8. Worksheet: Phylogenetic Diversity - Traits

Jonathan Enriquez Madrid; Z620: Quantitative Biodiversity, Indiana University

23 February, 2023

OVERVIEW

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

After completing this exercise you will be able to:

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

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo, at whatever stage you are. This will enable you to pull your work onto your own computer.
6. When you have completed the worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '8.BetaDiversity' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**11.PhyloTraits_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**11.PhyloTraits_Worksheet.pd**

The completed exercise is due on **Wednesday, February 22nd, 2023 before 12:00 PM (noon)**.

1) SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:
1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/8.PhyloTraits*" folder, and
4. load all of the required R packages (be sure to install if needed).

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

```
## [1] "C:/Users/jonat/GitHub/QB2023_Enriquez_Madrid/2.Worksheets/8.PhyloTraits"
```

```
setwd("C:/Users/jonat/GitHub/QB2023_Enriquez_Madrid/2.Worksheets/8.PhyloTraits")
```

2) DESCRIPTION OF DATA

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

3) SEQUENCE ALIGNMENT

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

Answer 1:

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

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

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

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

```
##
## Attaching package: 'phylobase'
```

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

##
## Attaching package: 'permute'

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

## This is vegan 2.6-4

##
## Attaching package: 'nlme'

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

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

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

## The following object is masked from 'package:seqinr':
##
##     count

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

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

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

##
## Attaching package: 'phytools'

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

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

##
## Attaching package: 'cluster'

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

## Registered S3 method overwritten by 'dendextend':
##   method     from
##   rev.hclust vegan

##
## --------------------
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## --------------------

##
## Attaching package: 'dendextend'

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

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

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

## The following objects are masked from 'package:phylobase':
##
##     labels<-, prune

## The following objects are masked from 'package:ape':
##
##     ladderize, rotate

## The following object is masked from 'package:stats':
##
##     cutree

##
## Attaching package: 'phylogram'

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

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

##
## Attaching package: 'scales'

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

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

```
##
## Attaching package: 'BiocGenerics'

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

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

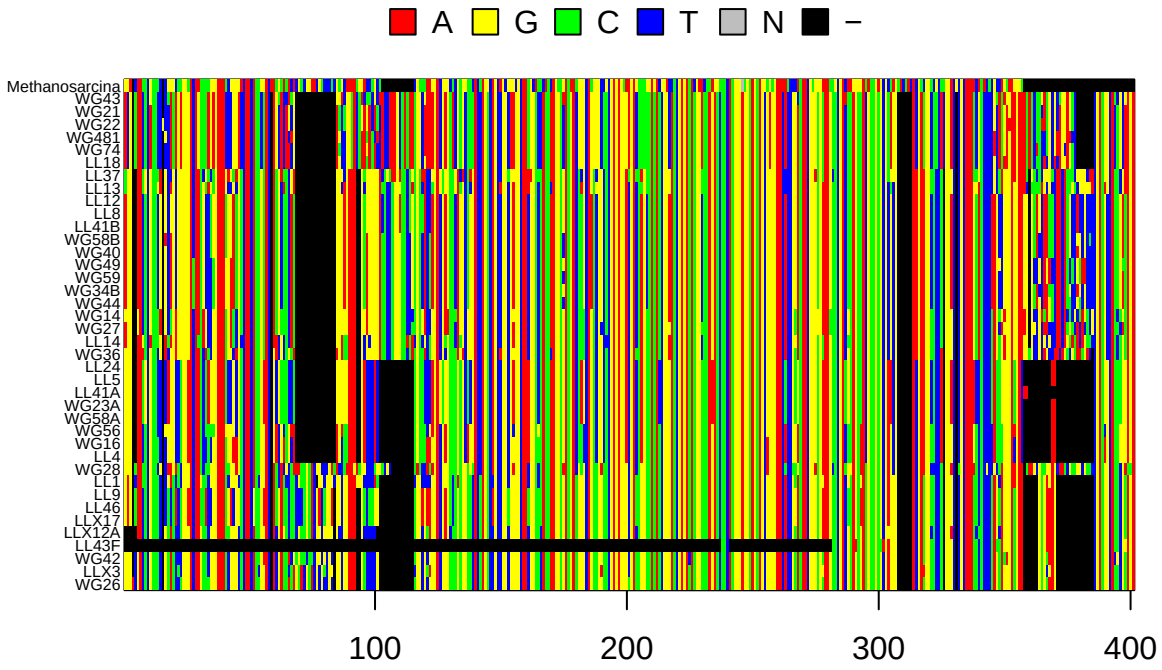
```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

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

##
## Attaching package: 'S4Vectors'

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

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

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

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

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

## The following object is masked from 'package:grDevices':
##
##     windows

## 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)

#Import unaligned sequence
seqs <- readDNAStringSet("data/p.isolates.fasta", format = 'fasta')
seqs
```

```
## DNAStringSet object of length 40:
##      width seq                                                names
##  [1]   619 ACACGTGAGCAATCTGCCCTTCT...TTCTCTGGGAATACCTGACGCT LL9
##  [2]   597 CGGCAGCGGGAAGTAGCTTGCTA...AACTGTTCAGCTAGAGTCTTGT WG14
##  [3]   794 CAGCGGCGGACGGGTGAGTAACA...GCTAACGCATTAAGCACTCCGC WG28
##  [4]   716 CTTCAGAGTTAGTGGCGGACGGG...TGCTAGTTGTCGGGATGCATGC LL24
##  [5]   803 ACGAACTCTTCGGAGTTAGTGGC...TAAAACTCAAAGGAATTGACGG LL41A
##  ...   ... ...
## [36]   652 TTCGGGAGTACACGAGCGGCGAA...TTCTCTGGGAATACCTGACGCT LL46
## [37]   661 GCGAACGGGTGAGTAACACGTGG...GAGCGAAAGCGTGGGTAGCGAA WG26
## [38]   694 GGCGAACGGGTGAGTAACACGTG...ACCCTGGTAGTCCACGCCGTAA WG42
## [39]   699 TACAGGTACCAGGCTCCTTCGGG...AAAGCATGGGTAGCGAACAGGA LLX17
## [40]  1426 TTCTGGTTGATCCTGCCAGAGGT...AACCTNAATTTTGCAAGGGGGG Methanosarcina
```

```
#Align sequence using MUSCLE
read.aln <- msaMuscle(seqs)

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

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

#Identify base pair region of 16s rRNA to visualize
window <- p.DNAbin[, 100:500]

#Command to visualize 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: Our sequence reads span from 597 base pairs to 1426 base pairs. All these seqences are from the 16S rRNA gene.
Answer 2b:Regions I think would be appropriate for phylogenetic inferences are regions that are most similar, as they are shared across the two species we are looking at. We can then identify the differences in these simialr regions and see how the two species vary in their sequences, and how this translates to variation in traits.

4) MAKING A PHYLOGENETIC TREE

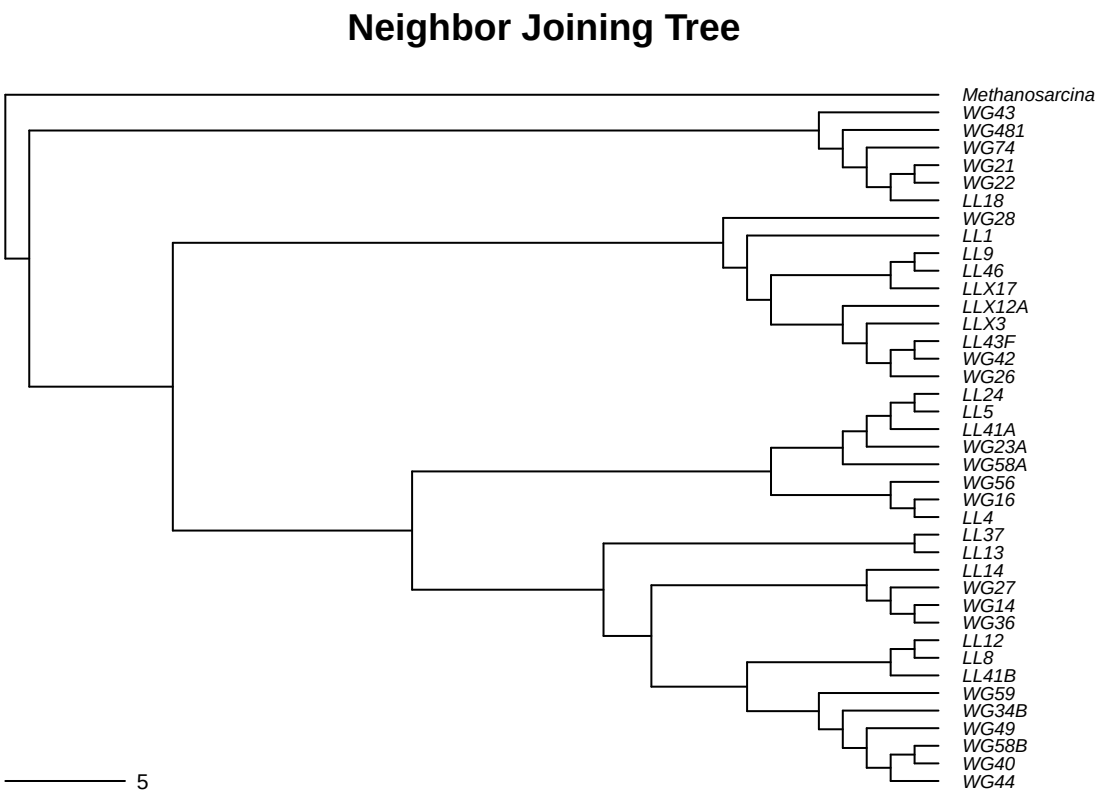
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

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 w/ "raw" model
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)

#neighbor joining algorithm to construct tree
nj.tree <- bionj(seq.dist.raw)

#identify outgroup sequence
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?

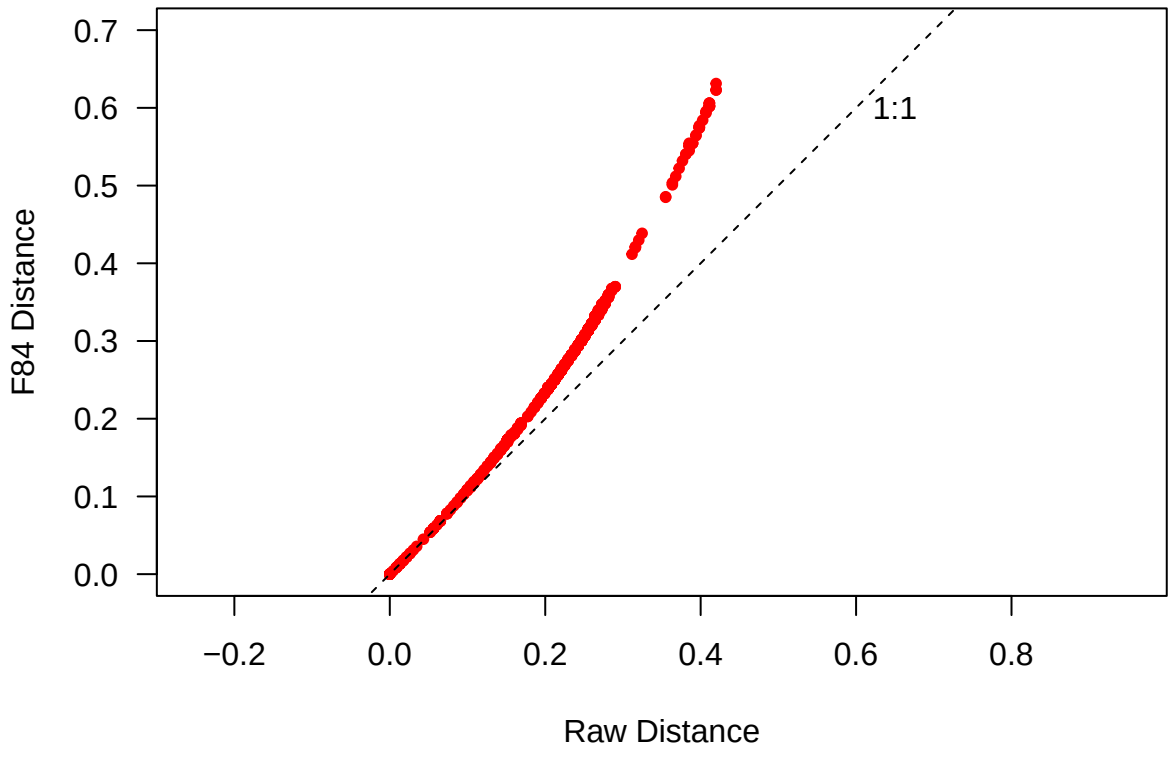
> ***Answer 3***:The advantage of making a neighbor joining tree is that it can be used as a "guide tree" that can be used in more sophisticated models. The disadvantage is that the tree starts

with a "star network" in which all species have the same common ancestor and all are equally related to each other, which is not true in reality. Other disadvantages are that the neighbor joining tree does not take into account multiple substitutions that may have occured at a site, nor does it take into account substitution bias for one nucleotide over the other.

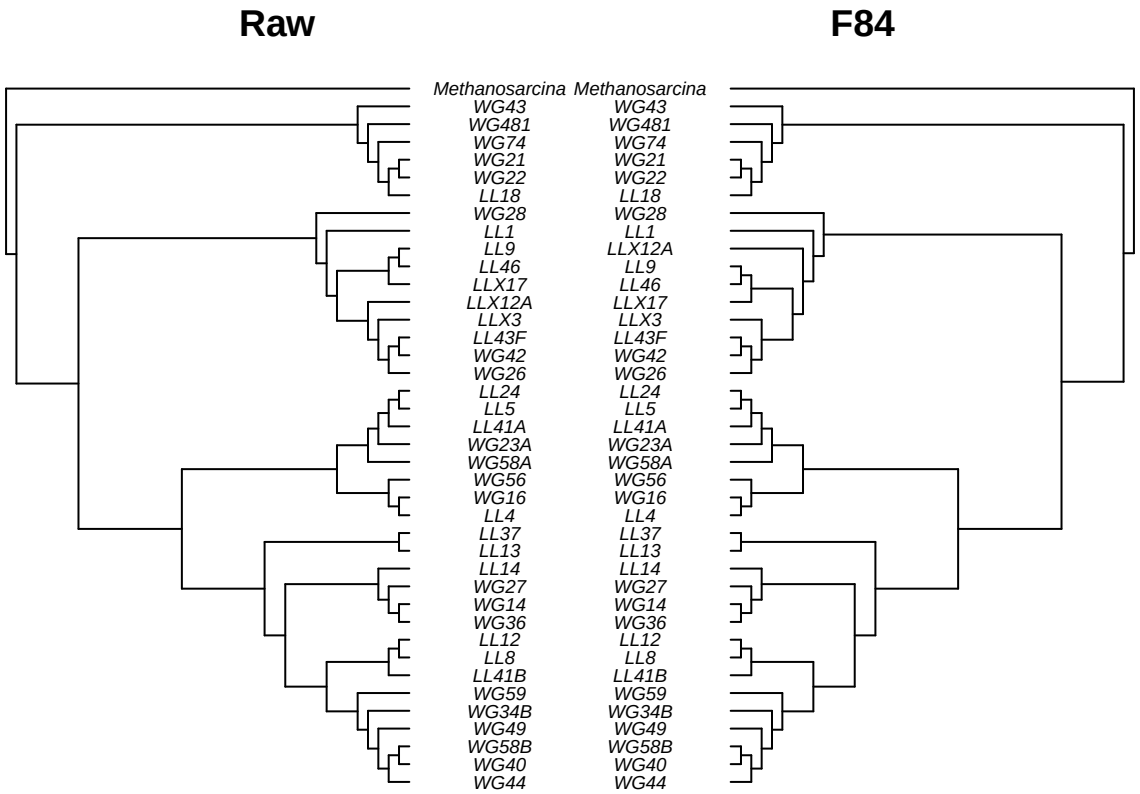
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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 w/ "F84" model
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", 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")
```



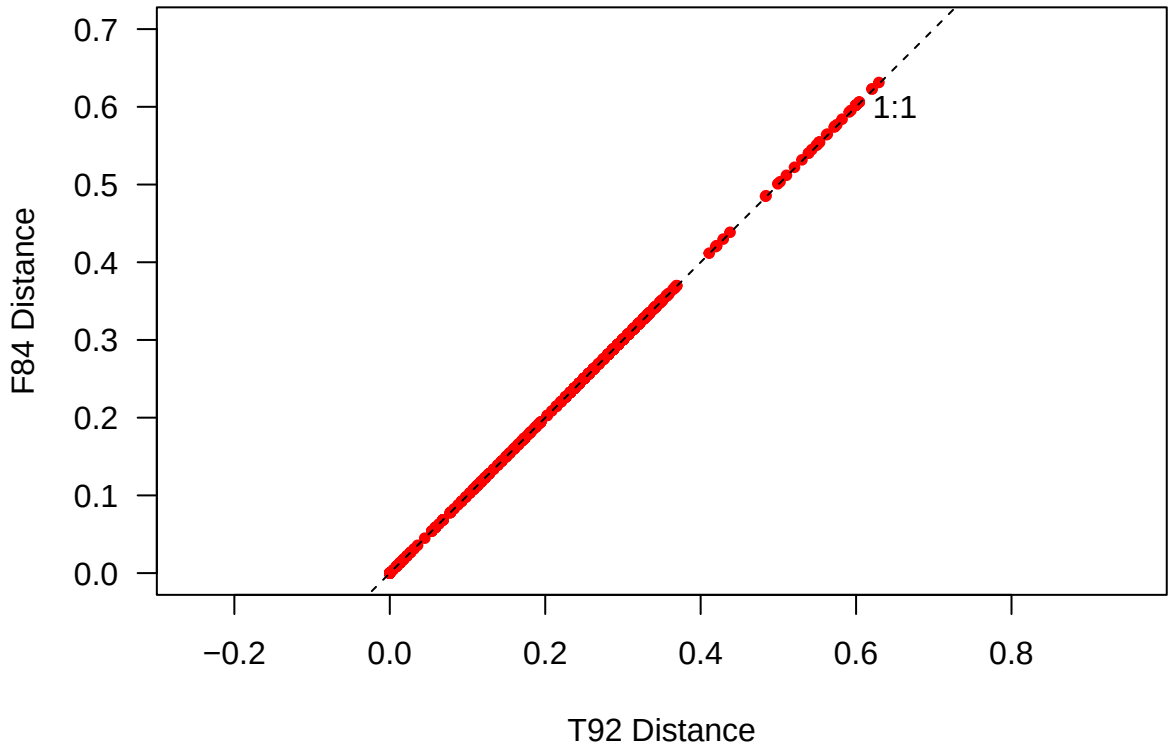
```
#Make neighbor joining tree using diff. DNA substitution models
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)

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

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

#make cophylogenetic plot
layout(matrix(c(1, 2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(raw.rooted, type = "phylogram", direction = "right", show.tip.label = TRUE, use.edge.length =
           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 =
           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,

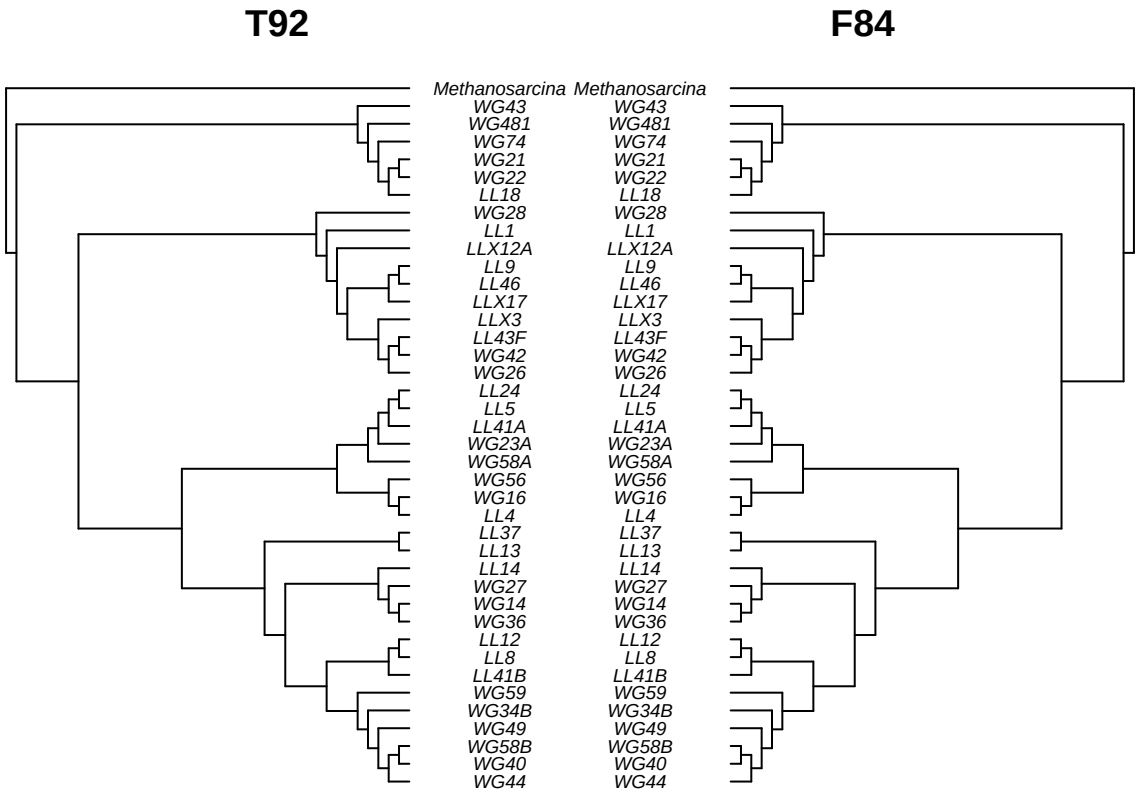
4. make a cophylogenetic plot that compares the topologies of both models, and
5. be sure to format, add appropriate labels, and customize each plot.

```
#create distance matrix w/ "T92" model
seq.dist.T92 <- dist.dna(p.DNAbin, model = "T92", pairwise.deletion = FALSE)

#Plot distances from different DNA substitution models (saturation plot)
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.T92, seq.dist.F84,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7),
     ylim = c(0, 0.7), xlab = "T92 Distance", ylab = "F84 Distance")
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1") #They align very well. One on top of the other.
```




```
#Make neighbor joining tree using diff. DNA substitution models
T92.tree <- bionj(seq.dist.T92)
F84.tree <- bionj(seq.dist.F84)

#Define outgroups
T92.outgroup <- match("Methanosarcina", T92.tree$tip.label)
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)

#Root the tree
T92.rooted <- root(T92.tree, T92.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)
```

```
#make cophylogenetic plot
layout(matrix(c(1, 2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(T92.rooted, type = "phylogram", direction = "right", show.tip.label = TRUE, use.edge.length =
           cex = 0.6, label.offset = 2, main = "T92")

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



Question 4:

a. Describe the substitution model that you chose. What assumptions does it make and how does it compare to the F84 model?
b. Using the saturation plot and cophylogenetic plots from above, describe how your choice of substitution model affects your phylogenetic reconstruction. If the plots are inconsistent with one another, explain why.
c. How does your model compare to the *F84* model and what does this tell you about the substitution rates of nucleotide transitions?

Answer 4a: The substitution model that I chose is Tamura model (T92). It assumes equal frequencies of nucleotides, but recognizes higher probability rates for transition mutations than for transversion mutations, in addition to accounting for G + C content.
Answer 4b: Choice of substitution model does affect the phylogenetic reconstruction. When comparing the "raw" model to the "F84" model we can see that there are differences in the two

phylogenetic trees. These differences have to do with how LL9, LL46, LLX17, and LLX12A are arranged in both trees. When comparing the "T92" model and "F84" model, I do not see any differences between trees, and the differences that are between the "raw" model and "F84" model disappear when comparing the "T92" and "F84" model. In addition, the saturation plots show that the "raw" model and "F84" model fit less well than the "T92" and "F84" models, as the "T92" and "F84" seem identical when viewed on the saturation plot.
Answer 4c: My model, "T92" compares very well with the "F84" model. This can be seen both with the saturation plot, as both models lay on top of each other, and when comparing the phylogenetic trees, as I see no differences between trees. This tells me that the substitution rates of nucleotide transitions are recognized in both the "T92" and "F84" model, but not in the "raw" model.

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

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

#Make NJ tree for 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 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 GTR model
fitGTR <- optim.pml(fit, model = "GTR", optInv = TRUE, optGamma = TRUE, rearrangement = "NNI", control =
```

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

```
#Perform model selection
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
```

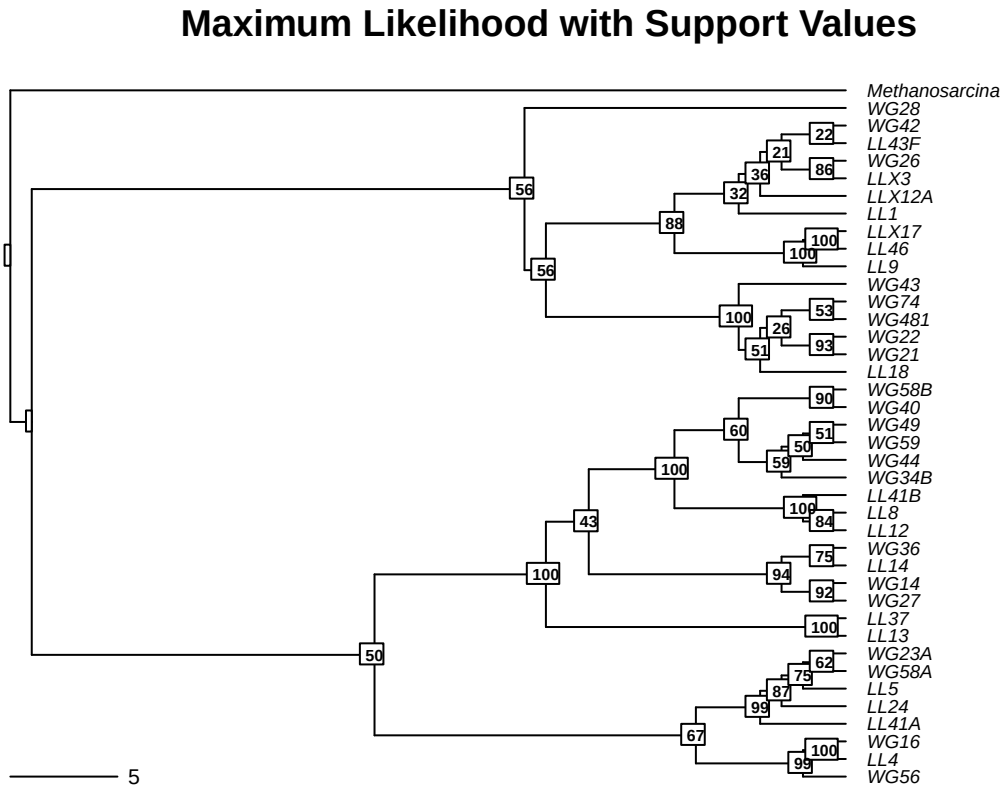
```
AIC(fitJC)
```

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

```
AIC(fitGTR)
```

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

```
#Bootstrapping
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
add.scale.bar(cex = 0.7)
nodelabels(ml.bootstrap$node.label, font = 2, bg = "white",
           frame = "r", cex = 0.5)
```



Question 5:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches?

Answer 5a: The maximum likelihood tree is different from the neighbor-joining tree. This is because the maximum liklihood tree uses statistical methods to find the best tree to fit the data, while the neighbor-joining tree does not. The maximum likelihood tree also takes into account nucleotide states unlike the neighbor-joining tree which only accounts for a distance matrix (a distance matrix that may or may not be correct). In addition, the maximum likelihood tree is also less effected by sampling error.
Answer 5b:The reason for bootstrapping is to determine how reliable the tree is. By bootstrapping we are making many different copies of our original tree and seeing how often the branches from these new trees match the branches of the original tree. If the branches from the copies match the branches from the original, than we are fairly certain that the branch represents the data correctly.
Answer 5c:Bootstrap values tell us how well the branch/tree fits the data, how reliable the tree is in explaining the data. Bootstrap values that are 95% and higher can be treated as correct, values from 94% to 70% are moderately supported, and values that are 50% and below are not supported.
Answer 5d: Bracnhes that have a bootstrapping value of 50% or lower have very low support. One example in our tree would be the node between WG42 and LL43F.
Answer 5e:Branches that can be trusted to show the correct structure of the tree are branches with values of 95% or higher. Any branhces that have values below this either have moderate support to them, or no support at all.

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 <- 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
nb <- setNames(as.vector(nb), as.matrix(row.names(p.growth)))
```

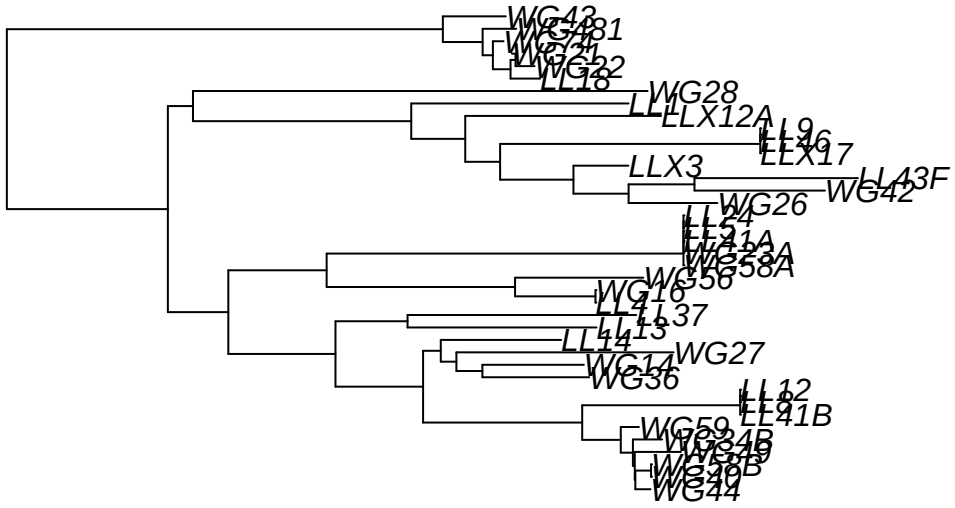
C. Visualizing Traits on Trees

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.

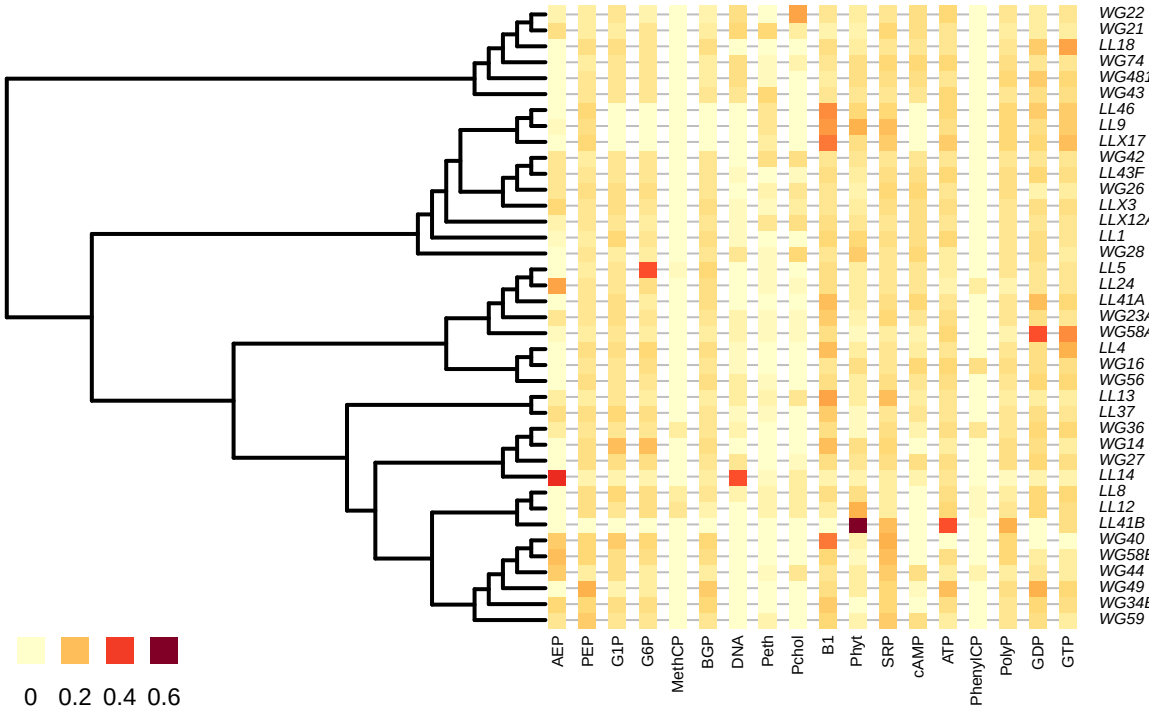
```
#Generate neighbor joining tree using F84 model
nj.tree <- bionj(seq.dist.F84)

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

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

#Keep rooted but drop outgroup
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).

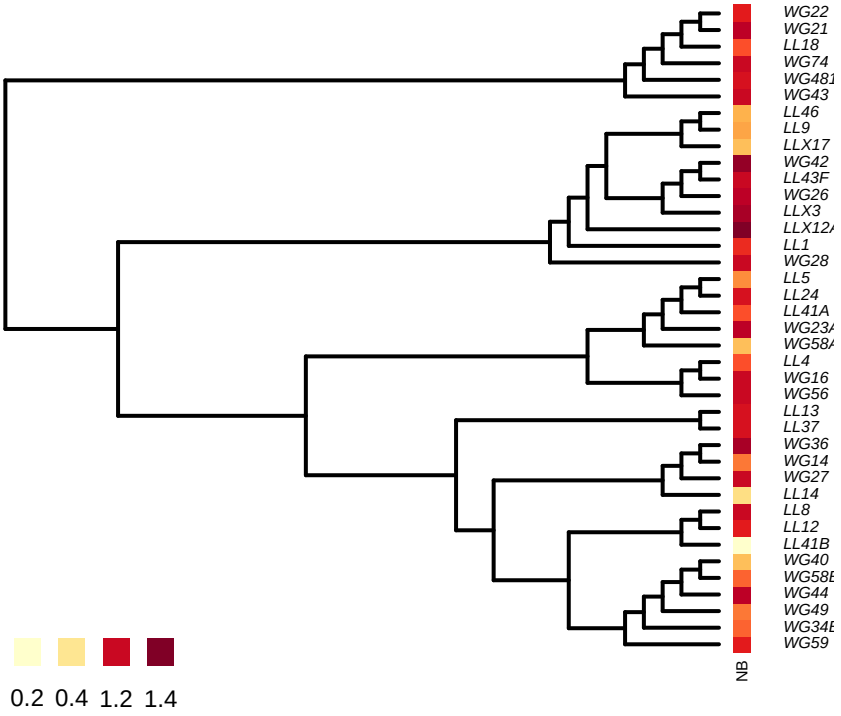
```
#Plot tree
plot(nj.rooted)
```




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

#Correct for zero branch length
nj.plot <- nj.rooted
nj.plot$edge.length <- nj.plot$edge.length + 10^-1

#Map phosphorus traits
par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(nj.plot, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FALSE
              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, nb)
table.phylo4d(x.nb, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = F
```



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?

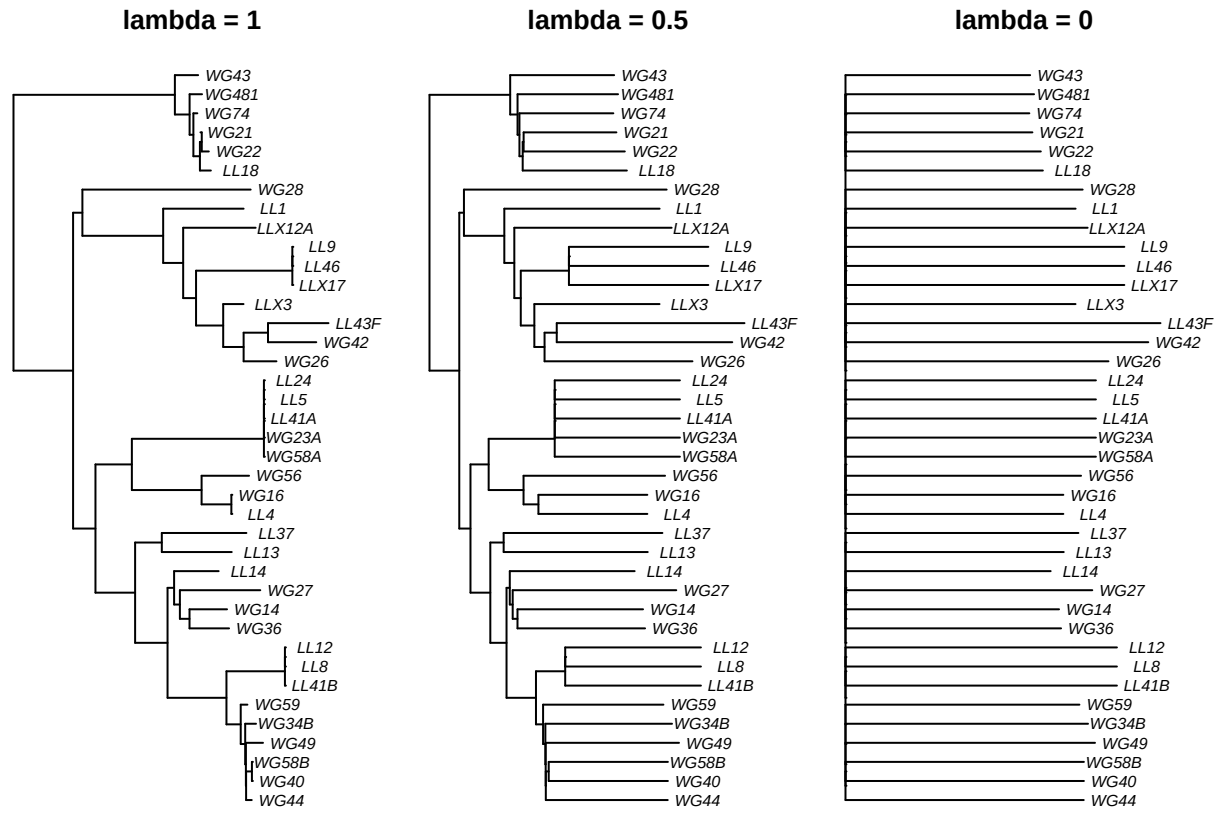
Answer 6a:Bacterial isolates that are generalists should have greater values for niche breadth, compared to specialists, as generalists can grow on many different resources (types of phosphorus), which is characteristic of a large niche breadth. In addition, generalists should also have lower maximum growth rates, compared to specialists, as there is a cost with allocating energy to being able to grow on many different types of resources (types of phosphorus).
Answer 6b:We would expect to see large niche breadth values for generalist isolates and small niche breadth values for specialist isolates. We would also expect to see low maximum growth rates for generalsts, and large maximum growth rates for specialists.

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.

```
#Visualize trees w/ diff. levels of phylogenetic signal
nj.lambda.5 <- geiger::rescale(nj.rooted, "lambda", 0.5)
nj.lambda.0 <- geiger::rescale(nj.rooted, "lambda", 0)
layout(matrix(c(1, 2, 3), 1, 3), width = c(1, 1, 1))
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.

```
#Statistic for comparing phylogenetic signal
fitContinuous(nj.rooted, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.006976
##  sigsq = 0.108060
##  z0 = 0.657697
##
##  model summary:
##  log-likelihood = 21.503414
##  AIC = -37.006827
##  AICc = -36.321113
```

```
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 42
##  number of iterations with same best fit = NA
##  frequency of best fit = NA
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.965171
##  sigsq = 0.108048
##  z0 = 0.656477
##
##  model summary:
##  log-likelihood = 21.502505
##  AIC = -37.005010
##  AICc = -36.319295
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 89
##  frequency of best fit = 0.89
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
#Compare lambda score w/ likelihood ratio
phylosig(nj.rooted, nb, method = "lambda", test = TRUE)
```

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

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

Answer 7a:The lambda value for the untransformed tree is 0, and the lambda value for the transformed tree (lambda = 0) is 0.965171. Is this correct? Being that the transformed tree is equal to lambda = 0, shouldn't it have a lambda value of 0 instead of 0.965171? The same question goes for the untransfomred tree. The untransformed tree is equal to lambda = 1, yet has a lambda value of 0 in the fitContinuos() function. ***Answer 7b***:Both models have the same AIC value of -37.005010, meaning that neither model is a better fit than the other.
Answer 7c: This result suggests that there is no phylogenetic signal as both models, the untransformed where lambda =1 and the transformed where lambda = 0, have the same AIC value meaning that neither model is a better fit of the data. Being that the two models are not different in how they fit the data, the transformed model which has no phylogenetic signal is no different from the untransformed model that shows phylogenetic signal, indicating that there is a phylogenetic signal.

B) Phylogenetic Signal: Blomberg's K

In the R code chunk below, do the following:
1. correct tree branch-lengths to fix any zeros,
2. calculate Blomberg's K for each phosphorus resource using the `phylosignal()` function,
3. use the Benjamini-Hochberg method to correct for false discovery rate, and
4. calculate Blomberg's K for niche breadth using the `phylosignal()` function.

```
#Correct for Zero branch-lengths
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7

#Create 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")

#For-loop to calculate Blomberg's K
for (i in 1:18){
  x <- setNames(as.vector(p.growth.std[,i]),row.names(p.growth))
  out <- phylosignal(x, nj.rooted)
  p.phylosignal[1:5, i] <- round(t(out), 6)
}

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

#Check results (growth rates)
print(p.phylosignal)
```

```
##                      AEP         PEP         G1P         G6P     MethCP
## K               0.000007    0.000008    0.000006    0.000002   0.000005
## PIC.var.obs  4050.684525  659.174701  926.261779 5887.269602 350.858812
## PIC.var.mean 7299.704580 1407.029998 1678.238920 3302.812194 447.362651
## PIC.var.P       0.293000    0.098000    0.151000    0.807000   0.408000
## PIC.var.z      -0.720915   -1.202121   -1.002341    1.071484  -0.287832
## PIC.P.BH        0.586000    0.294000    0.388000    0.854000   0.668000
##                      BGP         DNA        Peth       Pchol          B1
## K               0.000011    0.000087    0.000037    0.000025    0.000005
## PIC.var.obs   510.560954  237.150194  192.519559  397.370385 3357.131085
```

```
## PIC.var.mean 1579.373932 4720.542396 1620.579680 3007.989010 4752.631634
## PIC.var.P       0.045000    0.004000    0.007000    0.007000    0.285000
## PIC.var.z      -1.541660   -1.191216   -1.736710   -1.522639   -0.647970
## PIC.P.BH        0.162000    0.032000    0.032000    0.032000    0.586000
##                     Phyt         SRP        cAMP         ATP    PhenylCP
## K               0.000003    0.000005    0.000017    0.000002    0.000002
## PIC.var.obs  9230.268766 1166.315676  678.817257 3942.591218 1224.017444
## PIC.var.mean 8009.518247 1432.694255 2692.423386 2829.204895  676.389392
## PIC.var.P       0.631000    0.340000    0.007000    0.644000    0.871000
## PIC.var.z       0.159114   -0.489058   -2.242245    0.502959    1.201671
## PIC.P.BH        0.773000    0.612000    0.032000    0.773000    0.871000
##                    PolyP         GDP         GTP
## K               0.000004    0.000002    0.000003
## PIC.var.obs  1081.902144 4469.581412 2714.559889
## PIC.var.mean 1100.273554 3290.518387 2591.720178
## PIC.var.P       0.551000    0.700000    0.584000
## PIC.var.z      -0.032784    0.550455    0.093854
## PIC.P.BH        0.773000    0.787000    0.773000
```

```
#Phylogenetic signal for Niche breadth
signal.nb <- phylosignal(nb, nj.rooted)
signal.nb
```

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 3.608803e-06         48546.48              45186.09          0.598
##   PIC.variance.Z
## 1      0.1750324
```

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 does not seem to be a significant phylogenetic signal for niche breadth. However, when it comes to growth rates on different phosphorus resources, there are some significant phylogenetic signals. Specifically, these signals are seen in BGP, DNA, Peth, Pchol, and cAMP.
Answer 8b:The significant phylogenetic signals are suggestive of overdispersion (closely related species being less similar than expected by chance) as the K values for these phylogenetic signals are less than 1.

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.

```
#Turn continuous data to categorical
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
```

```
#Add names columns
p.growth.pa$name <- rownames(p.growth.pa)

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  AEP
##   Counts of states:  0 = 19
##                      1 = 20
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.5332703
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.017
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0128
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  PhenylCP
##   Counts of states:  0 = 33
##                      1 = 6
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.8944482
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.3676
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0129
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  DNA
##   Counts of states:  0 = 20
##                      1 = 19
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.5192799
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0135
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.014
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  cAMP
##   Counts of states:  0 = 10
##                      1 = 29
##   Phylogeny :  nj.rooted
##   Number of permutations :  10000
##
## Estimated D :  0.09766372
## Probability of E(D) resulting from no (random) phylogenetic structure :  3e-04
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.3664
```

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:The three phosphorus growth traits I choose are AEP, PhenylCP, and DNA. These three growth traits are overdisperesed as they have a positive D value.
Answer 9b:These results agree with the Blomberg's K analysis as K values for the Blomberg's analysis were less than 1, indicating the traits are overdispersed.
Answer 9c: One factor that might cause differences between these metrics is that Blomberg's K compares observed trait distributions under Brownian motion, while the Dispersion test calculates dispersion of traits under both Brownian motion and random phylogenetic structure. In addition, Blomberg's K uses continuous data while the Dispersion test uses categorical data.

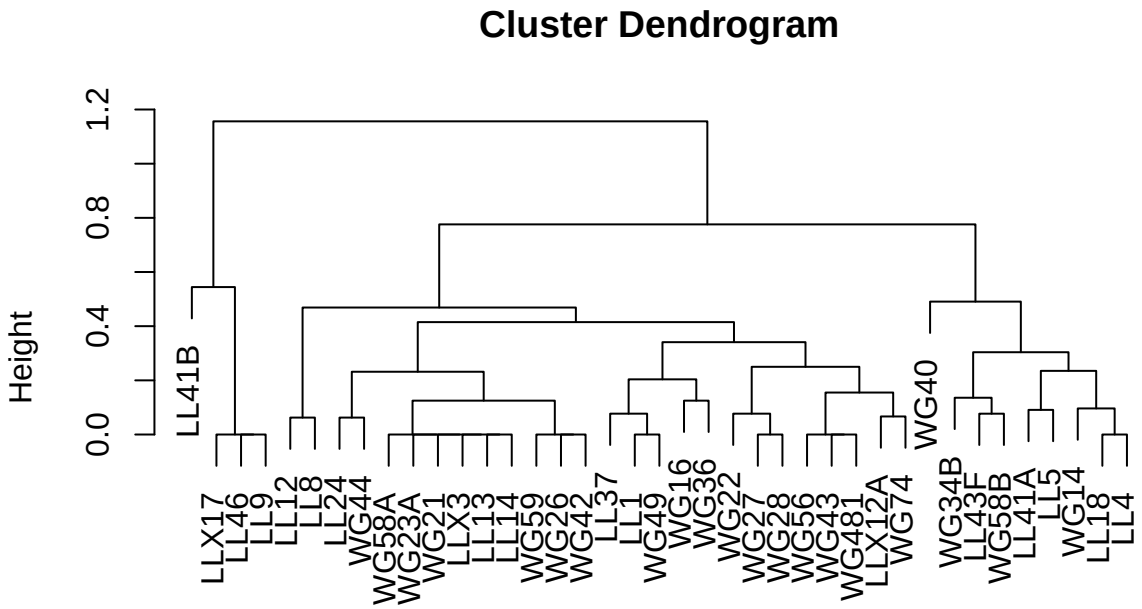
D) Correspondence between evolutionary history & ecology

1. calculate Jaccard index on resource use incidence matrix 2. create hierarchal cluster.
2. map resource use cluster onto phylogeny for each environment
3. use "RF.dist" and and "mantel" to measure the degree of correspondence between each dendogram

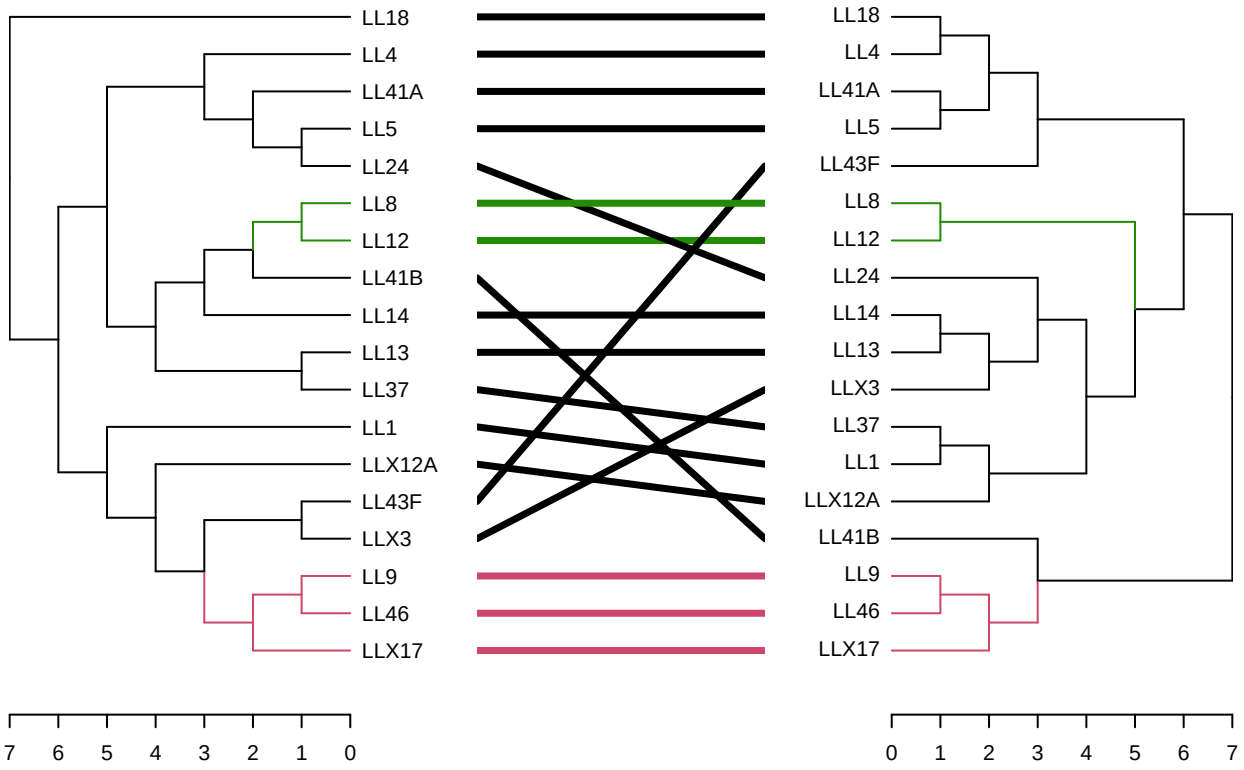
```
#Jaccard index
no <- vegdist(p.growth.pa[,1:18], method = 'jaccard', binary = TRUE)

#test clustering method that best fist data
m <- c("average", "single", "complete", "ward")
names(m) <- c("average", "single", "complete", "ward")

#compute agglomerative coefficient function
ac <- function(x) {
  agnes(no, method = x)$ac
}

#Calculate agglomerative coefficient
sapply(m, ac)
```

```
##   average    single  complete      ward
## 0.9064731 0.8881997 0.9207206 0.9470011
```

```
#Generate hierarchical cluster
no.tree <- hclust(no, method = "ward.D2")
plot(no.tree)
```

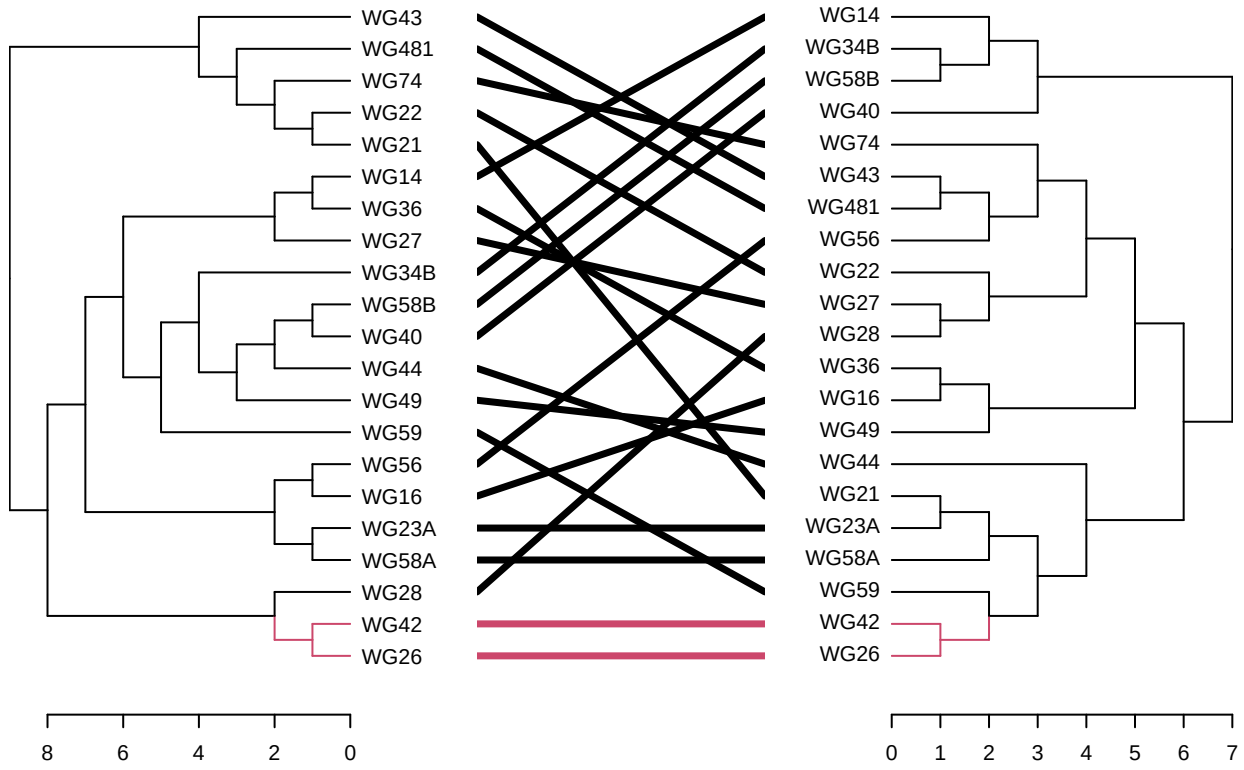



```
#Visualize diff between lakes
LL.tree <- drop.tip(nj.rooted,c(nj.rooted$tip.label[grepl("WG",
                                                          nj.rooted$tip.label)]))

LL.function <- drop.tip(as.phylo(no.tree),
                        c(no.tree$labels[grepl("WG", no.tree$labels)]))

WG.tree <- drop.tip(nj.rooted, c(nj.rooted$tip.label[grepl("LL",
                                                           nj.rooted$tip.label)]))

WG.function <- drop.tip(as.phylo(no.tree),
                        c(no.tree$labels[grepl("LL", no.tree$labels)]))

#Plot dendograms and link tips
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") %>%
  tanglegram(common_subtrees_color_branches = TRUE,
             highlight_distinct_edges = FALSE, highlight_branches_lwd = FALSE, margin_inner = 5) %>%
  entanglement()
```




```
## [1] 0.1124409
```

```
#other plot
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") %>%
  tanglegram(common_subtrees_color_branches = TRUE,
             highlight_distinct_edges = FALSE, highlight_branches_lwd = FALSE, margin_inner = 5) %>%
  entanglement()
```




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

```
#Measuer the degree of correspondance between dendograms
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 (LL.tree)
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: 0.0784
##       Significance: 0.222
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.146 0.195 0.232 0.256
## Permutation: free
## Number of permutations: 999
```

```
#Mantel test (WG.tree)
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: -0.08174
##       Significance: 0.769
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.151 0.203 0.232 0.286
## Permutation: free
## Number of permutations: 999
```

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 correction results. How does each analysis differ and shape our interpretation of correlating niche overlap with phylogeny

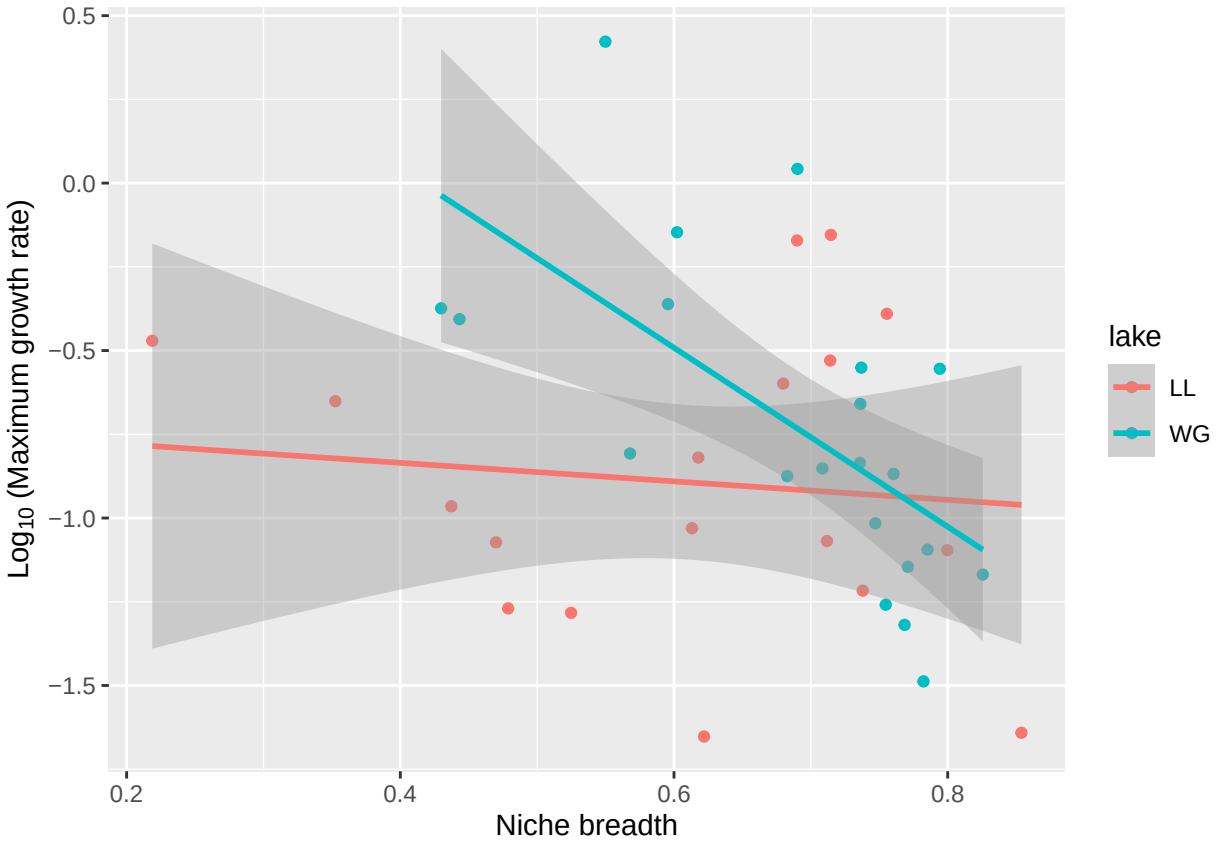
Answer9a: The two sets of tangelgrams differ in that isolates of one lake (LL) have more shared resources (niche overlap) than isolates in the other lake (WG). This can bee seen by the greater number of links there are in the LL tree.
Answer9b: The LL tree has a Robinson-Fould index (RF) of 0.8, while the WG tree has RF of 0.9. These values tell us that the WG tree is closer to complete incongruence than the LL tree. As for the Mantel values, the LL tree has a value of 0.0784 and the WG tree has a value of -0.08174, telling us that there is lower correlation in the WG tree than the LL tree. Both the Rf and Mantel values tell us that there is less correlation between niche overlap and phylogeny in the WG tree than the LL tree.

7) PHYLOGENETIC REGRESSION

In the R code chunk below, do the following:
1. Clean the resource data set 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

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

#Calculate 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'
```



```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
## -0.7557 -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
```

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

```
## [1] 48.413
```

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

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = nj.rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  41.08 -14.54
##
## Raw residuals:
##      Min       1Q   Median       3Q      Max
## -0.75804 -0.18999 -0.07425  0.32496  0.95857
##
## Mean tip height: 0.1814508
## Parameter estimate(s) using ML:
## lambda : 0.4861386
## sigma2: 0.9184409
##
## Coefficients:
```

```
##                Estimate     StdErr t.value p.value
## (Intercept) -0.8912676  0.3700360 -2.4086 0.02142 *
## NB          -0.0048049  0.5213029 -0.0092 0.99270
## lakeWG       1.4389308  0.5772311  2.4928 0.01755 *
## NB:lakeWG   -1.9663889  0.8487018 -2.3169 0.02648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1935    Adjusted R-squared: 0.1243
##
## Note: p-values and R-squared are conditional on lambda=0.4861386.
```

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

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

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

Answer 10a: We must correct for evolutionary history to see how much of the differences seen in niche breadth between the two lakes is due to the bacterial isolates' shared evolution, and how much is due to the environment (lake). ***Answer 10b***: A phylogenetic regression differs from a linear regression in that the residual errors in the linear regression are assumed to be independent, while the residual errors in the phyogenetic regression take into account the branch lengths of the phylogeny (shared evolutionary history). ***Answer 10c***: There is a greater fit (lower AIC score) for the phylogeny-corrected regression model than for the linear regression model. Accounting for shared evolutionary history improved the fit of the model to the data showing a trade-off for specialist and generalists in their maximum growth rate (generalists have higher maximum growth rate with a higher niche breadth, and specialist have a higher maximum growth rate with lower niche breadth.
Answer 10d: If two variables, such as number of offspring per birth and age at first birth, showed a similar pattern between two populations of different species within the same genus, this pattern could potentially disappear if the underlying phylogeny was included. This is because the two populations could differ in their shared evolutionary background from each other due to the populations being made up of different species within the same genus.

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?

```
#fas <- "ncbi_dataset/data/GCA_022539655.2/"
 #dna <- readDNAStringSet(fas)

#file.exists("ncbi_dataset/data/GCA_022539655.2/")

#getwd()

#choose.file("ncbi_dataset/data/GCA_022539655.2/")
```