8. Worksheet: Phylogenetic Diversity - Traits

Maddy Spencer; 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_Spencer/Week6-PhyloTraits"
```

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

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

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

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

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

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

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

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

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

```
##
## 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.
```

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: The FASTA file contains the raw sequence reads of each isolate. the AFA file is the aligned version of the FASTA file, so the presence of bases indicates alignment with other isolates. Where there is no alignment between isolates, there are dashes.

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.

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("msa")
```

```
## 'getOption("repos")' replaces Bioconductor standard repositories, see
## 'help("repositories", package = "BiocManager")' for details.
## Replacement repositories:
##     CRAN: http://rspm/default/__linux__/focal/latest
```

```
## Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.2 (2024-10-31)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'msa'
```

```
## Installation paths not writeable, unable to update packages
##   path: /opt/R/4.4.2/lib/R/library
##   packages:
##     class, cluster, foreign, KernSmooth, MASS, Matrix, nlme, nnet, rpart,
##     spatial, survival
```

```
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

##
## Attaching package: 'msa'

## The following object is masked from 'package:BiocManager':
##
##     version
```

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

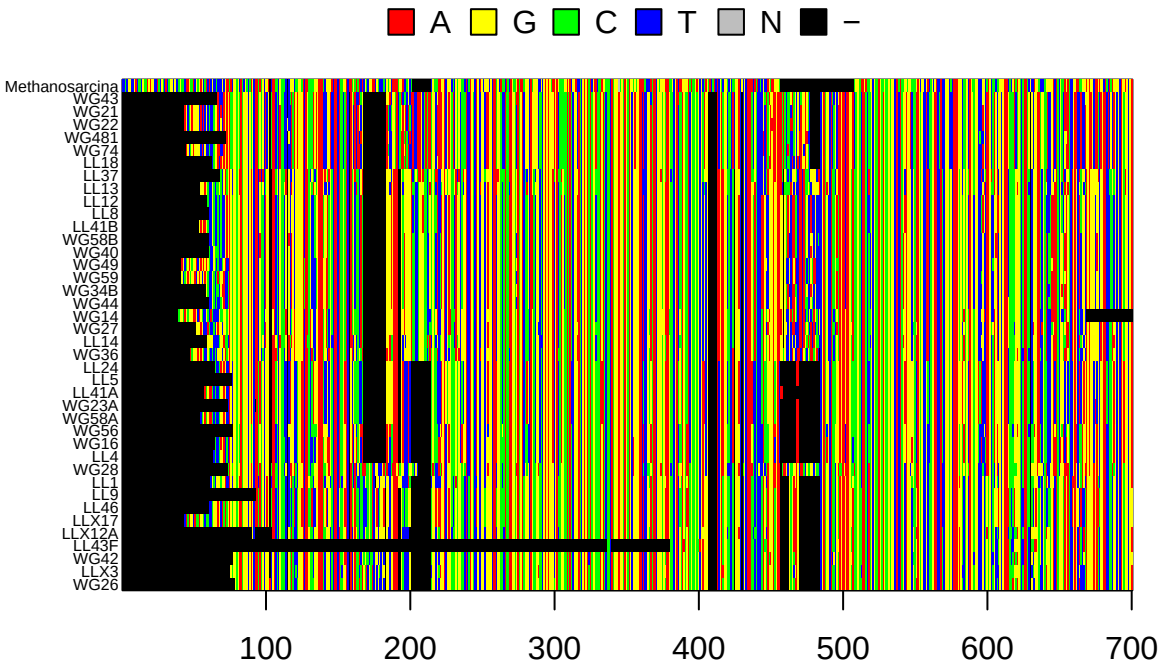
```
## DNAStringSet object of length 40:
##      width seq                                               names
##  [1]   619 ACACGTGAGCAATCTGCCCTTCT...TTCTCTGGGAATACCTGACGCT LL9
##  [2]   597 CGGCAGCGGGAAGTAGCTTGCTA...AACTGTTCAGCTAGAGTCTTGT WG14
##  [3]   794 CAGCGGCGGACGGGTGAGTAACA...GCTAACGCATTAAGCACTCCGC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

```
read.aln <- msaMuscle(seqs)
read.aln
```

```
## MUSCLE 3.8.31
##
## Call:
##    msaMuscle(seqs)
##
## MsaDNAMultipleAlignment with 40 rows and 1492 columns
##      aln                                                names
##  [1] TTCTGGTTGATCCTGCCAGAGGTTA...TCGAACCTNAATTTTGCAAGGGGGG Methanosarcina
##  [2] -------------------------...------------------------- WG43
##  [3] -------------------------...------------------------- WG21
##  [4] -------------------------...------------------------- WG22
##  [5] -------------------------...------------------------- WG481
##  [6] -------------------------...------------------------- WG74
##  [7] -------------------------...------------------------- LL18
##  [8] -------------------------...------------------------- LL37
```

```
##  [9] -------------------------...------------------------- LL13
##  ... ...
## [33] -------------------------...------------------------- LL9
## [34] -------------------------...------------------------- LL46
## [35] -------------------------...------------------------- LLX17
## [36] -------------------------...------------------------- LLX12A
## [37] -------------------------...------------------------- LL43F
## [38] -------------------------...------------------------- WG42
## [39] -------------------------...------------------------- LLX3
## [40] -------------------------...------------------------- WG26
##  Con -------------------------...------------------------- Consensus
```

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

p.DNAbin <- as.DNAbin(read.aln)
window <- p.DNAbin[, 1:700]
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: Each of the sequence reads is around 700bp, with the exception of the outgroup which is 1400bp. ***Answer 2b***: Region 100:200 and 600:700 would be good for phylogenetic inference because there seems to be a lot of variance in the sequences across isolates. The rest of the 16s rRNA is well conserved.

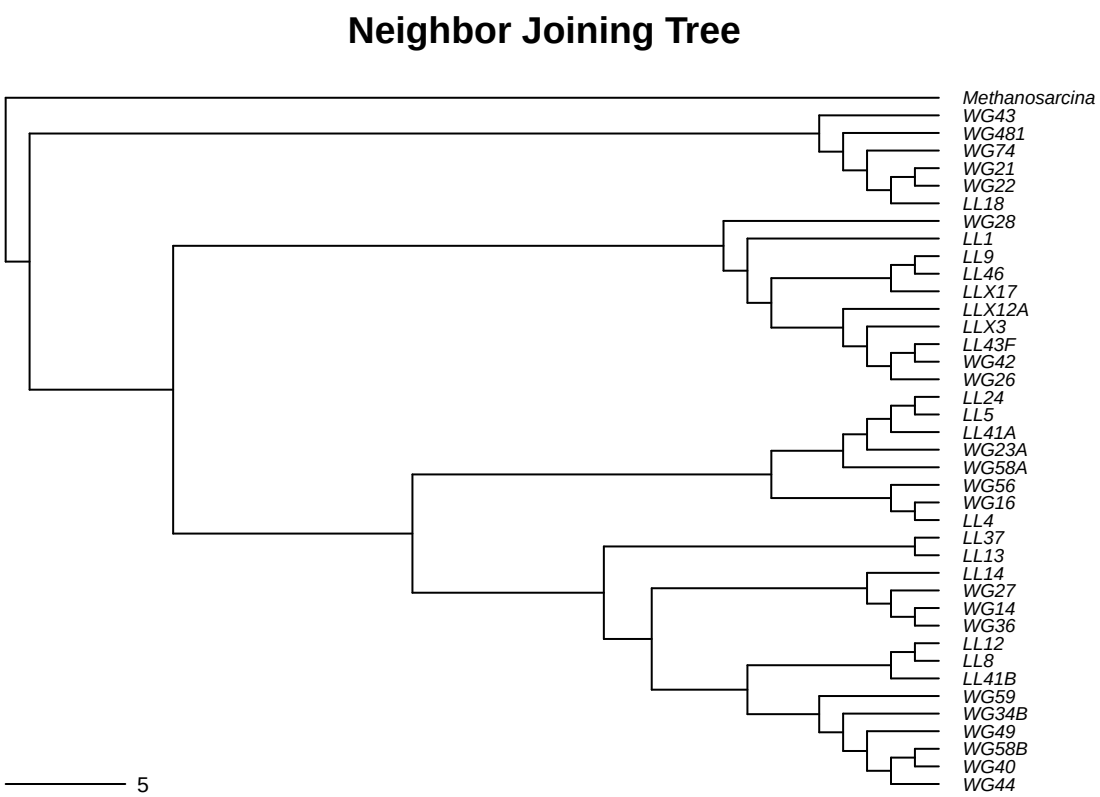
4) MAKING A PHYLOGENETIC TREE

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

A. Neighbor Joining Trees

In the R code chunk below, do the following:
1. calculate the distance matrix using `model = "raw"`,
2. create a Neighbor Joining tree based on these distances,
3. define "Methanosarcina" as the outgroup and root the tree, and
4. plot the rooted tree.

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

nj.tree <- bionj(seq.dist.raw)

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

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

par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(nj.rooted, main = "Neighbor Joining Tree", "phylogram",
           use.edge.length = FALSE, direction = "right", cex = 0.6,
           label.offset = 1)
add.scale.bar(cex = 0.7)
```



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

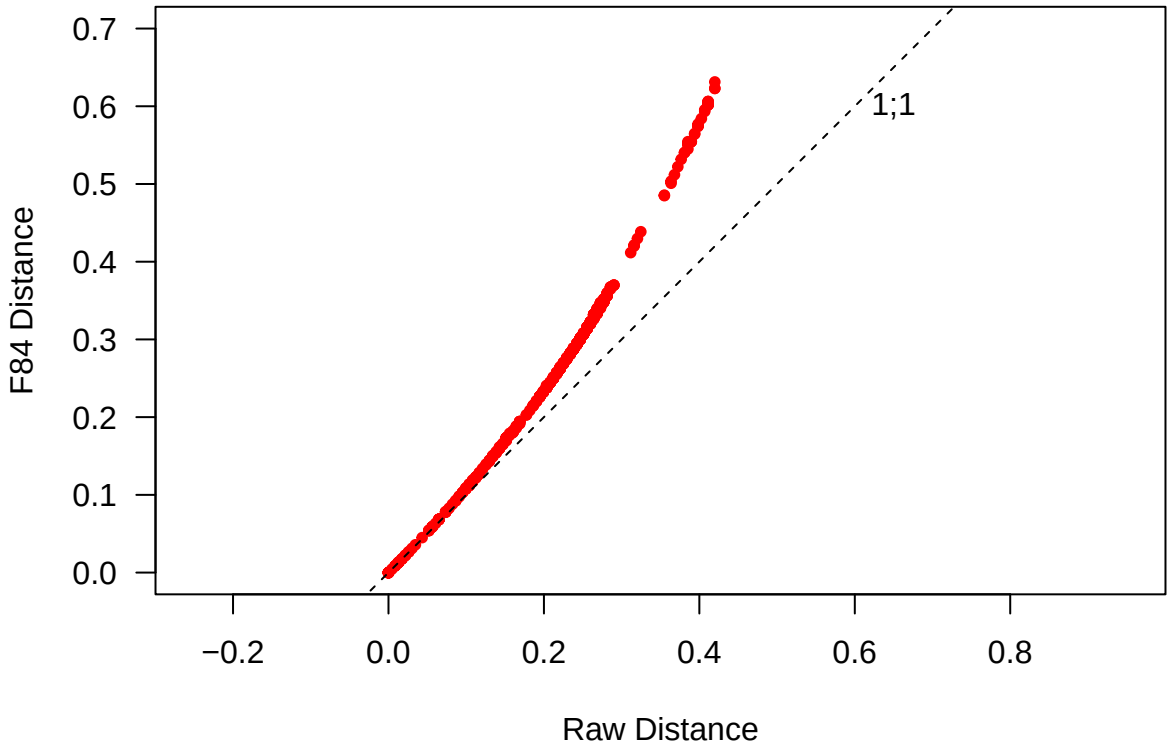
Answer 3: Making a neighbor joining tree is advantageous as it is very easy and efficient to

create, making it a good basis to start generating more sophisticated trees. Typically it is not the terminal tree output because it assumes all taxa are equally related, which we know is untrue. Neightbor joining also does not take into account the base pair states of each taxa as it is solely based on a distance matrix. This means it does not take into account DNA evolution, such as the emergence of substitutions/reversions/etc. over time.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

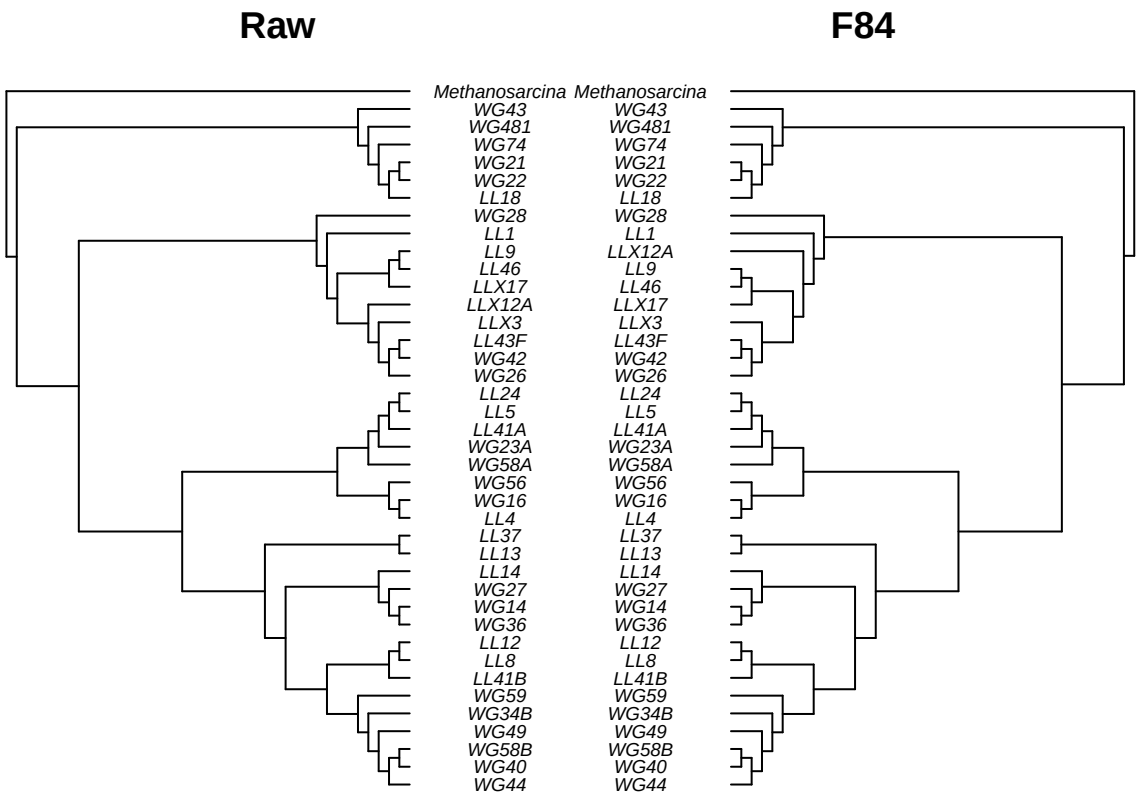
In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,
3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare the topologies of the trees.

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

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




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

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

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

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

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



C) ANALYZING A MAXIMUM LIKELIHOOD TREE

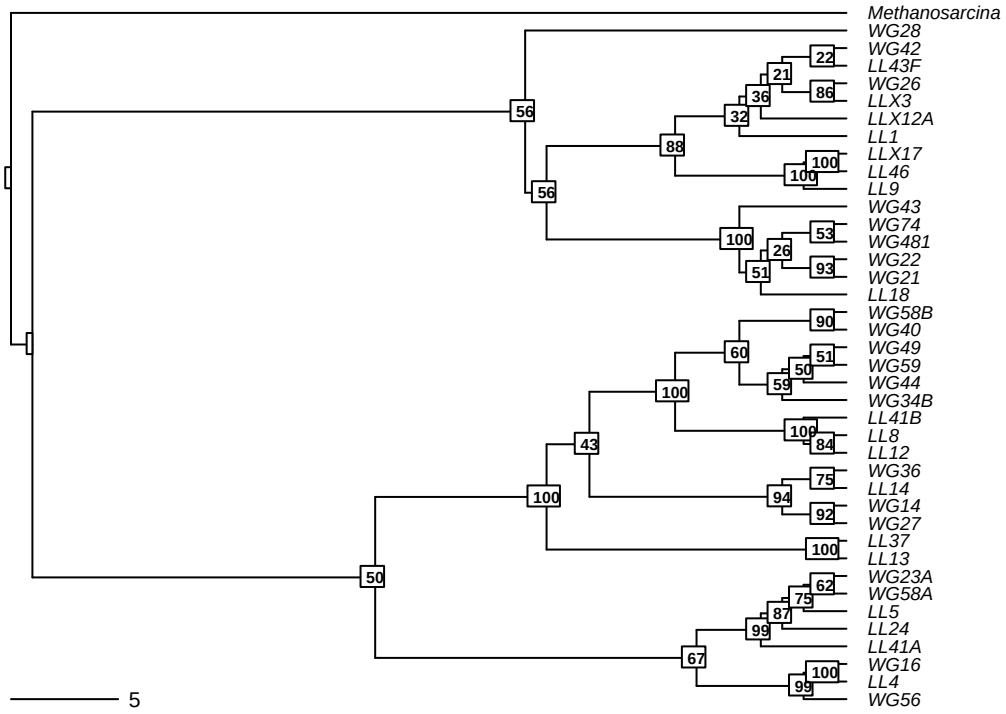
In the R code chunk below, do the following:
1. Read in the maximum likelihood phylogenetic tree used in the handout. 2. Plot bootstrap support values onto the tree

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

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

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

```
          cex = 0.5)
```




```
dist.topo(raw.rooted, ml.bootstrap, method = "score")
```

```
##           tree1
## tree2 0.4794552
```

Question 4:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

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

Answer 4a: The two trees are dinstinct as demonstrated by the BLS of 0.48, which is much larger than zero. This is unsurprising as the NJ tree is the most simple and computationally easy method. The NJ method does not consider base pair states because it is based only on a distance matrix, and it also does not using any statistical testing to improve the quality of the tree. The ML tree is more computationally intensive and does use statistical testing to determine the most likely tree, which makes it a more reliable method. ***Answer 4b***: We bootstrap our tree because we do not know the true tree and thus cannot compare to it to see the accuracy of our own. To compensate, we bootstrap our data many times over by resampling subsets of the alignments and refitting to a new tree to compare to the original one created. ***Answer 4c***: The bootstrap values will range from 0 to 100 (percentages), with larger percentages meaning the original tree is acccurate. This would be the case if that particular node is consistently seen across bootstrapped trees. ***Answer 4d***: Several of the early ancestor nodes have low bootsrapping values of 50% and

43%. Some nodes in the top right also have very low scores around 20%. ***Answer 4e***: Generally, any scores lower than 50% should not be trusted and therefore we should not trust these branches. This is because these nodes/branches change a lot between different bootstrapped trees.

5) INTEGRATING TRAITS AND PHYLOGENY

A. Loading Trait Database

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

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

p.growth.std <- p.growth / (apply(p.growth, 1, sum))
```

B. Trait Manipulations

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

```
umax <- (apply(p.growth, 1, max))

levins <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
  }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}

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

```
##       LL1      LL12      LL13      LL14      LL18      LL24      LL37       LL4
## 0.6798191 0.6899362 0.7146458 0.3525101 0.6178110 0.7117767 0.7141804 0.6131567
##     LL41A     LL41B     LL43F      LL46       LL5       LL8       LL9    LLX12A
## 0.6219701 0.2187649 0.7379376 0.4699429 0.5248238 0.7555647 0.4788159 0.8539080
##     LLX17      LLX3      WG14      WG16      WG21      WG22     WG23A      WG26
## 0.4372624 0.7996862 0.5678840 0.7358387 0.7852797 0.6827565 0.7709106 0.7823286
##      WG27      WG28     WG34B      WG36      WG40      WG42      WG43      WG44
## 0.7362067 0.7547562 0.6022315 0.7942277 0.4298220 0.8256545 0.7604551 0.7685069
##     WG481      WG49      WG56     WG58A     WG58B      WG59      WG74
## 0.7085050 0.5498899 0.7368923 0.4432747 0.5955820 0.6902266 0.7471288
```

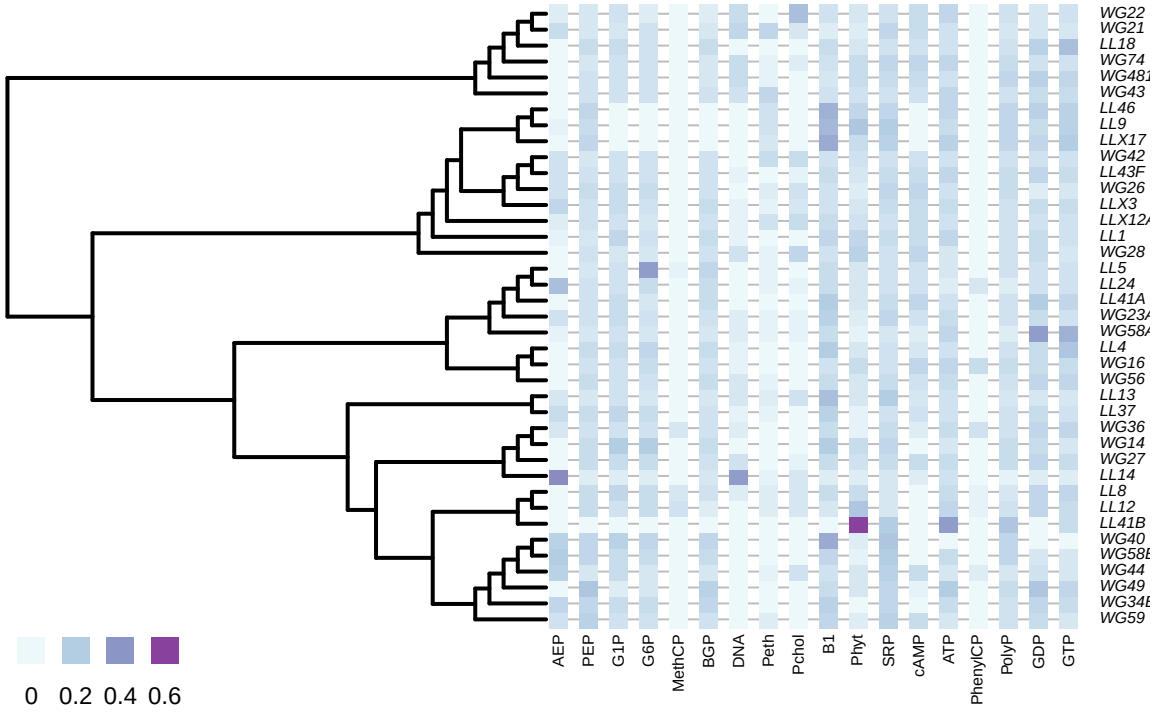
C. Visualizing Traits on Trees

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

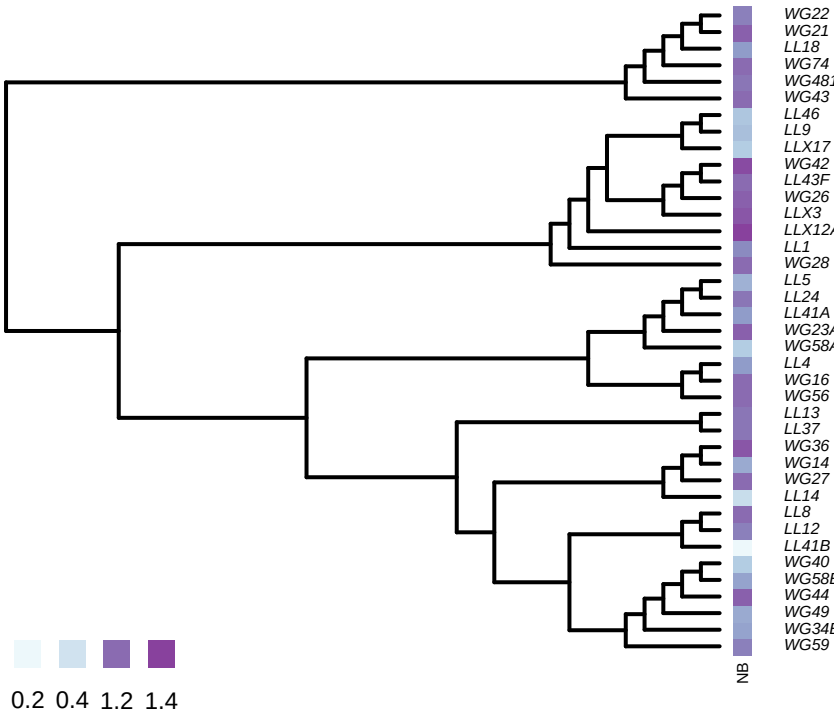
```
nj.tree <- bionj(seq.dist.F84)

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

nj.rooted<- root(nj.tree, outgroup, resolve.root = TRUE)
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

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

```
help("colorRampPalette")

mypalette <- colorRampPalette(c("#edf8fb", "#b3cde3", "#8c96c6", "#88419d"))

nj.plot <- nj.rooted
nj.plot$edge.length <- nj.plot$edge.length + 10^-1
par(mar = c(1, 1, 1, 1) + 0.1)
x <- phylo4d(nj.plot, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FALSE
              edge.color = "black", edge.width = 2, box = FALSE, col = mypalette(25), pch = 15, cex.symb
              cex.legend = 1.5, center = FALSE)
```




```
par(mar = c(1, 5, 1, 5) + 0.1)
x.nb <- phylo4d(nj.plot, nb)
table.phylo4d(x.nb, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FA
              use.edge.length = FALSE, edge.color = "black", edge.width = 2, box = FALSE, col = mypalett
              pch = 15, cex.symbol = 1.25, var.label = ("NB"), ratio.tree = 0.9, cex.legend = 1.5, cente
```



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?

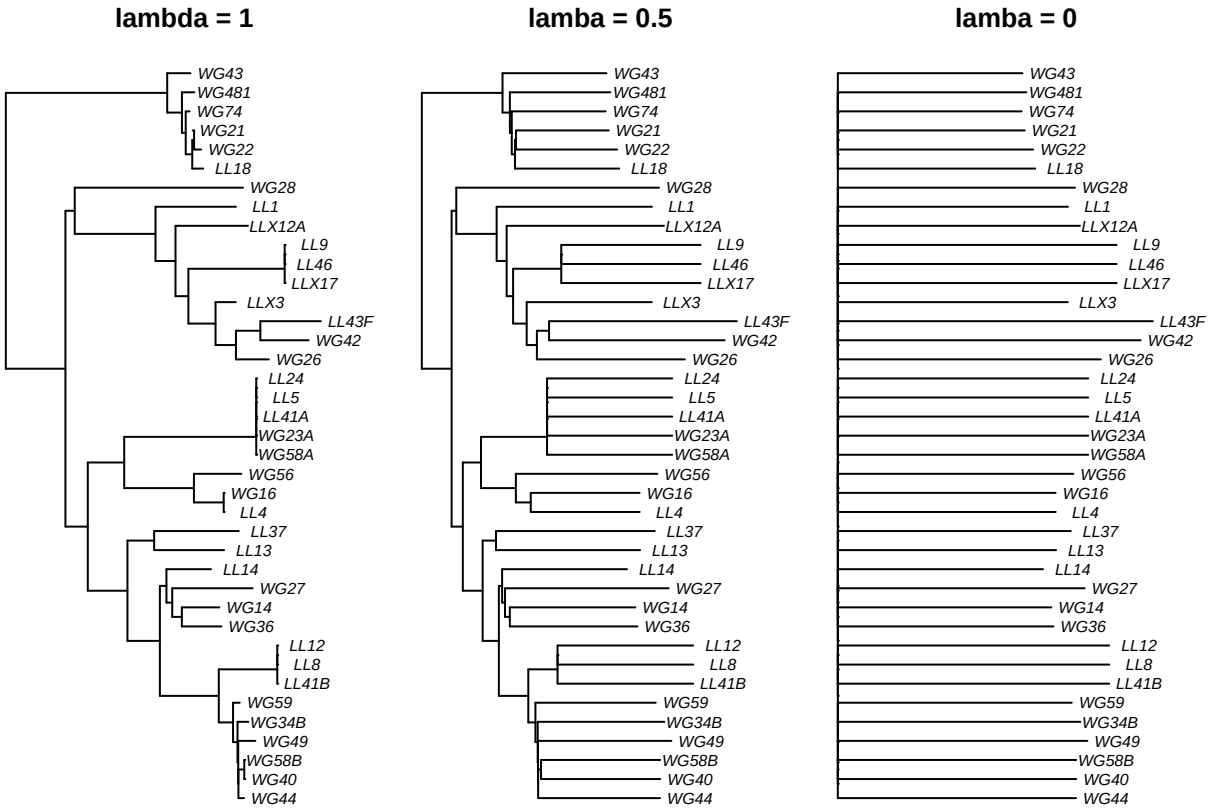
Answer 5a: I hypothesize that specialists will grow more efficiently on a given substrate than a generalist using the same subtrate as the generalist is adapted to use many different carbon sources. ***Answer 5b***: If my hypothesis were to be supported, we would see a negative correlation between niche breadth and grwoth rate. Those with a smaller niche breadth, i.e. specialists, would have a higher growth rate on their consumable substrate.

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.

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



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

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.006975
##  sigsq = 0.108060
##  z0 = 0.657697
##
##  model summary:
##  log-likelihood = 21.503414
##  AIC = -37.006827
##  AICc = -36.321113
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 47
##  number of iterations with same best fit = NA
##  frequency of best fit = NA
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.108048
##  z0 = 0.656477
##
##  model summary:
##  log-likelihood = 21.502505
##  AIC = -37.005010
##  AICc = -36.319295
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 90
##  frequency of best fit = 0.900
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

```
##
## Phylogenetic signal lambda : 0.00699105
## logL(lambda) : 21.5034
## LR(lambda=0) : 0.00181763
## P-value (based on LR test) : 0.965994
```

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?

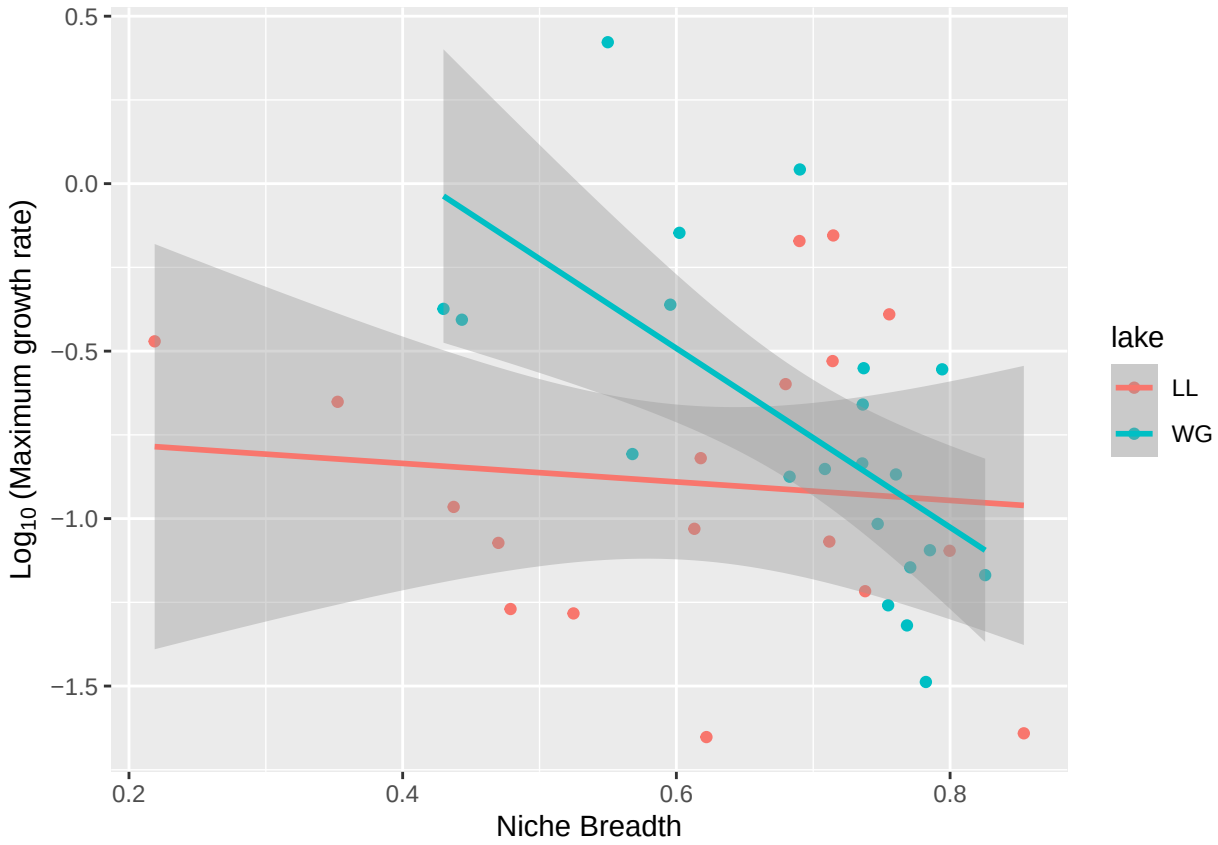
> ***Answer 6a***: Both the transformed and untransformed tree have very similar lambda values. The untransformed has a lambda of 0.006974. whereas the transformed has a value of 0. ***Answer 6b***: Both transformed and untransformed trees have AIV scores of -37. ***Answer 6c***: Both transformed and untransformed have both similar lambda and similar ACI scores, which all suggests there is not much phylogenetic signal. Additionally, the likelihood ratio test found a phylogenetic signal of only 0.00699105. The p-value between the two trees is insignificant (0.965994).

7) PHYLOGENETIC REGRESSION

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

```
nb.lake = as.data.frame(as.matrix(nb))
nb.lake$lake = rep('A')

for(i in 1:nrow(nb.lake)) {
  ifelse(grepl("WG", row.names(nb.lake)[i]), nb.lake[i, 2] <- "WG",
         nb.lake[i, 2] <- "LL")
}

colnames(nb.lake)[1] <- "NB"

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

ggplot(data = nb.lake, aes(x = NB, y = log10(umax), color = lake)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlab("Niche Breadth") +
  ylab(expression(Log[10]~"(Maximum growth rate)"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
fit.lm <- lm(log10(umax) ~ NB * lake, data = nb.lake)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = log10(umax) ~ NB * lake, data = nb.lake)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7557 -0.3108 -0.1077  0.3102  0.7800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7247     0.3852  -1.882   0.0682 .
## NB           -0.2763     0.6097  -0.453   0.6533
## lakeWG        1.8364     0.6909   2.658   0.0118 *
## NB:lakeWG    -2.3958     1.0234  -2.341   0.0251 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.418 on 35 degrees of freedom
## Multiple R-squared:  0.2595, Adjusted R-squared:  0.196
## F-statistic: 4.089 on 3 and 35 DF,  p-value: 0.01371
```

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

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

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

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nb.lake, phy = nj.rooted,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  41.08 -14.54
##
## Raw residuals:
##      Min       1Q   Median       3Q      Max
## -0.75804 -0.18999 -0.07425  0.32496  0.95857
##
## Mean tip height: 0.1814501
## Parameter estimate(s) using ML:
## lambda : 0.4861372
## sigma2: 0.9184437
##
## Coefficients:
##              Estimate    StdErr t.value p.value
## (Intercept) -0.891268  0.370036 -2.4086 0.02142 *
## NB          -0.004805  0.521303 -0.0092 0.99270
## lakeWG       1.438930  0.577231  2.4928 0.01755 *
## NB:lakeWG   -1.966388  0.848702 -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=0.4861372.
```

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: We needed to correct for shared evolutionary history because a simple regression assumes that all samples are independent of one another, but this is not the case as some may be more closely realted than others and vice-versa. Because evolutionary history can impact our results, its important to account for. ***Answer 7b***: A phylogenetic regression differs from a standard linear regression in that it does not require the input data to be independent nor normally distributed. A phylogenetic regression take into account the phylogenetic branch lengths, thus allowing proper intepretation of correlation between two variables. ***Answer 7c***: Niche breadth is not correlated with maximum growth rate in either model, as indicated by the insigificant p-value. The maximum growth rate is positively influenced by the lake the taxa reside in, as indicated by the significant p-value and positive estimate value. Lastly, the interaction between niche breadth and lake does influence the maximal growth rate negatively. In other words, the effect of location on maximal grwoth rate changes depending on the value of NB. In the standard regression model, our r-squared is 0.196, and in the phylogenetic regression, it is 0.1243. The AIC of the phylogenetic regression is lower, indicating it is a better fit. The lambda of the phylogenetic regression is 0.48, which indicates a moderately high level of phylogenetic signal. ***Answer 7d***: There could be a scenario in which the ability of an organism to consume a substrate is reliant entirely on the underlying phylogeny of species. In other words, the only deciding factor in consumption is whether or not the species retains an active copy of the enzyme that degrades it. More distantly related species have lower affinity for the substrate due to accumulations of mutations over evolutionary history, which can be seen in the phylogenetic trees. I can imagine taking into account phylogeny here would erase any effect that a secondary variable like environmental toxicity would have on substrate consumption.

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?

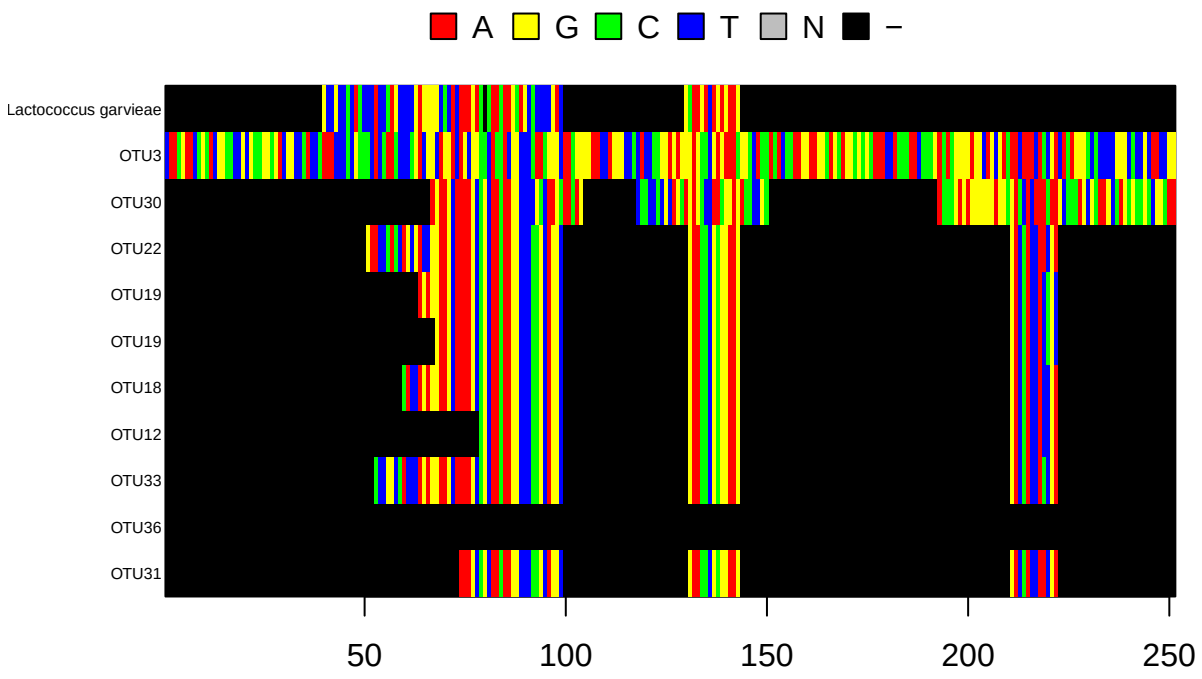
```
funseqs <- readDNAStringSet("./data/Fungal_seq.txt", format = 'fasta')
funseqs
```

```
## DNAStringSet object of length 11:
##      width seq                                               names
##  [1]   682 AAAGTCGTAACAAGGTTTCCGTA...GGACTACCCGCTGAACTTAAGC OTU31
##  [2]   663 CTTGGTCATTTAGAGGAAGTAAA...GTTCGATCTCAAATCAGGTAGG OTU33
##  [3]  3380 CTAAGTATAAGCAATCTATACGG...AACTTCTAAGGTTGACCTCGGA OTU3
##  [4]   712 CATTAGAGGAAGTAAAAGTCGTA...TAAGCATATCAATAAGCGGGAG OTU18
##  [5]   730 AGAGGAAGTAAAAGTCGTAACAA...AGACTACCCGCTGAACTTAAGC OTU19
##  ...   ... ...
##  [7]   630 CGTAACAAGGTTTCCGTAGGTGA...GACTTGGACAGGTTTTCATTAA OTU12
##  [8]   694 GAAGTAAAAGTCGTAACAAGGTT...ACACCCTCGAGCCCACGAACCC OTU19
##  [9]  1482 GAATTCACTAGTGATTGGAAGTA...AGGAAATCGAATTCCCGCGGCC OTU22
## [10]   896 AGAAGTAAAAGTCGTAACAAGGT...TACCCGCTGAACTTAAGCATAT OTU30
## [11]   506 GTTGTTCTACTTTATTCAGTTTT...GAATACATAGCTTACGCGAAGG Lactococcus garvieae
```

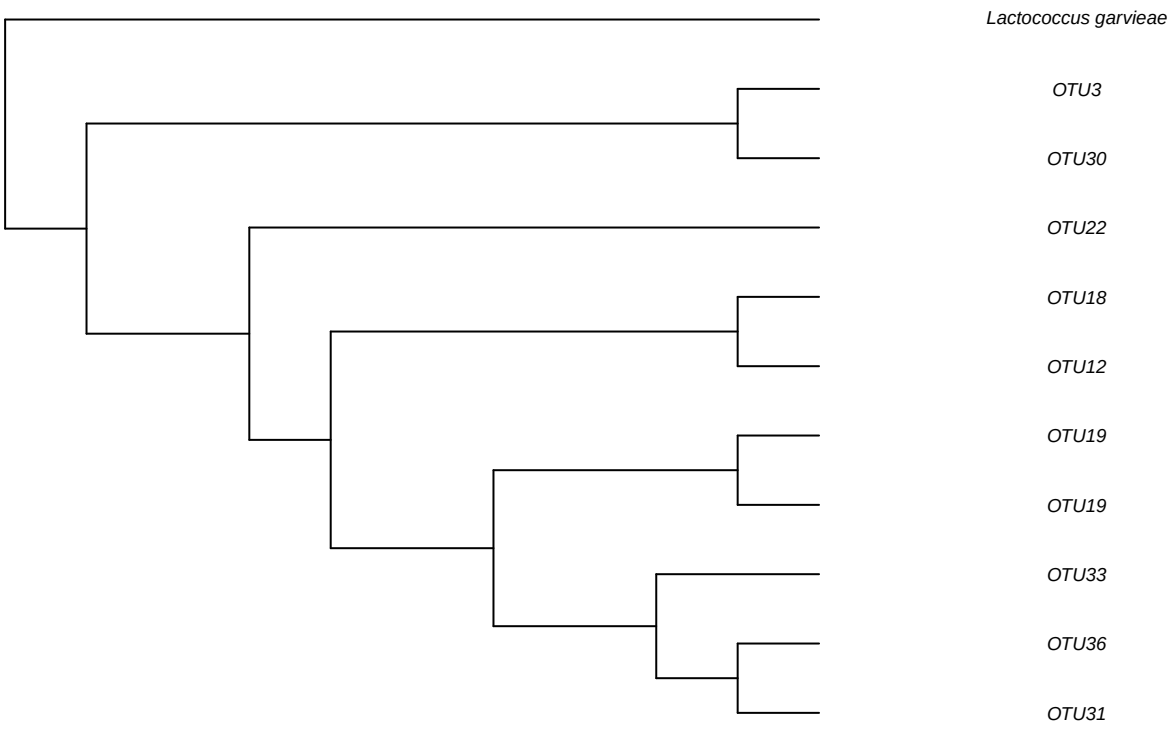
```
funread.aln <- msaMuscle(funseqs)
funread.aln
```

```
## MUSCLE 3.8.31
##
## Call:
##    msaMuscle(funseqs)
##
## MsaDNAMultipleAlignment with 11 rows and 3927 columns
##      aln                                                   names
##  [1] --------------------------...-------------------------- Lactococcus garvieae
##  [2] CTAAGTATAAGCAATCTATACGGTG...-------------------------- OTU3
##  [3] --------------------------...-------------------------- OTU30
##  [4] --------------------------...CGGAGGAAATCGAATTCCCGCGGCC OTU22
##  [5] --------------------------...-------------------------- OTU19
##  [6] --------------------------...-------------------------- OTU19
##  [7] --------------------------...CGGGAG------------------- OTU18
##  [8] --------------------------...-------------------------- OTU12
##  [9] --------------------------...-------------------------- OTU33
## [10] --------------------------...-------------------------- OTU36
## [11] --------------------------...-------------------------- OTU31
##  Con --------------------------...-------------------------- Consensus
```

```
funp.DNAbin <- as.DNAbin(funread.aln)
funwindow <- funp.DNAbin[, 200:450]
image.DNAbin(funwindow, cex.lab = 0.50)
```




```
seq.dist.fun <- dist.dna(funp.DNAbin, model = "F84", pairwise.deletion = FALSE)

fun.tree <- bionj(seq.dist.fun)

fun.outgroup <- match("Lactococcus garvieae", fun.tree$tip.label)

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

par(mar = c(1, 0, 2, 1))
plot.phylo(fun.rooted, type = "phylogram", direction = "right",
           show.tip.label = TRUE, use.edge.length = FALSE, adj = 0.5,
           cex = 0.6, label.offset = 2, main = "Fungal Phylogeny")
```



The original group that had performed this had the accession numbers for each OTU recorded, so we were easily able to find these. internal transcribers spacers (ITS) are very common in determining fungal phylogeny, so we decided to stick with these. The tree does seem to align with what is known about these organisms. Our outgroup, a bacteria *Lactococcus garviaea* is separate from everything else as expected. Each cluster typically represents a family or order. I accidentally included OTU19 twice in the FASTA file, but it serves as a nice sanity check that the tree was calibrated properly. Since all the OTUs are publically available and we only have 44 in total, for the final project I would like to make a full phylogenetic tree. It may also be interesting to perform a phylogenetic regression for each treatment using some of the other varaibles the researchers measured (enzyme activity, etc.).

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)**.