11. Worksheet: Phylogenetic Diversity - Traits

Joshua Jones; Z620: Quantitative Biodiversity, Indiana University

04 May, 2021

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, April 28th, 2021 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())
dev.off ()
```

```
## null device
##           1
```

```
getwd()
```

```
## [1] "C:/Users/joshu/quantbio/QB2021_Jones/2.Worksheets/11.PhyloTraits"
```

```
setwd("C:/Users/joshu/quantbio/QB2021_Jones/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)
  }
}
```

```
## Warning: package 'ape' was built under R version 4.0.5
```

```
## Warning: package 'seqinr' was built under R version 4.0.5
```

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

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

```
## Warning: package 'phylobase' was built under R version 4.0.5
```

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

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

```
## Warning: package 'adephylo' was built under R version 4.0.5
```

```
## Warning: package 'ade4' was built under R version 4.0.5
```

```
## Registered S3 method overwritten by 'spdep':
##   method   from
##   plot.mst ape

## Warning: package 'geiger' was built under R version 4.0.5

## Warning: package 'picante' was built under R version 4.0.5

## Warning: package 'vegan' was built under R version 4.0.4

## Warning: package 'permute' was built under R version 4.0.4

##
## Attaching package: 'permute'

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

## This is vegan 2.5-7

##
## Attaching package: 'nlme'

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

## Warning: package 'caper' was built under R version 4.0.5

## Warning: package 'phylolm' was built under R version 4.0.5

## Warning: package 'pmc' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.4

## Warning: package 'dplyr' was built under R version 4.0.4

##
## 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

## Warning: package 'phangorn' was built under R version 4.0.5

##
## Attaching package: 'phangorn'

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

## Warning: package 'pander' was built under R version 4.0.5
```

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

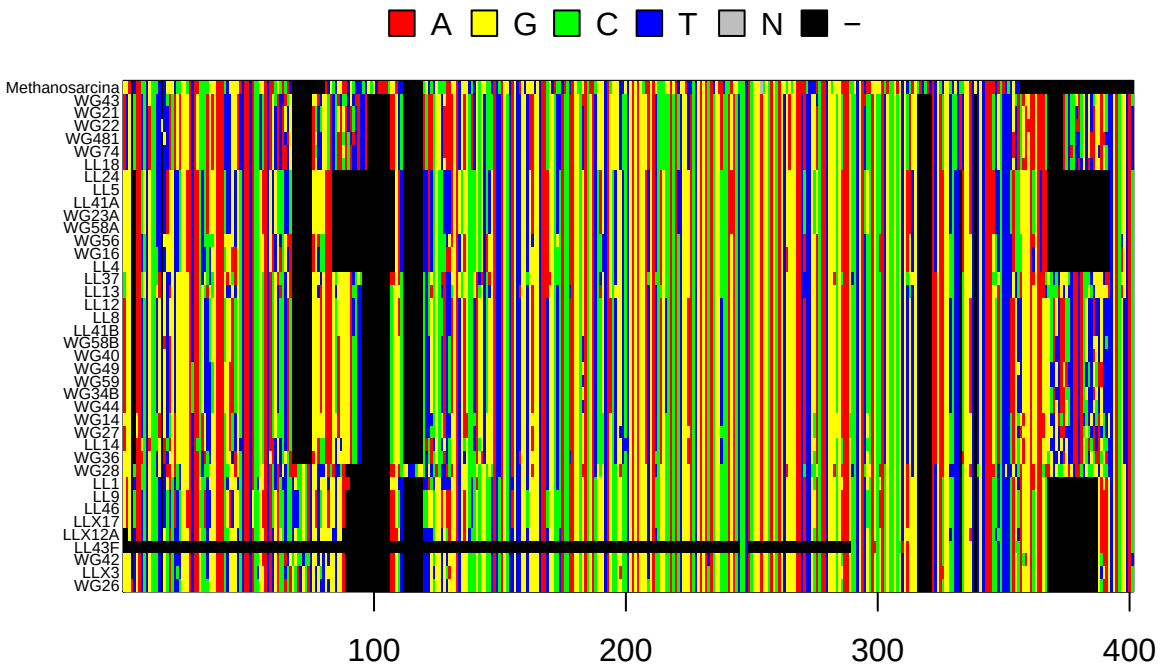
Question 1: Using your favorite text editor, compare the `p.isolates.fasta` file and the `p.isolates.afa` file. Describe the differences that you observe between the two files.

> ***Answer 1***: the .afa file is in all caps and has dashes, which I presume means that it has aligned sequences and these dashes are standing in for sections of DNA that are present in other samples but absent within the sample with the 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.

```
#Reading alignment file into R
read.aln <- read.alignment("data/p.isolates.afa", format = 'fasta')

#Converting alignment into a DNAbin object
p.DNAbin <- as.DNAbin(read.aln)
```

```
#Selecting a region to visualize
window <- p.DNAbin[,100:500]

#Visualizing
image.DNAbin(window, cex.lab = .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 be appropriate for phylogenetic inference and why?

Answer 2a: The longest sequence reads seems to be 1500bps since the DNAbin object contains 1500 total columns and I assume each column represents a different site along the longest sequence
Answer 2b: The most appropriate region would be one that is represented across all samples but isn't so conserved that there is no diffrentiation to draw conclusions about divergence from, such as the 16S rRNA gene in this case.

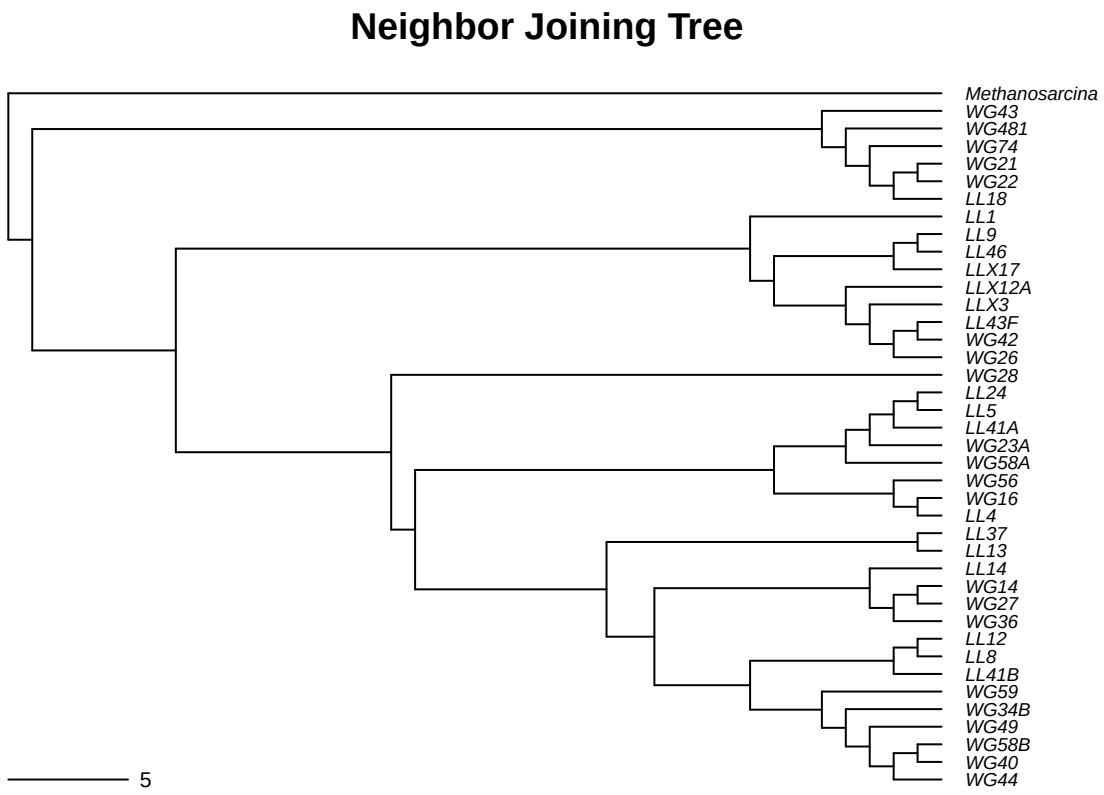
4) MAKING A PHYLOGENETIC TREE

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.

```
# Calculating the distance matrix
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)

# Create neighbor joining tree
nj.tree <- bionj(seq.dist.raw)

# Define "Metanosarcina" as the outgroup and rooting the tree
outgroup <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)

# Plotting 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 =
add.scale.bar(cex = .7)
```



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

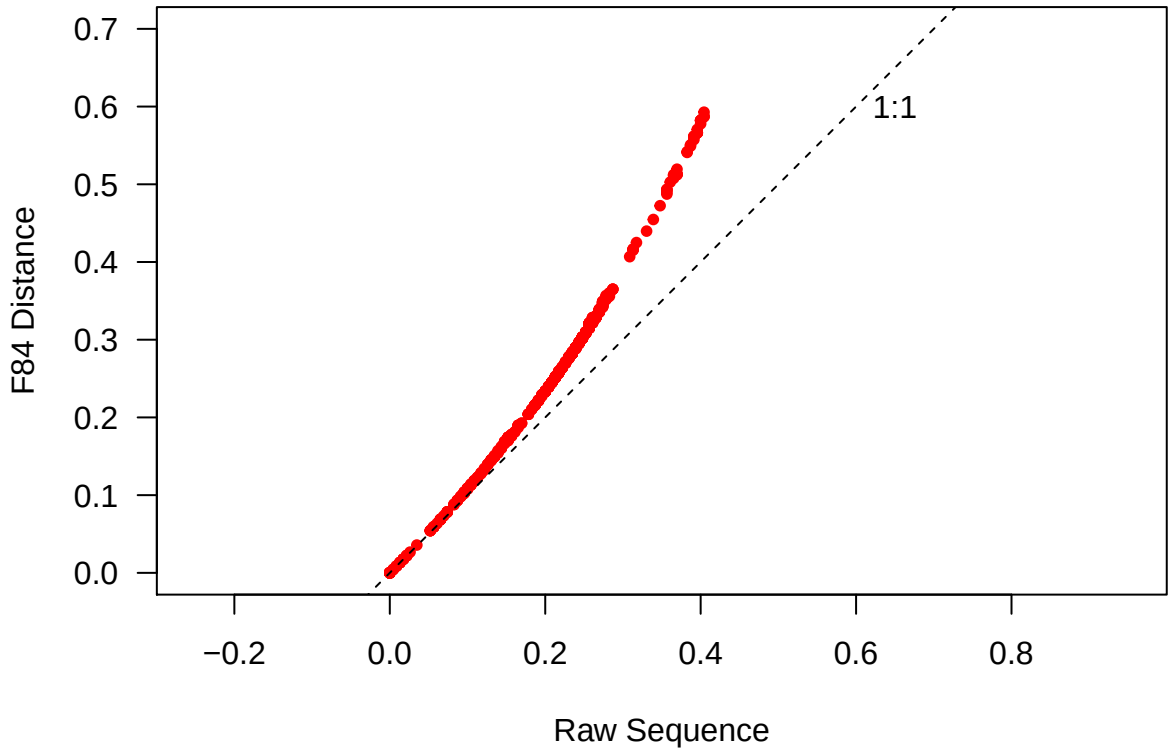
> ***Answer 3***: Neighbor Joining is far fasted than the Maximum Likelihood methods and far simpler than the Bayesian methods but simply grouping into nodes iteratively doesn't necessarily result

in the most accurate tree as there are also values to the weights and parsimony considered in more complex methods

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.

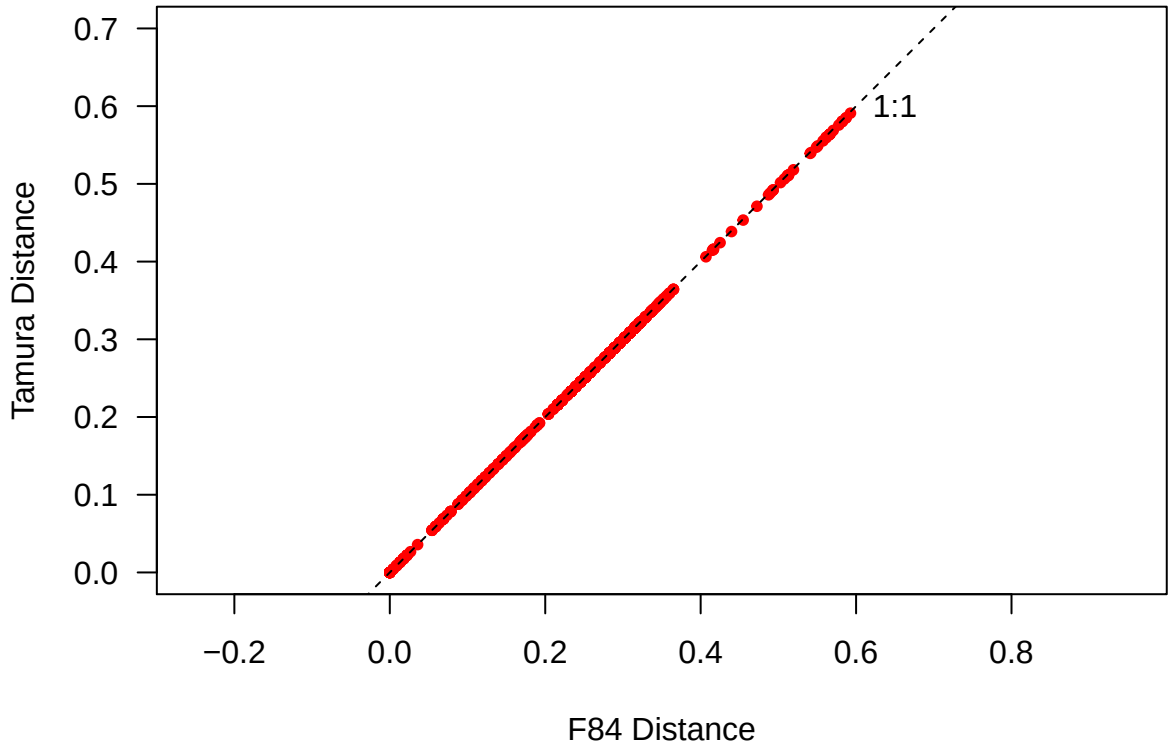
```
# Making distance matrix using Felsenstein 84 model:
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise.deletion = FALSE)

# Creating saturation plot comparing Raw and Felsenstein 84 models:
par(mar = c(5,5,2,1) + .1)
plot(seq.dist.raw, seq.dist.F84,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, .7), ylim = c(0, .7),
     xlab = "Raw Sequence", ylab = "F84 Distance")
abline(b = 1, a = 0, lty = 2)
text(.65, .6, "1:1")
```



In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,

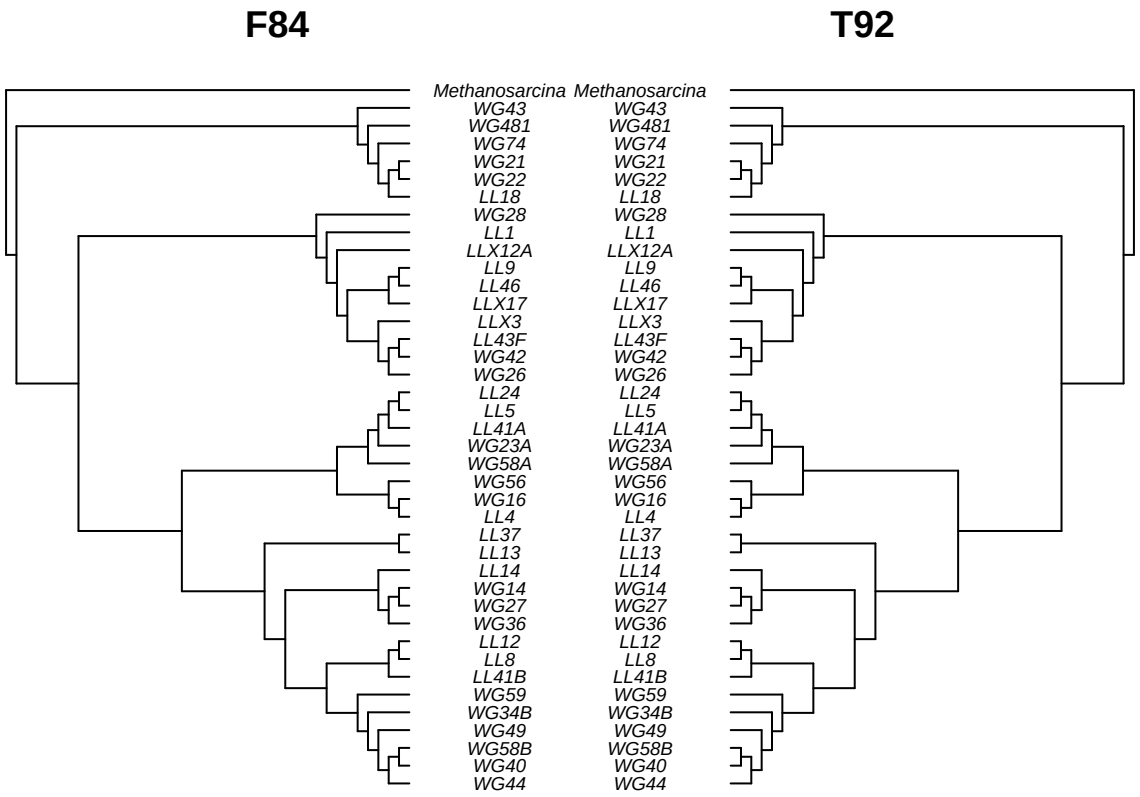
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5. be sure to format, add appropriate labels, and customize each plot.

```
# Creating distance for Tamura model (T92)
seq.dist.T92 <- dist.dna(p.DNAbin, model = "T92", pairwise.deletion = FALSE)

# Saturation plat comparing F84 to T92
par(mar = c(5,5,2,1) + .1)
plot(seq.dist.F84, seq.dist.T92,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, .7), ylim = c(0, .7),
     xlab = "F84 Distance", ylab = "Tamura Distance")
abline(b = 1, a = 0, lty = 2)
text(.65, .6, "1:1")
```




```
## Making cophylogenetic plot  comparing topologies of F84 and T92

# Neighbot Joining
F84.tree <- bionj(seq.dist.F84)
T92.tree <- bionj(seq.dist.T92)

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

# Rooting trees
```

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

# Cophylogenetic plot
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 = .5, cex = .6, label.offset = 2, main = "F84")

par(mar = c(1, 0, 2, 1))
plot.phylo(T92.rooted, type = "phylogram", direction = "left", show.tip.label = TRUE, use.edge.length =
```



Question 4:

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

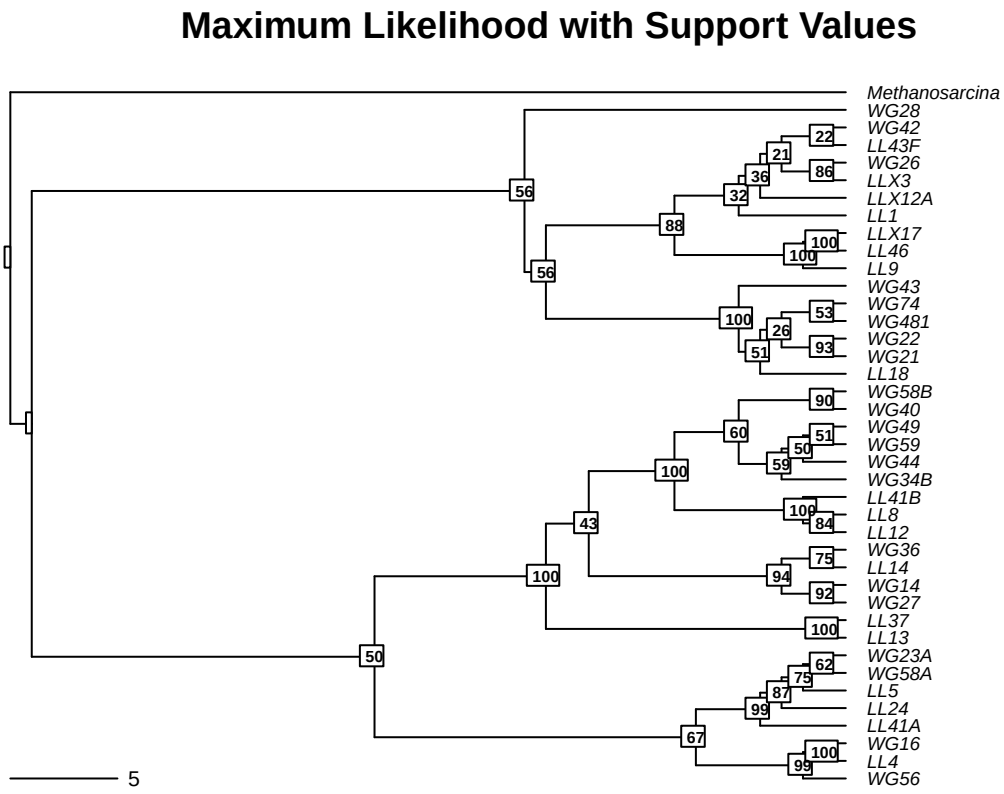
Answer 4a: The Tamura model factors in the higher probability of purine to purine or pyrimidine to pyrimidine mutations than purine to pyrimidine and vice-versa as well as the overall G+C content of across sequences. ***Answer 4b***: There is no difference from switching between

these two models. ***Answer 4c***: Since the increased complexity of the T92 model yields the same topology we can conclude that across these microbes there isn't a large effect of differential mutation rates across purines and pyrimidines or GC count.

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

```
# Reading in maximum likelihod phylogenetic tree
ml.bootsrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")

# Plotting bootstrap support values onto the tree
par(mar = c(1,1,2,1) + .1)
plot.phylo(ml.bootsrap, type = "phylogram", direction = "right", show.tip.label = TRUE, use.edge.length
          main = "Maximum Likelihood with Support Values")
add.scale.bar(cex = .7)
nodelabels(ml.bootsrap$node.label, font = 2, bg = "white", frame = "r", cex = .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: This tree seems to be considerably topologicallty dissimilar to the original neighbor-joining tree, this is reasonable since the maximum likelihood tree is a lot more complex, and theoretically more accurate, than the neighbor-joining one. ***Answer 5b***: Bootstrapping the tree allows the viewer to assess the statistical confidence in each node on the tree and understand which relationships are and are not likely to be correct. ***Answer 5c***: The bootstrap values tell you how statistically probable it is that that node is correct based on random resampling of the data. ***Answer 5d***: There are quite a few branches with low support, but the branches closest to WG42 and LL43F are the least supported of the whole tree. ***Answer 5e***: We should be particularly skeptical of those branches, as well as some others with fairly low values.

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.

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

# Standardizing 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.

```
# Calculating Maximum Growth Rate
umax <- apply(p.growth, 1, max)

# Creating function to calculate 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)
}

# Calculating Niche Breadth
nb <- as.matrix(levins(p.growth.std))
rownames(nb) <- row.names(p.growth)
colnames(nb) <- c("NB")
```

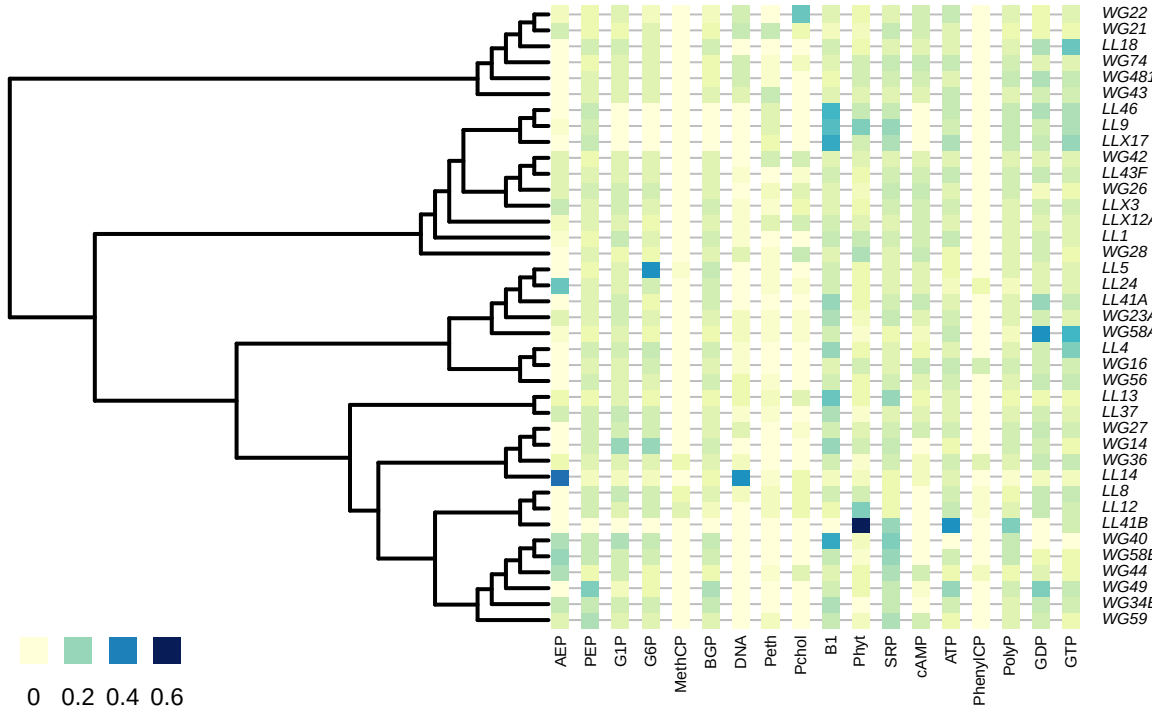
C. Visualizing Traits on Trees

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.

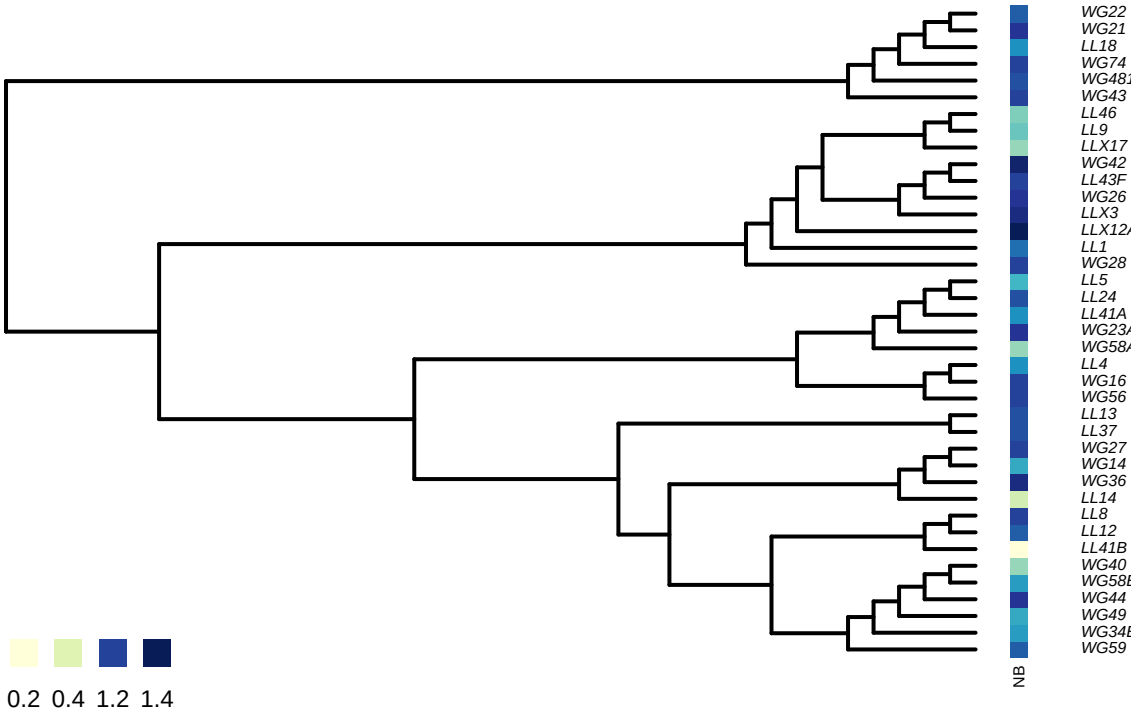
```
# Creating Neighbor Joining tree
nj.tree <- bionj(seq.dist.F84)

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

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

#Removing Outgroup
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

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

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

# Map P traits onto phylogeny
par(mar = c(1,1,1,1) + .1)
x <- phylo4d(nj.rooted, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = .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 = .5, cex.legend = 1.5, center = FALSE)
```




```
# Mapping Niche Breadth trait onto phylogeny
par(mar = c(1,1,1,1) + .1)
x <- phylo4d(nj.rooted, nb)
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE,
              cex.label = .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 = .9, 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: I would hypothesize that specialist have a high growth rate for a few select phosphorous resources and generalists have a moderate growth rate for a broader range of phosphorous resources. ***Answer 6b***: I would expect them to be negatively correlated if my hypothesis is correct, with strains with high niche breadth having a lower maximum growth rate and strains with low niche breadth having a high maximum growth rate.

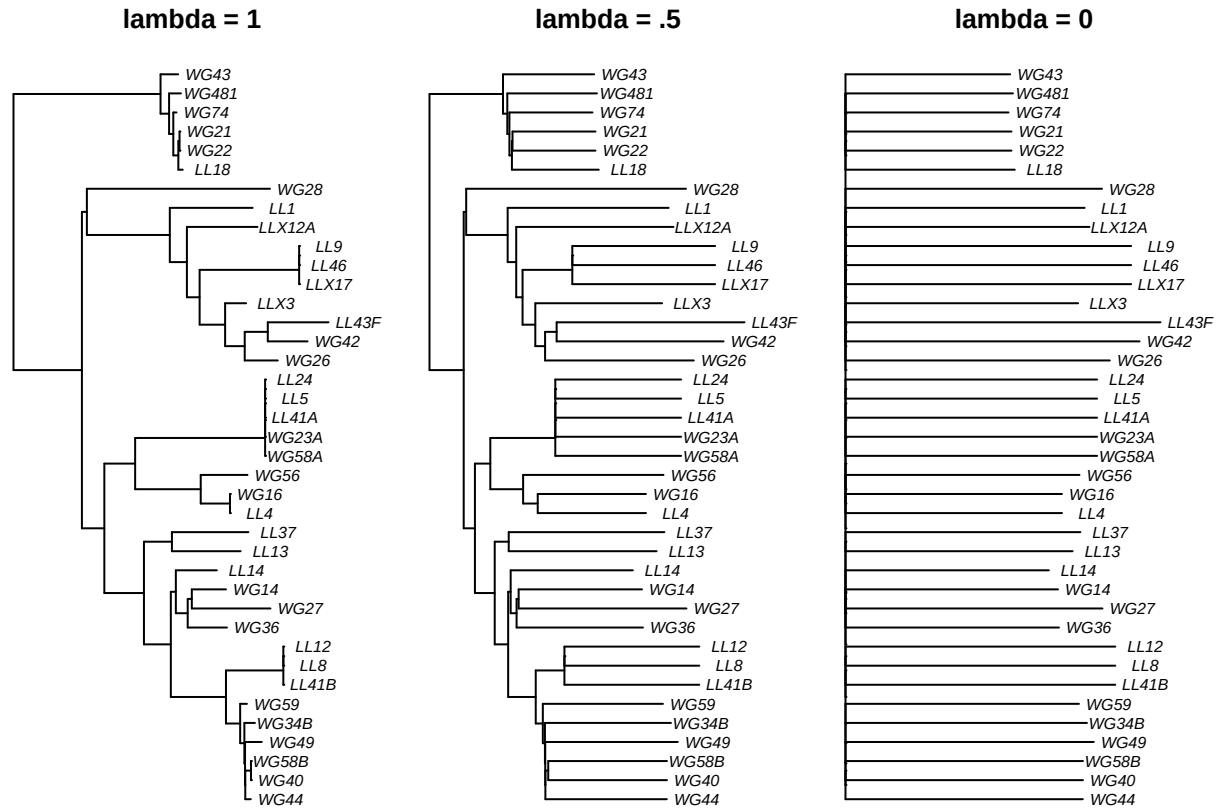
6) HYPOTHESIS TESTING

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.

```
# Creating rescaled trees
nj.lambda.5 <- rescale(nj.rooted, "lambda", .5)
nj.lambda.0 <- rescale(nj.rooted, "lambda", 0)
```

```
# Plotting trees
layout(matrix(c(1,2,3), 1,3), width = c(1,1,1))
par(mar = c(1,.5,2,.5) + .1)
plot(nj.rooted, main = "lambda = 1", cex = .7, adj = .5)
plot(nj.lambda.5, main = "lambda = .5", cex = .7, adj = .5)
plot(nj.lambda.0, main = "lambda = 0", cex = .7, adj = .5)
```



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

```
# Comparing trees
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 = 59
##  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 = 0.106395
##  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
##  number of iterations with same best fit = 85
##  frequency of best fit = 0.85
##
##  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 both trees is 0 ***Answer 7b***: And the AIC scores are identical
> ***Answer 7c***: This result suggests that there is no phylogenetic signal for niche breadth

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 tree branch-lengths
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7

# Calculating Blomberg's K for P resources
p.phylosignal <- matrix(NA, 6, 18)
colnames(p.phylosignal) <- colnames(p.growth.std)
rownames(p.phylosignal) <- c("K", "PIC.var.obs", "PIC.var.mean", "PIC.var.P", "PIC.var.z", "PIC.P.BH")

for (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)
}
```

```
## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
```

```
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used

## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used
```

```
# Using Benjamini-Hochberg method to correct for false discovery rate
p.phylosignal[6,] <- round(p.adjust(p.phylosignal[4,], method = "BH"), 3)

# Calculating Blomberg's K for niche breadth
signal.nb <- phylosignal(nb, nj.rooted)
```

```
## Warning in if (dataclass == "data.frame") {: the condition has length > 1 and
## only the first element will be used
```

```
signal.nb
```

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 3.427719e-06         49966.78              50295.92          0.535
##   PIC.variance.Z
## 1    -0.01585404
```

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 no significant phylogenetic signal for niche breadth but there is for several of the P resources. ***Answer 8b***: The K values are all 0 which suggest that the traits are heavily overdispersed

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
# Turning the Continuous growth data in categorical data
p.growth.pa <- as.data.frame((p.growth > .01) * 1)

#Adding column for isolate name
p.growth.pa$name <- rownames(p.growth.pa)

# Combining the tree and trait data
p.traits <- comparative.data(nj.rooted, p.growth.pa, "name")

# Calculating D on three phosphorus traits
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 :  0.6108828
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.035
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.004
```

```r
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 :  0.1429074
## Probability of E(D) resulting from no (random) phylogenetic structure :  0
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.311
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  PEP
##   Counts of states:  0 = 1
##                      1 = 38
##   Phylogeny :  nj.rooted
```

```
##   Number of permutations :  1000
##
## Estimated D :  -0.2823354
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.287
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.525
```

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.

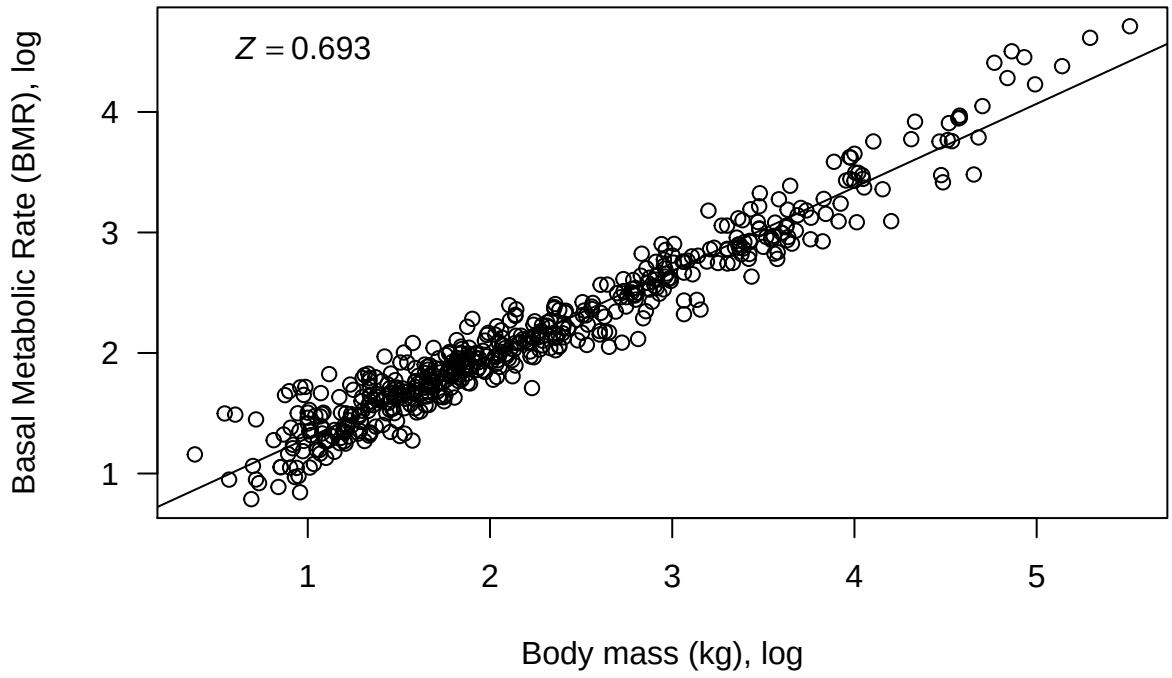
Answer 9a:
Growth on DNA has a D value of .598 which means it seems to be moderately overdispersed and isn't likely to be resulting from Brownian or random phylogenetic structure Growth on cAMP has a D value of .159 which means it is slightly overdispersed and also isn't likely to be resulting from Brownian or random phylogenetic structure and growth on PEP has a D value of -.253 which is slightly clustered but doesn't seem to be significantly explained by Brownian or random phylogenetic structure ***Answer 9b***: These results are consistent with Blomberg's K where we can see that the traits seem to be overdispersed but here none of these seem to singnificant in while two of them are significant in the prior analysis ***Answer 9c***: There are a few differences that could lead to this change of results, a main one is most likely the fact that this analysis is using catagorical data instead of the original continuous ones.

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, 3. Fit a phylogenetic regression to the trait dataset, taking into account the mammal supertree

```
# Loading mammal phylogeny and trait dataset
mammal.Tree <- read.tree("./data/mammal_best_super_tree_fritz2009.tre")
mammal.data <- read.table("./data/mammal_BMR.txt", sep = "\t", header = TRUE)

# Selecting variables to analyze
mammal.data <- mammal.data[,c("Species", "BMR_.mlO2.hour.", "Body_mass_for_BMR_.gr.")]
mammal.species <- array(mammal.data$Species)

# Selecting the tips of the mammal tree that are in the dataset
pruned.mammal.tree <- drop.tip(mammal.Tree, mammal.Tree$tip.label[-na.omit(match(mammal.species, mammal

# Selecting 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,]

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

```
# Running a 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 = 1,
     xlab = "Body mass (kg), log",
     ylab = "Basal Metabolic Rate (BMR), log")
abline(a = fit$coefficients[1], b = fit$coefficients[2])
b1 <- round(fit$coefficients[2],3)
eqn <- bquote(italic(Z) == .(b1))

# Plotting slope
text(.5, 4.5, eqn, pos = 4)
```

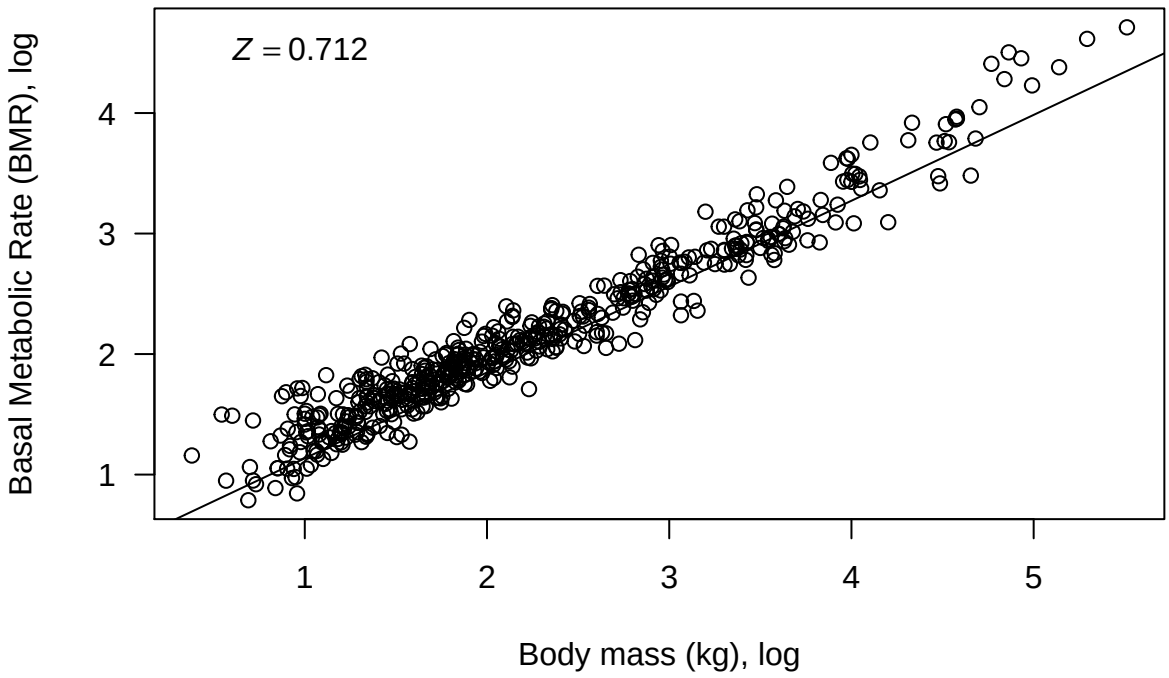



```
# Running a phylogeney-corrected regression with no bootstrap replicates
fit.phy <- phylolm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), data = pruned.mammal.data, pi
```

```
## Warning in phylolm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), :
## will drop from the tree 4502 taxa with missing data
```

```
plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.),
     log10(pruned.mammal.data$BMR_.mlO2.hour.), las = 1,
     xlab = "Body mass (kg), log",
     ylab = "Basal Metabolic Rate (BMR), 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))

# Plotting slope
text(.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: We have to correct for shared evolutionary history because our analysis is based off of the assumption that all the traits across individuals are independent and that cannot be valid unless that shared evolutionary history is considered in the analysis. ***Answer 10b***: The main difference is that instead of assuming that the risiduals are random we assume that they covary with the shared evolutionary history between the tips of the phylogeny. ***Answer 10c***: Both of the slopes show a strong relationship between ml O2 and BMR but accounting for shared evolutionary history seems to of worsened the fit slightly (0.019). ***Answer 10d***:

7) SYNTHESIS

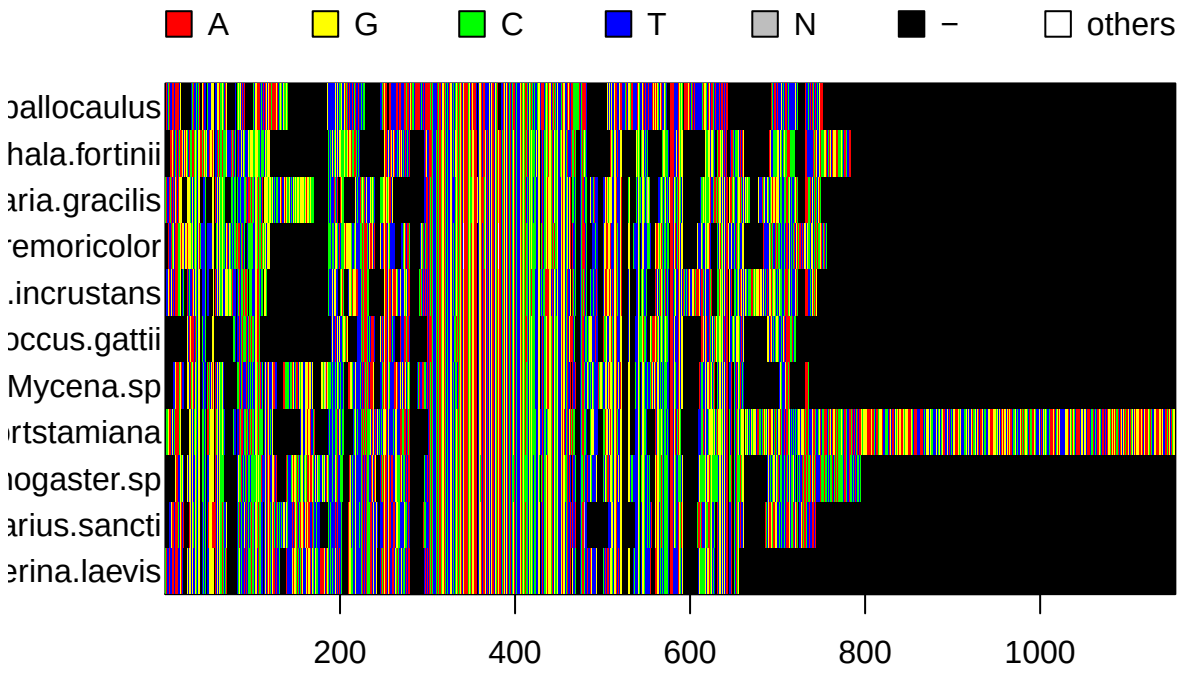
Work with members of your Team Project to obtain reference sequences for 10 or more taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the

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.

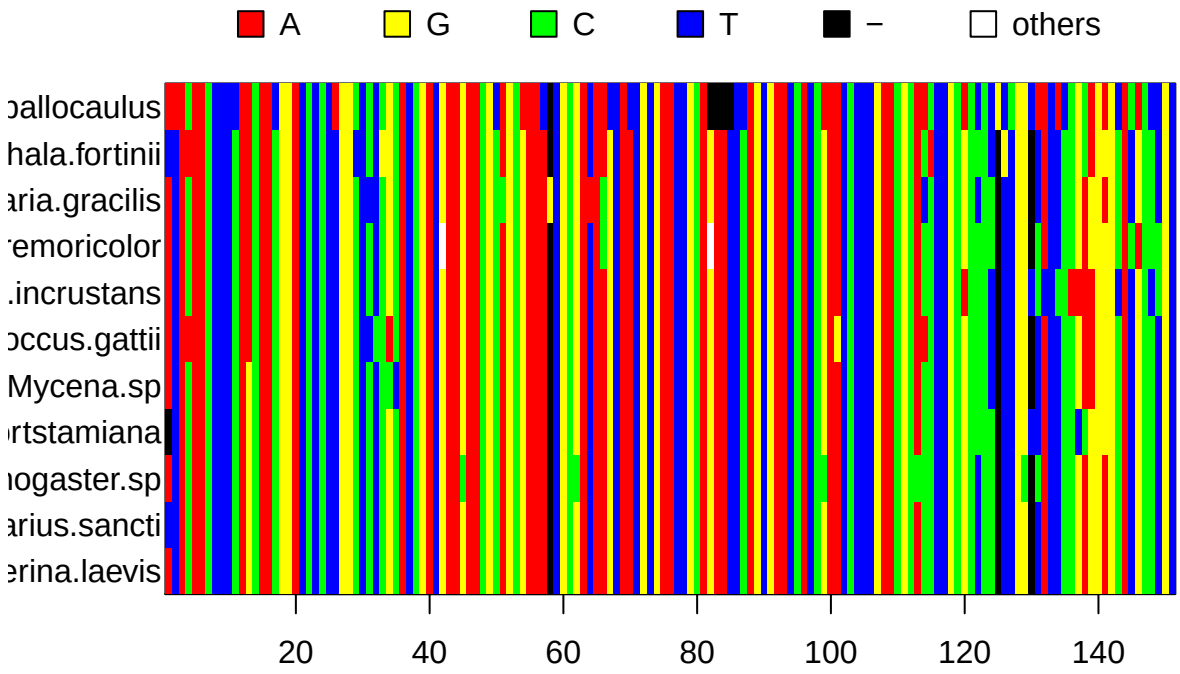
But before your team proceeds, you need to give some thought to which gene you want to focus on. For microorganisms like the bacteria we worked with above, many people use the ribosomal gene (i.e., 16S rRNA). This has many desirable features, including it is relatively long, highly conserved, and identifies 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?

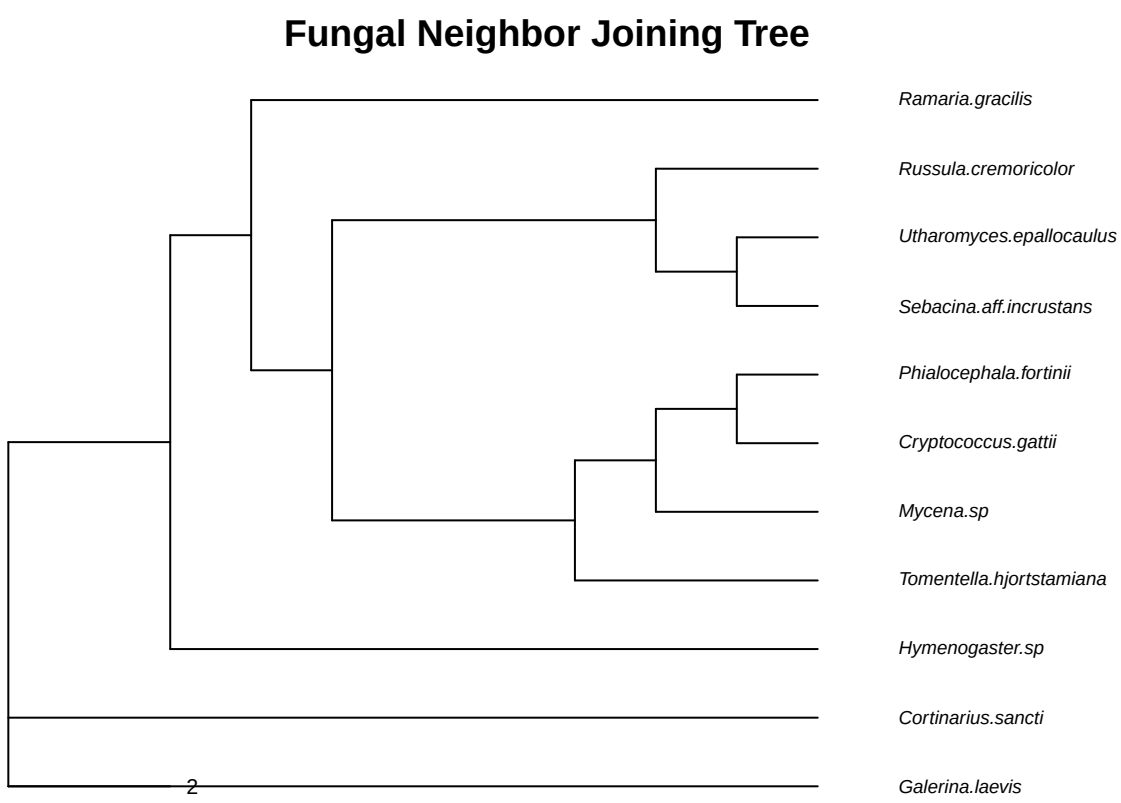
```
fungal.aln <- read.alignment(file = "./fungal2.aln.fasta", format = "fasta")

f.DNAbin <- as.DNAbin(fungal.aln)

image.DNAbin(f.DNAbin)
```




```
window <- f.DNAbin[,300:450]

image.DNAbin(window)
```




```
f.seq.dist <- dist.dna(window, model = "T92", pairwise.deletion = FALSE)
f.nj.tree <- bionj(f.seq.dist)

par(mar = c(1,1,2,1) + .1)
plot.phylo(f.nj.tree, main = "Fungal Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, direc
add.scale.bar(cex = .7)
```



Without going indepth there are a few clear irregularities between this tree and the tree expected from the evolutionary history of these organisms. As one example, R. gracilis, R. cremoricolor, and S. incrustans are within the same class and therefore should be more closer related to each other than to U. epallocaulus, which is in a different division. These, and other, irregularities could be solved with a more complex program for topological determinant, or perhapse a more difinitive DNA section for determining distance. I originally attempted to use ITS1 for the alignment but the quality was far too bad, so I switched to using the entire rRNA, ITS1, and ITS2 region, where there was a good section that comprised a mix of what I think was the ITS2 and RNA regions. However, this isn't a problem for our project since I've decided to not use phylogeny for our analysis of the fungal communities. Mainly because it would require accessing and working with the raw sequences, but also because I don't know the functional relevance of relatedness in fungus (ie, does relatedness have any functional relevance in related fungi)