8. Worksheet: Phylogenetic Diversity - Traits

Jaeyoung Yoo; Z620: Quantitative Biodiversity, Indiana University

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_Yoo/Week6-PhyloTraits"
```

```
#setwd("/Cloud/project/QB2025_Yoo/Week6-PhyloTraits")
```

2) DESCRIPTION OF DATA

The maintenance of biodiversity is thought to be influenced by **trade-offs** among species in certain functional traits. One such trade-off involves the ability of a highly specialized species to perform exceptionally well on a particular resource compared to the performance of a generalist. In this exercise, we will take a phylogenetic approach to mapping phosphorus resource use onto a phylogenetic tree while testing for specialist-generalist trade-offs.

3) SEQUENCE ALIGNMENT

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

> ***Answer 1***: `p.isolates.afa` contains aligned sequences which creates gaps in the sequences, while `p.isolates.fasta` file is unaligned one with no gap in the sequences.

In the R code chunk below, do the following: 1. read your alignment file, 2. convert the alignment to a DNAbin object, 3. select a region of the gene to visualize (try various regions), and 4. plot the alignment using a grid to visualize rows of sequences.

```
package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger', 'picante', 'stats', 'RColorBrewer
                  "pak", "forcats", "phytools")
for (package in package.list) {
  if (!require(package, character.only=TRUE, quietly=TRUE)) {
    install.packages(package)
    library(package, character.only=TRUE)
  }
}
```

```
##
## Attaching package: 'seqinr'

## The following objects are masked from 'package:ape':
##
##     as.alignment, consensus

##
## Attaching package: 'phylobase'

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

##
## Attaching package: 'phytools'

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

##
## Attaching package: 'permute'

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

```
## This is vegan 2.6-8
##
## Attaching package: 'vegan'
## The following object is masked from 'package:phytools':
##
##     scores
##
## Attaching package: 'nlme'
## The following object is masked from 'package:seqinr':
##
##     gls
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:MASS':
##
##     select
## The following object is masked from 'package:nlme':
##
##     collapse
## The following object is masked from 'package:seqinr':
##
##     count
## The following object is masked from 'package:ape':
##
##     where
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
##
## Attaching package: 'phangorn'
## The following objects are masked from 'package:vegan':
##
##     diversity, treedist
##
## 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 1.19.0
## 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:permute':
##
##     shuffle
## The following object is masked from 'package:geiger':
##
##     is.phylo
## The following object is masked from 'package:phytools':
##
##     untangle
## 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: 'amap'
## The following object is masked from 'package:vegan':
##
##     pca
##
```

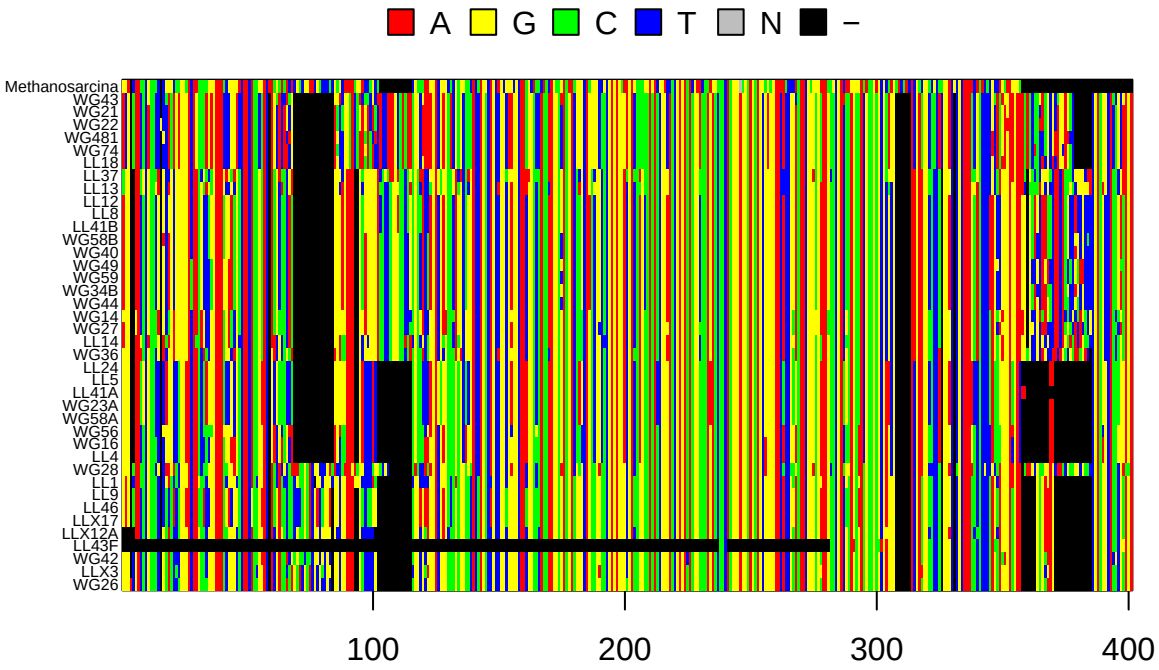
```
## Attaching package: 'scales'

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

## 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.
```

```
#install.packages("pak")
#comment out after first run to prevent continuous reinstall/update
#pak::pkg_install("msa")

library(msa)
```

```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

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

## The following object is masked from 'package:ade4':
##
##     score

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

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

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:tidyr':
##
##     expand

## The following object is masked from 'package:utils':
##
```

```
##     findMatches
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
## The following object is masked from 'package:nlme':
##
##     collapse
## Loading required package: XVector
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package: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
```

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

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

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

# Convert Alignment to DNAbin Object {ape}
p.DNAbin <- as.DNAbin(read.aln)

# Visualize 16S rRNA Gene
window <- p.DNAbin[, 100:500]
image.DNAbin(window, cex.lab = 0.5)
```



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: The 16S rRNA gene sequences are about 400 base pairs. ***Answer 2b***: The sequences from around 250-350 have both conservative and variable base pairs which can be appropriate for phylogenetic inference.

4) MAKING A PHYLOGENETIC TREE

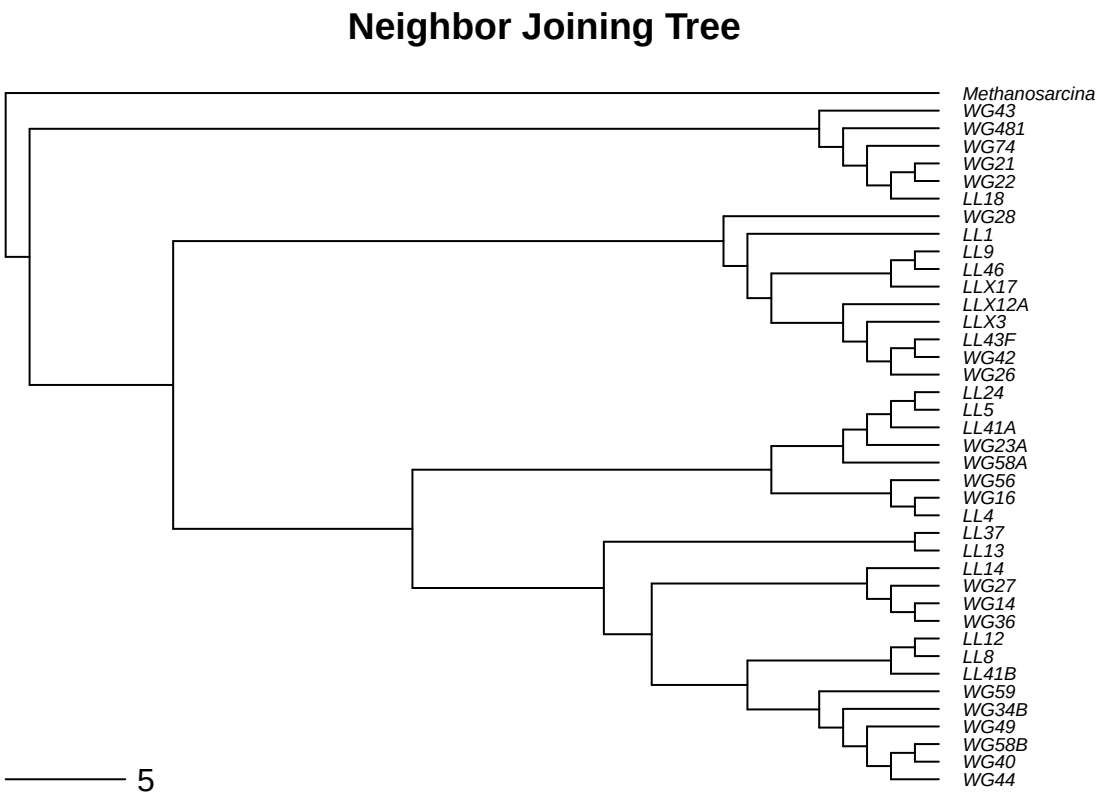
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

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 {ape}
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)

#Neighbor Joining Algorithm to Construct Tree, a 'phylo'
nj.tree <- bionj(seq.dist.raw)

#Identify Outgroup Sequence
outgroup <- match("Methanosarcina", nj.tree$tip.label)

#Root the Tree {ape}
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

#Plot the Rooted Tree {ape}
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(cx = 0.7)
```

```
## Warning in text.default(x + length * 1.1, y, as.character(length), adj = c(0, :
## "cx" is not a graphical parameter
```



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

Answer 3: The advantage of making a neighbor joining tree is visualizing the grouped individuals and check using outgroup to check. The method is less burden in computations than other methods. However, the disadvantage of it is that the tree is less accurate than other trees using Maximum likelihood or Bayesian method.

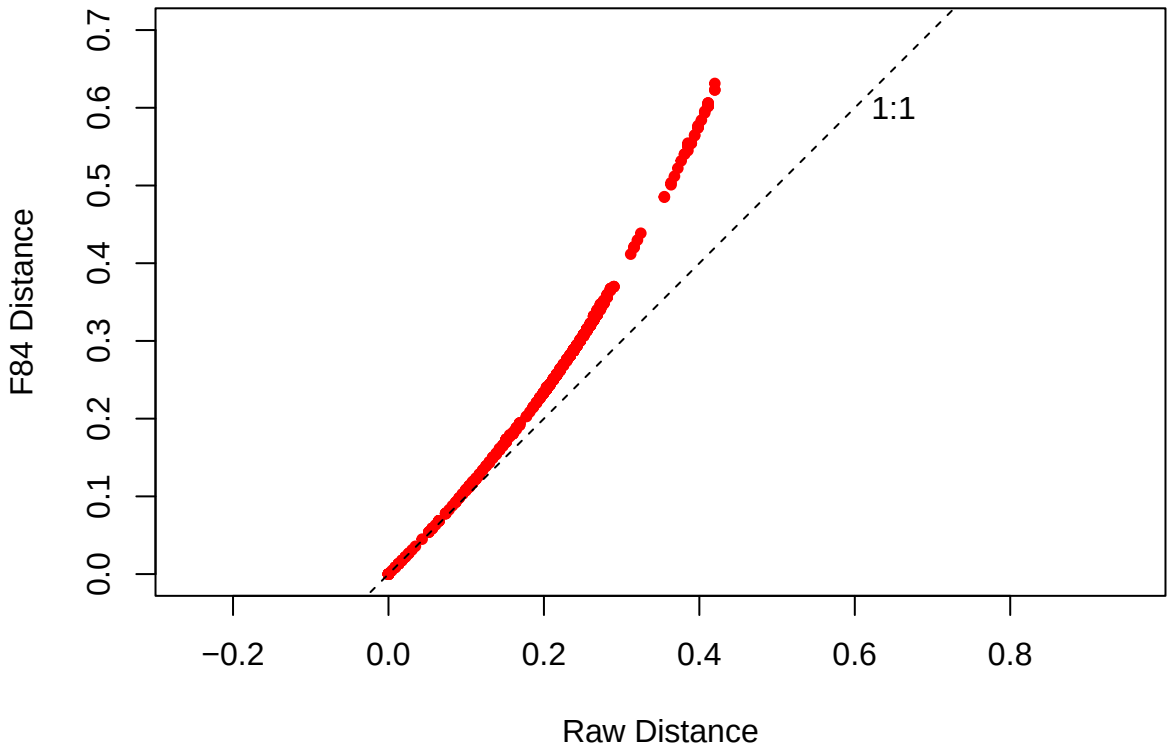
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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.

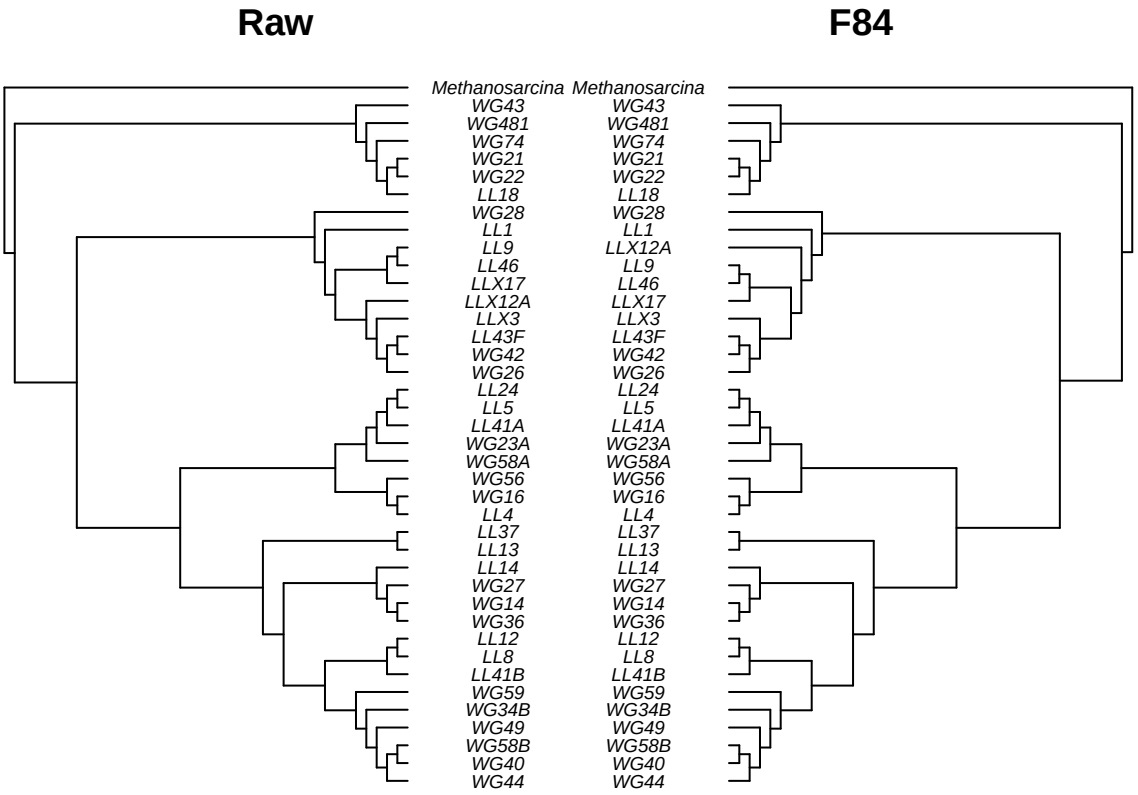
```
#Create distance matrix with "F84" model {ape}
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise.deletion = FALSE)

#Plot Distances from Different DNA Substitution Models
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.raw, seq.dist.F84, pch = 20, col = "red", msa = 1, asp = 1,
     xlim = c(0, 0.7), ylim = c(0, 0.7),
     xlab = "Raw Distance", ylab = "F84 Distance")
```

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

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

## Warning in axis(side = side, at = at, labels = labels, ...): "msa" is not a
## graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "msa" is not a
## graphical parameter

## Warning in box(...): "msa" is not a graphical parameter

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

```
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")
```



C) ANALYZING A MAXIMUM LIKELIHOOD TREE

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

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

#Make the NJ tree for the maximum likelihood method.
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  -9790.4 86         9        1102.1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

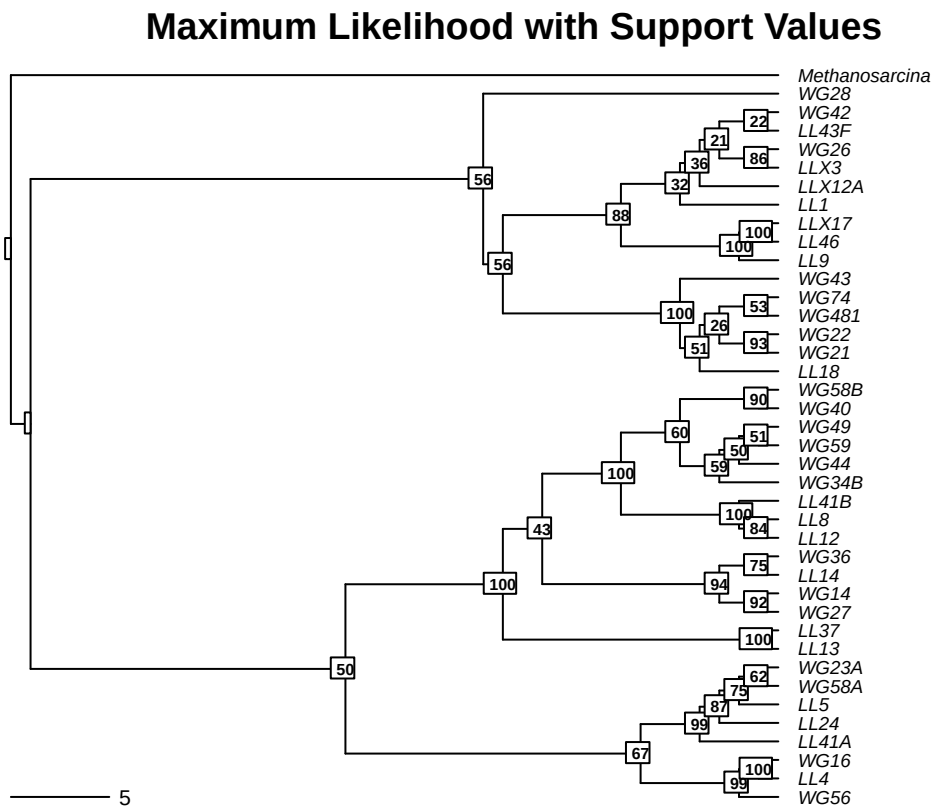
```
AIC(fitJC)
```

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

```
AIC(fitGTR)
```

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

```
#Bootstrap
ml.bootstrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")
par(mar = c(1, 2, 1, 2) + 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 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 trees are quite different. They are different because the neighbor-joining method uses simple distance between two nodes to create a tree, but the maximum likelihood method uses bootstraps and probability to create a model which makes difference. The maximum likelihood tree used bootstrapping to find a best fitted model and it contains percentage of bootstrap value in percentage which makes the tree more reliable than the neighbor-joining tree. ***Answer 4b***: We bootstrap our tree because we can assess the reliability of each branch in the tree by sampling the data and making trees multiple times. ***Answer 4c***: Bootstrap values show the reliability of each branch with statistical support. With higher value, the branch is more reliable. ***Answer 4d***: Branches with low support (bootstrap value<=50%) includes the branch that separates WG42 and LL43F, and the branch that separates WG28 and other individuals. ***Answer 4e***: No, we should not trust branches because low bootstrap values means that the branch could occur by chance and may change if the data changes slightly.

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))

# Levins Index for 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)
}

# Apply Niche Breadth for Each Isolate
nb <- as.data.frame(levins(p.growth.std))
```

C. Visualizing Traits on Trees

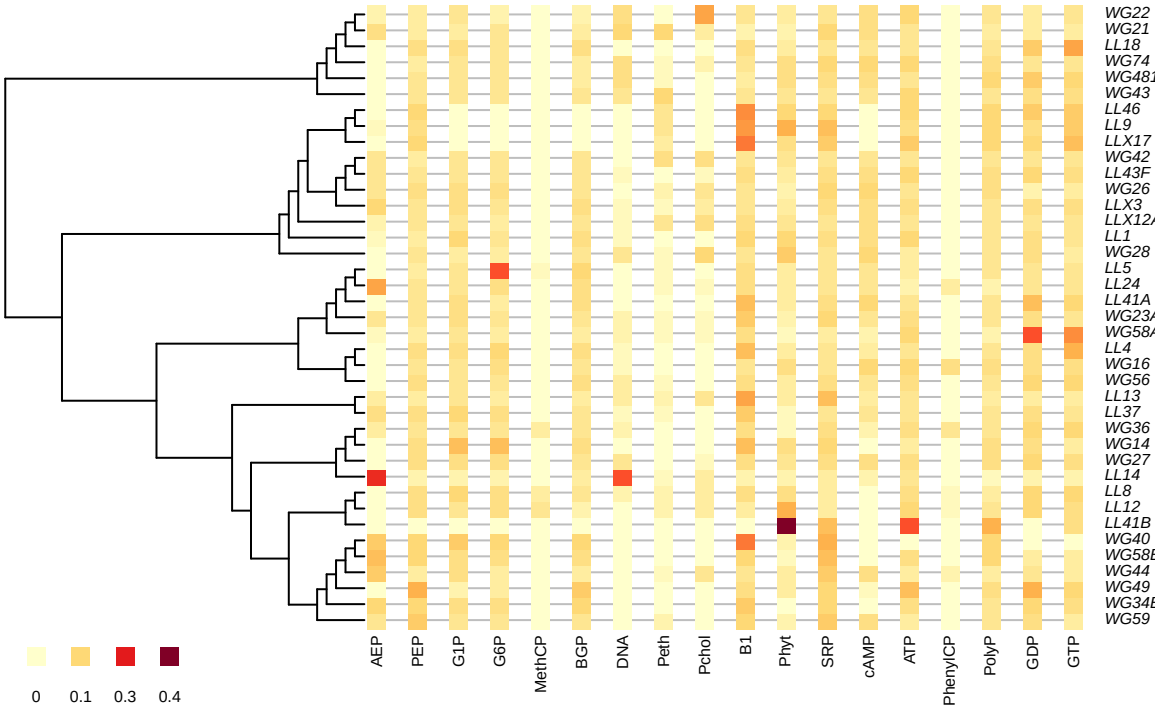
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```
# Generate Neighbor Joining Tree Using F84 DNA Substitution Model [ape]
nj.tree <- bionj(seq.dist.F84)

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

# Create a Rooted Tree [ape]
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

# Keep Rooted but Drop Outgroup Branch
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

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

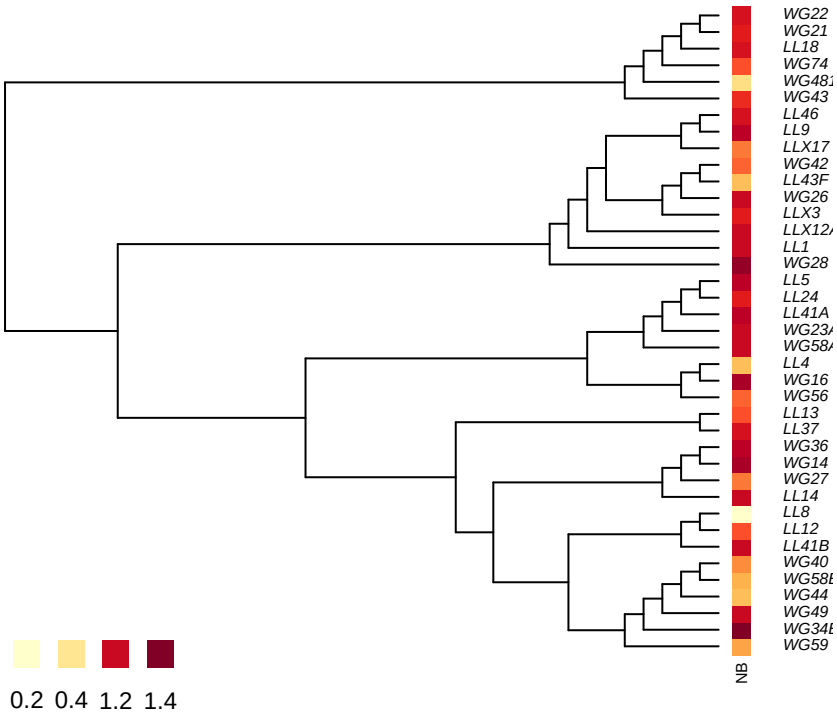
```
# Define Color palette
mypalette <- colorRampPalette(brewer.pal(9, "YlOrRd"))

# First, correct for zero branch-lengths on our tree
nj.plot <- nj.rooted
nj.plot$edge.length <- nj.plot$edge.length + 10^-1

# Map Phosphorus Traits [adephylo]
par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(nj.plot, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = 0.5, scale = FALSE, use.edge.length = FALSE,
              edges.color = "black", edges.width = 2, box = FALSE,
              col = mypalette(25), pch = 15, cex.symbol = 1.25,
              center = FALSE)
```

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

## Warning in plot.window(...): "edges.width" is not a graphical parameter

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

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

## Warning in title(...): "edges.color" is not a graphical parameter

## Warning in title(...): "edges.width" is not a graphical parameter
```




```
# Niche Breadth
par(mar = c(1, 5, 1, 5) + 0.1)
x <- phylo4d(nj.plot, nb)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = 0.5, scale = FALSE, use.edge.length = FALSE,
              edges.color = "black", edges.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)
```

```
## Warning in plot.window(...): "edges.color" is not a graphical parameter
## Warning in plot.window(...): "edges.width" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "edges.color" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "edges.width" is not a graphical parameter
## Warning in title(...): "edges.color" is not a graphical parameter
## Warning in title(...): "edges.width" is not a graphical parameter
```



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: The specialists with narrow niche breadth have higher growth rate in a specific condition with the preferred phosphorus subtrate than the generalist which have a broad niche breath. ***Answer 5b***: I expect to see a negative correlation between growth rate and maximum niche breadth.

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.

```
nj.lambda.5 <- lambdaTree(nj.rooted, lambda = 0.5)
```

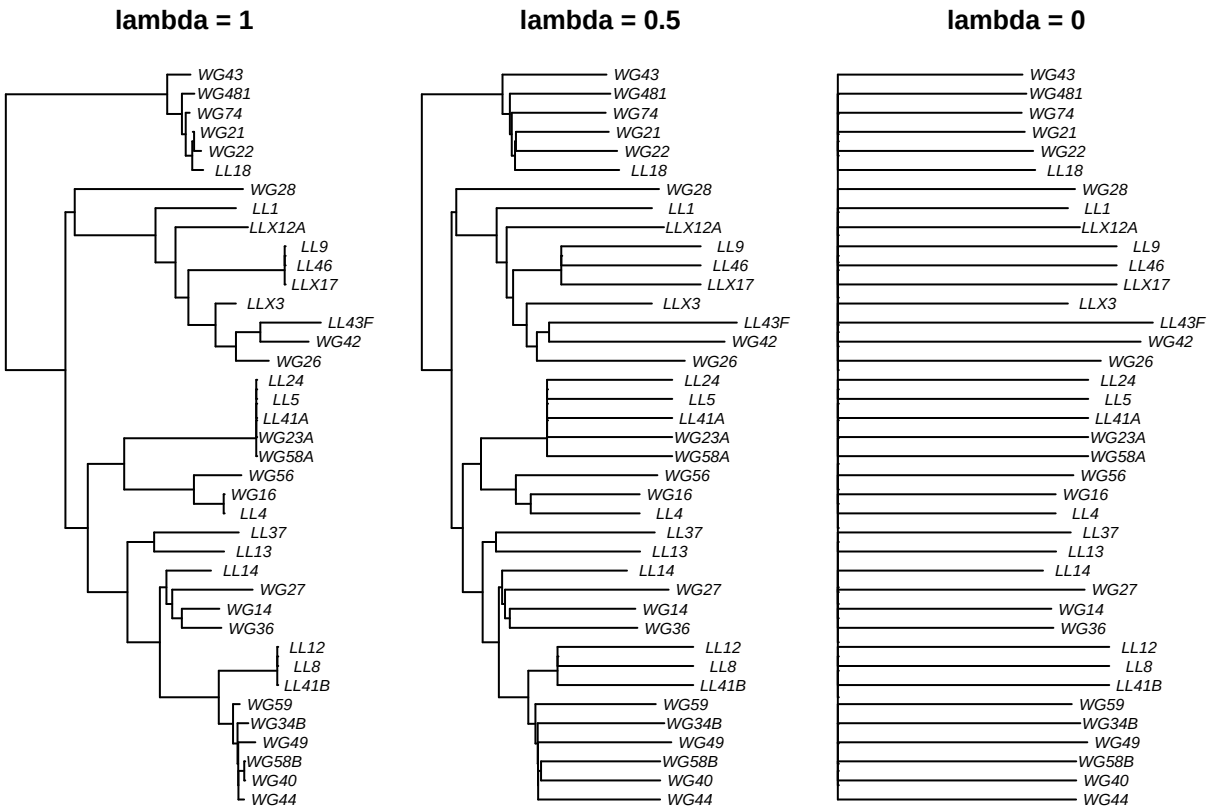
```
## Warning in .deprecate("lambdaTree", "rescale.phylo"): 'lambdaTree' is being
## deprecated: use 'rescale.phylo' instead
```

```
nj.lambda.0 <- lambdaTree(nj.rooted, lambda = 0)
```

```
## Warning in .deprecate("lambdaTree", "rescale.phylo"): 'lambdaTree' is being
## deprecated: use 'rescale.phylo' instead
```

```
layout(matrix(c(1, 2, 3), 1, 3), width = c(1, 1, 1))

par(mar=c(1, 0.5, 2, 0.5) + 0.1)
plot(nj.rooted, main = "lambda = 1", cex = 0.7, adj = 0.5)
```

```
plot(nj.lambda.5, main = "lambda = 0.5", cex = 0.7, adj = 0.5)
plot(nj.lambda.0, main = "lambda = 0", cex = 0.7, adj = 0.5)
```



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

```
# First, correct for zero branch-lengths on our tree
nj.rooted$edge.length <- nj.rooted$edge.length +10^-7

# Generate Test Statistics for Comparing Phylogenetic Signal {geiger}
rownames(nb) <- nj.rooted$tip.label
fitContinuous(nj.rooted, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.110841
##  z0 = 0.654176
##
##  model summary:
##  log-likelihood = 21.004830
##  AIC = -36.009661
##  AICc = -35.323946
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 67
```

```
##  frequency of best fit = 0.670
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
fitContinuous(nj.lambda.0, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.110841
##  z0 = 0.654176
##
##  model summary:
##  log-likelihood = 21.004829
##  AIC = -36.009658
##  AICc = -35.323944
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 85
##  frequency of best fit = 0.850
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

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***: Lambda is 0 in both models, which means that there is no phylogenetic signal in the tree. ***Answer 6b***: AIC for both models are -36.009658, so we cannot decide which model is better. ***Answer 6c***: Since lambda is 0 and AIC for both models are identical, there is no significant phylogenetic signal in the tree. However, I am not sure my codes are right, because visually the trees are very different.

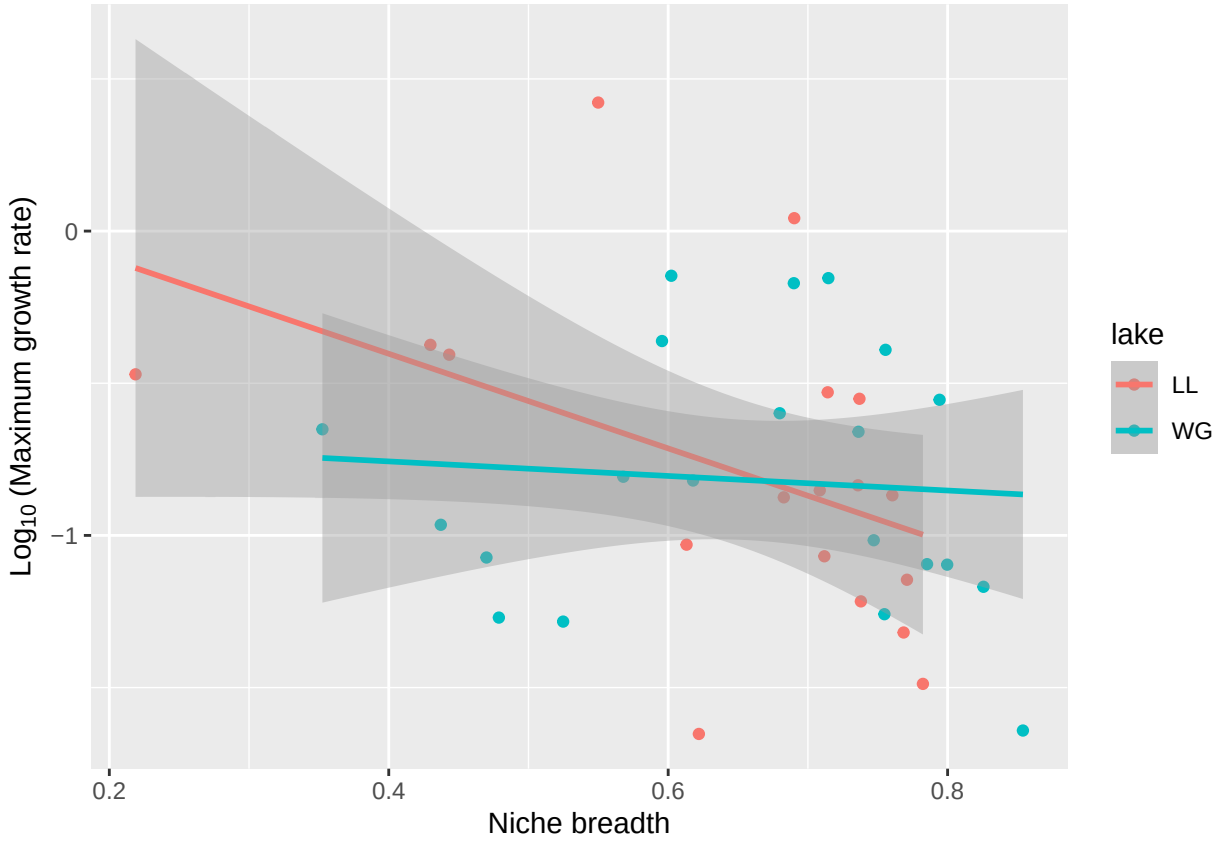
7) PHYLOGENETIC REGRESSION

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

the trait dataset, taking into account the bacterial phylogeny

```
# Using the niche breadth data, create a column indicating lake origin
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 maximum 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'
```



```r
# 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
## -0.90404 -0.29747 -0.01068  0.26099  1.05846
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2189     0.4897   0.447   0.6576
## NB           -1.5550     0.7365  -2.111   0.0419 *
## lakeWG       -0.8801     0.6875  -1.280   0.2089
## NB:lakeWG     1.3159     1.0295   1.278   0.2096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4566 on 35 degrees of freedom
## Multiple R-squared:  0.1163, Adjusted R-squared:  0.04055
## F-statistic: 1.535 on 3 and 35 DF,  p-value: 0.2225
```

```r
AIC(fit.lm)
```

```
## [1] 55.30868
```

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

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = nj.rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  52.31 -20.15
##
## Raw residuals:
##     Min      1Q  Median      3Q     Max
## -0.9650 -0.3661 -0.1143  0.1585  0.9827
##
## Mean tip height: 0.1814508
## Parameter estimate(s) using ML:
## lambda : 0.2794888
## sigma2: 1.027255
##
## Coefficients:
##             Estimate   StdErr t.value p.value
## (Intercept)  0.40814  0.48068  0.8491 0.40161
## NB          -1.76128  0.69000 -2.5526 0.01521 *
## lakeWG      -0.95998  0.62212 -1.5431 0.13180
```

```
## NB:lakeWG    1.51288  0.92192  1.6410 0.10975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1597    Adjusted R-squared: 0.08766
##
## Note: p-values and R-squared are conditional on lambda=0.2794888.
```

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

```
## [1] 52.30788
```

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***: It is because the species are not independent because of shared ancestry, so their traits can be from inheritance, not only by traits themselves. ***Answer 7b***: The standard linear regression simply assumes the species are independent while the phylogenetic regression assumes the connection among species and corrects the effect of phylogenetic similarity by shared ancestry. ***Answer 7c***: The slopes for both models are negative, which means that the individuals with higher niche breath have low maximum growth rate. However, the phylogenetic regression model ($R^2 = 0.0877$, AIC = 52.31) is better fit than the standard linear regression model ($R^2 = 0.0406$, AIC = 55.31). ***Answer 7d***: If the phylogenetic tree contains species which live only in one ecosystem with few variation in the environmental conditions, so there is no character displacement by different environment, there traits such as body size can be mostly due to their evolutionary history and genes, not by other environmental factors or other traits.

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?

> Answer: Sorry, but Ashish and I are still working on getting the gene for tree species, so it will take more time. We will add this part on next week worksheet!

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)**.