11. Worksheet: Phylogenetic Diversity - Traits

Richard Hull; Z620: Quantitative Biodiversity, Indiana University

27 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] "C:/Users/Rich Hull/GitHub/QB2021_Hull/2.Worksheets/11.PhyloTraits"
```

```
setwd("C:/Users/Rich Hull/GitHub/QB2021_Hull/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 has sequences that are not capitalized and identify indels via the symobl "n", and the afa file has sequences that are capitalized and identify indels via the symbol "-."

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)
  }
}
```

```
## 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
```

```
##
## 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
```

```
##
## 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
```

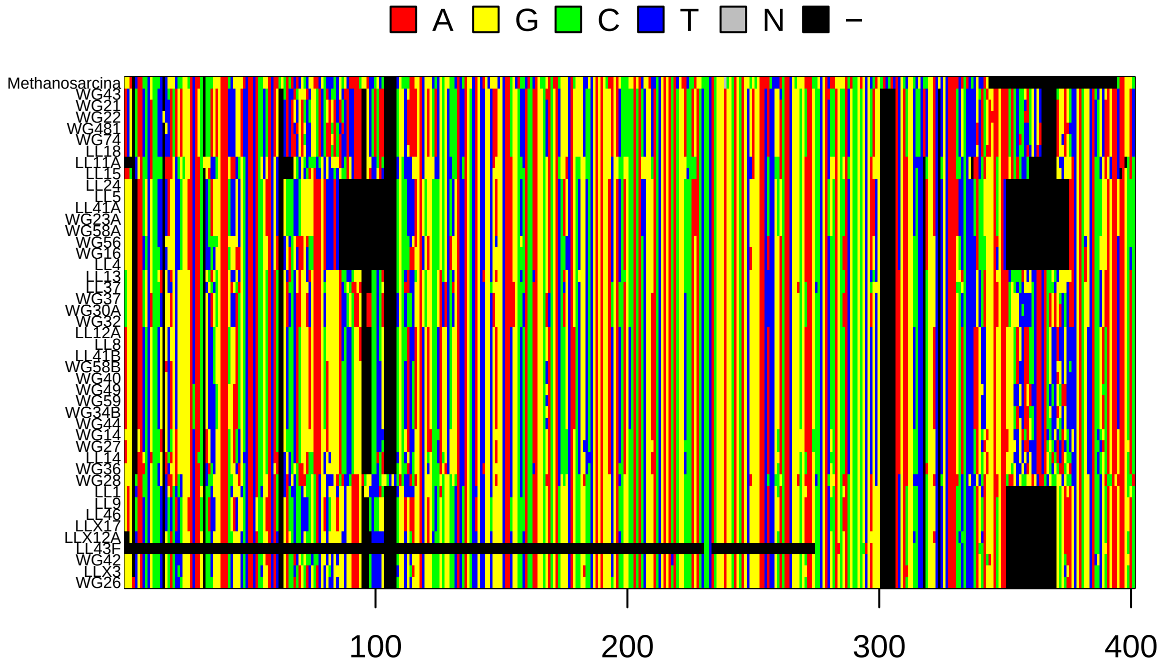
```
## Warning: package 'phangorn' was built under R version 4.0.5
```

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

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

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

```r
# read alignment file (unable to perform alignment)
read.aln <- read.alignment(file = "./data/out.aln.muscle.afa", format = "fasta")
# convert alignment file to DNAbin
p.DNAbin <- as.DNAbin(read.aln)
# identify base pair region of 16S rRNA gene to visualize
window <- p.DNAbin[, 100:500]
# visualize sequence alignment
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***: Approximately 400 base pairs ***Answer 2b***: Regions that are consistently different across all taxa are of specific importance; this includes regions such as LL11A, LL15, and WG28.

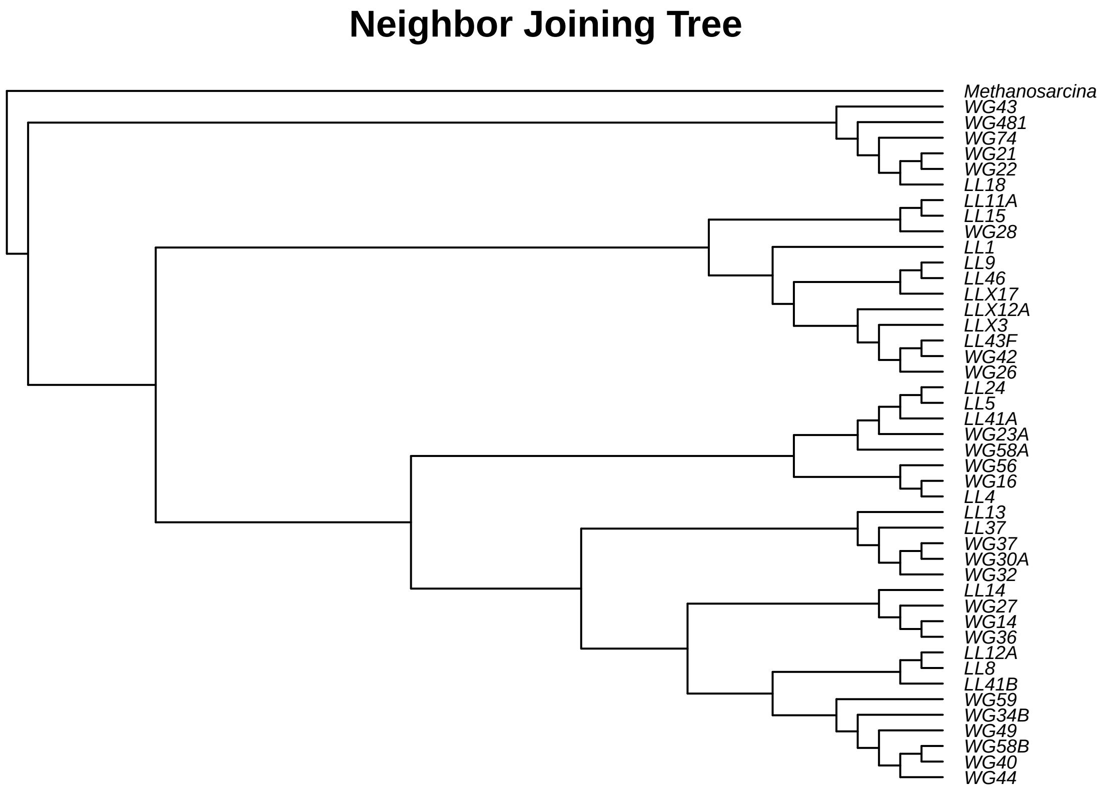
4) MAKING A PHYLOGENETIC TREE

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.

```r
# create distance matrix with raw model
seq.dist.raw <- dist.dna(p.DNAbin, model = "raw", pairwise.deletion = FALSE)
# neighbor joining algorithm to construct tree
nj.tree <- bionj(seq.dist.raw)
# identify outgroup seq
outgroup <- match("Methanosarcina", nj.tree$tip.label)
# root tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
# plot rooted tree
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(nj.rooted, main = "Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, direction =
```



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

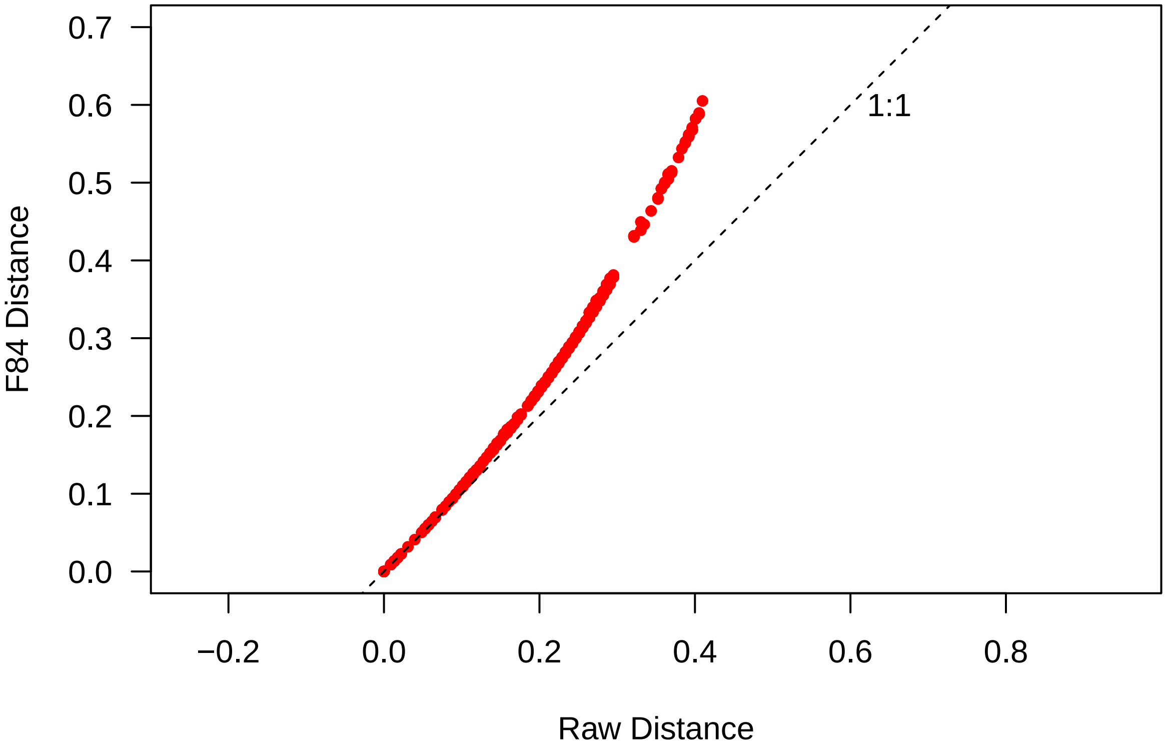
> ***Answer 3***: One main advantage of neighbor-joining is that it is very fast and can therefore be used on large datasets and can easily undergo bootstrap analysis. It also allows lineages to have differing branch lengths and allows for correction of multiple substitutions. However, disadvantages of this method are that sequence information is limited and it only provides one possible tree.

B) SUBSTITUTION MODELS OF DNA EVOLUTION

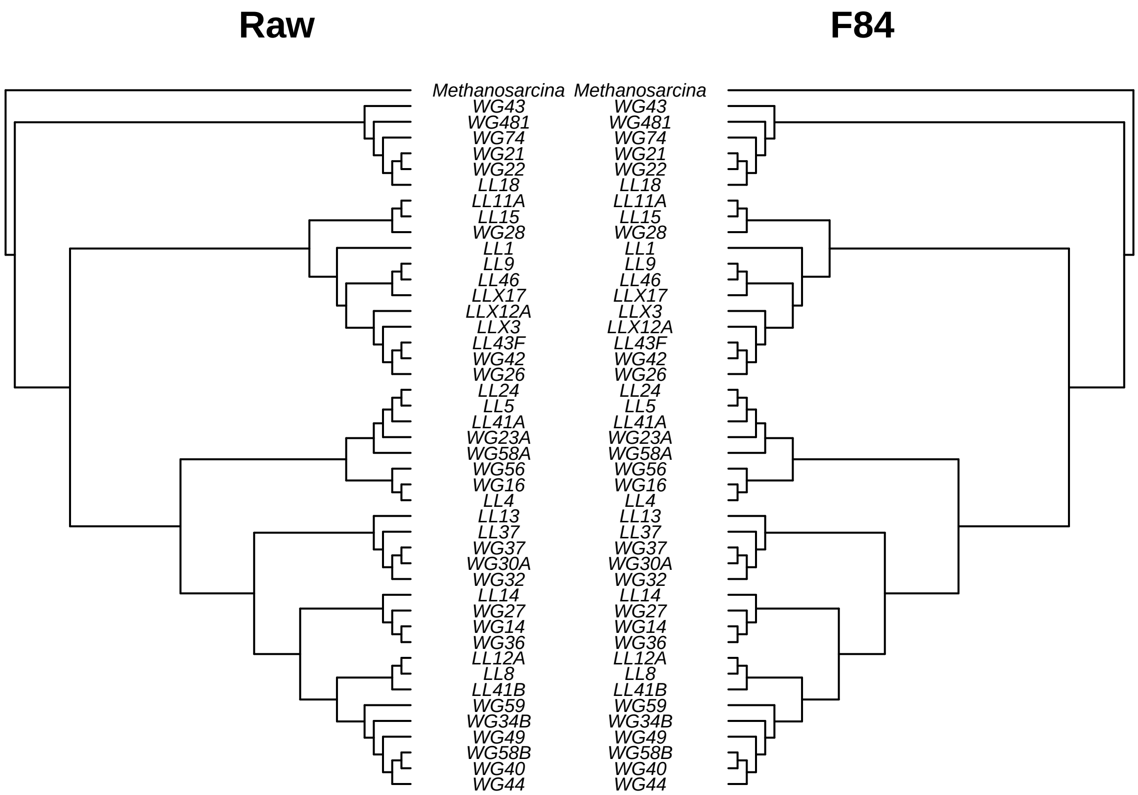
In the R code chunk below, do the following:
1. make a second distance matrix based on the Felsenstein 84 substitution model,
2. create a saturation plot to compare the *raw* and *Felsenstein (F84)* substitution models,

3. make Neighbor Joining trees for both, and
4. create a cophylogenetic plot to compare the topologies of the trees.

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

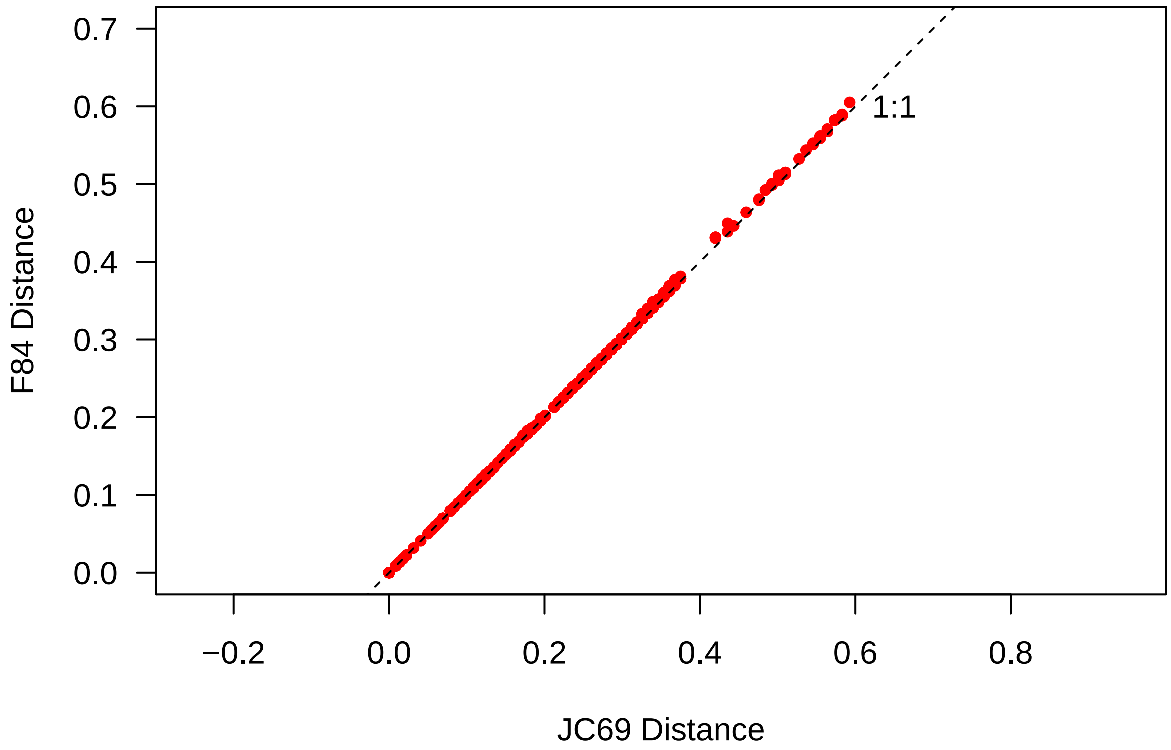


```
# make neighbor joining trees using different DNA substitution models
raw.tree <- bionj(seq.dist.raw)
F84.tree <- bionj(seq.dist.F84)
# define outgroups
raw.outgroup <- match("Methanosarcina", raw.tree$tip.label)
F84.outgroup <- match("Methanosarcina", F84.tree$tip.label)
# root trees
raw.rooted <- root(raw.tree, raw.outgroup, resolve.root = TRUE)
F84.rooted <- root(F84.tree, F84.outgroup, resolve.root = TRUE)
# create cophylogenetic plot
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 =
par(mar = c(1, 0, 2, 1))
plot.phylo(F84.rooted, type = "phylogram", direction = "left", show.tip.label = TRUE, use.edge.length =
```

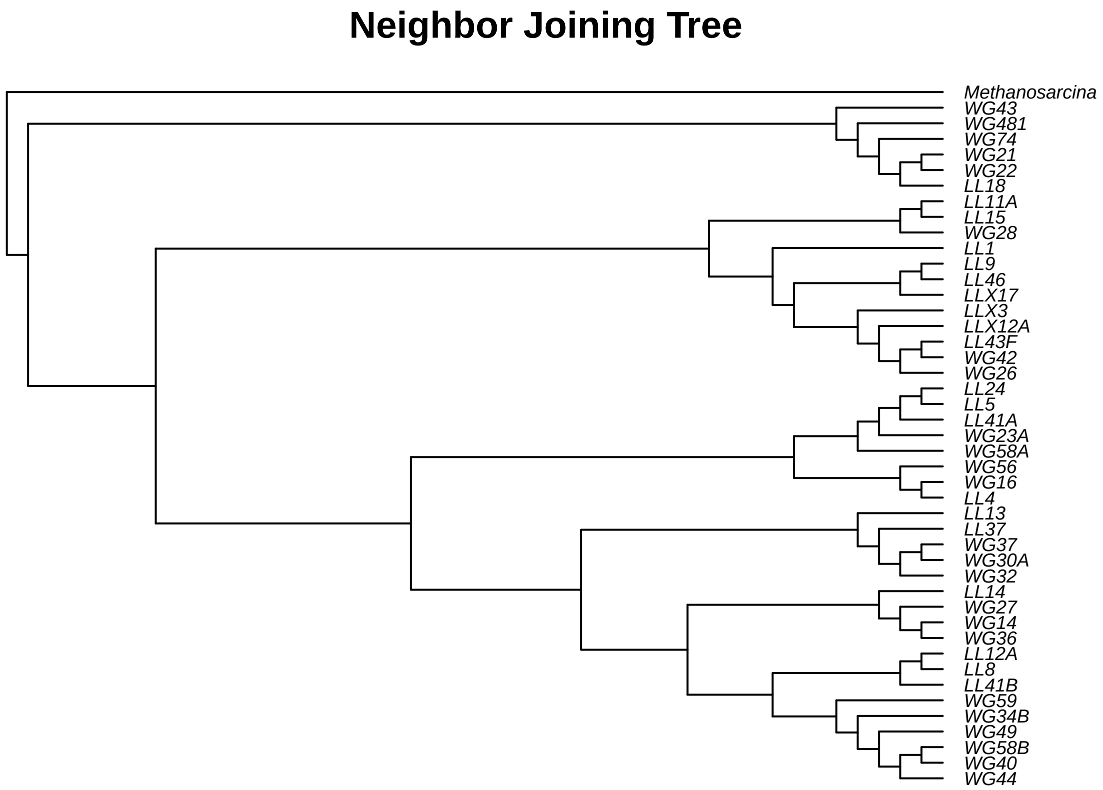


In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5.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

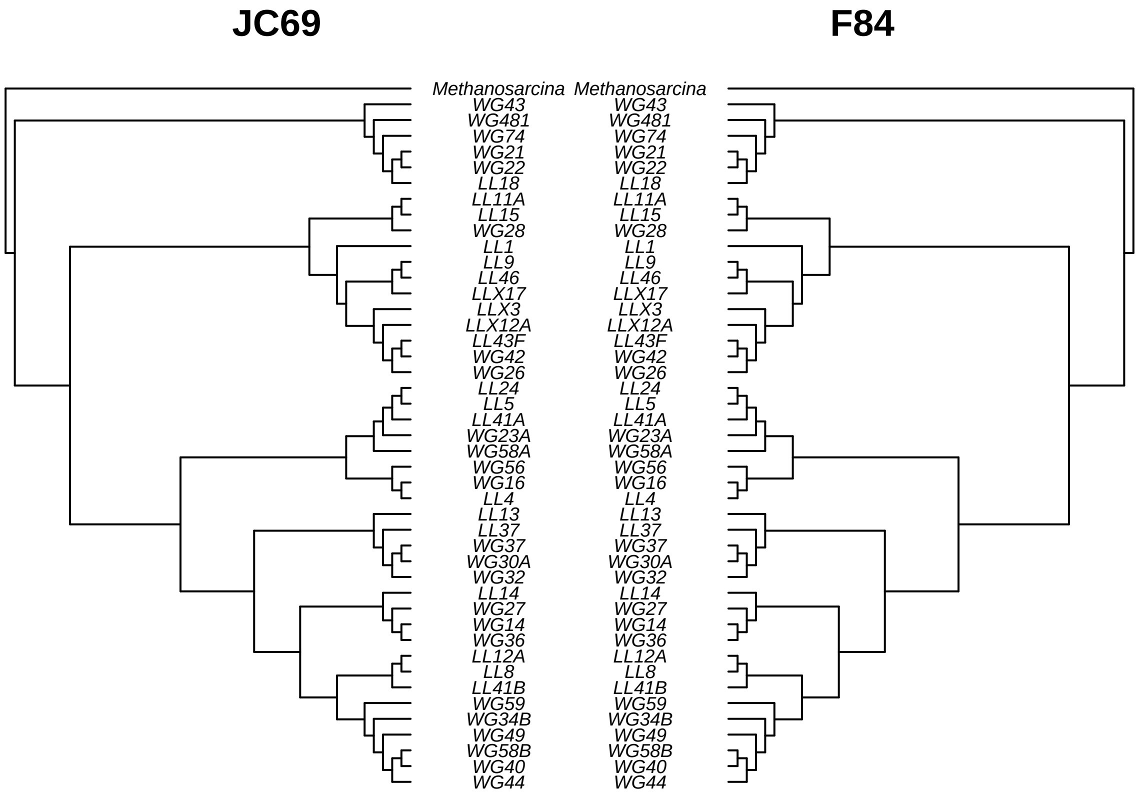
```
# create distance matrix
seq.dist.JC69 <- dist.dna(p.DNAbin, model = "JC69", pairwise.deletion = FALSE)
# make neighbor joining trees using JC69
JC69.tree <- bionj(seq.dist.JC69)
# define outgroup
JC69.outgroup <- match("Methanosarcina", JC69.tree$tip.label)
# root tree
JC69.rooted <- root(JC69.tree, JC69.outgroup, resolve.root = TRUE)
# plot differences from different DNA substitution models
par(mar = c(5,5,2,1) + 0.1)
plot(seq.dist.JC69, seq.dist.F84,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0, 0.7), ylim = c(0, 0.7), xlab = "JC69 Distance
abline(b = 1, a = 0, lty = 2)
text(0.65, 0.6, "1:1")
```



```
# plot rooted tree
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(JC69.rooted, main = "Neighbor Joining Tree", "phylogram", use.edge.length = FALSE, direction
```



```
# create cophylogenetic plot
layout(matrix(c(1,2), 1, 2), width = c(1, 1))
par(mar = c(1, 1, 2, 0))
plot.phylo(JC69.rooted, type = "phylogram", direction = "right", show.tip.label = TRUE, use.edge.length
par(mar = c(1, 0, 2, 1))
plot.phylo(F84.rooted, type = "phylogram", direction = "left", show.tip.label = TRUE, use.edge.length =
```



Question 4:

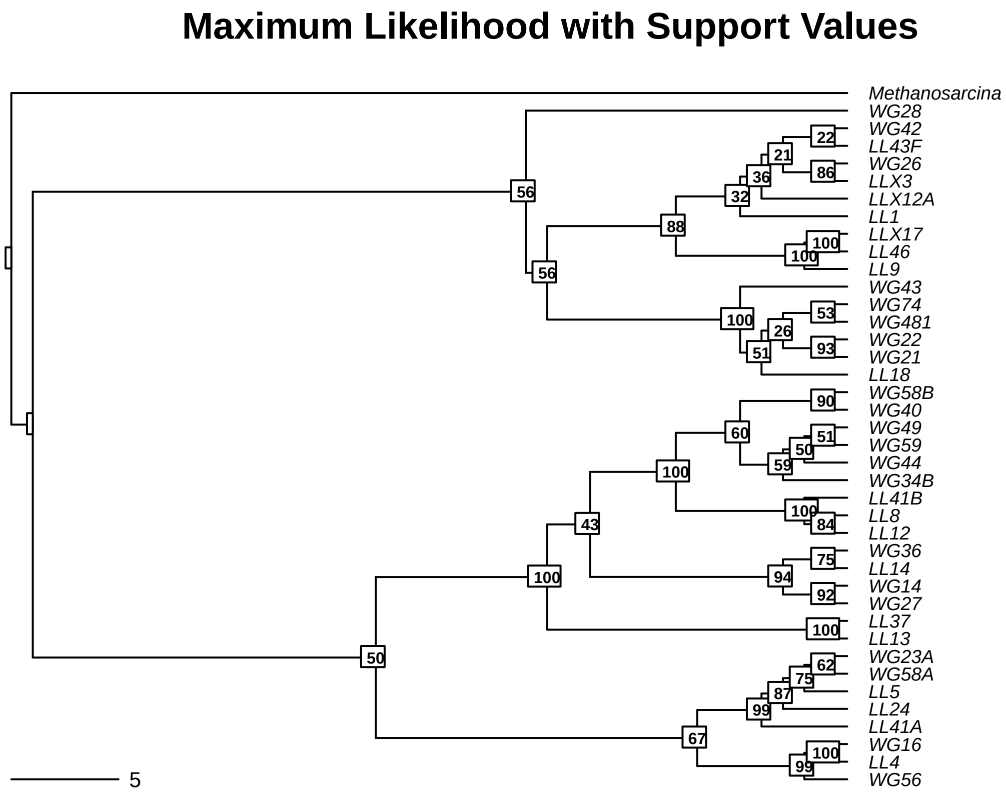
a. Describe the substitution model that you chose. What assumptions does it make and how does it compare to the F84 model?
b. Using the saturation plot and cophylogenetic plots from above, describe how your choice of substitution model affects your phylogenetic reconstruction. If the plots are inconsistent 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***: I utilized the JC69 model, which assumes that all nucleotides occur at the same frequency and can mutate from one to another with equal probability. This greatly contradicts the F84 sequence model, which assumes different rates of base transitions and transversions while also allowing for differences in base frequencies. ***Answer 4b***: It does not appear to alter the end product, which is the phylogenetic tree itself, and the distances in the matrix are roughly equivalent as well. ***Answer 4c***: It is very similar, which means that bases likely occur at equal frequencies and are mutating to other bases at equivalent rates.

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

```r
# read tree
ml.bootstrap <- read.tree("./data/ml_tree/RAxML_bipartitions.T1")
# plot
par(mar = c(1,1,2,1)+0.1)
plot.phylo(ml.bootstrap, type = "phylogram", direction = "right", show.tip.label = TRUE, use.edge.lengt
add.scale.bar(cex = 0.7)
nodelabels(ml.bootstrap$node.label, font = 2, bg = "white", frame = "r", cex = 0.5)
```



Question 5:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches?

Answer 5a: It is quite different, as the nodes in the neighbor joining tree descend quite orderly while the maximum likelihood tree has two large groupings prior to descending to smaller groups. This difference is due to the process used to make the trees, in that the neighbor joining method assesses only the next step (one step at a time) while the maximum likelihood method is instead based on probabilities, which are not calculated one step at a time. ***Answer 5b***: It provides support values for each node, or tells us how certain we should be in our tree. ***Answer 5c***: They provide the chance that the tree is correct, or the percentage of times the particular node was observed in the overall set of trees. ***Answer 5d***: Nodes with WG42, LL43F, et cetera. ***Answer 5e***: No, they occur together less than 50% of the time.

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.

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

```r
# calc max growth rate
umax <- (apply(p.growth, 1, max))
# quantify gen or sp
levins <- function(p_xi = ""){
  p = 0
  for (i in p_xi){
    p = p + i^2
  }
  nb = 1 / (length(p_xi) * p)
  return(nb)
}
# calc nb for each isolate
nb <- as.matrix(levins(p.growth.std))
# add row and column names
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.

```r
# generate neighbor joining tree using f84 dna subs mod
nj.tree <- bionj(seq.dist.F84)
# define outgroup
```

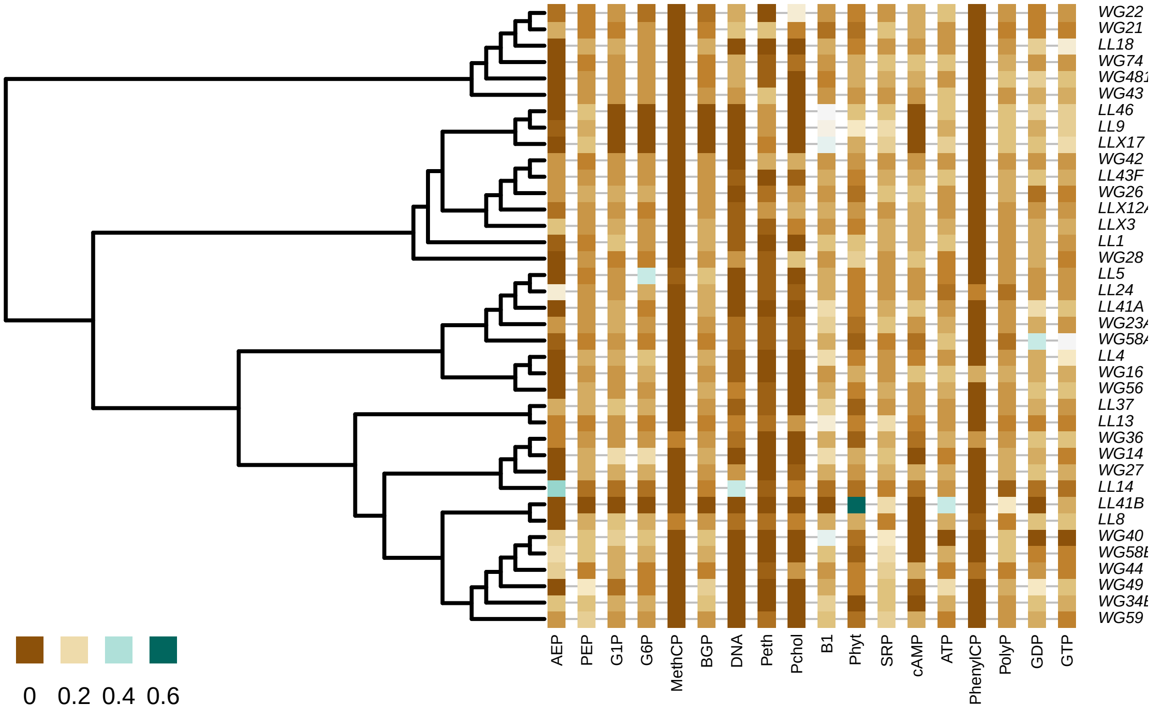
```
outgroup <- match("Methanosarcina", nj.tree$tip.label)
# root tree
nj.rooted <- root(nj.tree, outgroup, resolve.root = TRUE)
# keep rooted but drop outgroup branch
nj.rooted <- drop.tip(nj.rooted, "Methanosarcina")
nj.rooted <- drop.tip(nj.rooted, "LL11A")
nj.rooted <- drop.tip(nj.rooted, "LL15")
nj.rooted <- drop.tip(nj.rooted, "WG37")
nj.rooted <- drop.tip(nj.rooted, "WG30A")
nj.rooted <- drop.tip(nj.rooted, "WG32")
nj.rooted <- drop.tip(nj.rooted, "LL12A")
```

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 palette
mypalette <- colorRampPalette(brewer.pal(9, "BrBG"))
# map traits
par(mar = c(1,1,1,1) + 0.1)
x <- phylo4d(nj.rooted, p.growth.std)
```

```
## Warning in formatData(phy = x, dt = tip.data, type = "tip", ...): The following
## names are not found in the tree: LL12
```

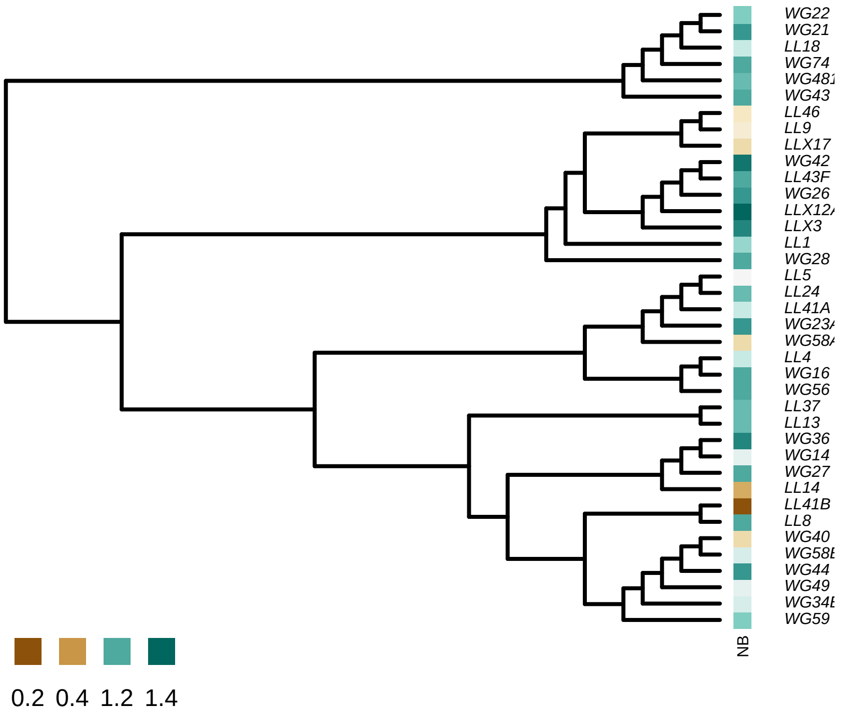
```
table.phylo4d(x, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = FALSE
```



```r
# map nb trait onto phylogeny
par(mar=c(1,5,1,5) + 0.1)
x.nb <- phylo4d(nj.rooted, nb)
```

```
## Warning in formatData(phy = x, dt = tip.data, type = "tip", ...): The following
## names are not found in the tree: LL12
```

```r
table.phylo4d(x.nb, treetype = "phylo", symbol = "colors", show.node = TRUE, cex.label = 0.5, scale = F
```



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: Growth rate should be greater among specialists because they are experts at taking advantage of a specific resource, but they only occur in very specific niches where this resource is available. ***Answer 6b***: Growth rates among specialists should be greater than those among generalists, while generalists should have a wider niche breadth than specialists.

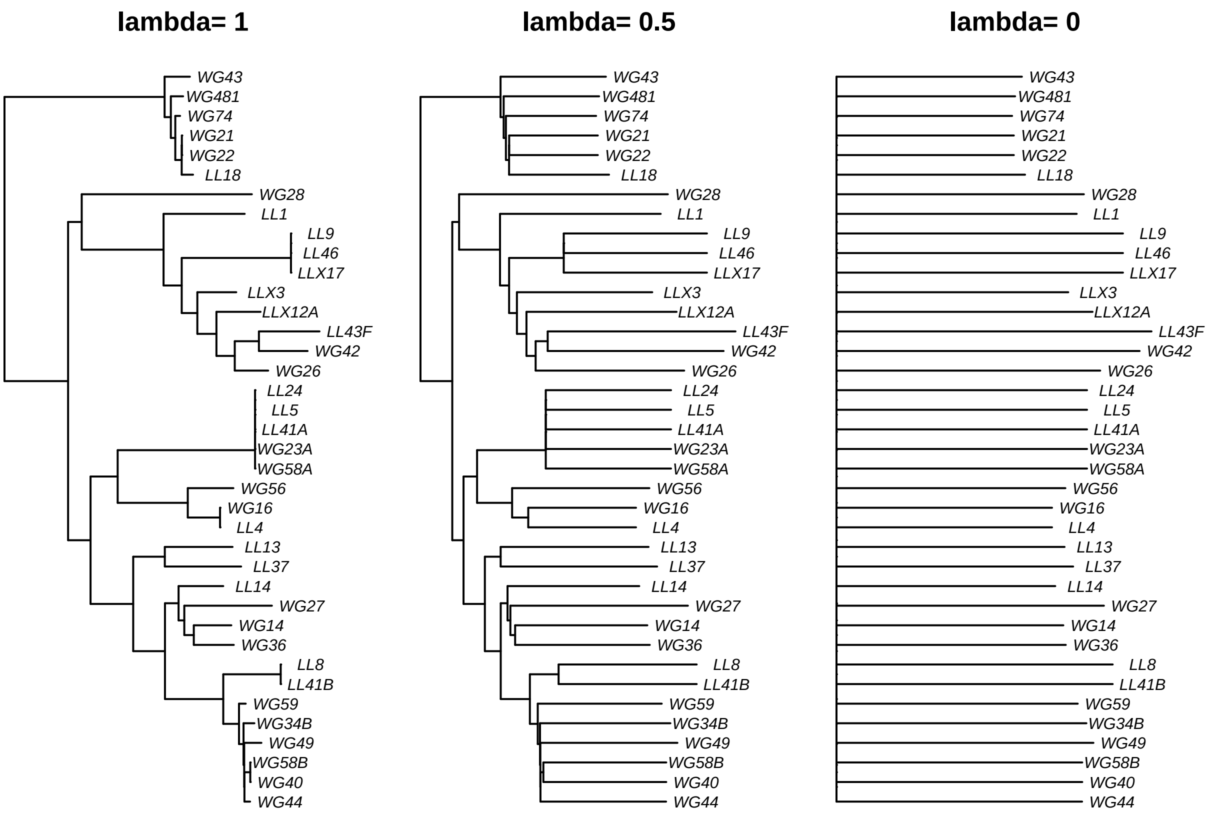
6) HYPOTHESIS TESTING

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 diff levels of phylo signal
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))
```

```r
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
# generate test stats for comparing phylogenetic signal
fitContinuous(nj.rooted, nb, model = "lambda")
```

```
## Warning in treedata(phy, dat): The following tips were not found in 'phy' and were dropped from 'dat
##  LL12

## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.061149
##  sigsq = 0.107501
##  z0 = 0.664846
##
##  model summary:
##  log-likelihood = 20.837985
##  AIC = -35.675971
##  AICc = -34.970088
##  free parameters = 3
```

```
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 49
##  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")
```

```
## Warning in treedata(phy, dat): The following tips were not found in 'phy' and were dropped from 'dat
##  LL12
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.424612
##  sigsq = 0.106866
##  z0 = 0.656763
##
##  model summary:
##  log-likelihood = 20.780666
##  AIC = -35.561332
##  AICc = -34.855449
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 77
##  frequency of best fit = 0.77
##
##  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***: 0.06 > 0, respectively ***Answer 7b***: -35.67597 & -34.970088, difference is less than 2, so the models are approximately equivalent ***Answer 7c***: This suggests that there is not phylogenetic signal

B) Phylogenetic Signal: Blomberg's K

In the R code chunk below, do the following:
1. correct tree branch-lengths to fix any zeros,
2. calculate Blomberg's K for each phosphorus resource using the `phylosignal()` function,
3. use the Benjamini-Hochberg method to correct for false discovery rate, and
4. calculate Blomberg's K for niche breadth using the `phylosignal()` function.

```
# correct for zero branch lengths on tree
nj.rooted$edge.length <- nj.rooted$edge.length + 10^-7
# create 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")
# calc 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 bh correction on pvalues
p.phylosignal[6, ] <- round(p.adjust(p.phylosignal[4, ], method = "BH"), 3)
# calc phylo signal for niche breadth
signal.nb <- phylosignal(nb, nj.rooted)
# Function did not work. Error: Error in data[res$phy$tip.label, ] : incorrect number of dimensions
```

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***: Not for niche breadth; getting code error, cannot tell. ***Answer 8b***: ^^^

C. Calculate Dispersion of a Trait

In the R code chunk below, do the following:
1. turn the continuous growth data into categorical data,
2. add a column to the data with the isolate name,
3. combine the tree and trait data using the `comparative.data()` function in `caper`, and
4. use `phylo.d()` to calculate D on at least three phosphorus traits.

```
# turn continuous data into categorical data
p.growth.pa <- as.data.frame((p.growth > 0.01) * 1)
# look at P use for each resource
apply(p.growth.pa, 2, sum)
# add names column to data
p.growth.pa$name <- rownames(p.growth.pa)
# run phylo d
p.traits <- comparative.data(nj.rooted, p.growth.pa, "name")
```

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

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?

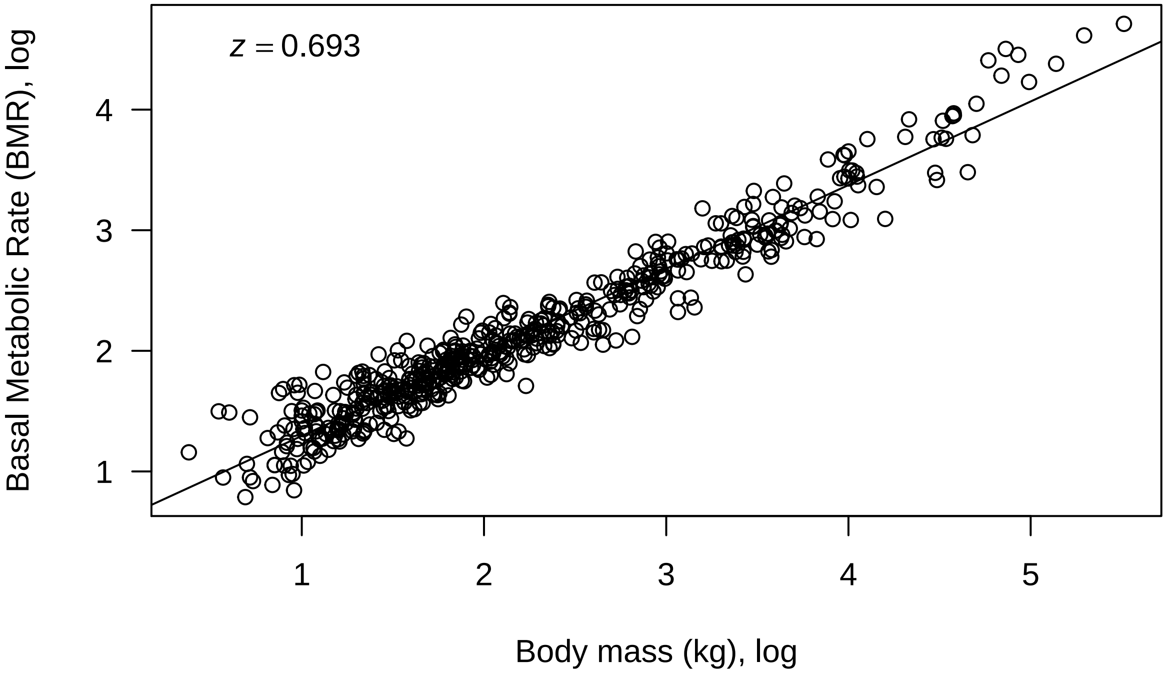
c. Discuss what factors might give rise to differences between the metrics.

> ***Answer 9a***: All 3 D values are positive, meaning the traits are overdispersed. However, two are closer to 0 than to 1 and only one is closer to 1. ***Answer 9b***: Unsureˆ code for Blomberg's K did not work ***Answer 9c***: Unsureˆ

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

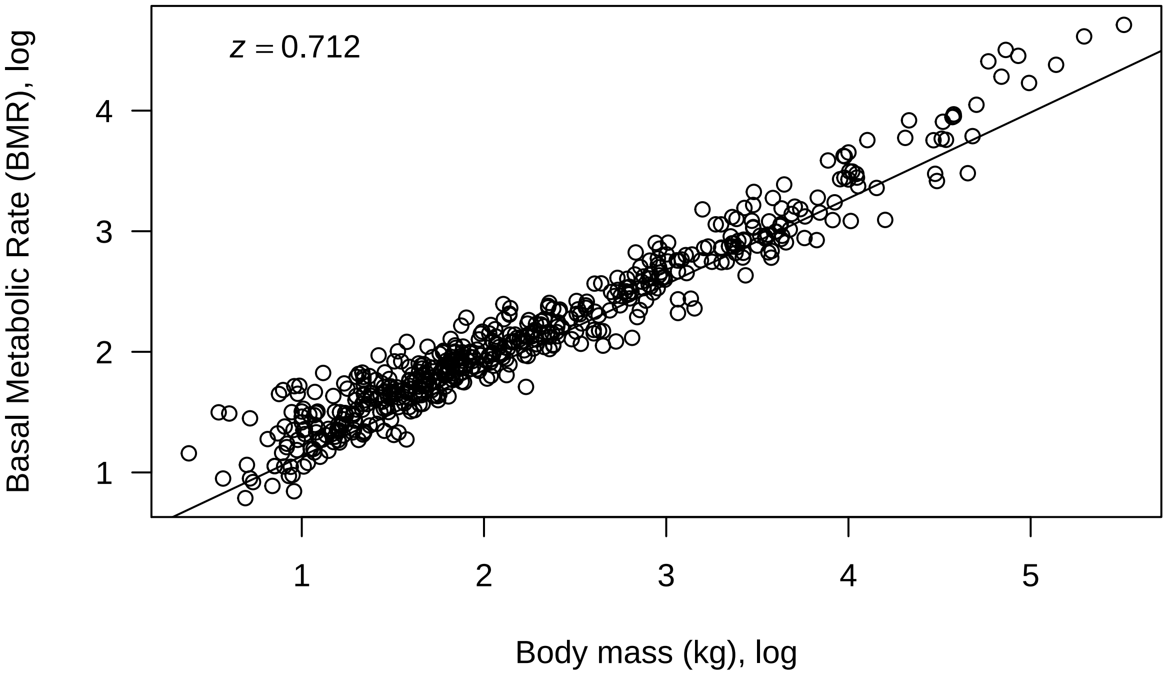
```
# input tree and dataset
mammal.Tree <- read.tree("./data/mammal_best_super_tree_fritz2009.tre")
mammal.data <- read.table("./data/mammal_BMR.txt", sep = "\t", header = TRUE)
# select variables to analyze
mammal.data <- mammal.data[,c("Species", "BMR_.mlO2.hour.", "Body_mass_for_BMR_.gr.")]
mammal.species <- array(mammal.data$Species)
# select the tips in mammal tree that are also in dataset
pruned.mammal.tree <- drop.tip(mammal.Tree, mammal.Tree$tip.label[-na.omit(match(mammal.species, mammal
# select species from dataset that are in pruned tree
pruned.mammal.data <- mammal.data[mammal.data$Species %in% pruned.mammal.tree$tip.label,]
# turn col of species names to row names
rownames(pruned.mammal.data) <- pruned.mammal.data$Species
# run simple linear regression
fit <- lm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), data = pruned.mammal.data)
plot(log10(pruned.mammal.data$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))
text(0.5, 4.5, eqn, pos = 4)
```



```
# run phylo regression
fit.phy <- phylolm(log10(BMR_.mlO2.hour.) ~ log10(Body_mass_for_BMR_.gr.), data = pruned.mammal.data, p
```

```
## 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 =
abline(a = fit.phy$coefficients[1], b = fit.phy$coefficients[2])
b1.phy <- round(fit.phy$coefficients[2],3)
eqn.phy <- bquote(italic(z) == .(b1.phy))
text(0.5, 4.5, eqn.phy, 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 need to correct for shared evolutionary history because this shared ancestry will inherently cause closely related taxa to share many of the same traits. In other words, the residuals are non-independent. ***Answer 10b***: The variance of the residual errors are described by a covariance matrix that accounts for the branch lengths of the phylogeny. ***Answer 10c***: Accounting for phylogenetic history actually slightly increased the slope and improved the fit of the model. The simple linear regression calculates a correspondence between mass and BMR of 0.69, while the phylogenetic linear regression calculates a correspondence between mass and BMR of 0.71. ***Answer 10d***: If you have two unlinked traits under different historical selection pressures you would expect the traits to be uncorrelated.

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?

```r
# Reference sequences were obtained from the NCBI for 10 species of woody tree species found in our dat
# read and align fasta file with sequences
library(msa)
```

```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

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

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following object is masked from 'package:ade4':
##
##     score
## The following objects are masked from ’package:stats’:
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from ’package:base’:
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: ’S4Vectors’

## The following objects are masked from ’package:dplyr’:
##
##     first, rename

## The following object is masked from ’package:tidyr’:
##
##     expand

## The following object is masked from ’package:base’:
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: ’IRanges’

## The following objects are masked from ’package:dplyr’:
##
##     collapse, desc, slice

## The following object is masked from ’package:nlme’:
##
##     collapse

## The following object is masked from ’package:grDevices’:
##
##     windows

## Loading required package: XVector
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
```

```
## Also defined by 'S4Vectors'
```

```
##
## Attaching package: 'Biostrings'
```

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

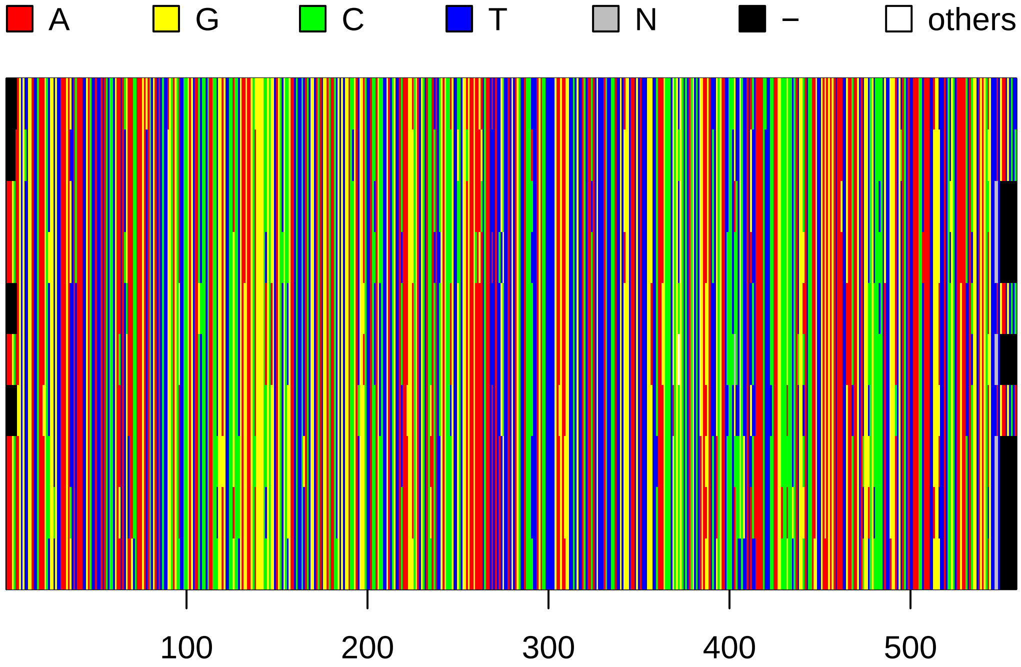
```
## The following object is masked from 'package:ape':
##
##     complement
```

```
## The following object is masked from 'package:base':
##
##     strsplit
```

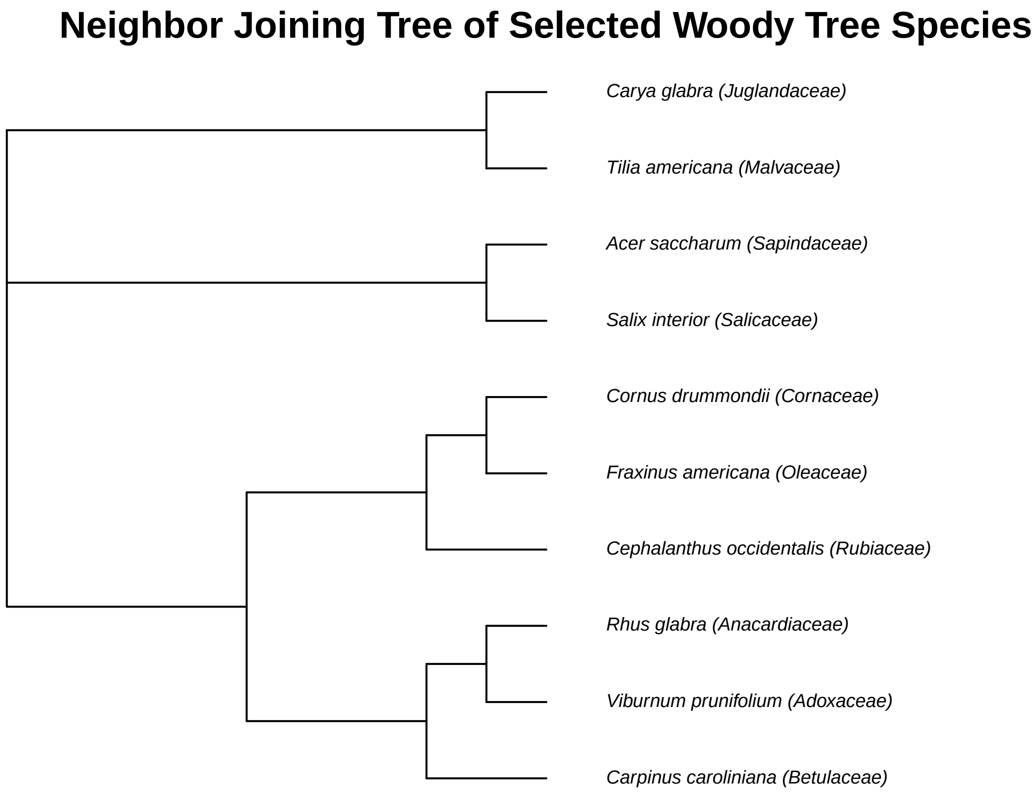
```r
alignedwt <- msa("./data/woodytreephylseq.fasta", type = "dna")
```

```
## use default substitution matrix
```

```r
# convert alignment file to DNAbin
dnabinwt <- as.DNAbin(alignedwt)
# visualize sequence alignment
image.DNAbin(dnabinwt, cex.lab = 0.50)
```



```
# alignments look good!
# # create distance matrix with raw model
wt.dm.jc69 <- dist.dna(dnabinwt, model = "JC69", pairwise.deletion = FALSE)
# neighbor joining algorithm to construct tree and        change node labels to species names with thei
wt.tree <- bionj(wt.dm.jc69)
wt.tree$tip.label <- c("Acer saccharum (Sapindaceae)", "Carya glabra (Juglandaceae)", "Tilia americana
# visualize tree
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(wt.tree, main = "Neighbor Joining Tree of Selected Woody Tree Species", "phylogram", use.edg
```


Neighbor Joining Tree of Selected Woody Tree Species
Carya glabra (Juglandaceae)
Tilia americana (Malvaceae)
Acer saccharum (Sapindaceae)
Salix interior (Salicaceae)
Cornus drummondii (Cornaceae)
Fraxinus americana (Oleaceae)
Cephalanthus occidentalis (Rubiaceae)
Rhus glabra (Anacardiaceae)
Viburnum prunifolium (Adoxaceae)
Carpinus caroliniana (Betulaceae)

```
# initial node is a polytomy probably because I did not specify an outgroup
# This tree does not necessarily agree with the most recent angiosperm phylogeny,
# This tree could be improved a number of different ways. First, I should use a monocot as an outgroup
```

“‘