11. Worksheet: Phylogenetic Diversity - Traits

Dustin Brewer; 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.pd**

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] "C:/Users/dusti/GitHub/QB2019_Brewer/2.Worksheets/11.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', 'caper', 'phylolm', 'pmc', 'ggplot2', 'tidyr', 'dplyr', 'phangorn', 'pander') for (package in package.list) { if (!require(package, character.only=TRUE, quietly=TRUE)) { install.packages(package) library(package, character.only=TRUE) } }

```
library("ape", lib.loc="~/R/win-library/3.5")
 library("seqinr", lib.loc="~/R/win-library/3.5")
```

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

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

```
library("phylobase", lib.loc="~/R/win-library/3.5")
```

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

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

```
library("adephylo", lib.loc="~/R/win-library/3.5")
```

```
## Loading required package: ade4
```

```
library("geiger", lib.loc="~/R/win-library/3.5")
library("picante", lib.loc="~/R/win-library/3.5")
```

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

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

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

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

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

```
## This is vegan 2.5-3
```

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

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

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

```
library("stats", lib.loc="C:/Program Files/R/R-3.5.2/library")
library("RColorBrewer", lib.loc="~/R/win-library/3.5")
library("caper", lib.loc="~/R/win-library/3.5")
```

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

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

```
library("phylolm", lib.loc="~/R/win-library/3.5")
library("pmc", lib.loc="~/R/win-library/3.5")
library("ggplot2", lib.loc="~/R/win-library/3.5")
library("tidyr", lib.loc="~/R/win-library/3.5")
library("dplyr", lib.loc="~/R/win-library/3.5")
```

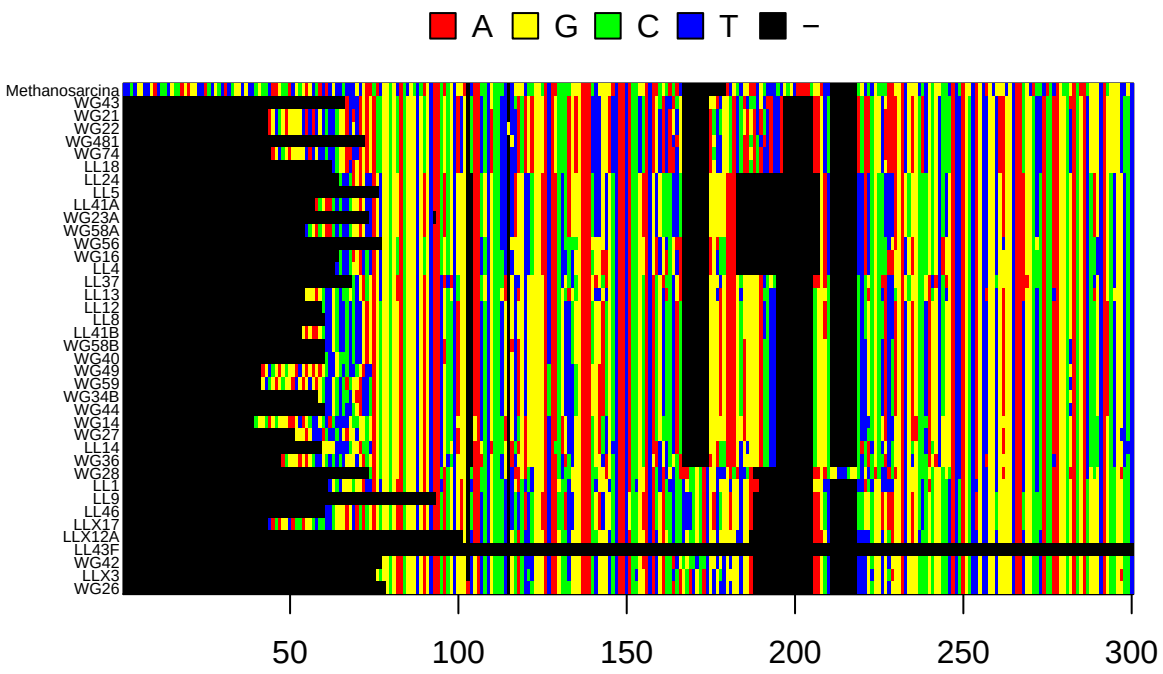
```
##
## Attaching package: 'dplyr'
```

```
## 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
```

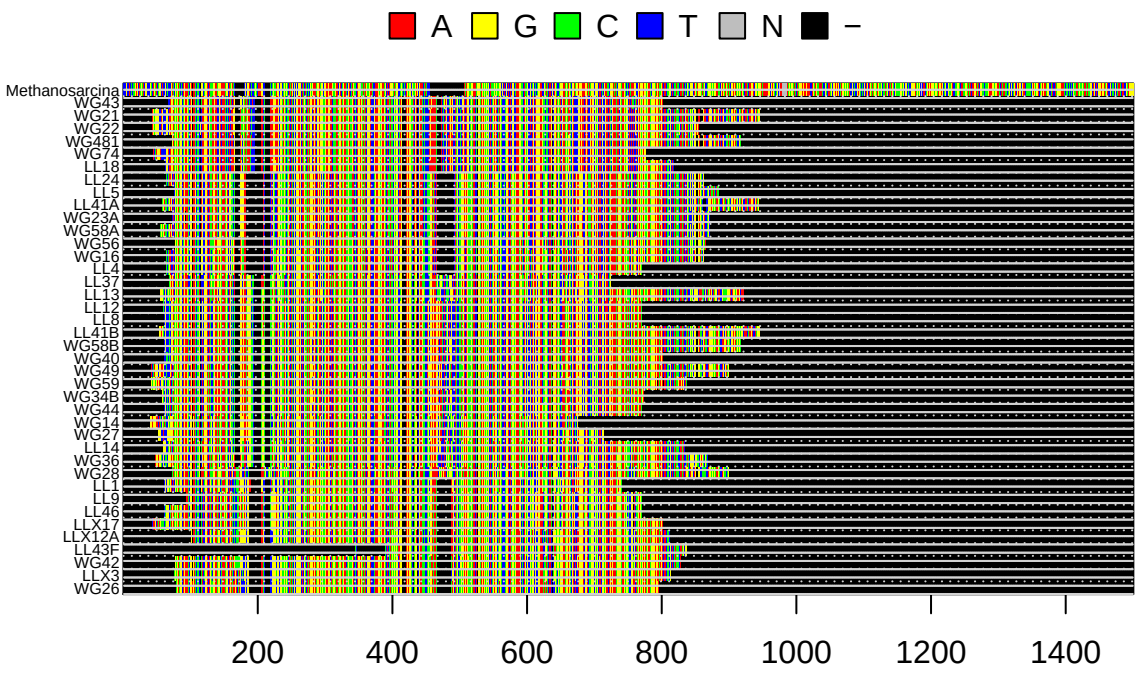
```
library("phangorn", lib.loc="~/R/win-library/3.5")
```

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

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

```
library("pander", lib.loc="~/R/win-library/3.5")

#read the alignment file from muscle (function from seqinr package)
read.aln <- read.alignment(file = "./data/p.isolates.afa", format = "fasta")

#Covert alignment file to DNAbin, which allows R to store and manipulate sequence data
p.DNAbin <- as.DNAbin(read.aln)

#ID a region of base pairs to visualize
window <- p.DNAbin[, 1:1500]
window2 <- p.DNAbin[, 1:300]


#Visualize the sequence alignment (function from ape package)
image.DNAbin(window2, cex.lab = .5)
```




```
image.DNAbin(window, cex.lab = .5)


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




```
#alignment looks good because many vertical lines suggest that a given nucleotide is shared at a site.
```

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***: It appears that most sequence reads for bacterial strains are around 800 nucleotides long, but the outgroup is about 1500 long. ***Answer 2b***: I think that from column 500 to 650 would be appropriate for phylogenetic inference, because all of the strains have nucleotides at these locations.

4) MAKING A PHYLOGENETIC TREE

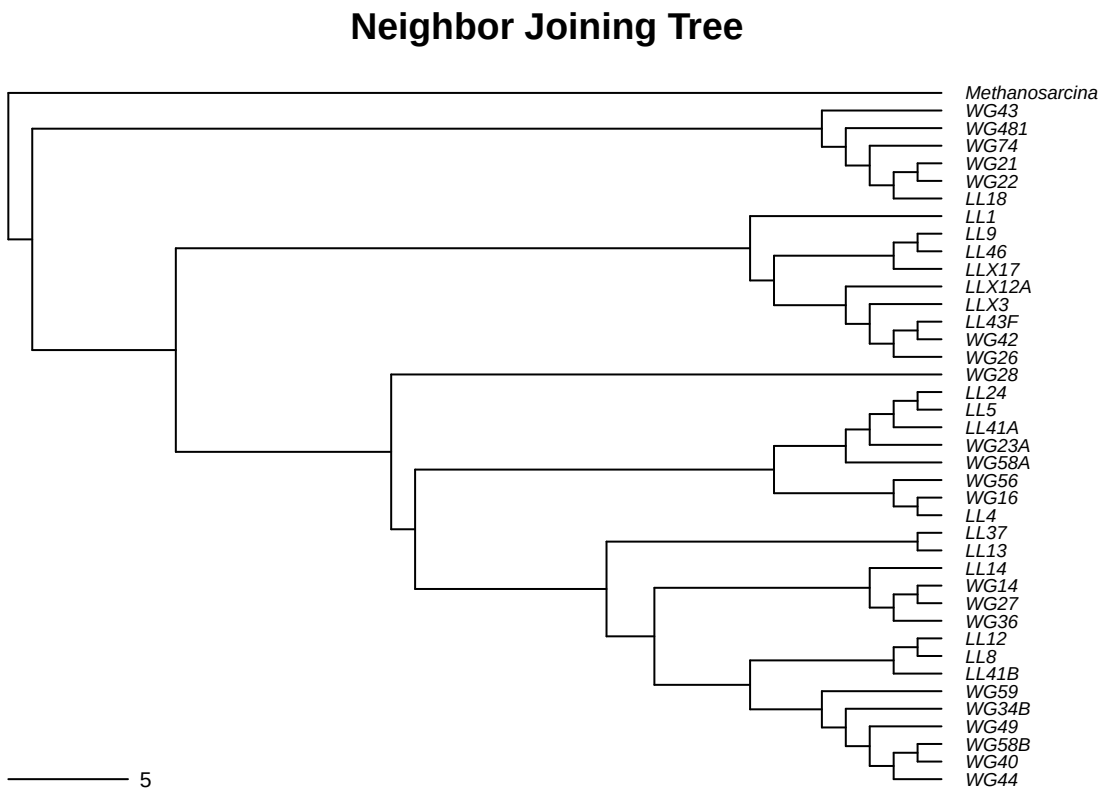
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

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 a distance matrix. "Pairwise = FALSE" means that columns in which a row is missing data are del
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)

#Create a neighbor joining tree based on these distances
nj.tree <- bionj(seq.dist.raw)

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

#Root the tree (compare the outgroup to everything else, i.e. the ingroup)
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

#Plot the rooted tree
par(mar = c(1, 1, 2, 1) + .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?

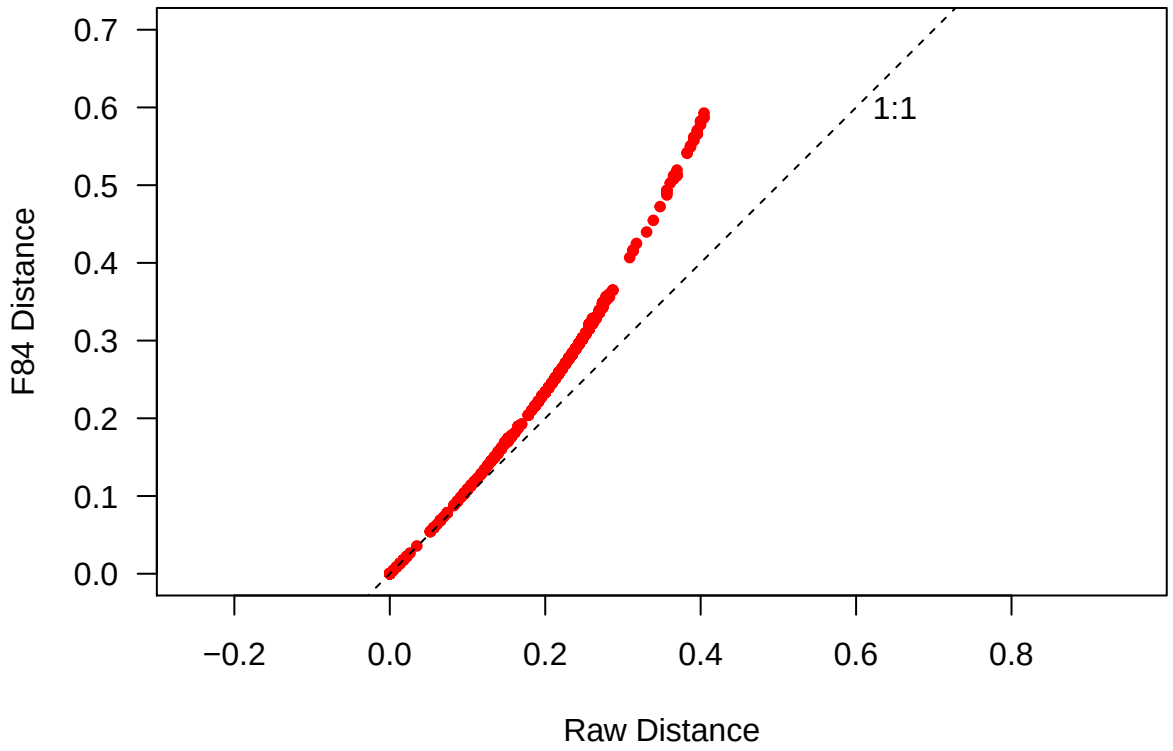
> ***Answer 3***: My understanding is that using the neighbor joining method is advantageous because it quickly produces a phylogeny that is generally correct. However, sequence data is disregarded

for columns if a single row or more is empty in that column, and multiple substitutions at the same site over time are not accounted for. It seems that there are more accurate, powerful techniques available than the neighbor joining method.

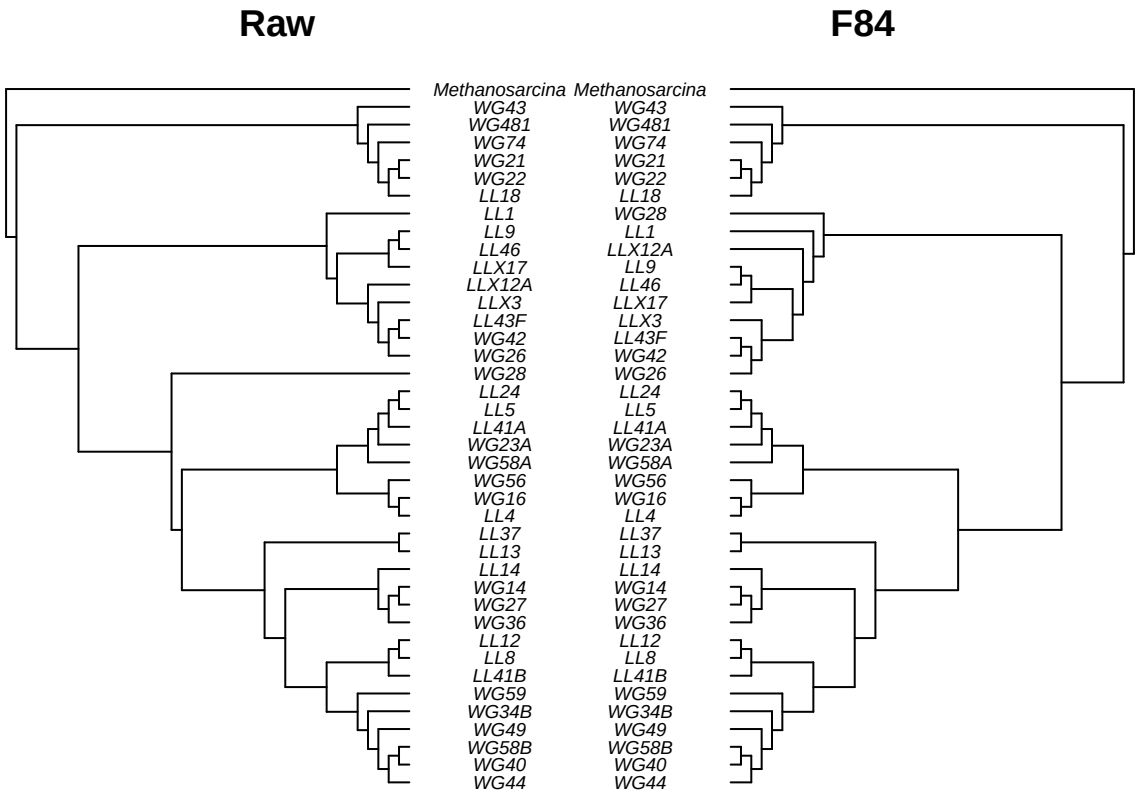
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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

```
#Distance matrix based on "Felsenstein 84" method
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise.deletion = FALSE)

#Create a 'saturation plot,' to compare 'raw' distance and 'F84' distance
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")
```



```
#Note that F84 distance tends to be longer, because it is correcting for substitutions

#Now, make neighbor joining trees for both
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
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root=TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root=TRUE)

# Make Cophylogenetic (to compare different models)
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")
```

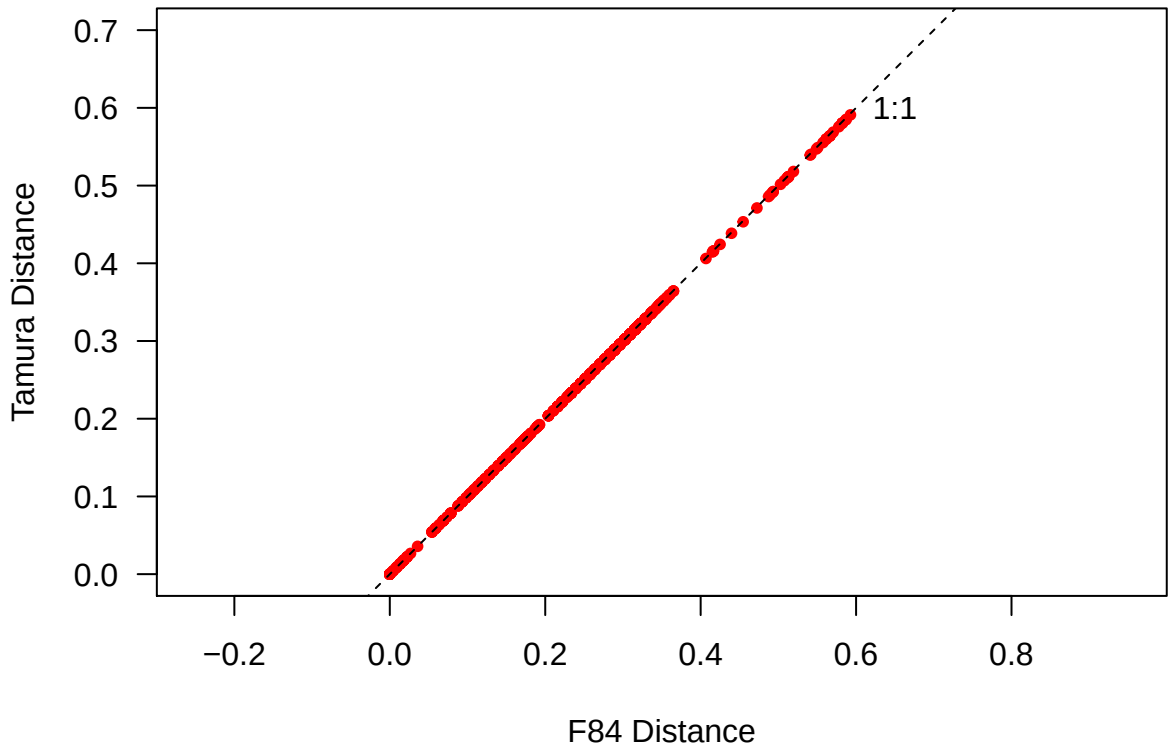



```
#Compare branch lengths between the two methods
dist.topo(raw.rooted, F84.rooted, method = "score")
```

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

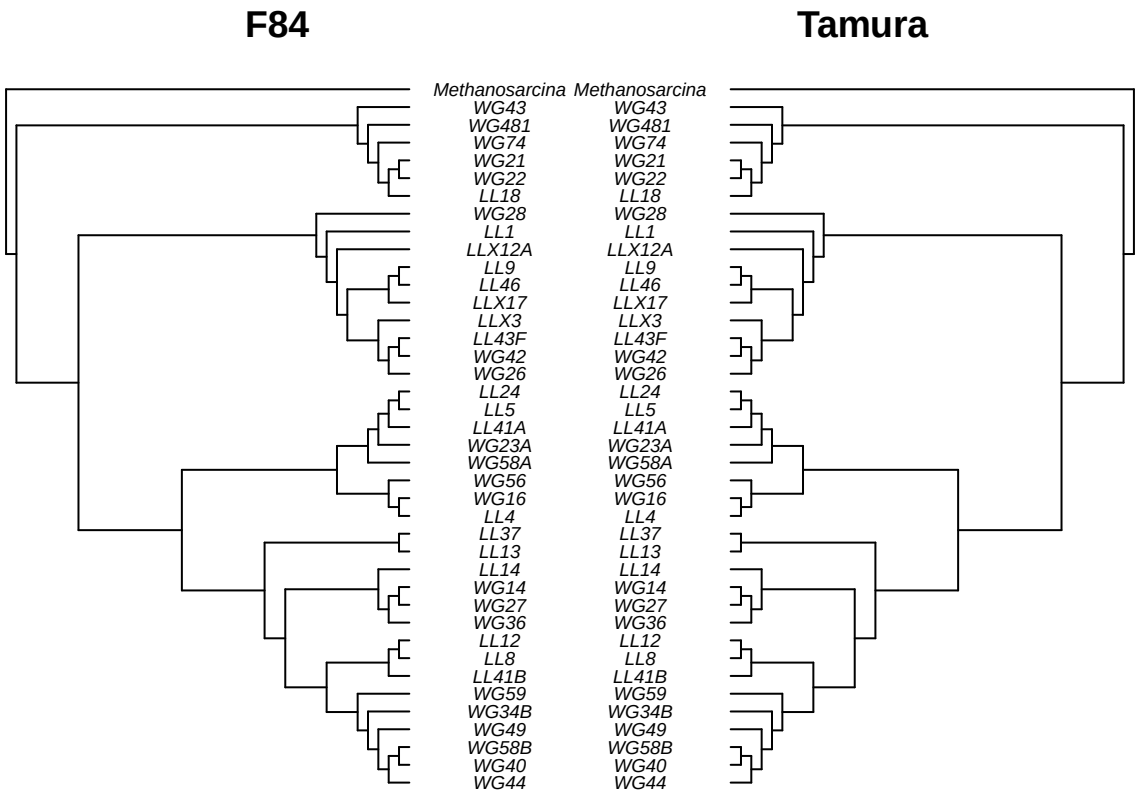
In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5. be sure to format, add appropriate labels, and customize each plot.

```
#Distance matrix based on Tamura method
seq.dist.T92 <- dist.dna(p.DNAbin, model = "T92", pairwise.deletion = FALSE)

#Create a 'saturation plot,' to compare 'raw' distance and 'F84' distance
par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.F84, seq.dist.T92,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7), ylim = c(0, 0.7),
     xlab = "F84 Distance", ylab = "Tamura Distance")
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1")
```




```
#Note that F84 distance tends to be longer, because it is correcting for substitutions

#Now, make neighbor joining trees for both
tamura.tree <- bionj(seq.dist.T92)
```

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

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

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

# Make Cophylogenetic (to compare different models)
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(tamura.rooted, type = "phylogram", direction = "left", show.tip.label=TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "Tamura")
```




```
#Compare branch lengths between the two methods
dist.topo(tamura.rooted, F84.rooted, method = "score")
```

```
##              tree1
## tree2 0.0005224877
```

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 Tamura model. This model assumes equal frequencies of nucleotides, but accounts for the fact that transition mutations occur more frequencly than transversion mutations and for G and C bias. The F84 appears similiar, but it allows for differences in frequencies of nucleotides. ***Answer 4b***: The saturation and cophylogenetic plots indicate that the two models produce virtually the same result.
Answer 4c: Given the sameness between the results of the models, I presume that nucleotide frequencies are about the same in the samples, like the Tamura model assumed. As for substitution rates, these models both assume that transition mutations occur more than transversion mutations, and are likely more accurate than the neighbor joining 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

```
# Requires alignment to be read in with as phyDat object

p.DNAbin.phyDat <- read.phyDat("./data/p.isolates.afa", format="fasta", type="DNA")

#Two times, when I tried to run the code from the handout to generate object 'fit,' it was running for
```

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: Two times, when I tried to run the code from the handout to generate object 'fit,' it was running for more than 30 mins at which point I terminated the R sessions. I'm not sure if my computer crashed. ***Answer 5b***: One would bootstrap a tree because doing so can help to determine if your tree is likely to be accurate. Creating multiple trees with the same data allows one to compare the different trees and to determine if a the tree-building process is reliable.
Answer 5c: The values tell you how reliable your model is. Values of 95% or more for a given node generally indicate that that node is reliable. ***Answer 5d***: In the handout, the branches with the lowest support are between WG42 and LL43F, and the node immediately above that.
Answer 5e: My impression is that we should be skeptical of those branches.

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

#Create a function that calculates niche breadth
levins <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
    }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}

#Calculate niche breadth for each isolate
nb <- as.matrix(levins(p.growth.std))

#Add row and column names to matrix
rownames(nb) <- row.names(p.growth)
colnames(nb) <- c("NB")
```

C. Visualizing Traits on Trees

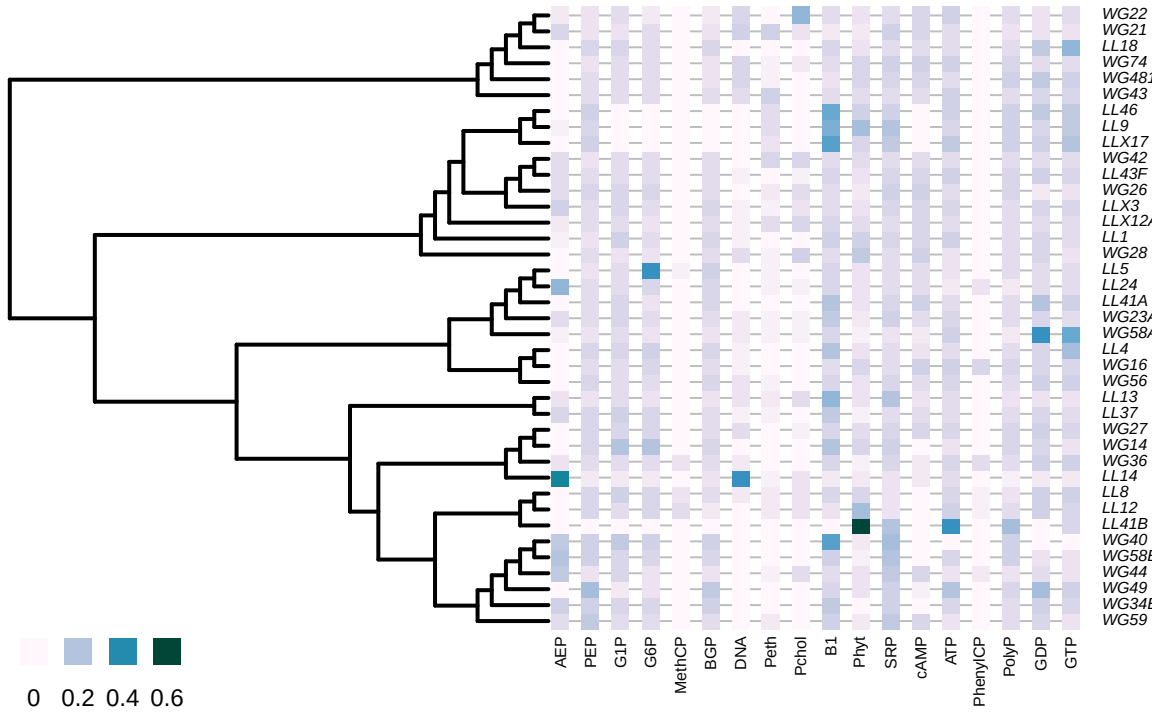
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

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

# Define the outgroup
```

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

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

# 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 phosporous traits
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
```

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?

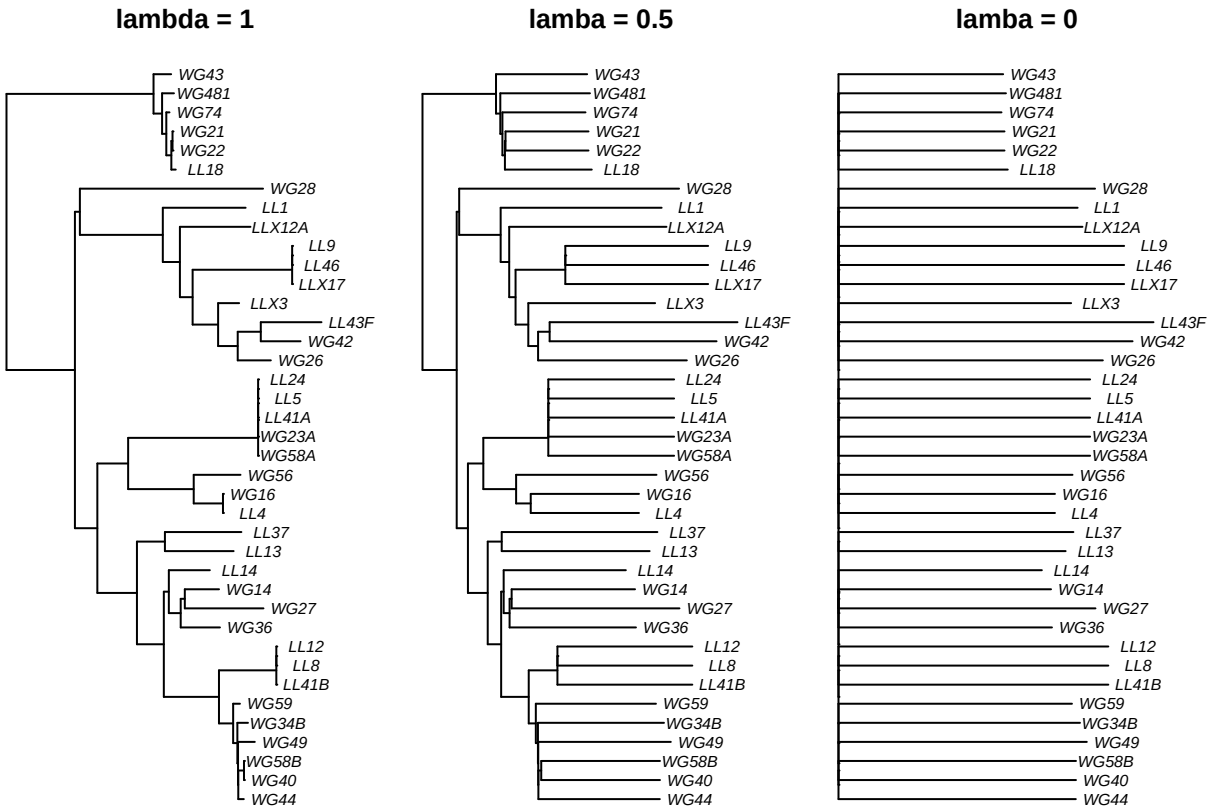
> ***Answer 6a***: As number of substrates that a bacterial strain is found on increases, its average growth rate at each site decreases. ***Answer 6b***: In support of this hypothesis would be to find that strains found on, say, 4 substrate types typically multiplied faster on those substrate types than did strains found on many substrate types (say, 8 or more).

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 'phylogentic signals' (i.e., similiarity based on relatedness?)
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 = "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.

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.020848
##  sigsq = 0.106492
##  z0 = 0.661368
##
##  model summary:
##  log-likelihood = 21.661104
##  AIC = -37.322208
##  AICc = -36.636494
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 49
##  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 = 0.86
##
##  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 lamda value of the untransformed tree is 0.02, compared to the 0 of the transformed tree. ***Answer 7b***: Both of the AIC values are -37, and so seem equally acceptable to choose. ***Answer 7c***: I'm not sure, but it seems that if the model in which all phylogenetic signal was removed from the tree is just as good the non-transformed tree according to AIC, then perhaps there is not phylogenetic signal amongst the strains.

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.

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

# Calculate phylogenetic signal for growth on all substrates
```

```
  #   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 Blomberg'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[6,]
```

```
##      AEP      PEP      G1P      G6P   MethCP      BGP      DNA     Peth
##    0.612    0.337    0.400    0.822    0.645    0.144    0.036    0.645
##    Pchol       B1     Phyt      SRP     cAMP      ATP PhenylCP    PolyP
##    0.645    0.612    0.716    0.645    0.045    0.719    0.822    0.694
##      GDP      GTP
##    0.719    0.676
```

```
#Calculate phylogenetic signal for nich 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              49703.75          0.541
##   PIC.variance.Z
## 1     0.01303725
```

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 appear to be phylogenetic signal with respect to growth, on DNA and cAMP. ***Answer 8b***: I think that the traits are overdispersed (K values less than 1), though it is possible that I was looking at the wrong #s.

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.

```r
# Turn continuous data into categorical data
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)

# Look at P 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
```

```r
# Add 'names' column
p.growth.pa$name <- rownames(p.growth.pa)

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

```
##
## 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 :  1000
##
## Estimated D :  0.4626351
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.009
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.027
```

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

```
##
## 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 :  1000
##
## Estimated D :  0.8664323
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.273
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.016
```

```
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 :  0.6038142
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.027
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.004
```

```
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 :  0.1503457
## Probability of E(D) resulting from no (random) phylogenetic structure :  0
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.313
```

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?

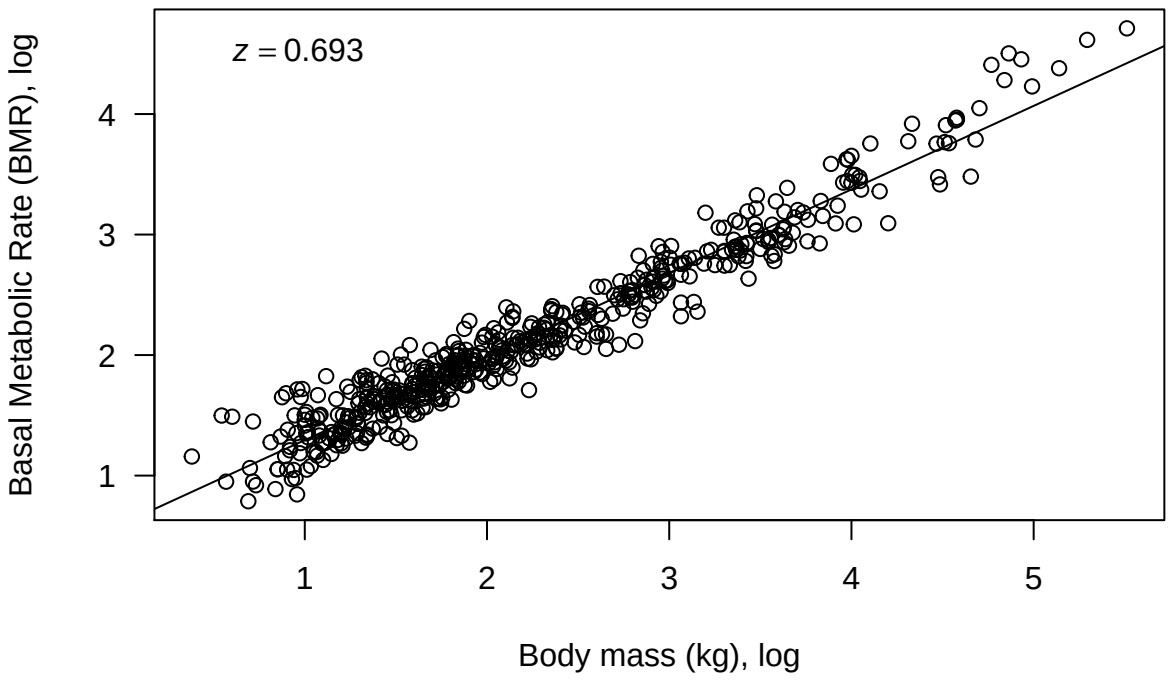
c. Discuss what factors might give rise to differences between the metrics.

Answer 9a: AEP, PhenylCP, and DNA all had D values greater than 0 and not close to 1, suggesting overdispersion. ***Answer 9b***: The Blomberg's K analysis p-values for AEP (.591) and PhenylCP (0.828) did not agree with the D values (i.e. Blombergs K suggested no phylogentic signal and D suggested a phylogenetic signal). The DNA p value (0.009) for Blomberg's K analyis did agree with the D value, suggesting a phylogenetic signal.
Answer 9c: The metrics may yield different results because they were designed for different purposes. My impression is the Blomberg's K might be more suitable for phylogenetic anlysis than 'D', though I certainly could be wrong.

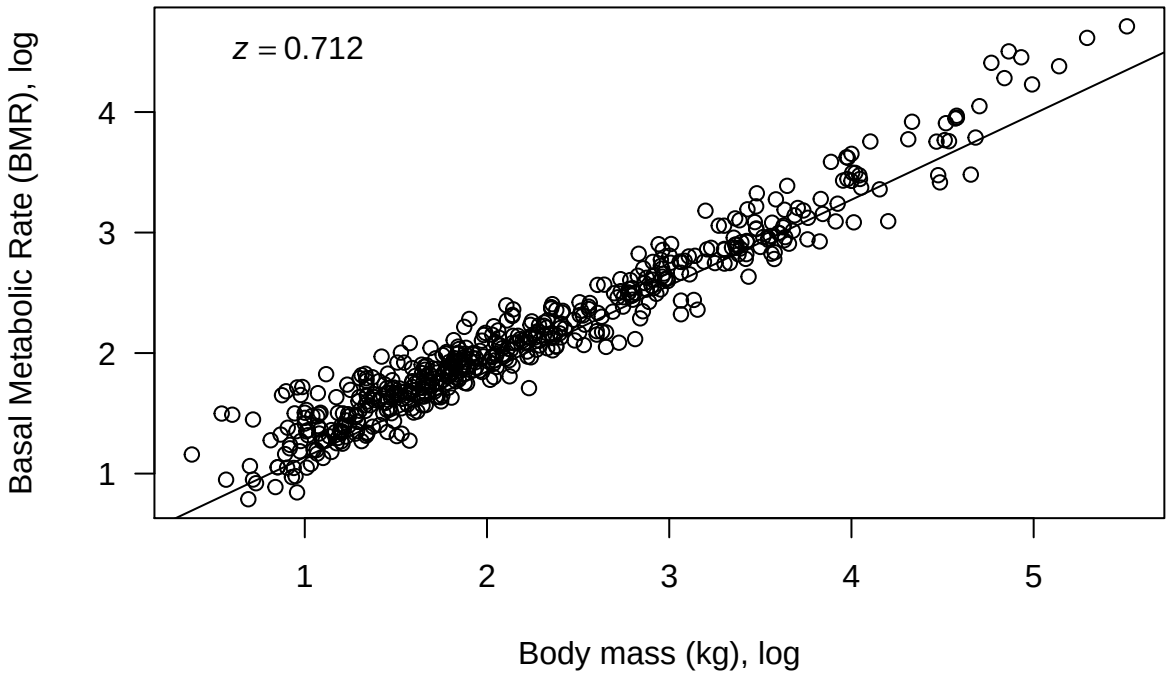
7) PHYLOGENETIC REGRESSION

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 mammal data set
mammal.Tree <- read.tree("./data/mammal_best_super_tree_fritz2009.tre")
mammal.data <- read.table("./data/mammal_BMR.txt", sep = "\t", header = TRUE)

# Select variables of interest
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 prunned tree
pruned.mammal.data <- mammal.data[mammal.data$Species %in% pruned.mammal.tree$tip.label,]

# Turn column of species names into rownames
rownames(pruned.mammal.data) <- pruned.mammal.data$Species

# 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 =
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 =
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 needs to be corrected for because otherwise the assumption of independence (say, in regression analysis) is violated. ***Answer 10b***: In a phylogenetic regression, the variance of the residual errors take into account the branch lengths of the phylogeny. This is unlike in a simple regression, in which the residual errors are assumed to be independently distributed while following a normal distribution. ***Answer 10c***: The slope of each model is positive, and approximately 0.7. It appears that accounting for shared evolutionary history improved the fit. The z value (why not r?) for simple regression was 0.693 and for phylogenetic regression it was 0.712. ***Answer 10d***: If for several fish species in different taxonomic families annual # of eggs produced and age were regressed, a positive, linear relationship between age and # of eggs produced might disappear when a phylogenetic regression is done. For example, taxa A might primarily produce eggs at one year of age, taxa B (bigger than A) might primarily produce eggs at two years of age, and taxa C (bigger than A and B) might primarily produce eggs at three years of age. While at first it might seem that as individual fish from different taxa get older they produce more eggs, it could be found that the species that tend to breed later are bigger fish that lay more eggs, but that any individual will likely lay the same amount of eggs from year to year.

7) SYNTHESIS

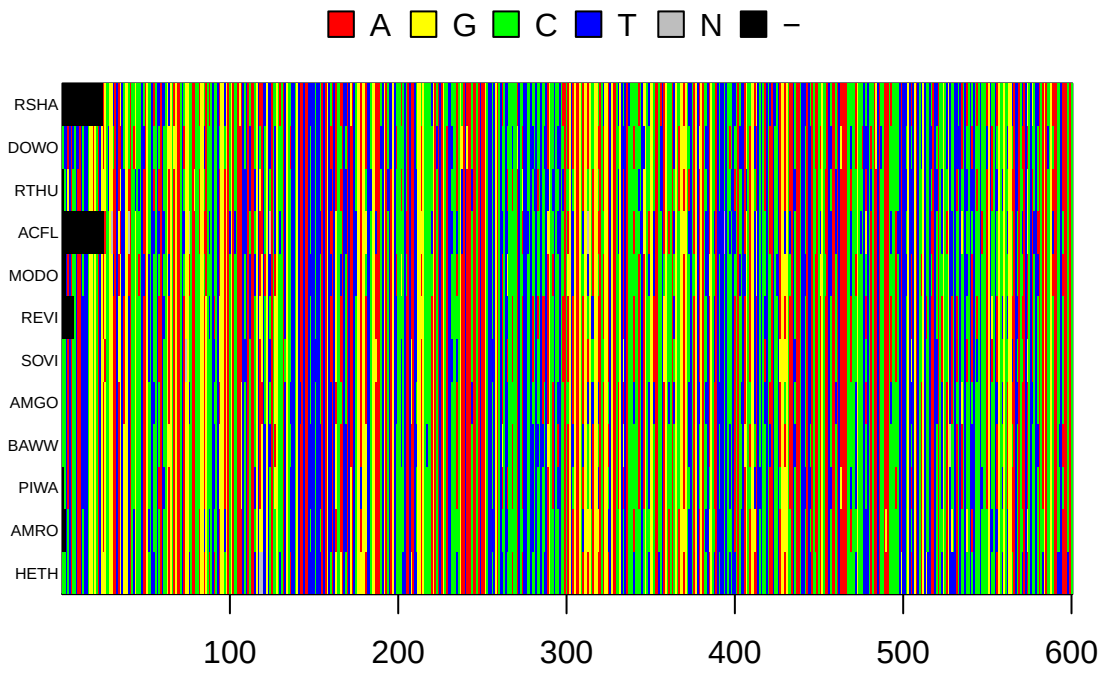
Work with members of your Team Project to obtain reference sequences for 10 or more taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/.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

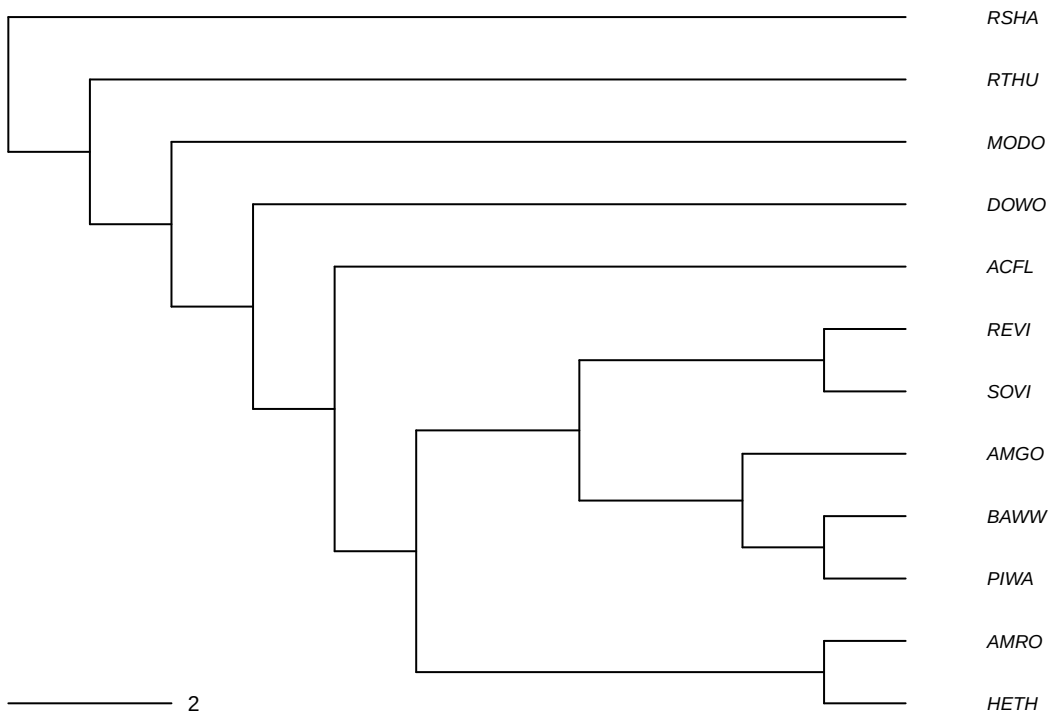
After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. Based on the decisions above and the output, does your tree jibe with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

> The tree very well matches what I would expect, phylogenetically. The vireos (REVI, SOVI) are grouped together, the thrushes (HETH and AMRO) are grouped together, and the non-passerines are shown to be distinct from the passerines. If I were to do something differently in the future, I'd use a more advanced method (rather than 'raw), when I have more time.

```
#read the alignment file from muscle (function from seqinr package)
read2.aln <- read.alignment(file = "./data/birds.afa", format = "fasta")

#Covert alignment file to DNAbin, which allows R to store and manipulate sequence data
p.DNAbin2 <- as.DNAbin(read2.aln)

#ID a region of base pairs to visualize
window.b <- p.DNAbin2[, 1:600]



#Visualize the sequence alignment (function from ape package)
image.DNAbin(window.b, cex.lab = .5)
```




```
#Create a distance matrix. "Pairwise = FALSE" means that columns in which a row is missing data are del
seq.dist.raw.b <- dist.dna(p.DNAbin2, model = "raw", pairwise.deletion = FALSE)

#Create a neighbor joining tree based on these distances
nj.tree.b <- bionj(seq.dist.raw.b)

#Identify the outgroup
outgroup.b <- match("RSHA", nj.tree.b$tip.label)

#Root the tree (compare the outgroup to everything else, i.e. the ingroup)
nj.rooted.b <- root(nj.tree.b, outgroup.b, resolve.root = TRUE)

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


Neighbor Joining Tree
RSHA
RTHU
MODO
DOWO
ACFL
REVI
SOVI
AMGO
BAWW
PIWA
AMRO
HETH
2