11. Worksheet: Phylogenetic Diversity - Traits

Eli Graber; Z620: Quantitative Biodiversity, Indiana University

29 April, 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 28$^{\text{th}}$, 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())
getwd()
```

```
## [1] "/Users/eligraber/GitHub/QB2021_Graber/2.Worksheets/11.PhyloTraits"
```

```
setwd("~/GitHub/QB2021_Graber/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.

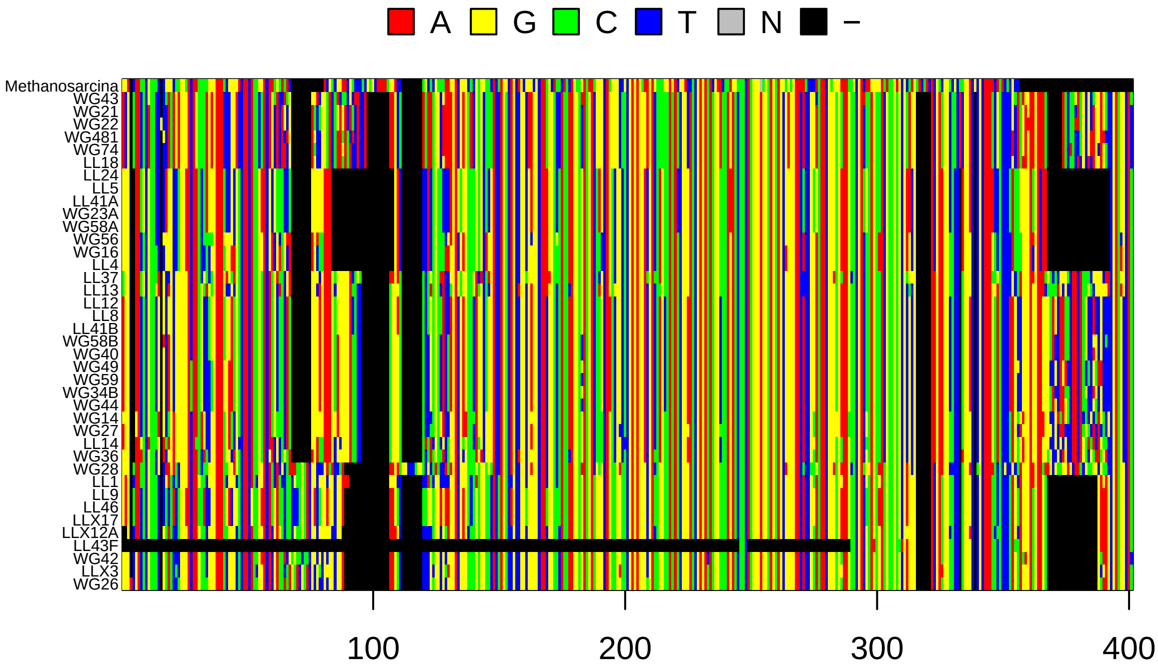
> ***Answer 1***: The fasta file is far more consolitated and doesn't iclude blank code areas that afa has. The code regions are in a different order. There is a difference in capitalization.

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

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

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

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

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

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

##
## 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

##
## 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
```

```
read.aln <- read.alignment(file = "./data/p.isolates.afa", format = "fasta")
p.DNAbin <- as.DNAbin(read.aln)
window <- p.DNAbin[, 100:500]
p.DNAbin <- p.DNAbin[, 500:650]
image.DNAbin(window, cex.lab = 0.50)
```



Question 2: Make some observations about the `muscle` alignment of the 16S rRNA gene sequences for our bacterial isolates and the outgroup, *Methanosarcina*, a member of the domain Archaea. Move along the alignment by changing the values in the `window` object.

a. Approximately how long are our sequence reads?

b. What regions do you think would are appropriate for phylogenetic inference and why?

> ***Answer 2a***: It is reading a 400 character window.
> ***Answer 2b***: So the 0-80, 330-380, and/or 120-170 all seem like regions that would be appropriate as they seems to be complex and differing across samples that could lead to interesting comparision and inferences.

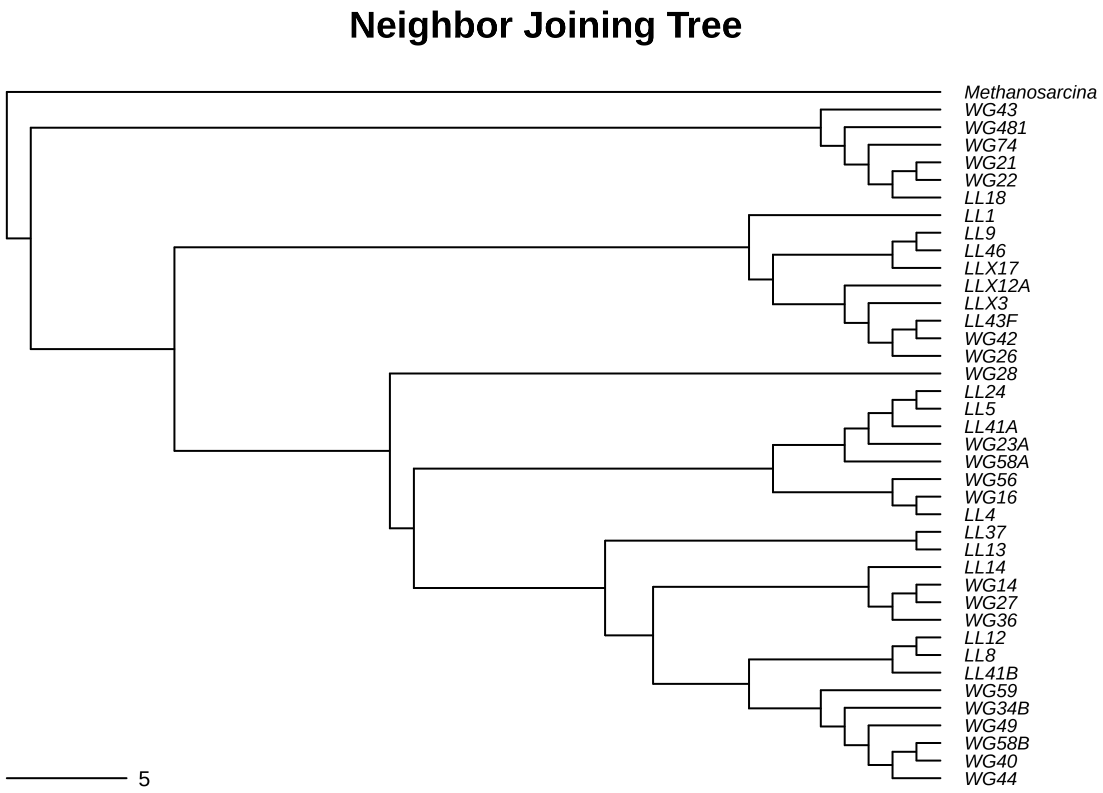
4) MAKING A PHYLOGENETIC TREE

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.

```
p.DNAbin <- as.DNAbin(read.aln)
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)
nj.tree <- bionj(seq.dist.raw)
outgroup <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(nj.rooted, main = "Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, direction =
add.scale.bar(cex = 0.7)
```



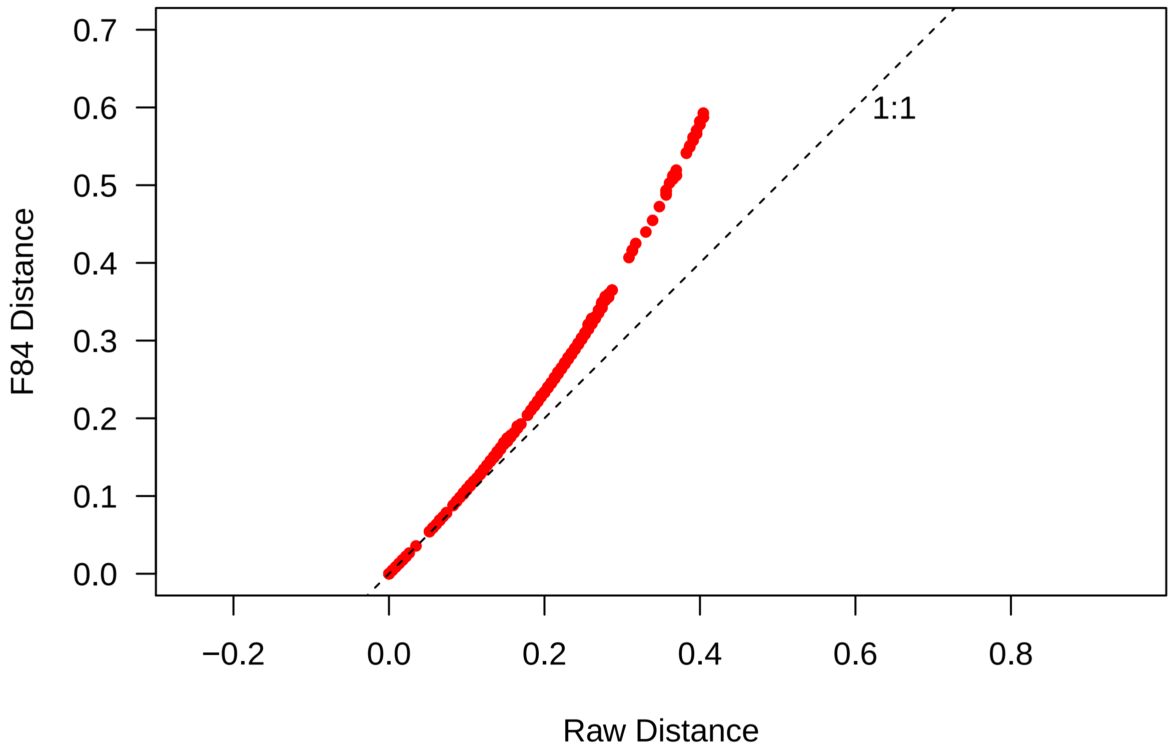
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: A disadvantage is that it does not give us a boot strap value. An advantge is it gives us an idea of how closely these related these sequences are, gives us an idea of length of time, and properly lists the outgroup

B) SUBSTITUTION MODELS OF DNA EVOLUTION

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

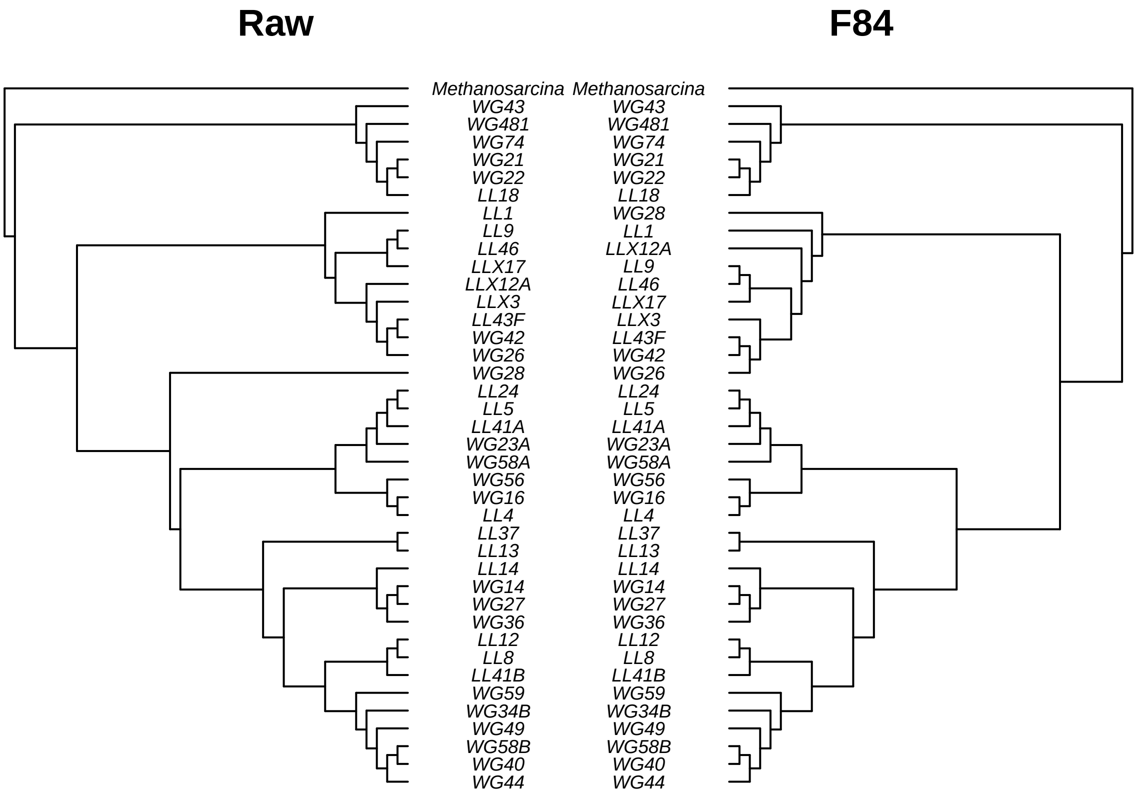
```
seq.dist.F84 <- dist.dna(p.DNAbin, model = "F84", pairwise.deletion = FALSE)
par(mar = c(5,5,2,1) + 0.1)
plot(seq.dist.raw, seq.dist.F84,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7), ylim = c(0, 0.7),
     xlab = "Raw Distance", ylab = "F84 Distance")
abline(b = 1, a =0, lty = 2)
text(0.65, 0.6, "1:1")
```



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.

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

```
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root=TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root=TRUE)
layout(matrix(c(1,2), 1,2), width = c(1,1))
par(mar = c (1,1,2,0))
plot.phylo(raw.rooted, type = "phylogram", direction = "right", show.tip.label=TRUE,
           use.edge.length = FALSE, adj = 0.5, cex = 0.6, label.offset = 2, main = "Raw")

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

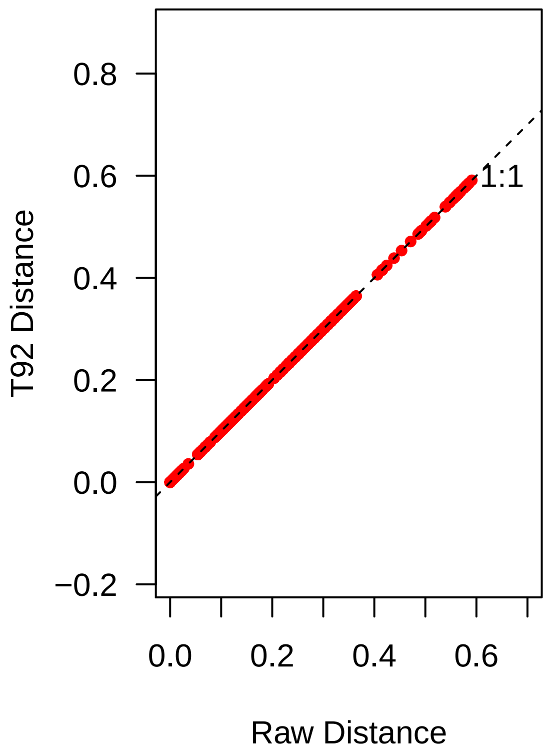


```
dist.topo(raw.rooted, F84.rooted, method = "score")
```

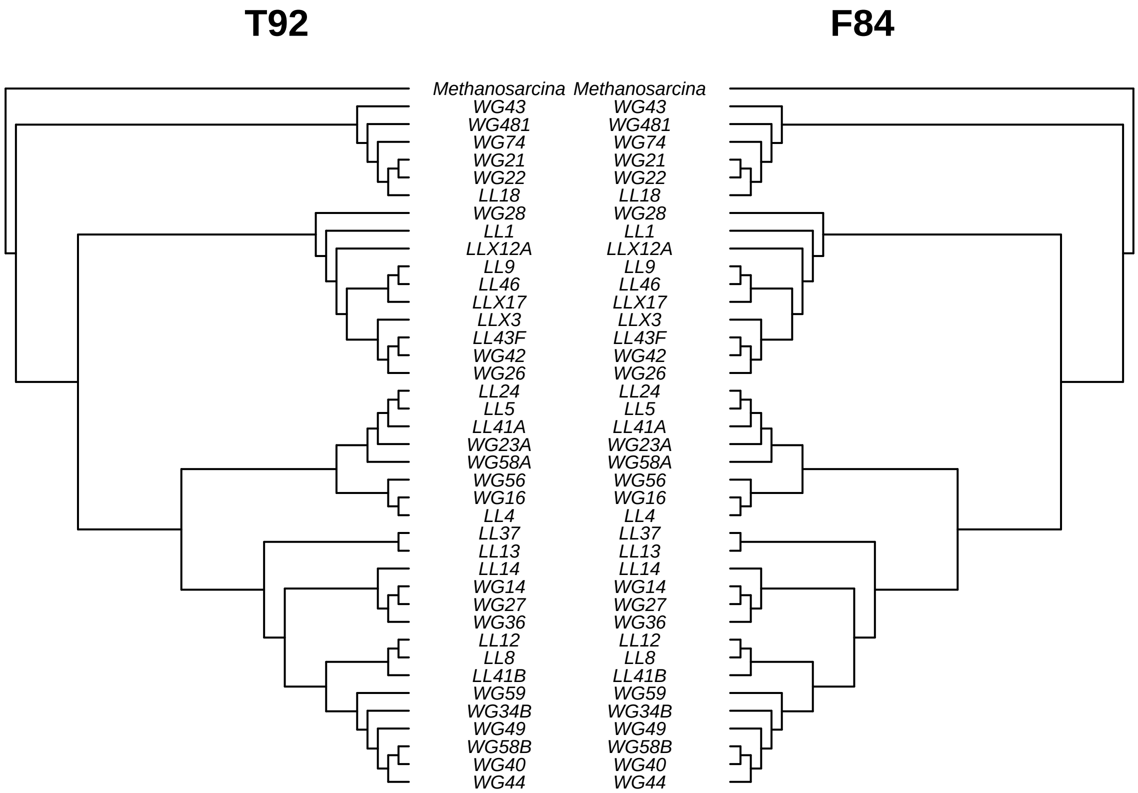
```
##            tree1
## tree2 0.04387426
```

```
seq.dist.T92 <- dist.dna(p.DNAbin, model = "T92", pairwise.deletion = FALSE)

par(mar = c(5, 5, 2, 1) + 0.1)
plot(seq.dist.T92, seq.dist.T92, pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7), ylim = c(0,
abline(b = 1, a =0, lty = 2)
text(0.65, 0.6, "1:1")

T92.tree <- bionj(seq.dist.T92)
```

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

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

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

layout(matrix(c(1,2), 1, 2), width = c(1,1))
```



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

par(mar = c(1, 0, 2, 1))
plot.phylo(F84.rooted, type = "phylogram", direction = "left", show.tip.label = T, use.edge.length = F,
```



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 which is based on the Kimura model which changes things to account for high C:G content and equal frequencies of nucelotides.
Answer 4b: When comparing to raw, F84 are similiar but not excalty the same while the F84 to T92 comparison shows that they are identical.
Answer 4c: Since T92 is identical to F84 is shows that the assumptions made by T92 are accurate to the actual sequences.

C) ANALYZING A MAXIMUM LIKELIHOOD TREE

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

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

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

```
#p.DNAbin.phyDat <- read.phyDat("./data/p.isolates.afa", format = "fasta", type = "DNA")
p_dnabin_phydat <- read.phyDat("./data/p.isolates.afa", format="fasta", type="DNA")
fit <- pml(nj.rooted, data=p_dnabin_phydat[, 500:650])
fitJC <- optim.pml(fit, TRUE)
```

```
## Warning: I unrooted the tree
```

```
## optimize edge weights:  -1543.036 --> -1499.186
## optimize edge weights:  -1499.186 --> -1499.121
## optimize topology:  -1499.121 --> -1482.392
## optimize topology:  -1482.392 --> -1473.938
## optimize topology:  -1473.938 --> -1470.402
## 8
## optimize edge weights:  -1470.402 --> -1470.402
## optimize topology:  -1470.402 --> -1468.915
## optimize topology:  -1468.915 --> -1467.281
## optimize topology:  -1467.281 --> -1467.114
## 3
## optimize edge weights:  -1467.114 --> -1467.114
## optimize topology:  -1467.114 --> -1467.114
## 0
## optimize edge weights:  -1467.114 --> -1467.114
```

```
fitGTR <- optim.pml(fit, model = "GTR", optInv=TRUE, optGamma = TRUE)
```

```
## Warning: I unrooted the tree
```

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

```
## optimize edge weights:  -1544.293 --> -1499.186
## optimize base frequencies:  -1499.186 --> -1495.324
## optimize rate matrix:  -1495.324 --> -1466.451
## optimize invariant sites:  -1466.451 --> -1379.13
## optimize edge weights:  -1379.13 --> -1374.461
## optimize base frequencies:  -1374.461 --> -1373.851
## optimize rate matrix:  -1373.851 --> -1371.697
## optimize invariant sites:  -1371.697 --> -1371.619
## optimize edge weights:  -1371.619 --> -1371.411
## optimize base frequencies:  -1371.411 --> -1371.075
## optimize rate matrix:  -1371.075 --> -1370.93
## optimize invariant sites:  -1370.93 --> -1370.908
## optimize edge weights:  -1370.908 --> -1370.831
## optimize base frequencies:  -1370.831 --> -1370.767
## optimize rate matrix:  -1370.767 --> -1370.743
## optimize invariant sites:  -1370.743 --> -1370.729
## optimize edge weights:  -1370.729 --> -1370.7
## optimize base frequencies:  -1370.7 --> -1370.689
## optimize rate matrix:  -1370.689 --> -1370.686
## optimize invariant sites:  -1370.686 --> -1370.679
## optimize edge weights:  -1370.679 --> -1370.669
## optimize base frequencies:  -1370.669 --> -1370.668
## optimize rate matrix:  -1370.668 --> -1370.667
## optimize invariant sites:  -1370.667 --> -1370.664
## optimize edge weights:  -1370.664 --> -1370.663
```

```
anova(fitJC, fitGTR)
```

```
## Likelihood Ratio Test Table
##   Log lik. Df Df change Diff log lik. Pr(>|Chi|)
## 1  -1467.1 70
## 2  -1370.7 86        16         192.9  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

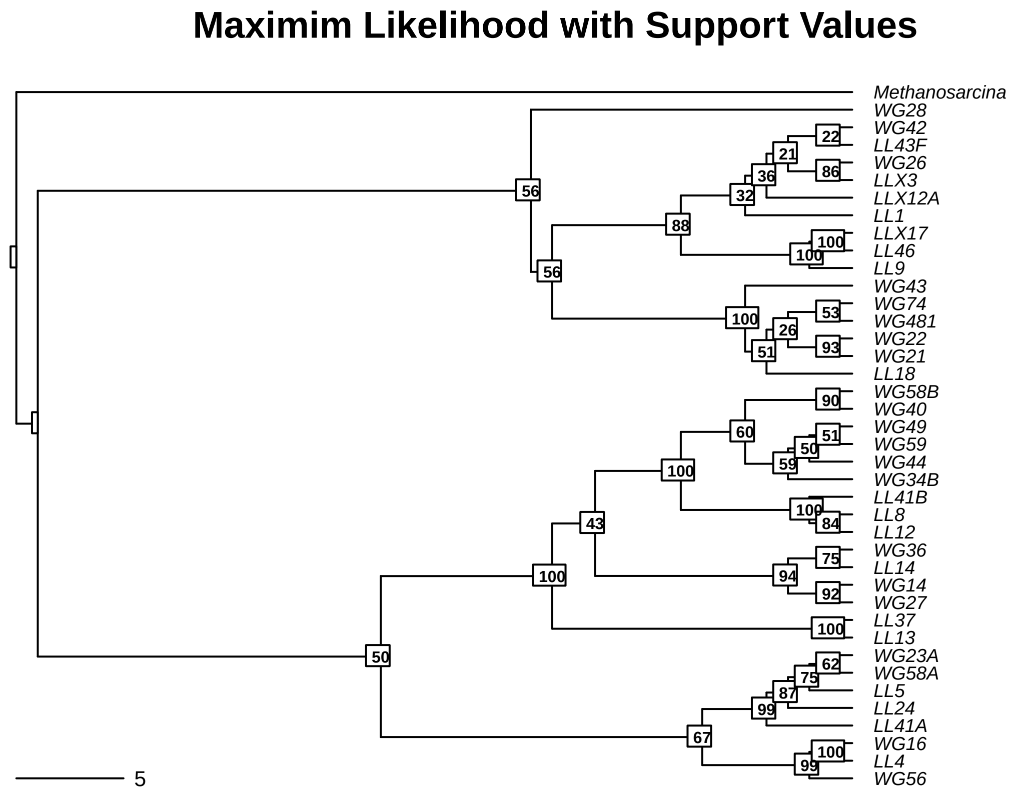
```
AIC(fitJC)
```

```
## [1] 3074.227
```

```
AIC(fitGTR)
```

```
## [1] 2913.325
```

```
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.lengt
           main = "Maximim Likelihood with Support Values")
add.scale.bar(cex = 0.7)
nodelabels(ml.bootstrap$node.label, font = 2, bg = "white", frame = "r", cex = 0.5)
```



Maximim Likelihood with Support Values

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 clustering is very different between the trees and even within the clusters their exact relationships also seem to be different. ***Answer 5b***: To determine how reliable/ how confident we are in the tree produced through producing a bunch of synthetic comparisons by drawing from the same data.
Answer 5c: With bootstrap values, higher is better as it shows that it is a constant reoccuring part of our data. Some of the nodes are very high in value and in turn that means those nodes are vary correct thought it seems like some of the specific branches are less certain. ***Answer 5d***: It seems like the lowest support branches are near the top from WG42 to LLX12A
Answer 5e:

5) INTEGRATING TRAITS AND PHYLOGENY

A. Loading Trait Database

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

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

B. Trait Manipulations

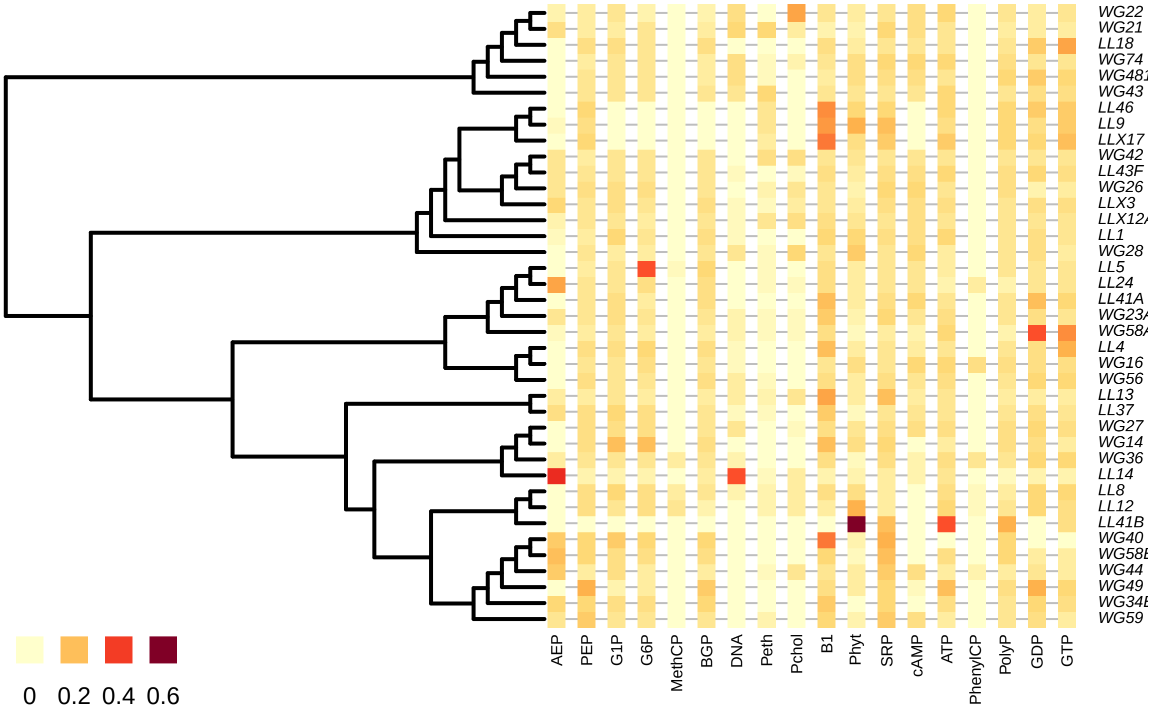
In the R code chunk below, do the following:
1. calculate the maximum growth rate (μ_{max}) of each isolate across all phosphorus types,
2. create a function that calculates niche breadth (*nb*), and
3. use this function to calculate *nb* for each isolate.

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

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

nb <- as.matrix(levins(p.growth.std))

rownames(nb) <- row.names(p.growth)
colnames(nb) <- c("NB")
```

C. Visualizing Traits on Trees

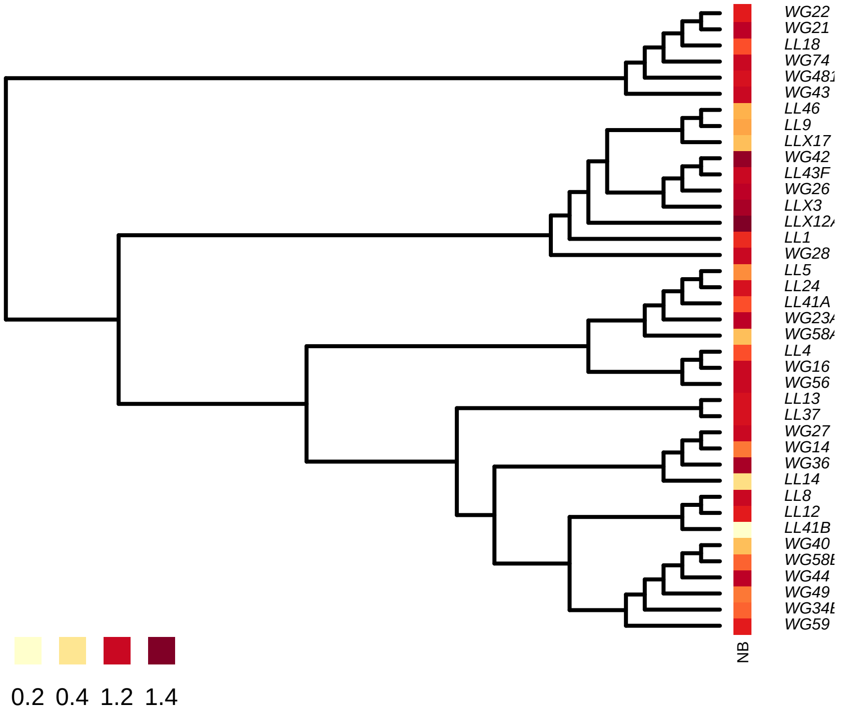
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```
nj.tree <- bionj(seq.dist.F84)
outgroup <- match("Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
mypalette <- colorRampPalette(brewer.pal(9, "YlOrRd"))
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
              col= mypalette(25), pch = 15, cex.symbol = 1.25, ratio.tree = 0.5, cex.legend = 1.5, cent
```



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).

```
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 = FA
```



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***: There will be more speicalist species because they can fill specific niches better while some generalist species will exist as they can fill any niche gaps within an area that the specialist aren't adapted to.
> ***Answer 6b***: That each species would have a high growth rate for at least one phorphorus condition with very low growth rates on the others. The generalist species (which there would be very few of) would have a moderate growth rate across the various phosphorus conditions.

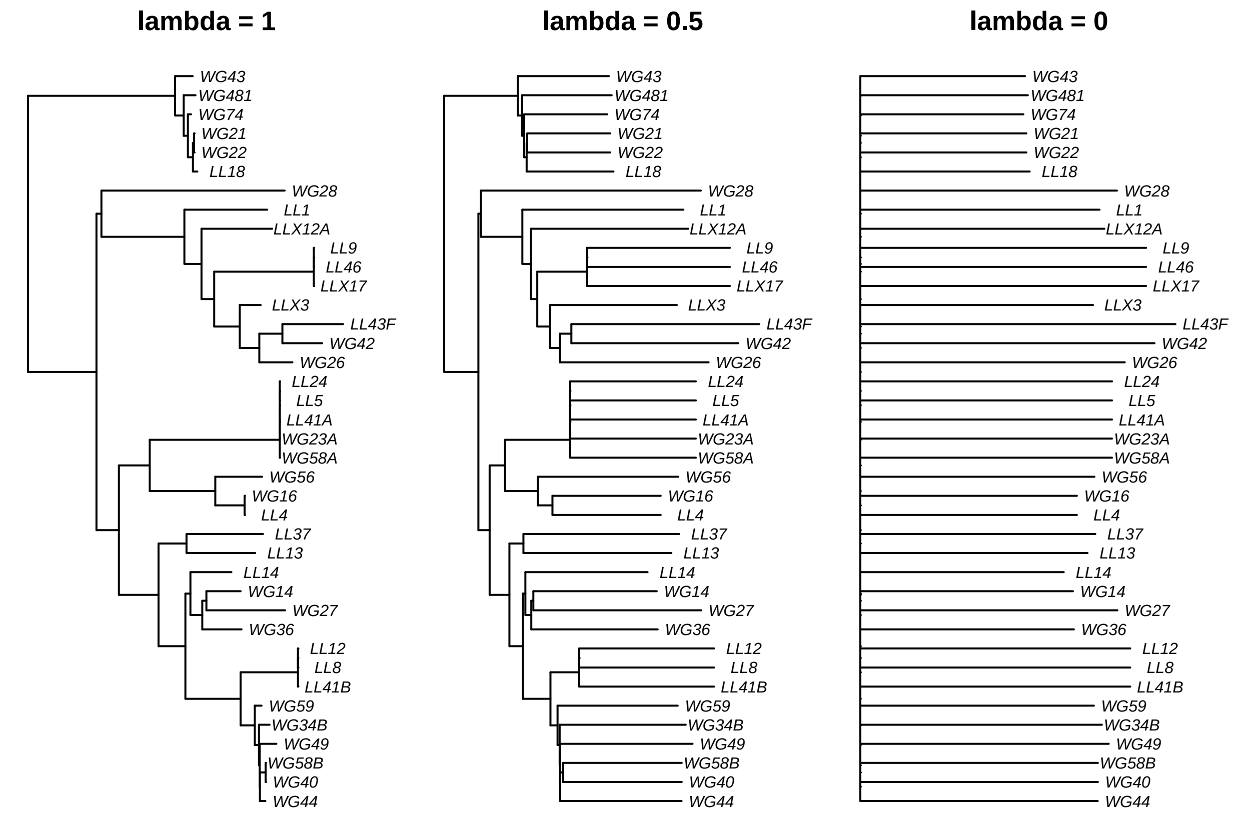
6) HYPOTHESIS TESTING

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.

```
nj.lambda.5 <- rescale(nj.rooted, "lambda", 0.5)
nj.lambda.0 <- rescale(nj.rooted, "lambda", 0)
```

```r
layout(matrix(c(1,2,3), 1,3), widths = 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.

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.020847
##  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 = 54
##  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 = 88
##  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***: Lambda 0 makes it look like all these genetic lines came about at the same time while 1 and 0.5 show clear differentiation, but 0.5 shows it on a shorter timeline than the lambda 1 tree. ***Answer 7b***: The models are considered equivalent because their AIC scores are very similar (-37.304587 and -37.304587). ***Answer 7c***: There is not a phylogenetic signal based on the close proximity of AIC scores and lambda values that were produced.

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.

```
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7
p.phylosignal <- matrix(NA, 6, 18)
colnames(p.phylosignal) <- colnames(p.growth.std)
rownames(p.phylosignal) <- c("K", "PIC.war.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
```

```
p.phylosignal [6, ] <- round(p.adjust(p.phylosignal[4,], method = "BH"), 3)

singal.nb <- phylosignal(nb, nj.rooted)
```

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

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 not as the output is very low ~10ˆ-5. ***Answer 8b***: There is not so this is not applicable.

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.

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

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
```

```
p.growth.pa$name <- rownames(p.growth.pa)

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 :  0.4737065
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.008
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.02
```

```
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 :  0.8662811
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.257
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.011
```

```
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 :  0.5992613
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.037
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.004
```

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?

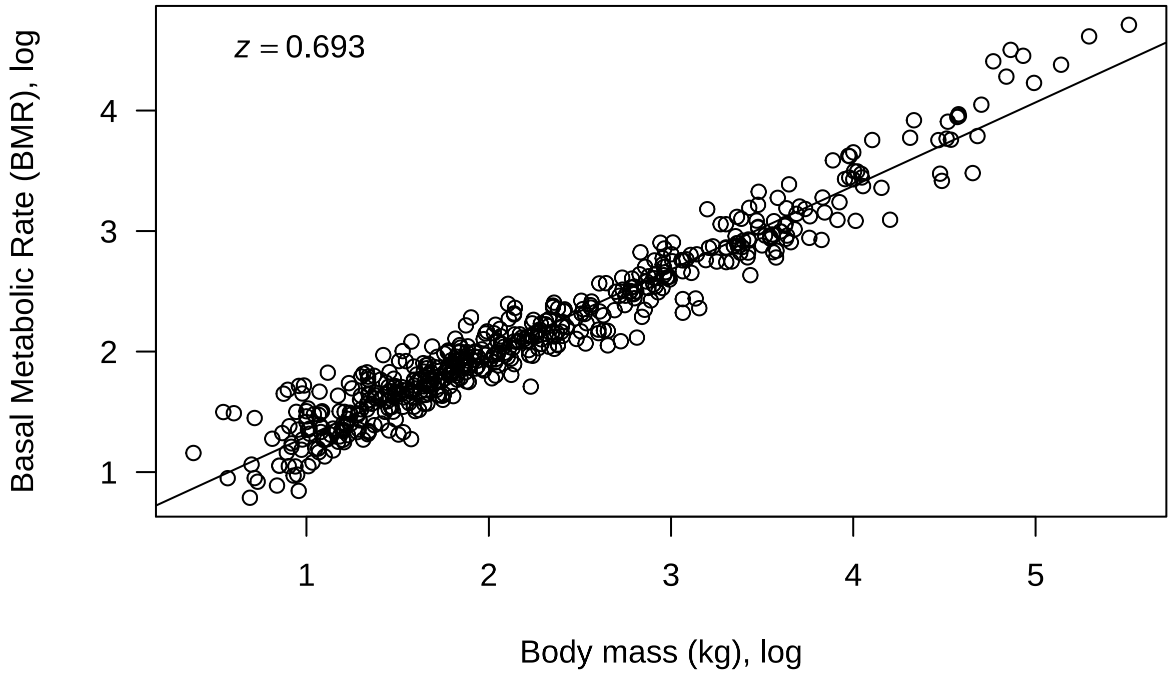
c. Discuss what factors might give rise to differences between the metrics.

> ***Answer 9a***: AEP and DNA are clearly clustered and it seems that phenylCP is as well though of the three traits I considered it is the most dispersed.
> ***Answer 9b***: This produces a different conclusion than Blomberg's K as it showed that there was not a phylogentic signal while here we see clustering. ***Answer 9c***: Blomberg's K is basedon variation in trait occurance, while D outputs are based on trait dispersion in space so considering the same traits in very different ways will produce different orders.

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

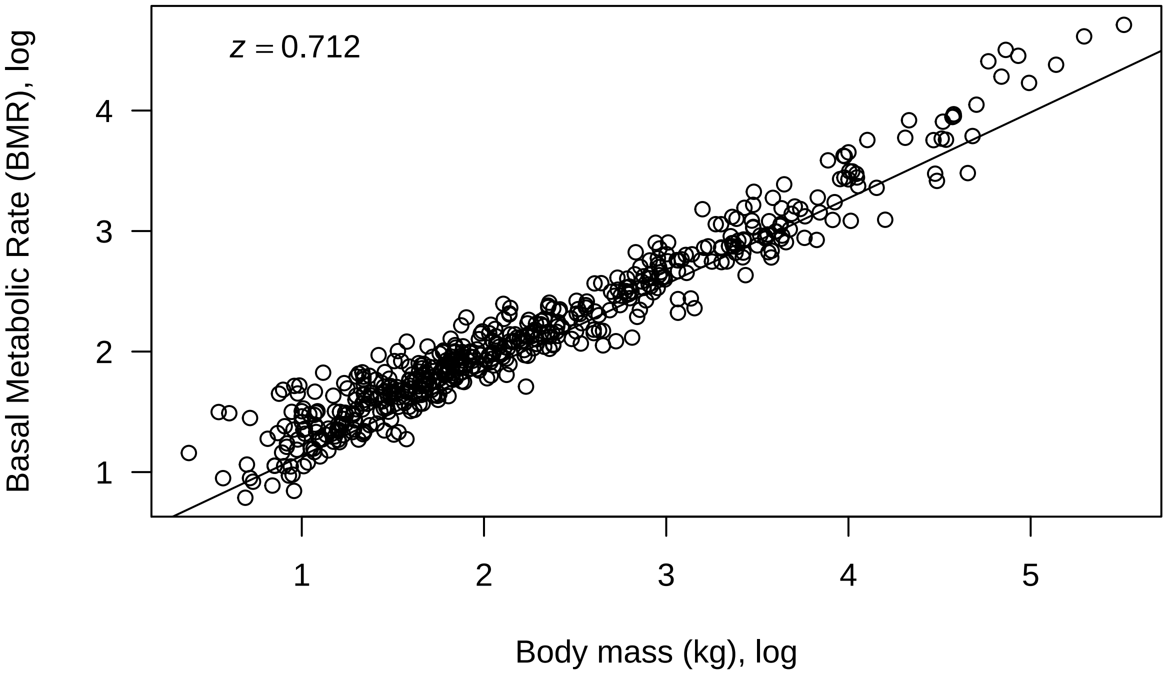
```r
mammal.Tree <- read.tree("./data/mammal_best_super_tree_fritz2009.tre")
mammal.data <- read.table("./data/mammal_BMR.txt", sep = "\t", header = T)

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

pruned.mammal.tree <- drop.tip(mammal.Tree, mammal.Tree$tip.label[-na.omit(match(mammal.species, mammal

pruned.mammal.data <- mammal.data[mammal.data$Species %in% pruned.mammal.tree$tip.label,]

rownames(pruned.mammal.data) <- pruned.mammal.data$Species

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))
text(0.5, 4.5, eqn, pos = 4)
```



```
fit.phy <- phylolm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.),
                   data = pruned.mammal.data, pruned.mammal.tree, model = 'lambda', boot = 0)
```

```
## 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 Metal
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***: They violate the assumption of indpendence because they have this shared evolutionary history. ***Answer 10b***: Phylogentic regression takes the branch lengths into account when producing residual errors while a standard regression treats them ad independent and identically distributed. ***Answer 10c***: The slope in which evolutionary history is accounted for is more accurate to the trend of the data, it is only a slight improvement but an improvement none the less. ***Answer 10d***: If there are two variables that are very not related, may look related because we are seeing correlation in the phylogeny that is not accurate to the reality of those two variables.

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.

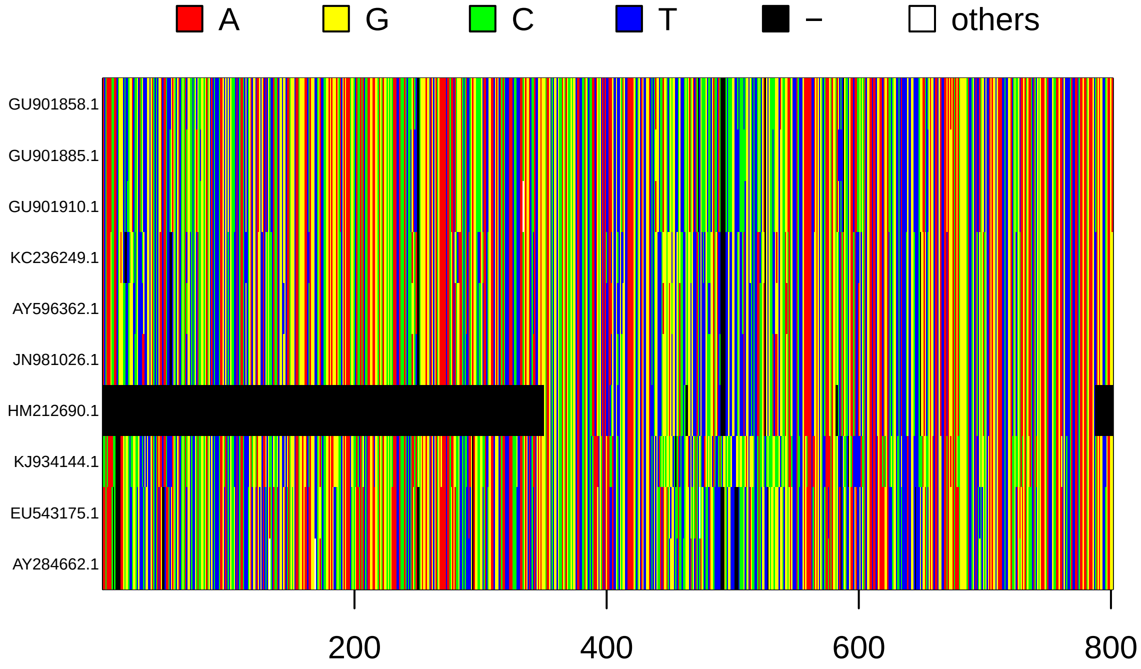
But before your team proceeds, you need to give some thought to which gene you want to focus on. For microorganisms like the bacteria we worked with above, many people use the ribosomal gene (i.e., 16S rRNA). This has many desirable features, including it is relatively long, highly conserved, and identifies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

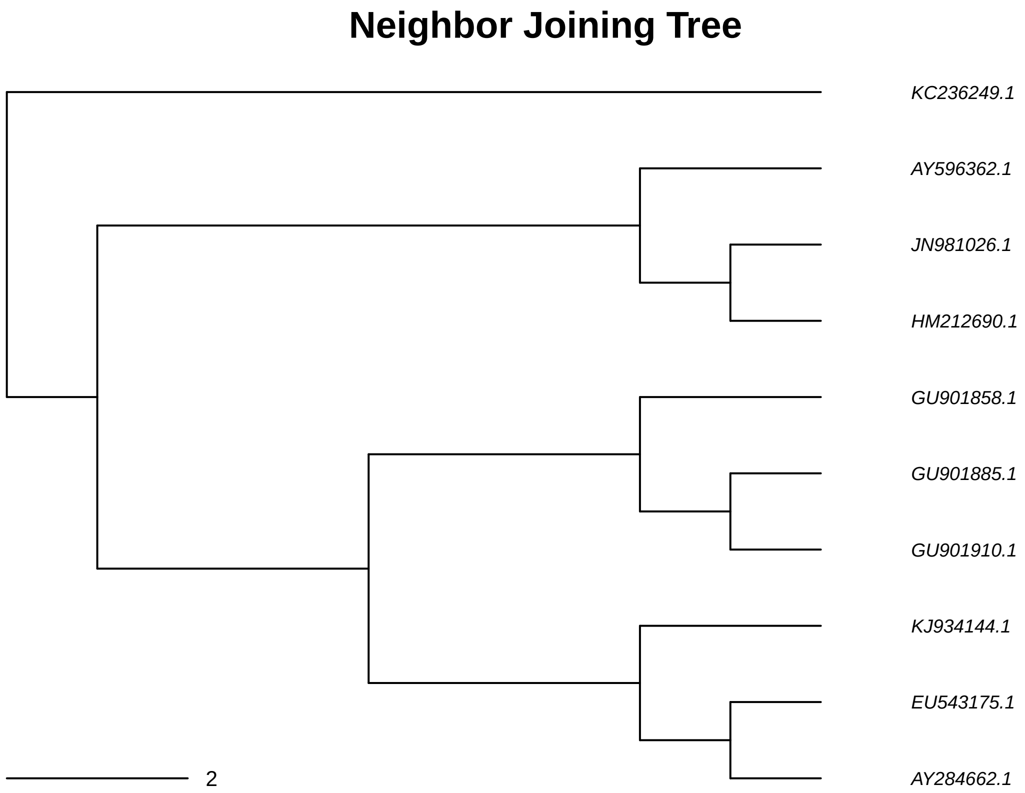
```
read.projectfasta <- read.alignment("./nbProjectFasta.txt", format = "fasta")
```

```
## Warning in readLines(file): incomplete final line found on './
## nbProjectFasta.txt'
```

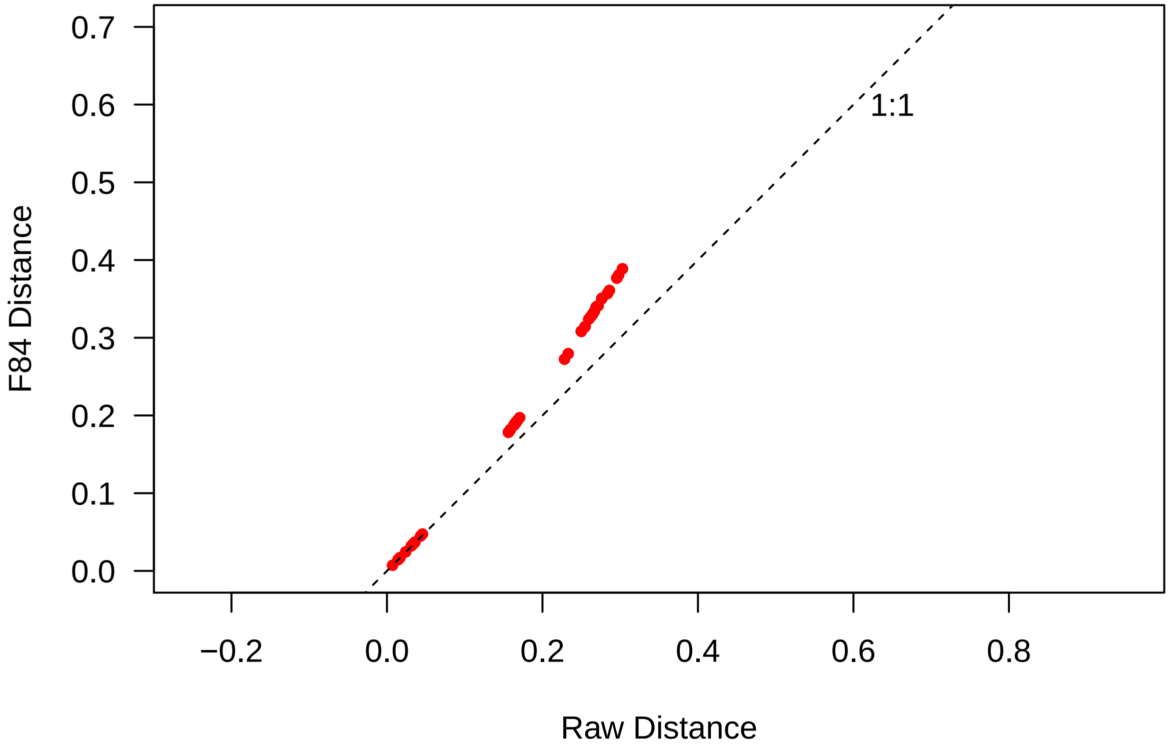
```
p.dat <- as.DNAbin(read.projectfasta)
windows <- p.dat[, 200:1000]
image.DNAbin(windows, cex.lab = 0.50)
```



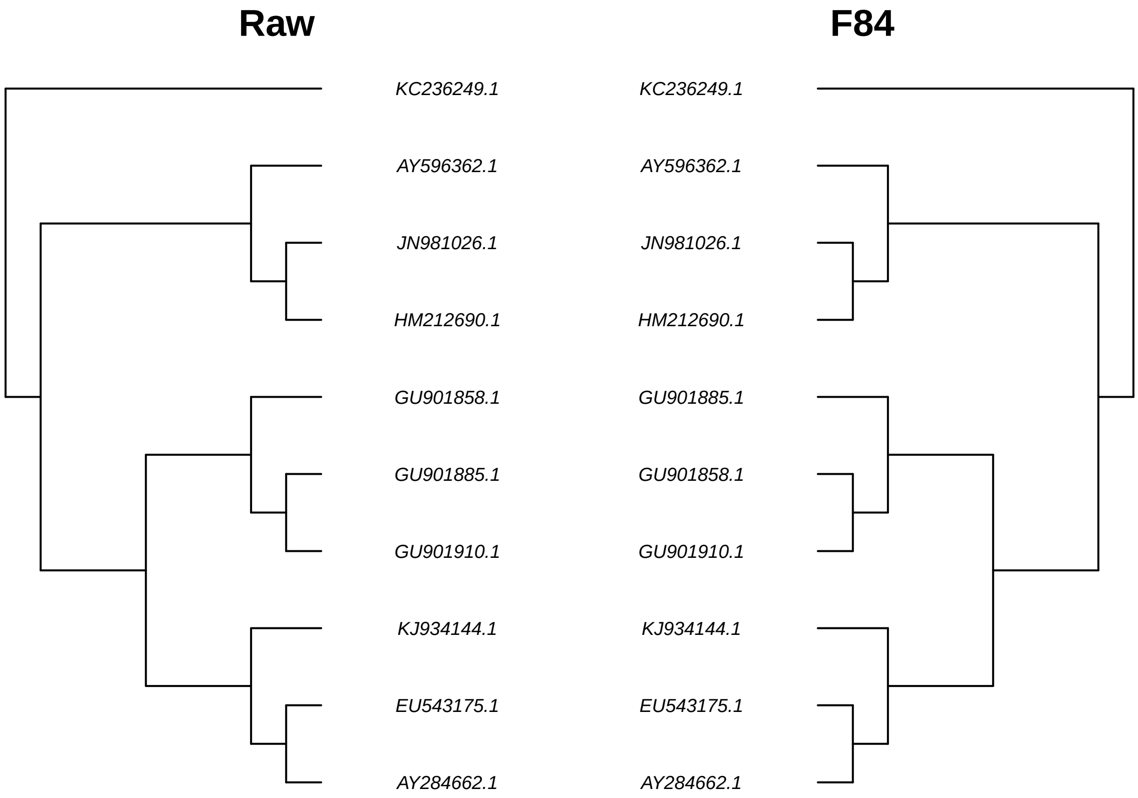
```
seq.dist.synth <- dist.dna(p.dat, model = "raw", pairwise.deletion = FALSE)

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

synthoutgroup <- match("KC236249.1", synth.tree$tip.label)

synth.rooted <- root(synth.tree, synthoutgroup, resolve.root = TRUE)

par(mar = c(1,1,2,1) + 0.1)
plot.phylo(synth.rooted, main = "Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, directio
add.scale.bar(cex = 0.7)
```



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

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



```
synthraw.tree <- bionj(seq.dist.synth)
synthF84.tree <- bionj(synth.dist.F84)

synthraw.outgroup <- match("KC236249.1", synthraw.tree$tip.label)
synthF84.outgroup <- match("KC236249.1", synthF84.tree$tip.label)

synthraw.rooted <- root(synthraw.tree, synthraw.outgroup, resolve.root=TRUE)
synthF84.rooted <- root(synthF84.tree, synthF84.outgroup, resolve.root=TRUE)

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

par(mar = c(1, 0, 2, 1))
plot.phylo(synthF84.rooted, type = "phylogram", direction = "left", show.tip.label = T, use.edge.length
```



The soil invertebrates used to construct this phylogeny are not particuluarly close in relation. Though they are all soil invertebrates that can and often do co-occur so a notable level of similarity in their 18S ribosomal genes would not be shocking. The comparison between our initial tree and F84 produce the same result.