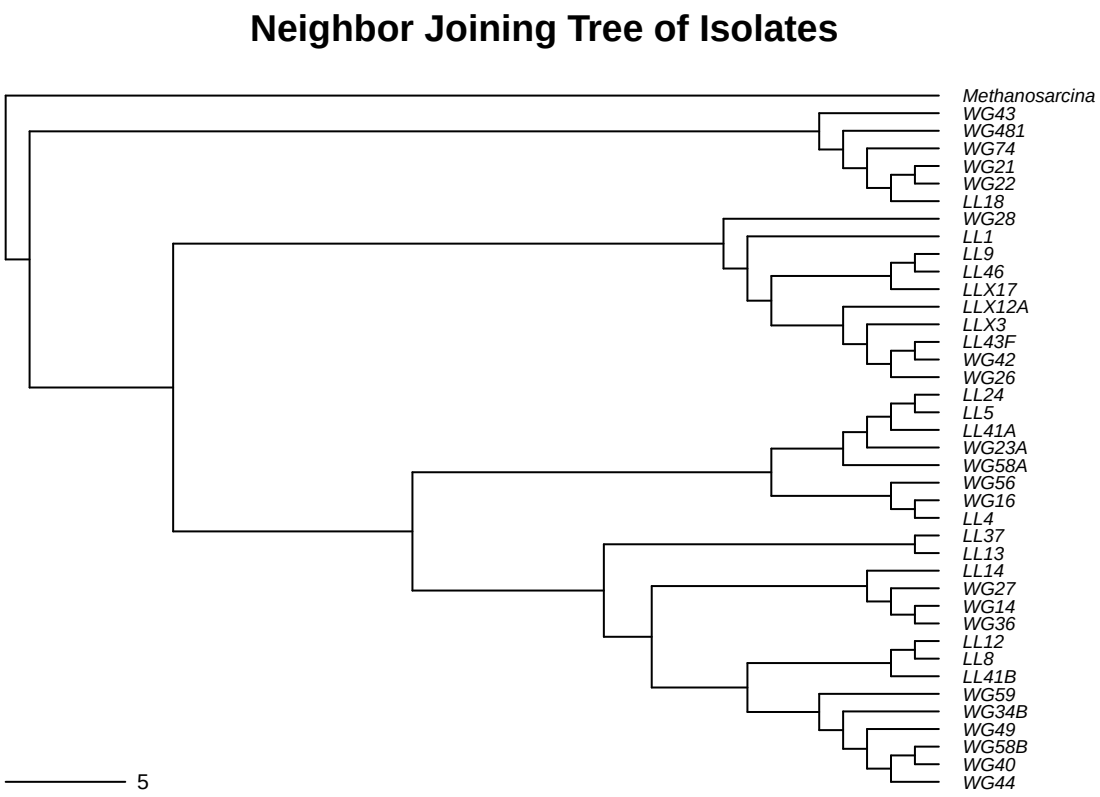
4) MAKING A PHYLOGENETIC TREE

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.fasta, model = "raw", pairwise.deletion = FALSE)
#tree
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 of Isolates", "phylogram",
           use.edge.length = FALSE, direction = "right", cex = 0.6,
           label.offset = 1)
add.scale.bar(cex = 0.7)
```



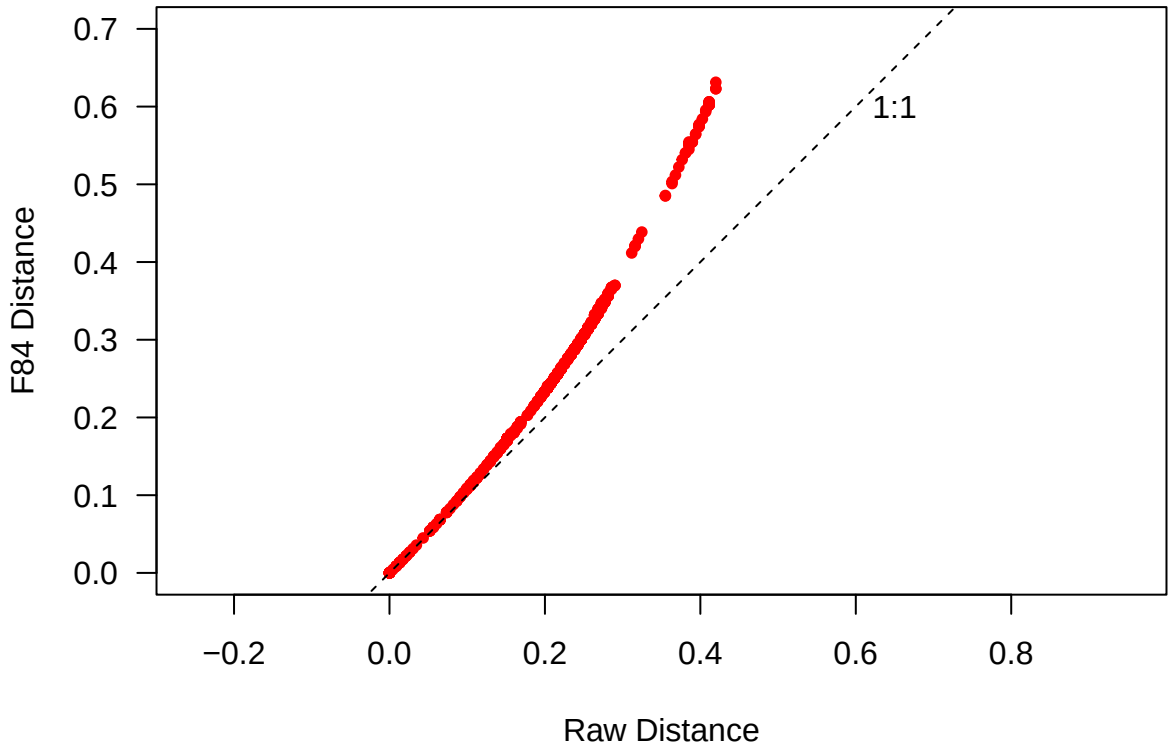
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: Neighbor-joining trees are the simplest form of phyolgenetic tree and very computationally light. These tree allow one to view broad patterns. However, the raw estimates of phylogeentic distance are used, meaning they assume that substitutions only occur once and at the same rate. This is not accurate to biology as base pairs can be substituted multiple different times and at different rates. The resulting tree may not be accurate to the actual life history as a result.

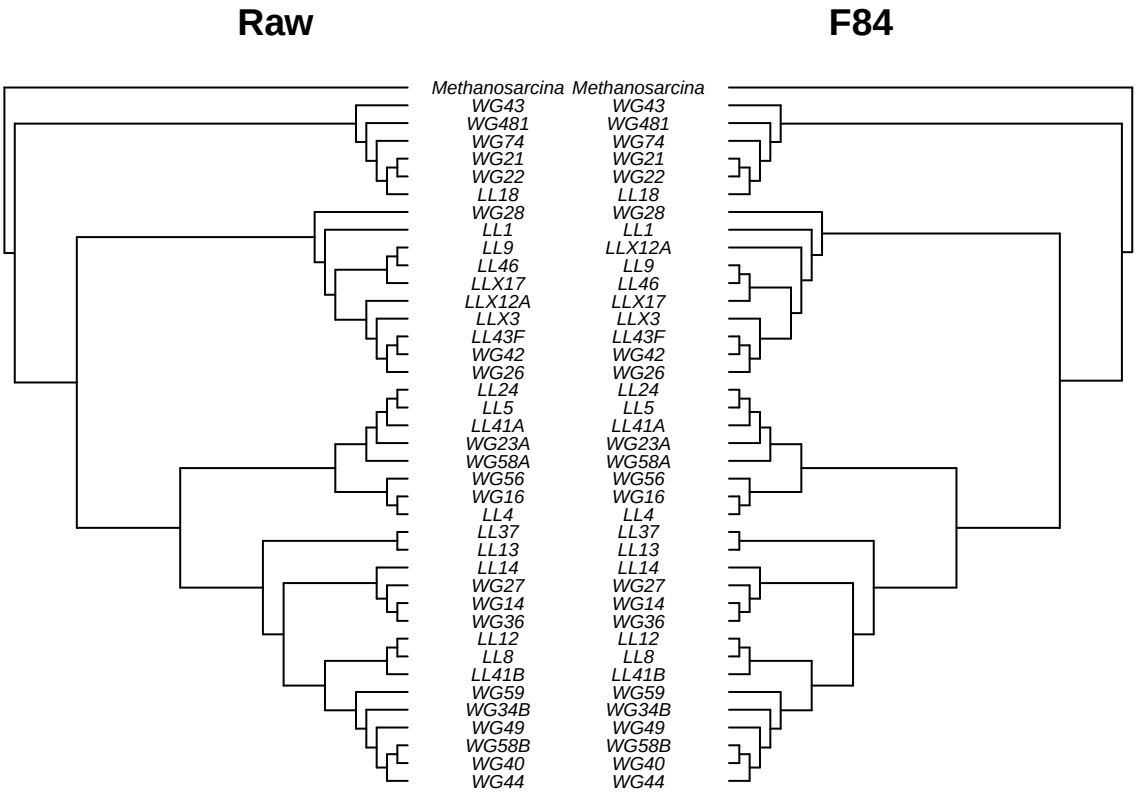
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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.

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




```
#trees
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")
```




```
#tree length diff
dist.topo(raw.rooted, F84.rooted, method = "score")
```

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

C) ANALYZING A MAXIMUM LIKELIHOOD TREE

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

```
phyDat.aln <- msaConvert(read.aln.fasta, type = "phangorn::phyDat")

# NJ-ML method
aln.dist <- dist.ml(phyDat.aln)
aln.NJ <- NJ(aln.dist)

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

# JC69 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
```

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

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

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

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

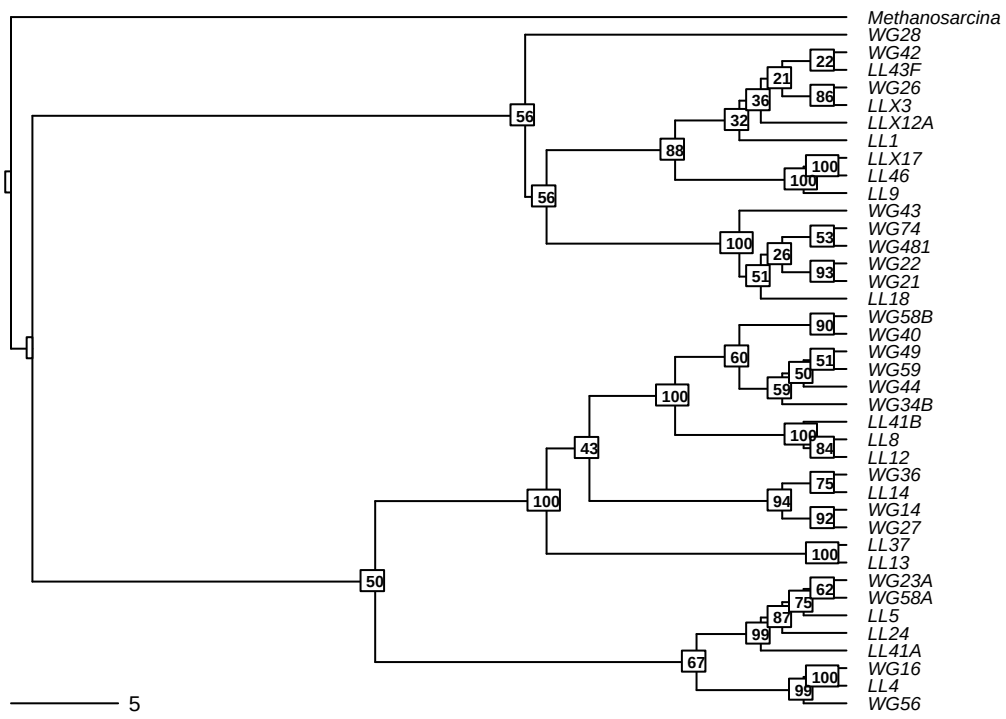
```
AIC(fitJC)
```

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

```
AIC(fitGTR)
```

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

```
#Bootstrap
ml.bootstrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(ml.bootstrap, type = "phylogram", direction = "right",
      show.tip.label = TRUE, use.edge.length = FALSE, cex = 0.6,
      label.offset = 1, main = "Maximum Likelihood with Support Values")
add.scale.bar(cex = 0.7)
nodelabels(ml.bootstrap$node.label, font = 2, bg = "white",
      frame = "r", cex = 0.5)
```



Question 4:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches? Why or why not?

Answer 4a: The maximum likelihood tree and the NJ tree vary from one another. Given that the NJ tree does not account for nucleotide states and not based on statistics compared to Neighborhood joining tree, trees produced by ML are typically more robust. ***Answer 4b***: Maximum likelihood trees are based on statistical computation that finds the tree with the highest probability of being the true phylogeny. Bootstrapping assigns values of support to indicate how reliable/resolve a branch is with hgher values indicating more resolved branches. ***Answer 4c***: High bootstrap values indicate more reliable/resolved branches. Therefore, values with 95% or higher are considered resolved compared to values beteern 94-70 having some support and less than 50 are considered unresolved. ***Answer 4d***: There are several key brnaches with low support. In particular, there is a cluster of branches at the top of the tree analyzing WG42-LL1 relationship, the node between WG74-WG21, WG58B-WG27.
Answer 4e: The low values indicate that there is low resolution for these branches. Therefore, these branches should be handeled with caution. Overall, the tree has relatively high bootstrap values, so the tree as a whole is relatively resolved. However, these low resolution branches should not be taken as operationally correct and assumed as unresolved.

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)

# Standardize
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))

#function for niche breadth (nb)
nb.levins <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
    }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}
#Isolate NB
nb <- as.matrix(nb.levins(p.growth.std))
nb <- setNames(as.vector(nb), as.matrix(row.names(p.growth)))
```

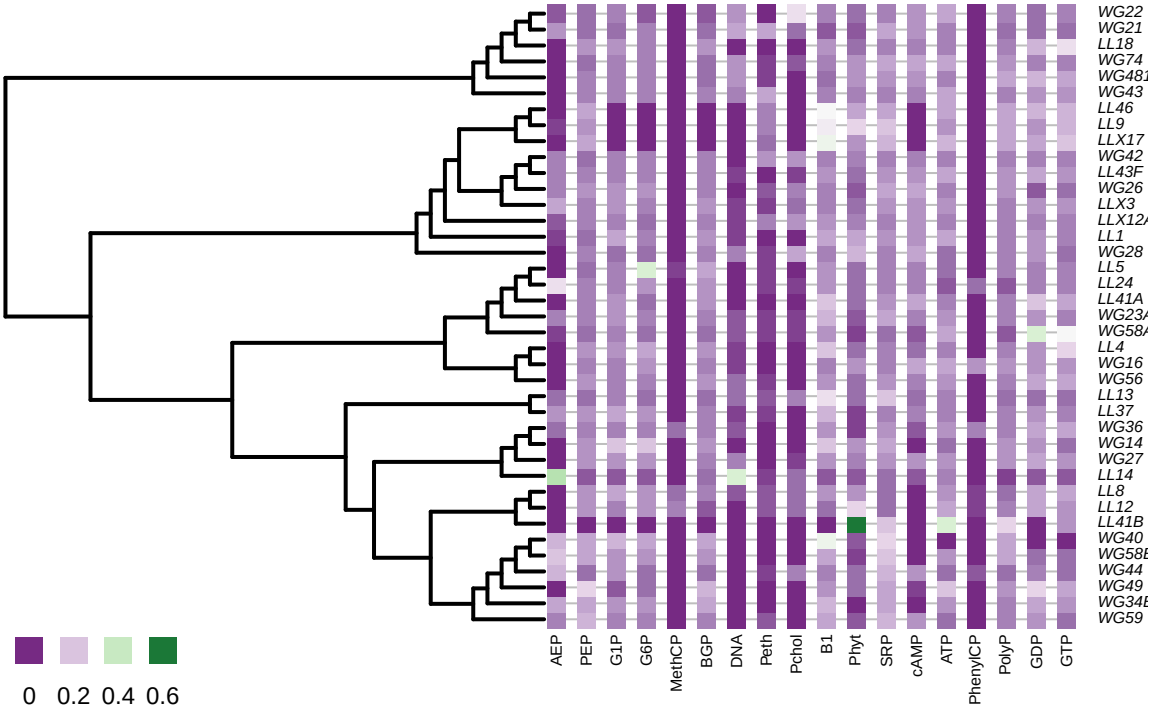
C. Visualizing Traits on Trees

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.

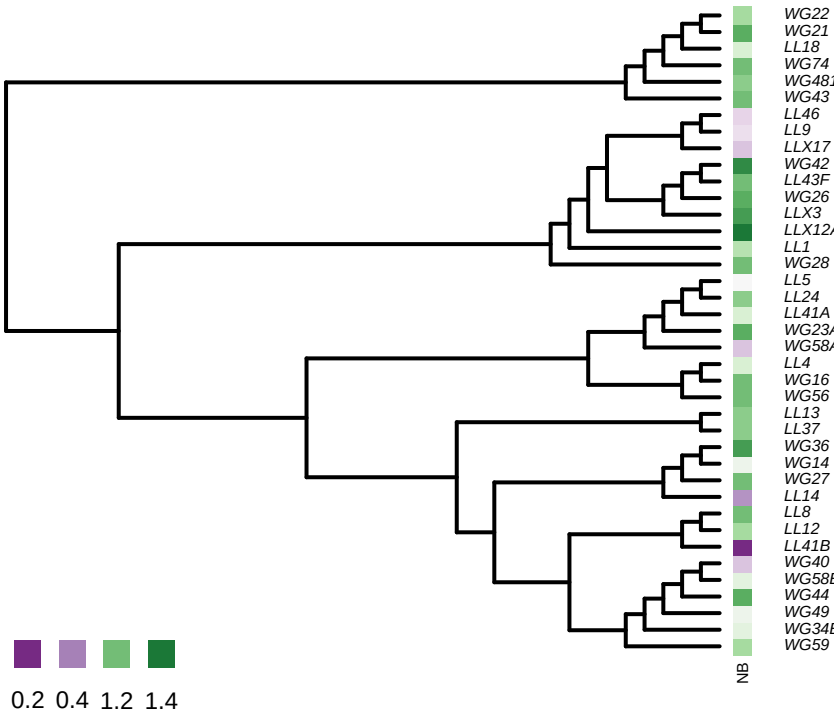
```
seq.dist.T92 <-dist.dna(p.DNAbin.fasta, model = "T92", pairwise.deletion = FALSE)
nj.tree.T92 <-bionj(seq.dist.T92)
outgroup.T92 <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted.T92 <- root(nj.tree.T92, outgroup.T92, resolve.root = TRUE)
nj.rooted.T92 <- drop.tip(nj.rooted.T92, "Methanosarcina")


#plot(nj.rooted)
```

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, "PRGn"))

nj.plot.T92 <- nj.rooted.T92
nj.plot.T92$edge.length <- nj.plot.T92$edge.length + 10^-1

par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(nj.plot.T92, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = 0.5, scale = FALSE, use.edge.length = FALSE,
              edge.color = "black", edge.width = 2, box = FALSE,
              col=mypalette(25), pch = 15, cex.symbol = 1.25,
              ratio.tree = 0.5, cex.legend = 1.5, center = FALSE)
```




```
# Niche Breadth
par(mar = c(1, 5, 1, 5) + 0.1)
x.nb <- phylo4d(nj.plot.T92, 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 5:

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

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

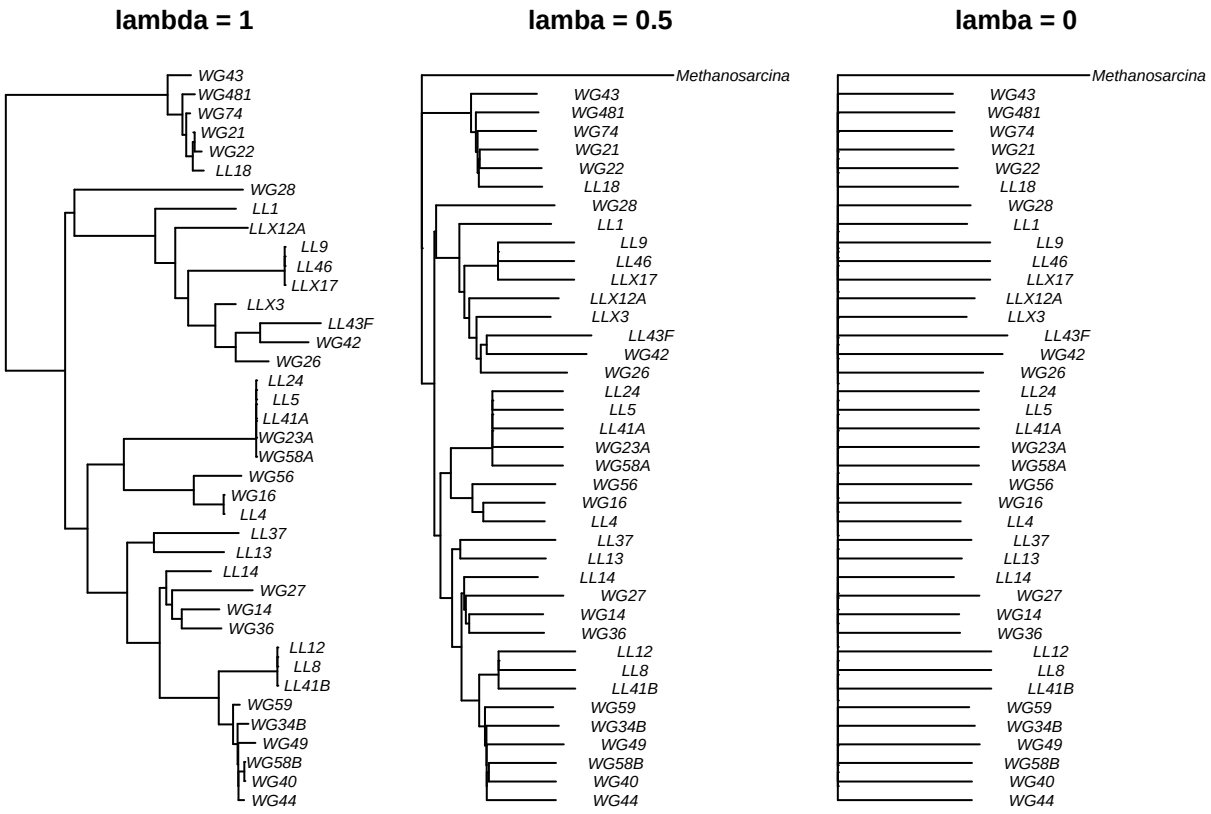
Answer 5a: Generalist species that occupy a wide variety traits will be better fit to a highly variable environments and be less fit in stable environments whereas specialist species will be better fit in highly stable environments and signifcantly less fit in highly variable environments. Therefore, species that occupy similar niches (generalist or specialist) will be more closely related.
Answer 5b: When testing ths hypothesis, individuals that are closely related should have similar niche breadth values. However, the growth rate for individuals with similar niche breadth values may be constrainted as species compete for similar resources. This may cause certain species to diverage into more specialist roles despite having sister taxa that are generalist.

6) HYPOTHESIS TESTING

Phylogenetic Signal: Pagel's Lambda

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

```
#original tree: nj.rooted.T92
nj.lambda.5 <- geiger:::rescale.phylo(nj.rooted, model = "lambda", lambda = 0.5)
nj.lambda.0 <-geiger:::rescale.phylo(nj.rooted, model = "lambda", lambda = 0)
layout(matrix(c(1, 2, 3), 1, 3), width = c(1, 1, 1))
par(mar=c(1, 0.5, 2, 0.5) + 0.1)
plot(nj.rooted.T92, main = "lambda = 1", cex = 0.7, adj = 0.5)
plot(nj.lambda.5, main = "lamba = 0.5", cex = 0.7, adj = 0.5)
plot(nj.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.

```
fitContinuous(nj.rooted.T92, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.006930
##  sigsq = 0.108362
##  z0 = 0.657663
##
##  model summary:
##  log-likelihood = 21.499853
##  AIC = -36.999706
##  AICc = -36.313992
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 46
##  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")
```

```
## Warning in treedata(phy, dat): The following tips were not found in 'data' and were dropped from 'phy
##  Methanosarcina
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.082632
##  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 = 0.810
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
phylosig(nj.rooted.T92, nb, method = "lambda", test = TRUE)
```

```
##
## Phylogenetic signal lambda : 0.006947
## logL(lambda) : 21.4999
## LR(lambda=0) : 0.00179646
## P-value (based on LR test) : 0.966192
```

```
#Phylo signal
nj.rooted.T92$edge.length <- nj.rooted.T92$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.T92)
  p.phylosignal[1:5, i] <- round(t(out), 6)
}

p.phylosignal[6, ] <- round(p.adjust(p.phylosignal[4, ], method = "BH"), 3)

print(p.phylosignal)
```

```
##                      AEP         PEP         G1P         G6P     MethCP
```

```
## K               0.000007    0.000008    0.000006    0.000002   0.000005
## PIC.var.obs  4050.684457  659.174698  926.261775 5887.269606 350.858813
## PIC.var.mean 7325.232632 1353.411188 1639.998890 3287.914274 453.097231
## PIC.var.P       0.283000    0.125000    0.167000    0.812000   0.413000
## PIC.var.z      -0.733574   -1.063422   -0.985927    1.070566  -0.307508
## PIC.P.BH        0.598000    0.375000    0.429000    0.854000   0.676000
##                      BGP         DNA        Peth       Pchol          B1
## K               0.000011    0.000087    0.000037    0.000025    0.000005
## PIC.var.obs   510.560959  237.150186  192.519394  397.370402 3357.131069
## PIC.var.mean 1550.299992 4481.573445 1586.133171 2889.842635 4728.806206
## PIC.var.P       0.046000    0.003000    0.006000    0.007000    0.299000
## PIC.var.z      -1.522483   -1.169044   -1.732546   -1.497970   -0.634199
## PIC.P.BH        0.166000    0.032000    0.032000    0.032000    0.598000
##                     Phyt         SRP        cAMP         ATP    PhenylCP
## K               0.000003    0.000005    0.000017    0.000002    0.000002
## PIC.var.obs  9230.268770 1166.315669  678.817262 3942.591218 1224.017445
## PIC.var.mean 8159.726937 1398.118485 2715.708877 2796.920533  687.223095
## PIC.var.P       0.629000    0.372000    0.006000    0.648000    0.854000
## PIC.var.z       0.137054   -0.425623   -2.396843    0.515631    1.155270
## PIC.P.BH        0.778000    0.670000    0.032000    0.778000    0.854000
##                    PolyP         GDP         GTP
## K               0.000004    0.000002    0.000003
## PIC.var.obs  1081.902135 4469.581414 2714.559870
## PIC.var.mean 1084.982441 3093.771938 2646.186095
## PIC.var.P       0.576000    0.732000    0.572000
## PIC.var.z      -0.005603    0.653869    0.050698
## PIC.P.BH        0.778000    0.824000    0.778000
```

Question 6: There are two important outputs from the `fitContinuous()` function that can help you interpret the phylogenetic signal in trait data sets. a. Compare the lambda values of the untransformed tree to the transformed (lambda = 0). b. Compare the Akaike information criterion (AIC) scores of the two models. Which model would you choose based off of AIC score (remember the criteria that the difference in AIC values has to be at least 2)? c. Does this result suggest that there's phylogenetic signal?

> ***Answer 6a***: The untransformed tree lambda is .006947 compared the the transformed lambda which is .00179646.
> ***Answer 6b***: The AIC value for the untransformed tree is -36.9997 while the AIC of the log tranformed model is -36.6647. Alithough the untranformed tree has a lower AIC value, the two models are considered equivalent because they do not have a difference of 2. ***Answer 6c***: These results do not suggest there is a phylogenetic signal.

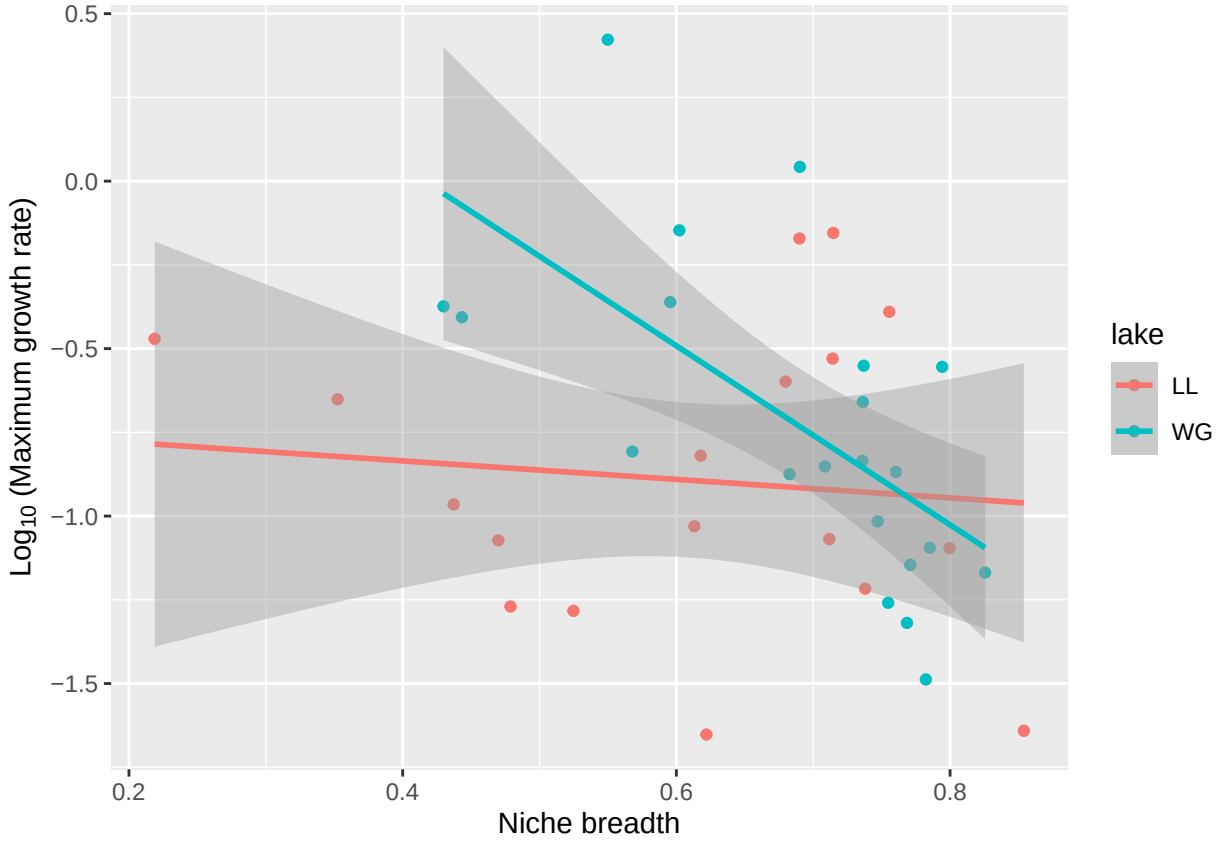
7) PHYLOGENETIC REGRESSION

Question 7: In the R code chunk below, do the following:
1. Clean the resource use dataset to perform a linear regression to test for differences in maximum growth rate by niche breadth and lake environment. 2. Fit a linear model to the trait dataset, examining the relationship between maximum growth rate by niche breadth and lake environment, 2. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny

```
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"

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

# Plot
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'
```




```
#Linear
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
```

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

```
## Warning in phylolm(log10(umax) ~ NB * lake, data = nb.lake, nj.rooted, model =
## "lambda", : will drop from the tree 1 taxa with missing data
```

```
summary(fit.plm)
```

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = nj.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: 0.1411147
## Parameter estimate(s) using ML:
## lambda : 0.4838753
## sigma2: 1.152639
##
## Coefficients:
##              Estimate    StdErr t.value p.value
## (Intercept) -0.908378  0.367115 -2.4744 0.01834 *
## NB           0.018987  0.523770  0.0363 0.97129
## lakeWG       1.464616  0.576672  2.5398 0.01569 *
## NB:lakeWG   -1.997261  0.846830 -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=0.4838753.
```

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

```
## [1] 41.12295
```

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

> ***Answer 7a***:Correcting for the shared evolutionary history is essential because the linear regression model assumes independence. However, in this data, the shared history means the variables are not independent so it needs to be corrected.
> ***Answer 7b***: Phylogenetic regression is the varience of residual errors by the covariance matrix, which allows branch length to be factored into the model. Phylogenetic regression also takes into account phylogenetic signal when determining the covariance matrix. Simple linear regression only analyzes the relationship between residual error by a vairable.
> ***Answer 7c***: For LL lake, the slope is near zero, therefore there is only a small portion of maximum growth explained by niche breadth. Comparatively, WG lake has a steep negative slope resulting in a negative relationship between niche breadth and maximum growth. Correcting for phylogeny made the fit worse (from 48.41 to 41.73) ***Answer 7d***: The relationship between two variables would completely disappear when doing a phylogenetic regression if one of the variables was highly correlated with phylogeny. For example, if analyzing the relationship between wing pattern of butterflies and niche breadth, wing patterns may be too similar among taxa that would result in no relationship if phylogeny was taken into account.

7) SYNTHESIS

Work with members of your Team Project to obtain reference sequences for 10 or more taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. In almost all cases, researchers must deposit their sequences in places like NCBI before a paper is published. Those sequences are checked by NCBI employees for aspects of quality and given an **accession number**. For example, here an accession number for a fungal isolate that our lab has worked with: JQ797657. You can use the NCBI program nucleotide **BLAST** to find out more about information associated with the isolate, in addition to getting its DNA sequence: https://blast.ncbi.nlm.nih.gov/. Alternatively, you can use the `read.GenBank()` function in the `ape` package to connect to NCBI and directly get the sequence. This is pretty cool. Give it a try.

But before your team proceeds, you need to give some thought to which gene you want to focus on. For microorganisms like the bacteria we worked with above, many people use the ribosomal gene (i.e., 16S rRNA). This has many desirable features, including it is relatively long, highly conserved, and identifies taxa with reasonable resolution. In eukaryotes, ribosomal genes (i.e., 18S) are good for distinguishing course taxonomic resolution (i.e. class level), but it is not so good at resolving genera or species. Therefore, you may need to find another gene to work with, which might include protein-coding gene like cytochrome oxidase (COI) which is on mitochondria and is commonly used in molecular systematics. In plants, the ribulose-bisphosphate carboxylase gene (*rbcL*), which on the chloroplast, is commonly used. Also, non-protein-encoding sequences like those found in **Internal Transcribed Spacer (ITS)** regions between the small and large subunits of of the ribosomal RNA are good for molecular phylogenies. With your team members, do some research and identify a good candidate gene.

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

about the evolutionary history of your organisms? If not, why? Is there anything you could do differently that would improve your tree, especially with regard to future analyses done by your team?

```
library(devtools)
```

```
## Loading required package: usethis
```

```
##
## Attaching package: 'devtools'
```

```
## The following object is masked from 'package:permute':
##
##     check
```

```
devtools::install_github("jinyizju/V.PhyloMaker")
```

```
## Using GitHub PAT from the git credential store.
```

```
## Skipping install of 'V.PhyloMaker' from a github remote, the SHA1 (9197490c) has not changed since la
##   Use `force = TRUE` to force installation
```

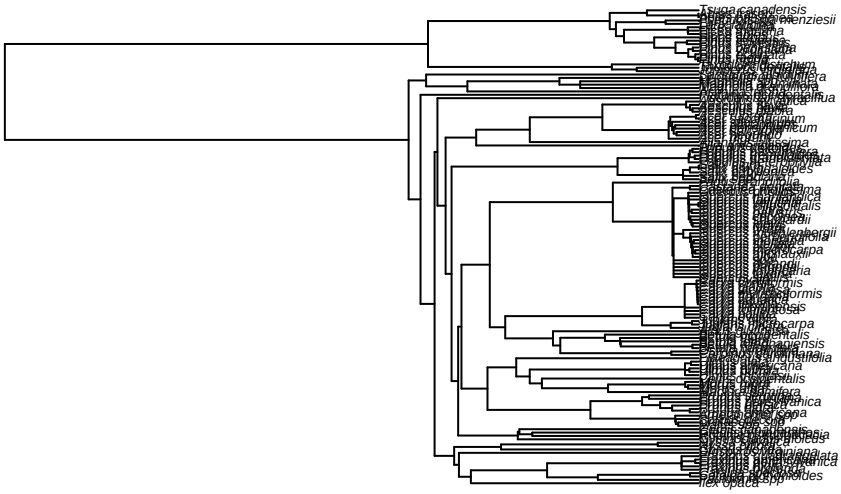
```
require(V.PhyloMaker)
```

```
## Loading required package: V.PhyloMaker
```

```
setwd("/cloud/project/QB_biodiversity_project_EH/")
SPCDinfo <- read.csv("/cloud/project/QB_biodiversity_project_EH/SPCDinfo.csv")

treesp <- SPCDinfo[c("Scientific.Name", "Genus", "Family")] #extract relevant cols
treephylo <- phylo.maker(treesp, scenarios = "S3", nodes = nodes.info.1) #make tree
```

```
## [1] "Duplicated species detected and removed."
##  [1] ""                   ""                   ""
##  [4] ""                   ""                   ""
##  [7] ""                   ""                   ""
## [10] ""                   ""                   "Quercus_palustris"
## [13] ""                   ""
## [1] "Taxonomic classification not consistent between sp.list and tree."
##      genus family_in_sp.list family_in_tree
## 1    Abies        Abietaceae       Pinaceae
## 3 Aesculus       Aesculaceae    Sapindaceae
## [1] "Note: 1 taxa fail to be binded to the tree,"
## [1] ""
```

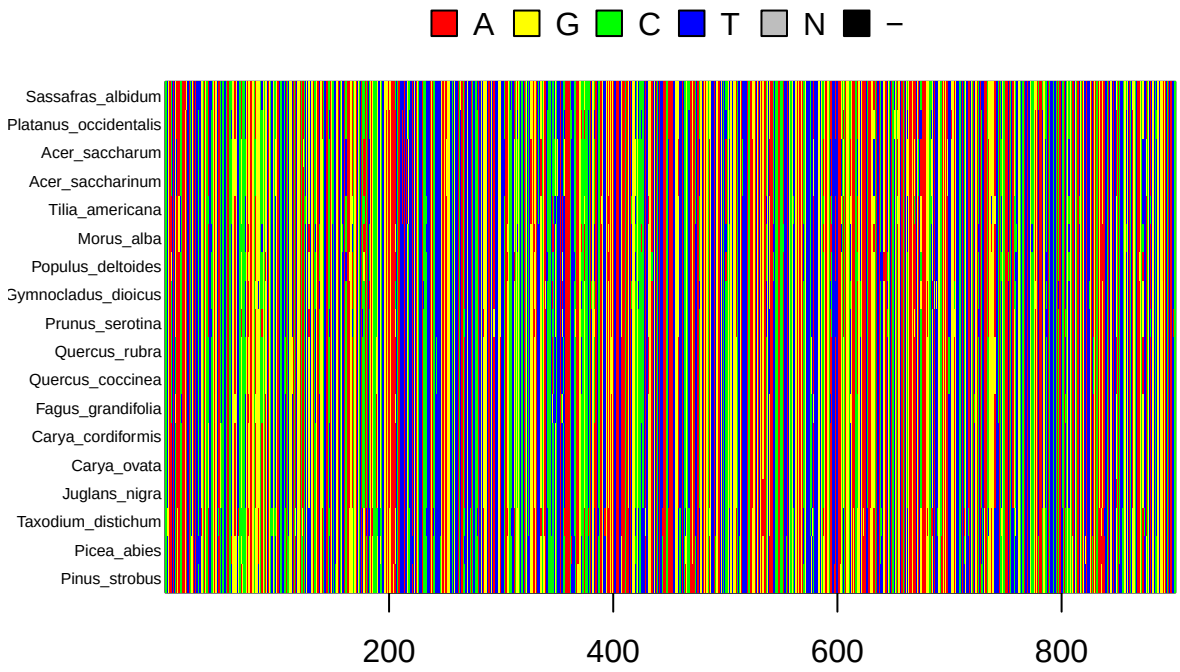
```
plot.phylo(treephylo$scenario.3, cex = .4)
```




```
treephylo$scenario.3
```

```
##
## Phylogenetic tree with 141 tips and 127 internal nodes.
##
## Tip labels:
##   Ilex_opaca, Paulownia_spp, Catalpa_bignonioides, Catalpa_speciosa, Fraxinus_profunda, Fraxinus_nig
## Node labels:
##   Spermatophyta, Mesangiospermae, mrcaott2ott969, Pentapetalae, mrcaott248ott27233, mrcaott248ott650
##
## Rooted; includes branch length(s).
```

```
setwd("/cloud/project/QB_biodiversity_project_EH/")
tree.fasta <-readDNAStringSet("/cloud/project/QB_biodiversity_project_EH/Tree_fasta.txt", format = "fas
read.aln.tree <-msaMuscle(tree.fasta)
save.aln.tree <-msaConvert(read.aln.tree, type = "bios2mds::align")

tree.DNAbin.fasta <-as.DNAbin(read.aln.tree)
window.L <-tree.DNAbin.fasta[,100:1000]
image.DNAbin(window.L, cex.lab = .5)
```




```
phyData.aln.tree <-msaConvert(read.aln.tree, type = "phangorn::phyDat")
aln.dist.tree <-dist.ml(phyData.aln.tree)
aln.NJ.tree <-NJ(aln.dist.tree)
fit.tree <- pml(tree = aln.NJ.tree, data = phyData.aln.tree)

# JC69 model
fitJC.tree <- optim.pml(fit.tree, TRUE)
```

```
## optimize edge weights:  -6136.802 --> -6103.154
## optimize edge weights:  -6103.154 --> -6103.154
## optimize topology:  -6103.154 --> -6101.874  NNI moves:  1
## optimize edge weights:  -6101.874 --> -6101.874
## optimize topology:  -6101.874 --> -6101.874  NNI moves:  0
```

```
# GTR model
fitGTR.tree <- optim.pml(fit.tree, 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.tree, fitGTR.tree)
```

```
## Likelihood Ratio Test Table
##   Log lik. Df Df change Diff log lik. Pr(>|Chi|)
## 1  -6101.9 33
## 2  -5592.1 42         9        1019.6  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fitJC.tree)
```

```
## [1] 12269.75
```

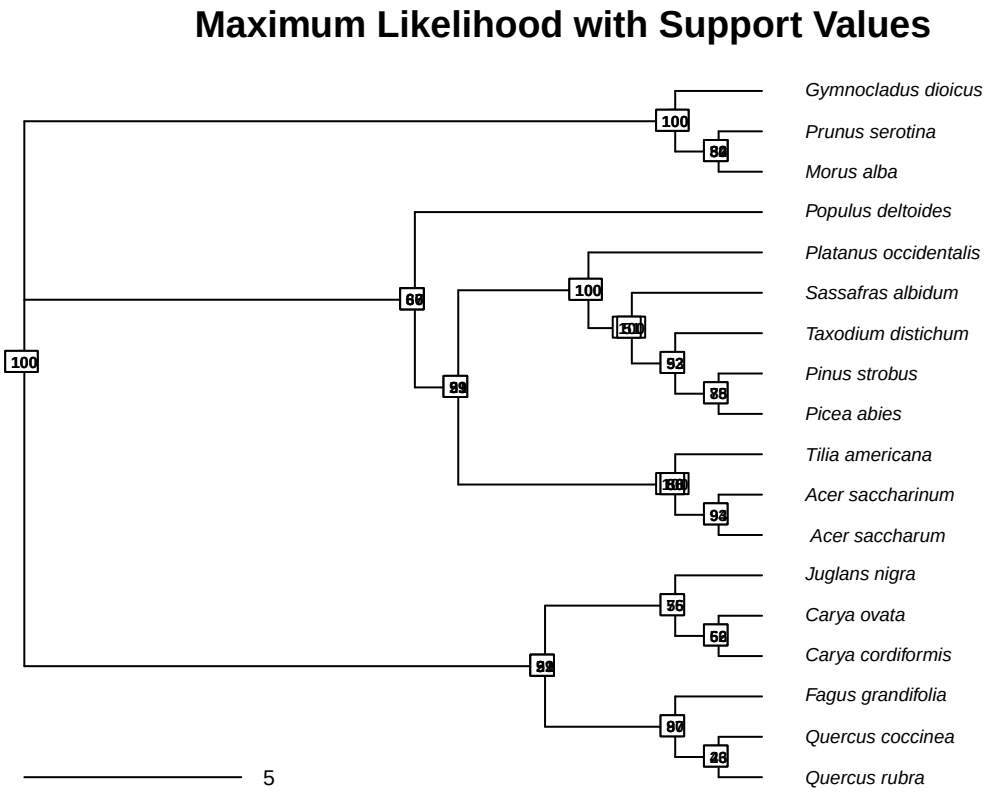
```
AIC(fitGTR.tree)
```

```
## [1] 11268.13
```

```
#phylo tree: aln.NJ.tree


par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(aln.NJ.tree, 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)
```

```
## Warning in XX - width * adj[1]: longer object length is not a multiple of
## shorter object length
```

```
## Warning in YY - height * adj[2]: longer object length is not a multiple of
## shorter object length
```



>tree matches known phylogeny. I do need to fix the labels.

SUBMITTING YOUR ASSIGNMENT

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