11. Worksheet: Phylogenetic Diversity - Traits

Mackenzie Caple; Z620: Quantitative Biodiversity, Indiana University

19 February, 2019

OVERVIEW

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

After completing this exercise you will be able to:

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

Directions:

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

The completed exercise is due on **Wednesday, February 20th, 2019 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 "*/11.PhyloTraits*" folder, and
4. load all of the required R packages (be sure to install if needed).

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

```
## [1] "/Users/mcaple/GitHub/QB2019_Caple/2.Worksheets/11.PhyloTraits"
```

```
setwd("~/GitHub/QB2019_Caple/2.Worksheets/11.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
```

```
## Warning: package 'phylobase' was built under R version 3.5.2
```

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

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

```
## Warning: package 'geiger' was built under R version 3.5.2
```

```
## Warning: package 'vegan' was built under R version 3.5.2
```

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

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

```
## This is vegan 2.5-4
```

```
##
## 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 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
```

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.

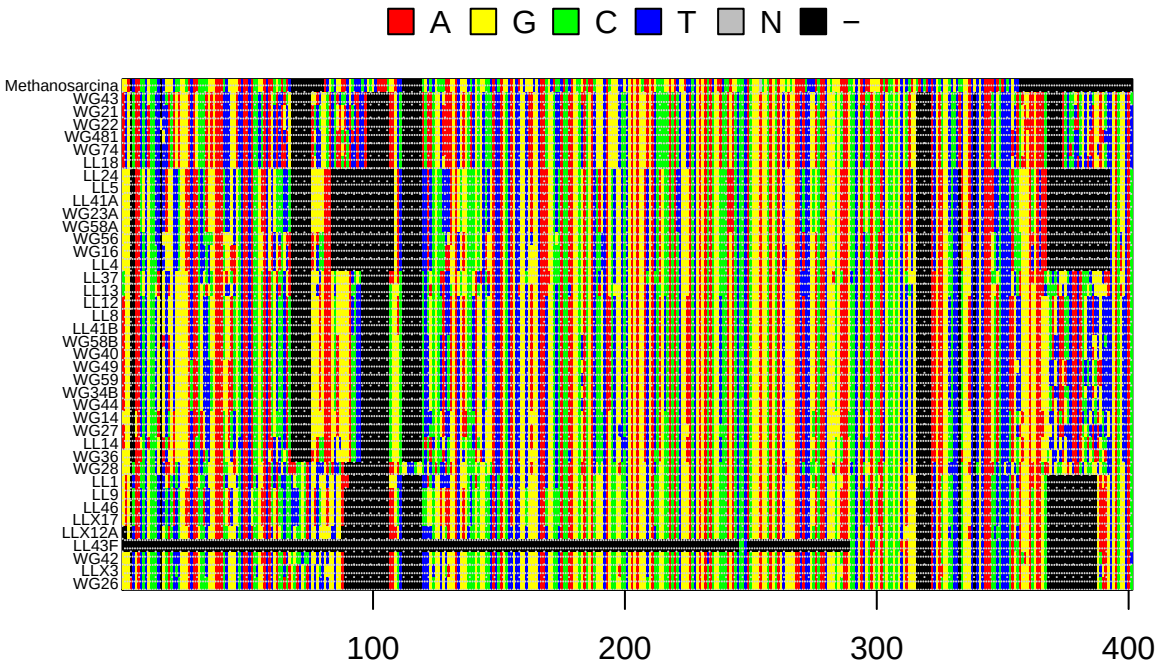
3) SEQUENCE ALIGNMENT

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 FASTA file, each species' sequence is a different length and has no gaps. In the .afa file, the sequences are now all the same length, with missing pieces filled in with dashes. Additionally, some sequences now have Ns to represent ambiguous nucleotides.

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

```
# read alignment file {seqinr}
read.aln <- read.alignment(file = "./data/p.isolates.afa", format = "fasta")

# convert alignment file to DNAbin object {ape}
p.DNAbin <- as.DNAbin(read.aln)

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

# command to visualize sequence alignment {ape}
image.DNAbin(window, cex.lab = 0.50)

# optional code to add grid to help visualize rows
grid(ncol(window), nrow(window), col = "lightgrey")
```



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: most sequences are between 800 and 1000 base pairs (outgroup reference sequence is 1500 bp) ***Answer 2b***: the regions from approximately 400-450 and 600-700; these regions would be good for phylogenetic analysis because they are complete (contain sequences from all isolates) yet (visually) seem to contain a large degree of diversification

4) MAKING A PHYLOGENETIC TREE

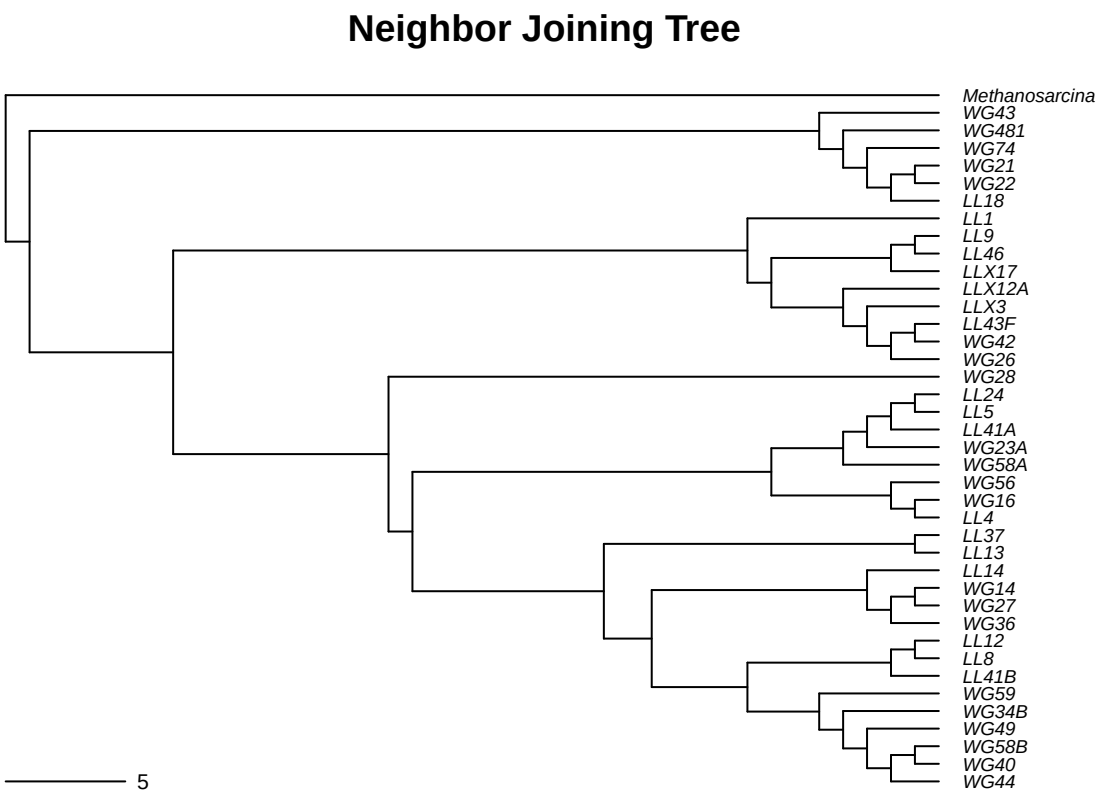
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

A. Neighbor Joining Trees

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

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

# neighbor joining algorithm to construct the tree, a 'phylo' object {ape}
nj.tree <- bionj(seq.dist.raw)
```

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

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

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



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

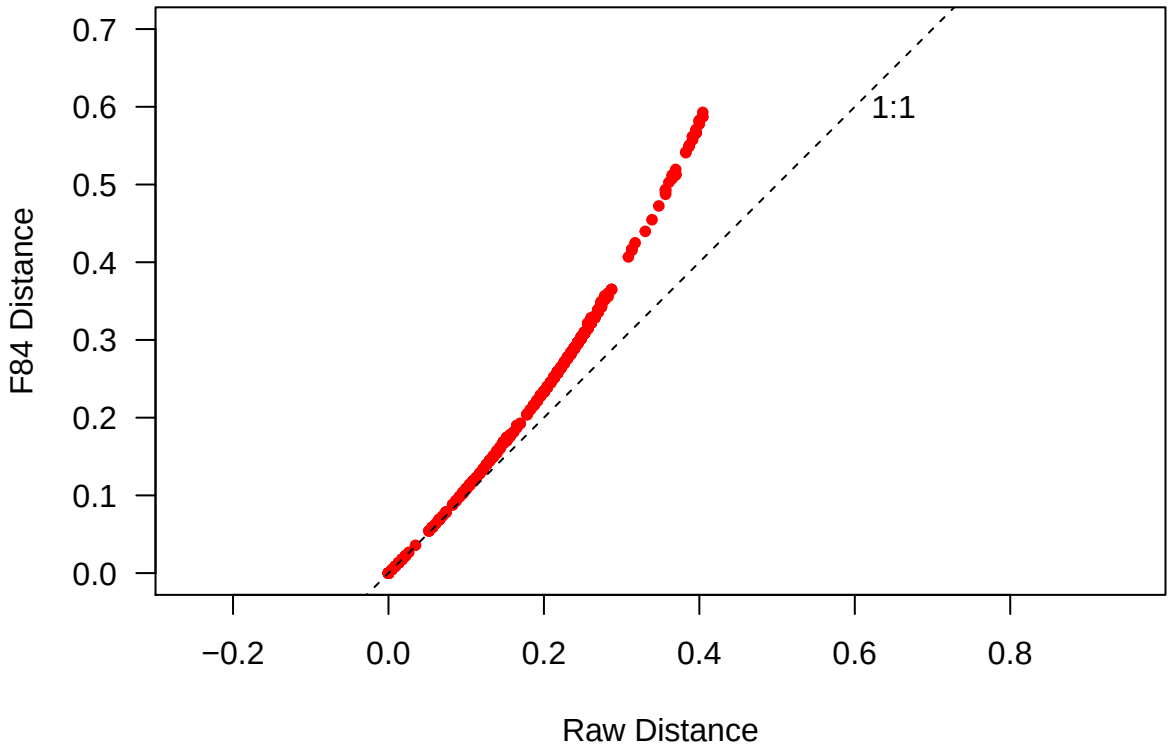
> ***Answer 3***: The advantages are that it is very fast to run and that it will give you a decent idea of relationships. The disadvantage is that it is not a very sophisticated algorithm (e.g. assumes single substitutions, does not account for substitution biases), so it may not be as accurate as methods such as Maximum Likelihood or Bayesian.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

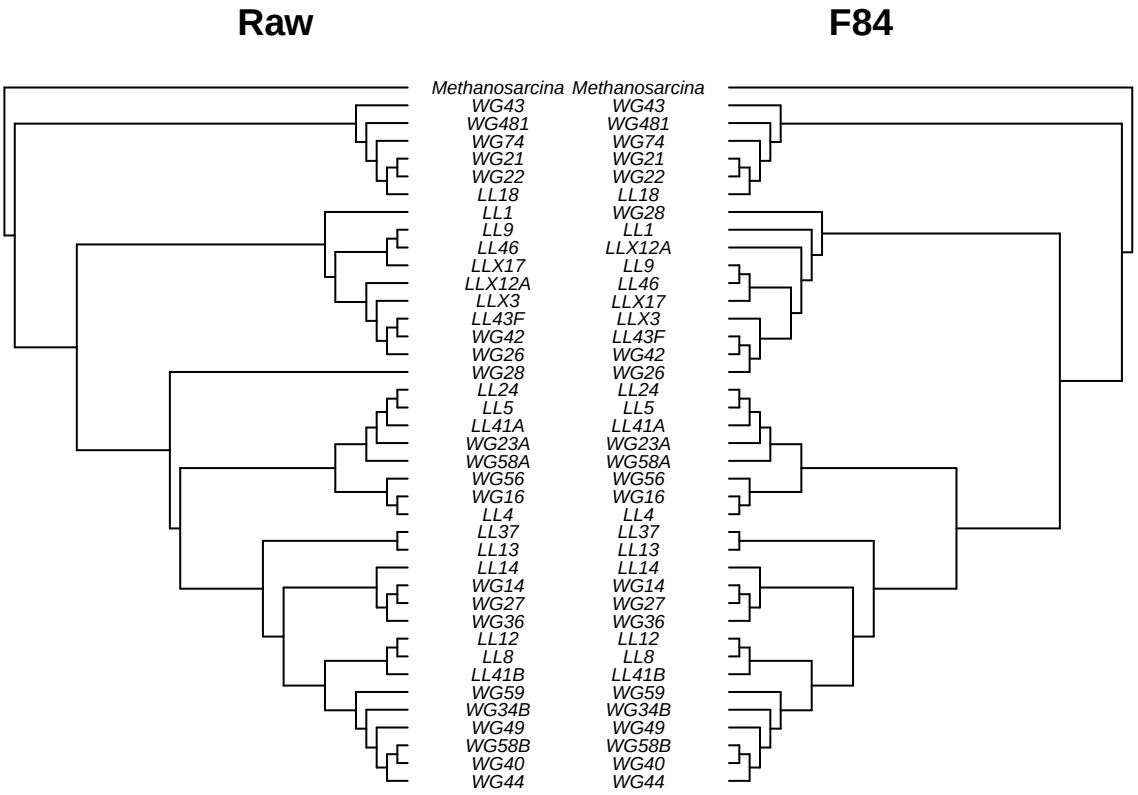
In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,
3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare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

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

```r
# use a saturation plot to 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")
```




```r
# make neighbor joining trees using different DNA substitution models {ape}
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)

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

# root the trees {ape}
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)

# make cophylogenetic plot {ape}
layout(matrix(c(1, 2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(raw.rooted, type = "phylogram", direction = "right", show.tip.label = TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "Raw")
par(mar = c(1, 0, 2, 1))
plot.phylo(F84.rooted, type = "phylogram", direction = "left", show.tip.label = TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "F84")
```

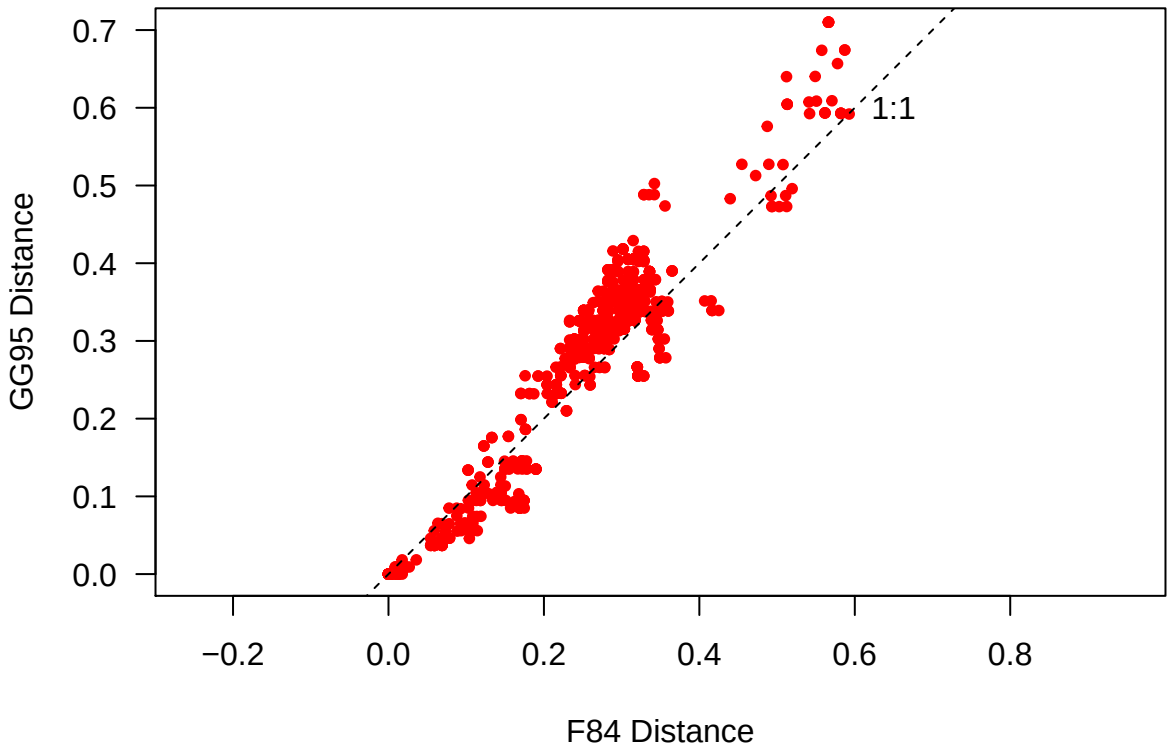



```
## quantitatively compare trees
# set method = "PH85" for the symmetric difference
# set method = "score" for the symmetric difference
# this function automatically checks for a root and unroots rooted trees
dist.topo(raw.rooted, F84.rooted, method = "score")
```

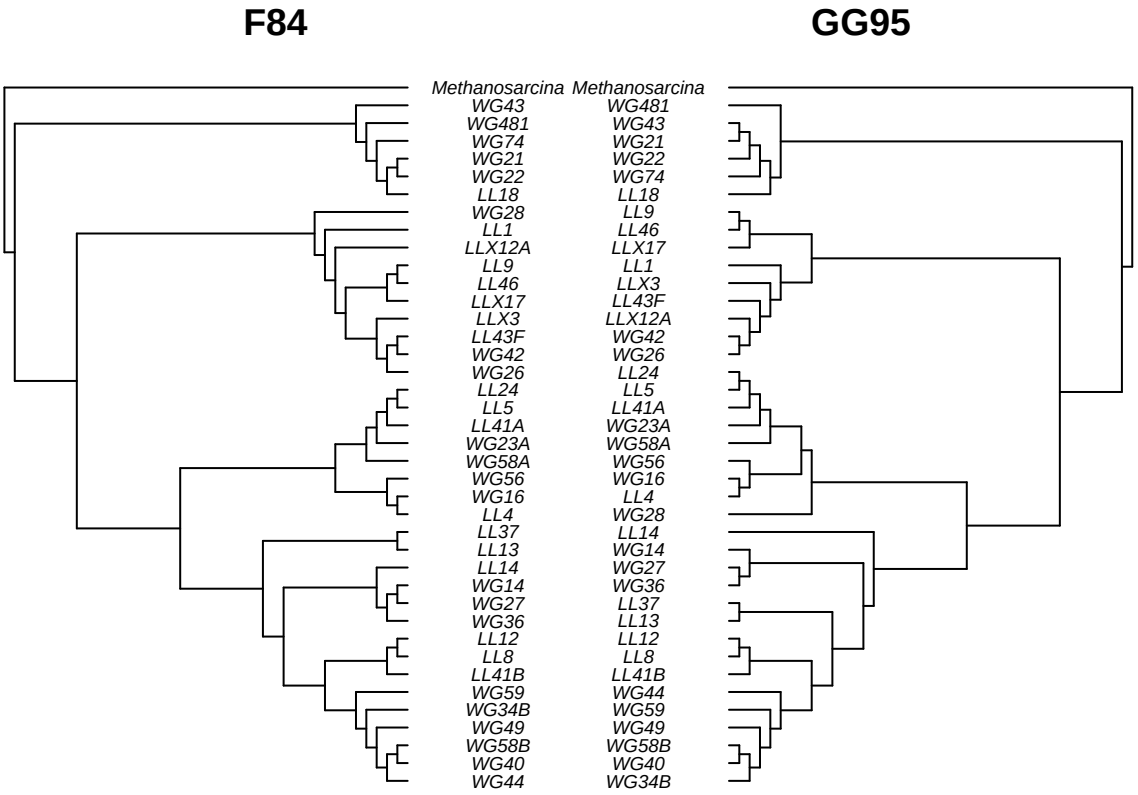
```
##            tree1
## tree2 0.04387426
```

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 with "F84" model {ape}
seq.dist.GG95 <- dist.dna(p.DNAbin, model = "GG95", pairwise.deletion = FALSE)

# use a saturation plot to plot distances from different DNA substitution models
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.F84, seq.dist.GG95,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7), ylim = c(0, 0.7),
     xlab = "F84 Distance", ylab = "GG95 Distance")
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1")
```




```
# make neighbor joining trees using different DNA substitution models {ape}
F84.tree <- bionj(seq.dist.F84)
GG95.tree <- bionj(seq.dist.GG95)

# define outgroups
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)
GG95.outgroup <- match("Methanosarcina", GG95.tree$tip.label)

# root the trees {ape}
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)
GG95.rooted <- root(GG95.tree, GG95.outgroup, resolve.root = TRUE)

# make cophylogenetic plot {ape}
layout(matrix(c(1, 2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(F84.rooted, type = "phylogram", direction = "right", show.tip.label = TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "F84")
par(mar = c(1, 0, 2, 1))
plot.phylo(GG95.rooted, type = "phylogram", direction = "left", show.tip.label = TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "GG95")
```



Question 4:

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

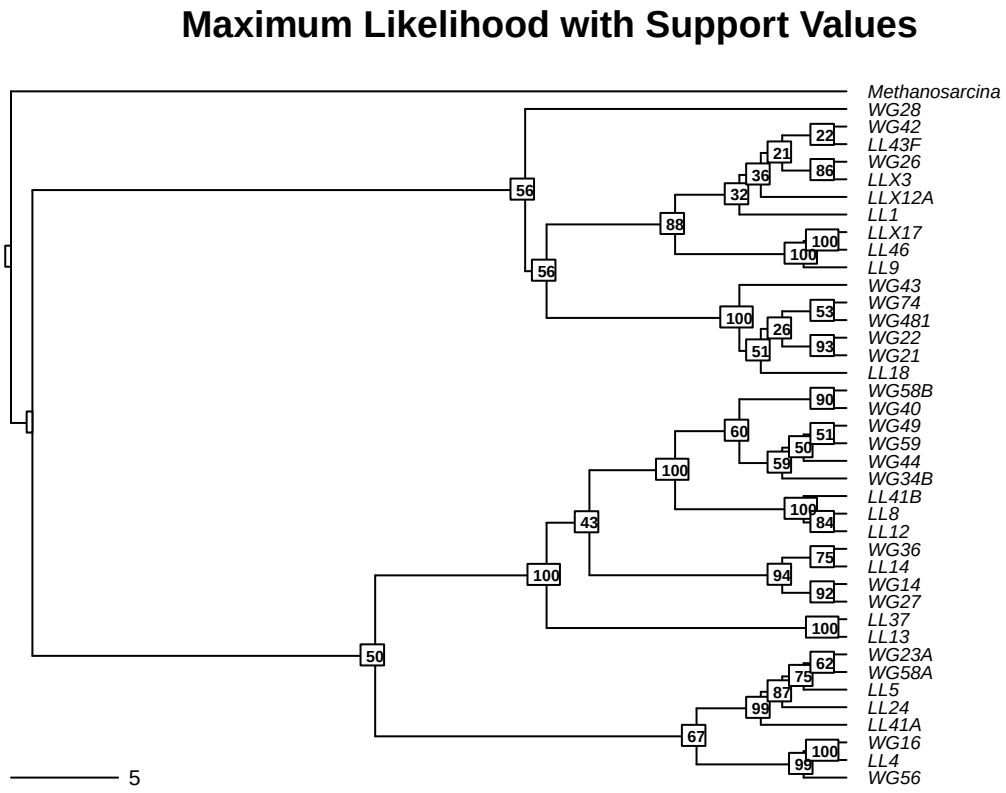
Answer 4a: I chose the GG95 model; this model accounts for the fact that G+C content may vary over time. It also accounts for differences in the rates of transitions and transversions. In contrast F84, although it does account for differences in transition and transversion rates and allows variation in base frequencies, assumes these rates to be constant through time. ***Answer 4b***: The saturation plot shows that, although neither model has a dramatic systematic difference from the other, the GG95 model introduces more variation into the difference in distances. ***Answer 4c***: The two trees have the same gross structure, but some phyla vary at the more fine levels. This is an indication that assumptions about substitution rates can have important effects on structuring phylogenetic trees.

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

```
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 Suppor
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 structure of these two trees appears quite different. This is because the neighbor-joining trees group taxa based on how close or distant the sequences are from each other; however, simple distance matrices are not always correct, so methods that take into account the probability of different trees arising can often get closer to the true tree. ***Answer 5b***: Bootstrapping gives us a measure of confidence for each node of the tree by resampling the data multiple times. ***Answer 5c***: Higher bootstrap values represent a higher chance that the node is accurate, i.e. more confidence in that node. ***Answer 5d***: The clade containing WG42, LL43F, WG26, LLX3, LLX12A, and LL1 has very low support. ***Answer 5e***: No, I do not trust the placements of these nodes

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 and column names to niche breadth matrix
rownames(nb) <- row.names(p.growth)
colnames(nb) <- c("NB")

nb
```

```
##               NB
## LL1    0.6798191
## LL12   0.6899362
## LL13   0.7146458
## LL14   0.3525101
## LL18   0.6178110
## LL24   0.7117767
## LL37   0.7141804
## LL4    0.6131567
## LL41A  0.6219701
## LL41B  0.2187649
```

```
## LL43F  0.7379376
## LL46   0.4699429
## LL5    0.5248238
## LL8    0.7555647
## LL9    0.4788159
## LLX12A 0.8539080
## LLX17  0.4372624
## LLX3   0.7996862
## WG14   0.5678840
## WG16   0.7358387
## WG21   0.7852797
## WG22   0.6827565
## WG23A  0.7709106
## WG26   0.7823286
## WG27   0.7362067
## WG28   0.7547562
## WG34B  0.6022315
## WG36   0.7942277
## WG40   0.4298220
## WG42   0.8256545
## WG43   0.7604551
## WG44   0.7685069
## WG481  0.7085050
## WG49   0.5498899
## WG56   0.7368923
## WG58A  0.4432747
## WG58B  0.5955820
## WG59   0.6902266
## WG74   0.7471288
```

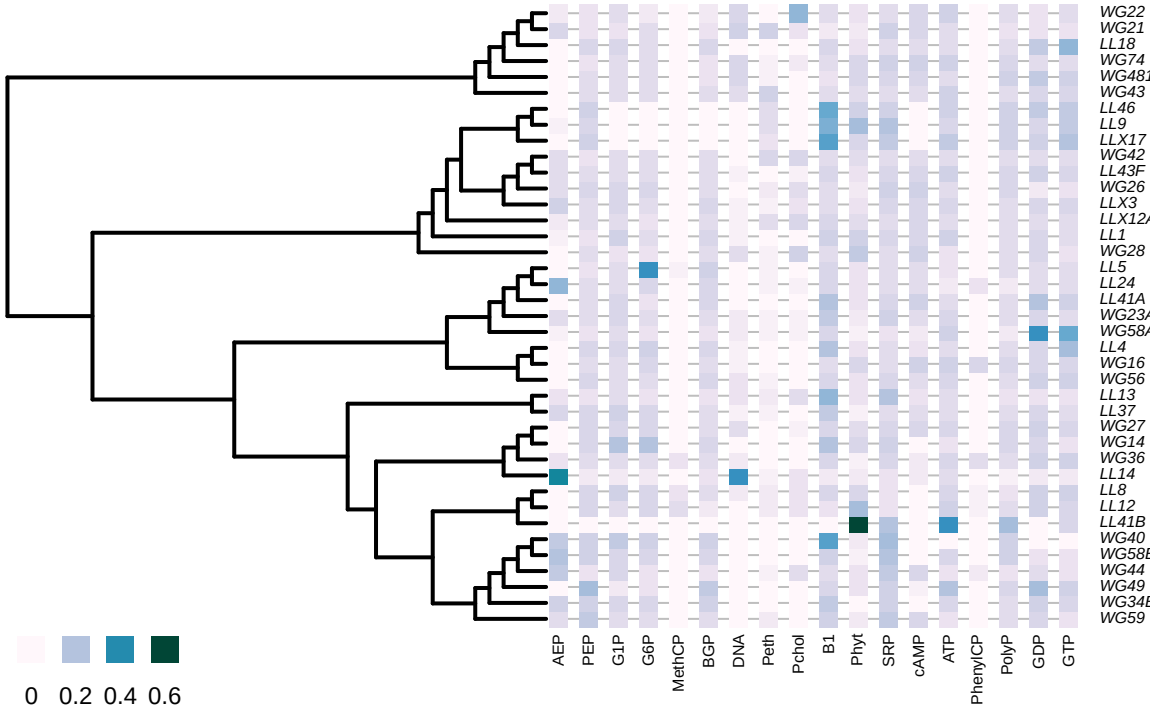
C. Visualizing Traits on Trees

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.

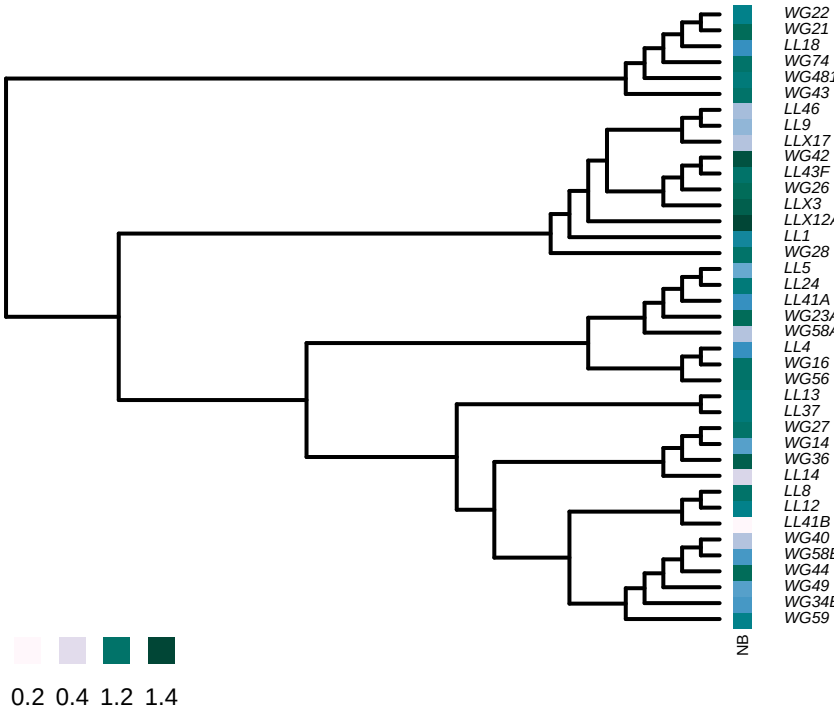
```
# generate neighbor joining tree using F84 DNA substitution model {ape}
nj.tree <- bionj(seq.dist.F84)

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

# create a rooted tree {ape}
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

# keep rooted but drop outgroup branch
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

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

```
# define color palette
mypalette <- colorRampPalette(brewer.pal(9, "PuBuGn"))

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




```
# niche breadth
par(mar = c(1, 5, 1, 5) + 0.1)
x.nb <- phylo4d(nj.rooted, nb)
table.phylo4d(x.nb, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = 0.5, scale = FALSE, use.edge.length = FALSE,
              edge.color = "black", edge.width = 2, box = FALSE,
              col = mypalette(25), pch = 15, cex.symbol = 1.25, var.label = ("       NB"),
              ratio.tree = 0.90, cex.legend = 1.5, center = FALSE)
```



Question 6:

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

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

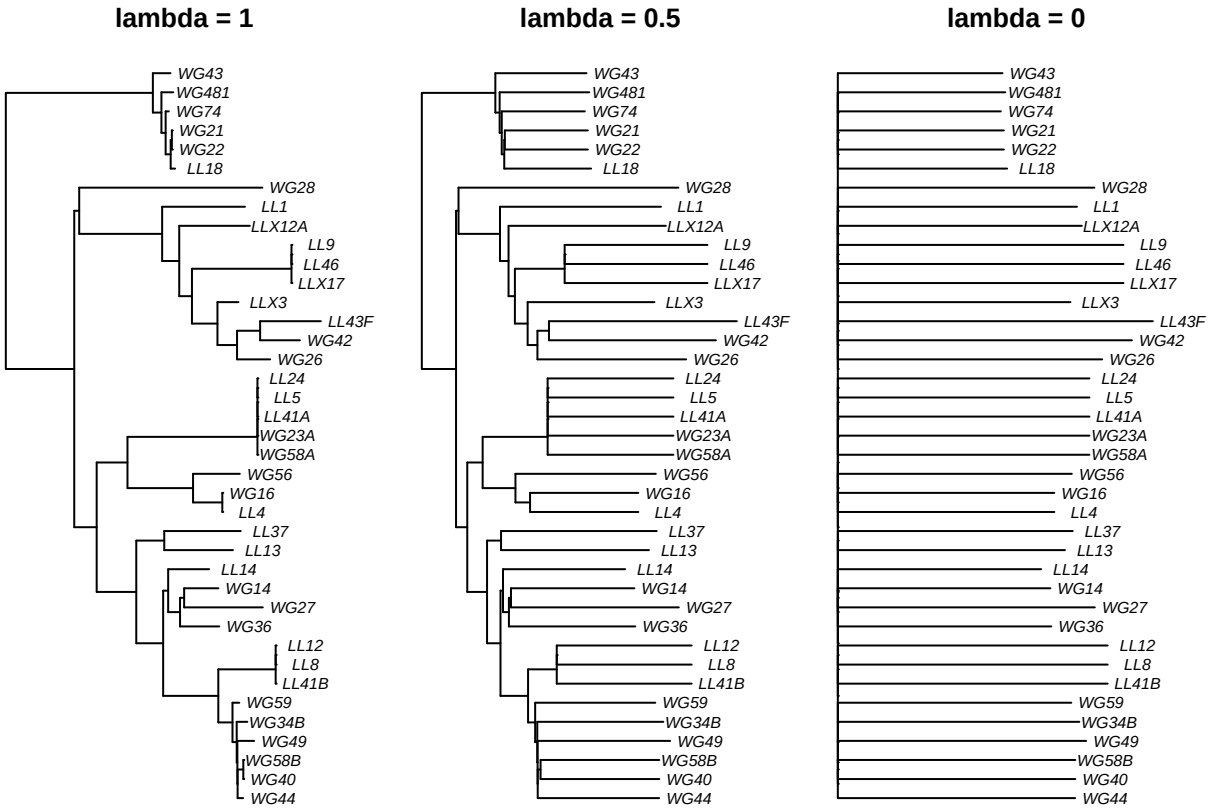
Answer 6a: Species that have lower (narrower) niche breadths will have a very high max growth rate on one or a few P sources, while species with higher (broader) niche breadth will have moderate max growth rates on many P sources ***Answer 6b***: On the figure, species with darker colored niche breadths should have a medium-colored band that doesn't vary wildy across all sources of P; species with lighter colored niche breadths should have bands that are very light on most P sources but are very dark on one (or a few)

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 with different levels of phylogenetic signal {geiger}
nj.lambda.5 <- rescale(nj.rooted, "lambda", 0.5)
nj.lambda.0 <- 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.

```
# generate test statistics for comparing phylogenetic signal {geiger}
fitContinuous(nj.rooted, nb, model = "lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.106395
##  z0 = 0.657777
##
##  model summary:
##  log-likelihood = 21.652293
##  AIC = -37.304587
##  AICc = -36.618872
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 52
##  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 = 0.106395
##  z0 = 0.657777
##
##  model summary:
##  log-likelihood = 21.652293
##  AIC = -37.304587
##  AICc = -36.618872
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  frequency of best fit = 0.88
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

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

> ***Answer 7a***: The lambda for the untransformed tree is approximately 0.021. Since lambda ranges from 0 to 1, this is quite small, meaning that the phylogenetic signal is quite weak. ***Answer 7b***: Both models have very similar AIC scores; I would pick the untransformed tree since its AIC score was slightly lower, but I recognize that this is not a significant difference. ***Answer 7c***: These results suggest that there is no significant 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 on our tree
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7

# calculate phylogenetic signal for growth on all P resources
# first, create a blank output matrix
p.phylosignal <- matrix(NA, 6, 18)
colnames(p.phylosignal) <- colnames(p.growth.std)
rownames(p.phylosignal) <- c("K", "PIC.var.obs", "PIC.var.mean", "PIC.var.P", "PIC.var.z", "PIC.P.BH")
```

```
# use a for loop to calculate Bloomberg's K for each resource
for(i in 1:18){
  x <- as.matrix(p.growth.std[ , i, drop = FALSE])
  out <- phylosignal(x, nj.rooted)
  p.phylosignal[1:5, i] <- round(t(out), 3)
}

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

p.phylosignal
```

```
##                   AEP      PEP      G1P      G6P  MethCP      BGP      DNA
## K               0.000    0.000    0.000    0.000   0.000    0.000    0.000
## PIC.var.obs  4373.157  664.095  948.941 5924.730 350.894  536.104  259.084
## PIC.var.mean 8338.335 1546.112 1894.472 3630.032 507.034 1772.141 5182.407
## PIC.var.P       0.237    0.071    0.104    0.759   0.334    0.026    0.001
## PIC.var.z      -0.851   -1.322   -1.248    0.929  -0.464   -1.709   -1.300
## PIC.P.BH        0.609    0.319    0.374    0.804   0.616    0.156    0.018
##                   Peth    Pchol       B1     Phyt      SRP     cAMP
## K               0.000    0.000    0.000    0.000    0.000    0.000
## PIC.var.obs  1446.463 2368.391 3517.018 9240.368 1307.025  690.723
## PIC.var.mean 1843.292 3270.438 5429.012 9163.794 1605.282 3042.116
## PIC.var.P       0.327    0.420    0.223    0.581    0.342    0.005
## PIC.var.z      -0.487   -0.513   -0.805    0.010   -0.513   -2.547
## PIC.P.BH        0.616    0.687    0.609    0.740    0.616    0.045
##                   ATP PhenylCP    PolyP      GDP      GTP
## K               0.000    0.000    0.000    0.000    0.000
## PIC.var.obs  4040.137 1224.017 1126.345 4473.878 2721.766
## PIC.var.mean 3058.625  748.817 1221.846 3587.044 2929.059
## PIC.var.P       0.617    0.810    0.481    0.661    0.497
## PIC.var.z       0.429    1.007   -0.169    0.405   -0.152
## PIC.P.BH        0.740    0.810    0.688    0.744    0.688
```

```
# calculate 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 3.427719e-06         49966.78              49573.92          0.549
##   PIC.variance.Z
## 1     0.02003935
```

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

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

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

Answer 8a: There is only phylogenetic signal for standardized growth on DNA and cAMP; there is extremely weak (essentially no) phylogenetic signal for niche breadth ***Answer 8b***: All the K values are 0, indicating overdispersion

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 into categorical data
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)

# look at P 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 column to data
p.growth.pa$name <- rownames(p.growth.pa)

# merge trait and phylogenetic data; run 'phylo.d'
p.traits <- comparative.data(nj.rooted, p.growth.pa, "name")
phylo.d(p.traits, binvar = ATP)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  ATP
##   Counts of states:  0 = 1
##                      1 = 38
##   Phylogeny :  nj.rooted
##   Number of permutations :  1000
##
## Estimated D :  4.046872
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.833
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.067
```

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

```
##
## 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 :  1000
##
## Estimated D :  0.6094273
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.03
```

```
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.003
```

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

```
##
## 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 :  1000
##
## Estimated D :  0.1480059
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.002
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.315
```

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.

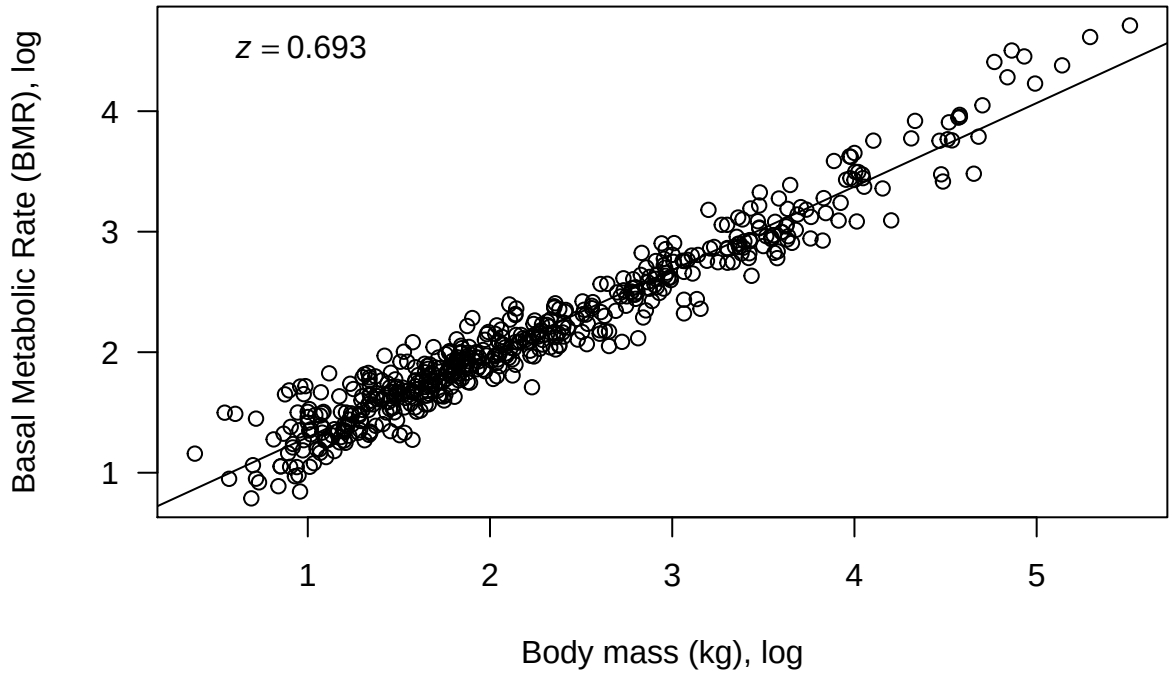
Answer 9a: The growth traits seem to be overdispersed, since all values of D are positive. ***Answer 9b***: Both models show overdispersion, but the D values show some significance, whereas Blomberg's K did not have significant p-values ***Answer 9c***: BLomberg's K compares only to Brownian motion as the null, whereas D used two different nulls, Brownian motion and random phylogenetic structure

7) PHYLOGENETIC REGRESSION

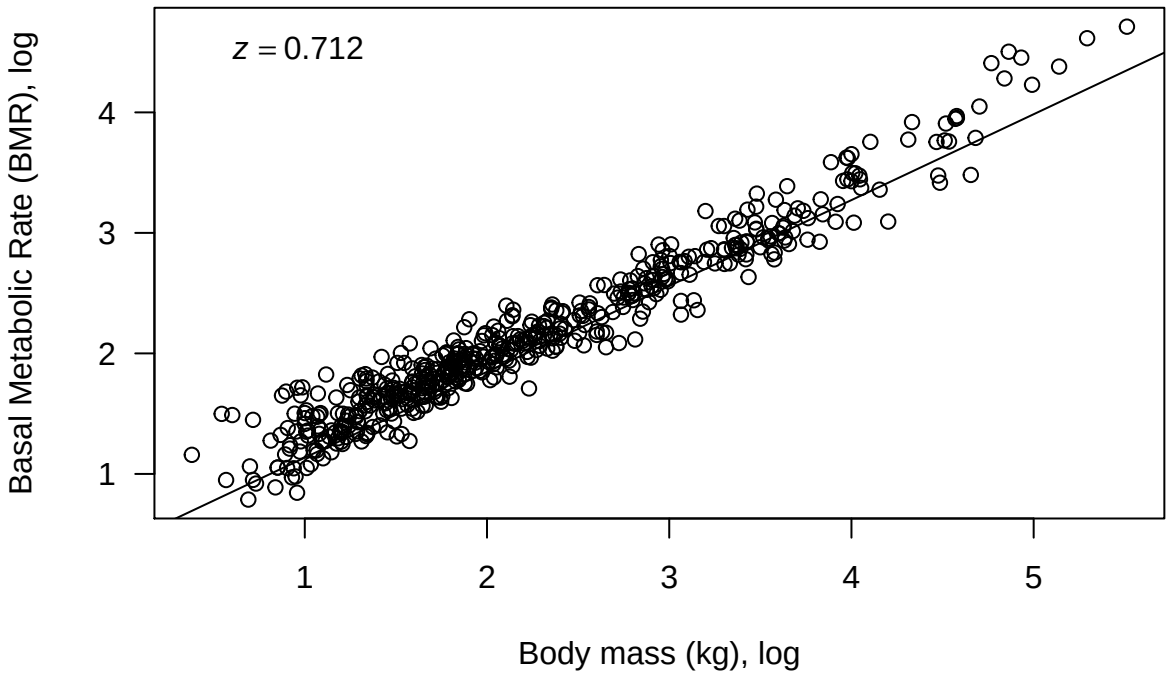
In the R code chunk below, do the following:
1. Load and clean the mammal phylogeny and trait dataset, 2. Fit a linear model to the trait dataset, examining the relationship between mass and BMR, 2. Fit a phylogenetic regression to the trait dataset, taking into account the mammal supertree

```
# input the tree and dataset
mammal.Tree <- read.tree("data/mammal_best_super_tree_fritz2009.tre")
mammal.data <- read.table("data/mammal_BMR.txt", sep = "\t", header = TRUE)
#select the variables we want to analyze
mammal.data <- mammal.data[ , c("Species", "BMR_.mlO2.hour.", "Body_mass_for_BMR_.gr.")]
mammal.species <- array(mammal.data$Species)
# select the tips in the mammal tree that are also in the dataset
pruned.mammal.tree <- drop.tip(mammal.Tree, mammal.Tree$tip.label[-na.omit(match(mammal.species, mammal
# select the species from the dataset that are in our pruned tree
pruned.mammal.data <- mammal.data[mammal.data$Species %in% pruned.mammal.tree$tip.label, ]
# turn column of species names in row nanes
rownames(pruned.mammal.data) <- pruned.mammal.data$Species

# examine mass vs BMR
# run a simple linear regression
fit <- lm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), data = pruned.mammal.data)
```

```
plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.), log10(pruned.mammal.data$BMR_.mlO2.hour.), las =
     ylab = "Basal Metabolic Rate (BMR), log", xlab = "Body mass (kg), log")
abline(a = fit$coefficients[1], b = fit$coefficients[2])
b1 <- round(fit$coefficients[2], 3)
eqn <- bquote(italic(z) == .(b1))
# plot the slope
text(0.5, 4.5, eqn, pos = 4)
```




```
# run a phylogeny-corrected regression with no bootstrap replicates
fit.phy <- phylolm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), data = pruned.mammal.data,
                   pruned.mammal.tree, model = 'lambda', boot = 0)
plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.), log10(pruned.mammal.data$BMR_.mlO2.hour.), las =
     ylab = "Basal Metabolic Rate (BMR), log", xlab = "Body mass (kg), log")
abline(a = fit.phy$coefficients[1], b = fit.phy$coefficients[2])
b1.phy <- round(fit.phy$coefficients[2], 3)
eqn <- bquote(italic(z) == .(b1.phy))
text(0.5, 4.5, eqn, pos = 4)
```



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: Shared evolutionary history means that traits are not independent ***Answer 10b***: In a linear regression the residuals are assumed to be independent and identically distributed, while in a phylogenetic regression the residuals follow variances defined by a covariance matrix which is defined by the underlying tree structure ***Answer 10c***: By accounting for shared evolutionary history the fit of the model was improved, which indicates that mass and BMR are even more strongly correlated than expected. ***Answer 10d***: If groups of study organisms are chosen poorly, response to a variable may be completely confounded by phylogeny. For example, if you're interested in height increase per month as a function of rainfall, the signal will be completely washed out by phylogeny if you only choose to study, say, one broad woody tree with deep roots, one slow-growing succulent, and one fast-growing annual forb.

7) SYNTHESIS

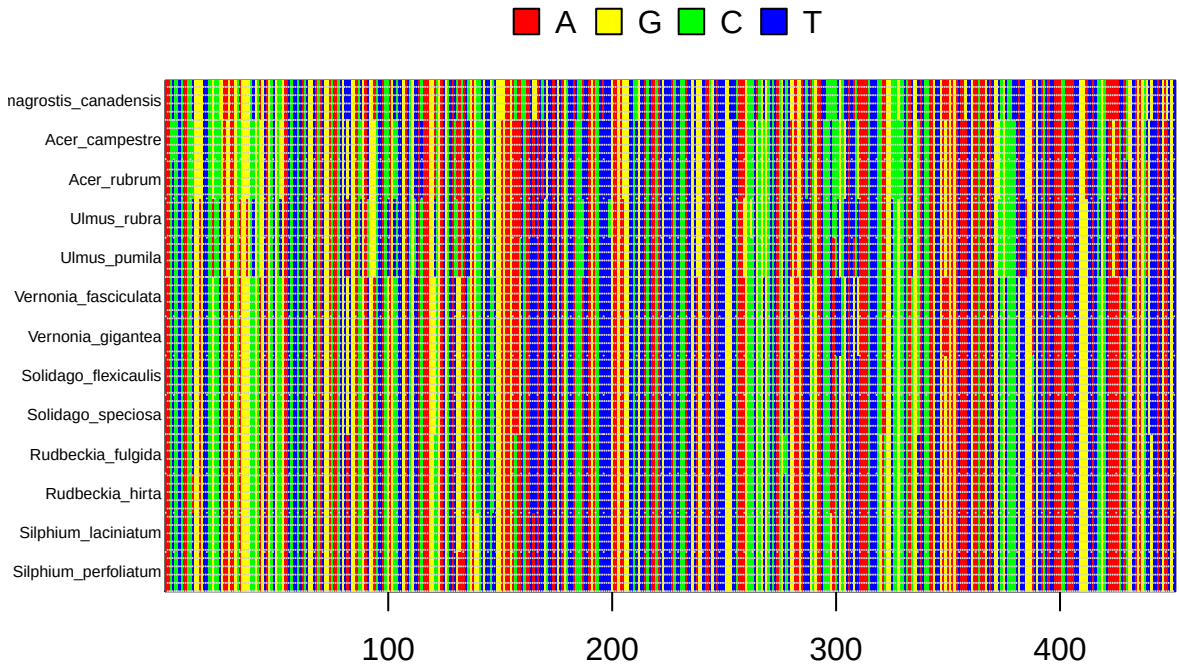
Work with members of your Team Project to obtain reference sequences for 10 or more taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. 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.

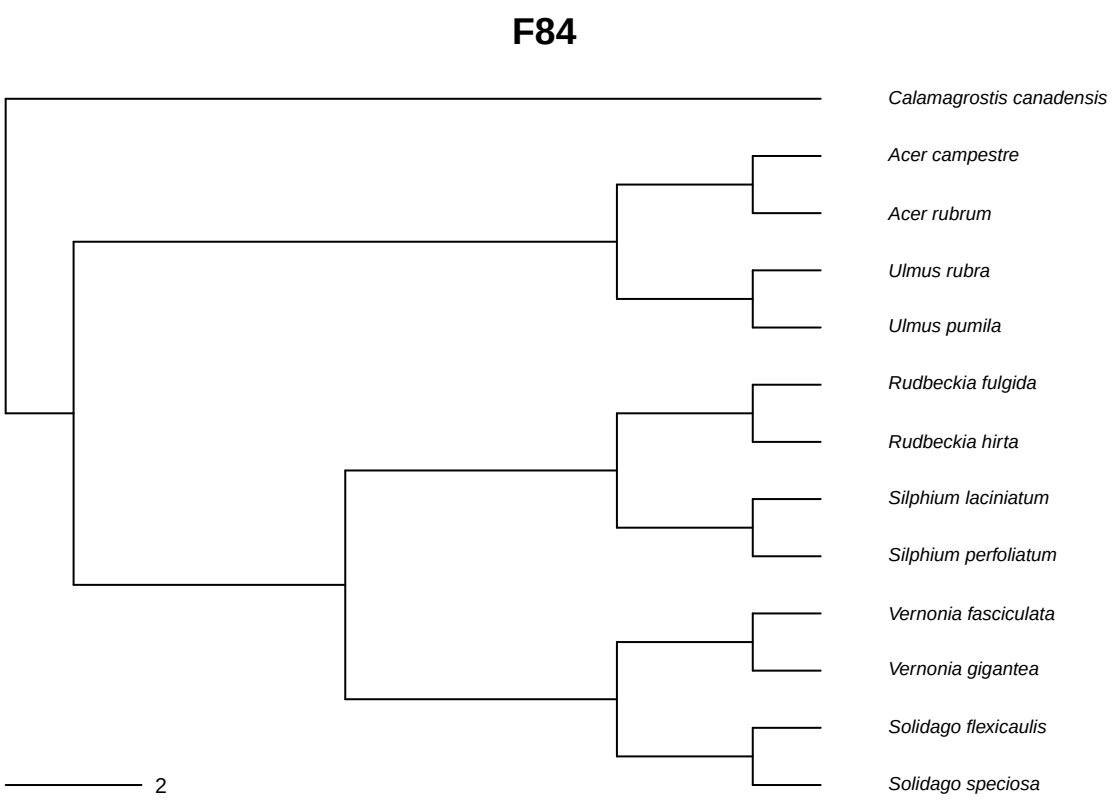
After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. Based on the decisions above and the output, does your tree jibe with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

```
## visualize to check alignment

# read alignment file {seqinr}
read.aln <- read.alignment(file = "plant_seq.afa", format = "fasta")
# convert alignment file to DNAbin object {ape}
p.DNAbin <- as.DNAbin(read.aln)
# identify base pair region of 16S rRNA gene to visualize
window <- p.DNAbin[ , 100:550]
# command to visualize sequence alignment {ape}
image.DNAbin(window, cex.lab = 0.50)
# optional code to add grid to help visualize rows
grid(ncol(window), nrow(window), col = "lightgrey")
```




```
## make a tree using F84 substitution model

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

```r
# neighbor joining algorithm to construct the tree, a 'phylo' object {ape}
nj.tree <- bionj(seq.dist.F84)

# identify outgroup structure
outgroup <- match("Calamagrostis_canadensis", nj.tree$tip.label)

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

# plot the rooted tree {ape}
par(mar = c(1, 1, 2, 1) + 0.1)
plot.phylo(nj.rooted, main = "F84", "phylogram", use.edge.length = FALSE,
           direction = "right", cex = 0.6, label.offset = 1)
add.scale.bar(cex = 0.7)
```



Our tree fits with what we know about our organisms. Our outgroup is a monocot whereas all other species are dicots. ALl of our congeneric species are sister to each other, so that's good. Additionally, we inadvertantly picked 8 species in Asteraceae, which grouped together, which gives me confidence in our tree.