8. Worksheet: Phylogenetic Diversity - Traits

Thomas Zambiasi; Z620: Quantitative Biodiversity, Indiana University

22 February, 2023

OVERVIEW

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

After completing this exercise you will be able to:

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

Directions:

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

The completed exercise is due on **Wednesday, February 22nd, 2023 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 "*/8.PhyloTraits*" folder, and
4. load all of the required R packages (be sure to install if needed).

```
#workspace setup
rm(list = ls())
getwd()
```

```
## [1] "C:/Users/tmzam/GitHub/QB2023_Zambiasi/2.Worksheets/8.PhyloTraits"
```

```
setwd("C:/Users/tmzam/GitHub/QB2023_Zambiasi/2.Worksheets/8.PhyloTraits")
#package loading
library(ape)
```

```
## Warning: package 'ape' was built under R version 4.2.2
```

```
library(seqinr)
```

```
## Warning: package 'seqinr' was built under R version 4.2.2
```

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

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

```
library(phylobase)
```

```
## Warning: package 'phylobase' was built under R version 4.2.2
```

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

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

```
library(adephylo)
```

```
## Warning: package 'adephylo' was built under R version 4.2.2
```

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

```
library(geiger)
```

```
## Warning: package 'geiger' was built under R version 4.2.2
```

```
library(picante)
```

```
## Warning: package 'picante' was built under R version 4.2.2

## Loading required package: vegan

## Warning: package 'vegan' was built under R version 4.2.2

## Loading required package: permute

##
## Attaching package: 'permute'

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

## Loading required package: lattice

## This is vegan 2.6-4

## Loading required package: nlme

##
## Attaching package: 'nlme'

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

```
library(stats)
library(RColorBrewer)
library(caper)
```

```
## Warning: package 'caper' was built under R version 4.2.2

## Loading required package: MASS

## Loading required package: mvtnorm
```

```
library(phylolm)
```

```
## Warning: package 'phylolm' was built under R version 4.2.2
```

```
library(pmc)
```

```
## Warning: package 'pmc' was built under R version 4.2.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(tidyr)
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.2.2
```

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

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

```
library(pander)
```

```
## Warning: package 'pander' was built under R version 4.2.2
```

```
library(phytools)
```

```
## Warning: package 'phytools' was built under R version 4.2.2
```

```
## Loading required package: maps
```

```
##
## Attaching package: 'phytools'
```

```
## The following object is masked from 'package:vegan':
##
##     scores
```

```
## The following object is masked from 'package:phylobase':
##
##     readNexus
```

```
library(vegan)
library(cluster)
```

```
##
## Attaching package: 'cluster'
```

```
## The following object is masked from 'package:maps':
##
##     votes.repub
```

```
library(dendextend)
```

```
## Registered S3 method overwritten by ’dendextend’:
##   method     from
##   rev.hclust vegan
```

```
##
## ---------------------
## Welcome to dendextend version 1.16.0
## Type citation(’dendextend’) for how to cite the package.
##
## Type browseVignettes(package = ’dendextend’) for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: ’dendextend’
```

```
## The following object is masked from ’package:phytools’:
##
##     untangle
```

```
## The following object is masked from ’package:permute’:
##
##     shuffle
```

```
## The following object is masked from ’package:geiger’:
##
##     is.phylo
```

```
## The following objects are masked from ’package:phylobase’:
##
##     labels<-, prune
```

```
## The following objects are masked from ’package:ape’:
##
##     ladderize, rotate
```

```
## The following object is masked from ’package:stats’:
##
##     cutree
```

```
library(phylogram)
```

```
## Warning: package ’phylogram’ was built under R version 4.2.2
```

```
##
## Attaching package: 'phylogram'

## The following object is masked from 'package:dendextend':
##
##     prune

## The following object is masked from 'package:phylobase':
##
##     prune
```

```
library(bios2mds)
```

```
## Warning: package 'bios2mds' was built under R version 4.2.2

## Loading required package: amap

## Loading required package: e1071

## Loading required package: scales

##
## Attaching package: 'scales'

## The following object is masked from 'package:geiger':
##
##     rescale

## Loading required package: rgl

## Warning: package 'rgl' was built under R version 4.2.2
```

```
library(BiocManager)
```

```
## Warning: package 'BiocManager' was built under R version 4.2.2
```

```
library(msa)
```

```
## Warning: package 'msa' was built under R version 4.2.2

## Loading required package: Biostrings

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'
## 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, aperm, 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

## Warning: package ’S4Vectors’ was built under R version 4.2.2

## Loading required package: stats4

##
## Attaching package: ’S4Vectors’

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

## The following objects are masked from ’package:base’:
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: ’IRanges’

## 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’

## Loading required package: GenomeInfoDb

## Warning: package ’GenomeInfoDb’ was built under R version 4.2.2

##
## Attaching package: ’Biostrings’

## The following object is masked from ’package:dendextend’:
##
##     nnodes

## 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

##
## Attaching package: ’msa’

## The following object is masked from ’package:BiocManager’:
##
##     version
```

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

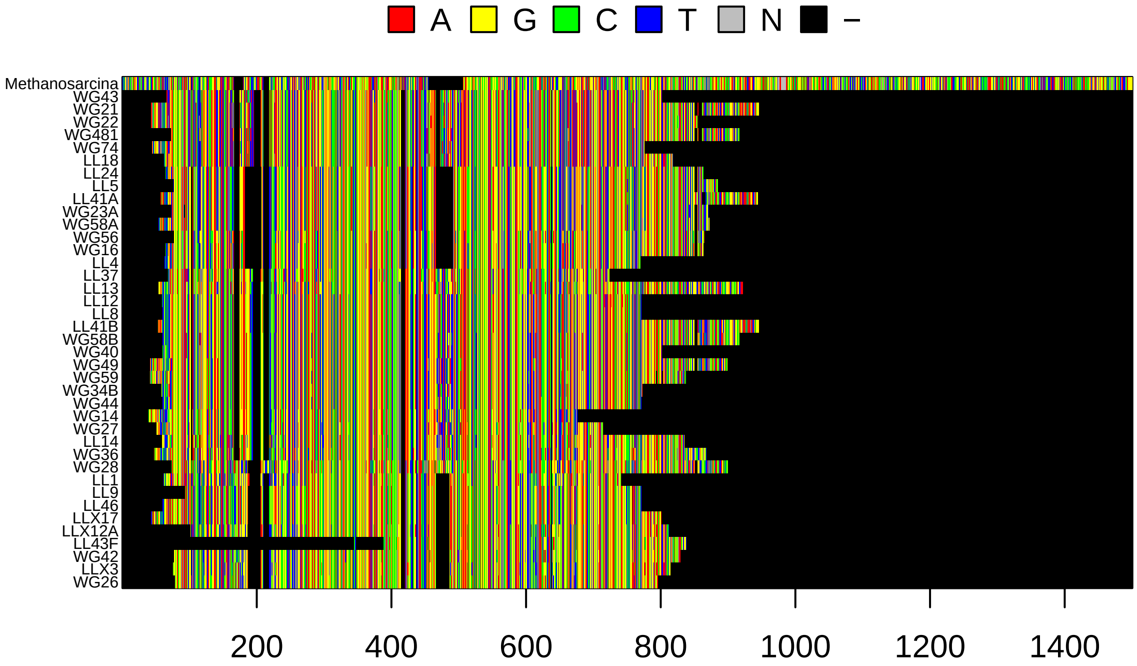
```
#Loading files to look at for Question 1
seqs.fasta<-readDNAStringSet("data/p.isolates.fasta",format="fasta")
seqs.afa<-read.alignment("data/p.isolates.afa",format="fasta")
```

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 much easier to look at; the individual nucleotides are color-coded, listed with one sequence per line, and have the width of the sequence listed. The .afa file does not format the sequences with colors or anything and lists them all in one long string. The order of the 40 sequences is given at the beginning of the file, and each sequence is then listed in order. Every sequence has each nucleotide in order and also has several dashes mixed in among the nucleotides. It seems like the dashes represent gaps in the sequences, so I would assume this means that all of the sequences in this file are aligned and the blanks represent parts of each sequence that would align with the others but do not have specific data in that region.

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.

```
#already read-in the .afa alignment file (see last code chunk)
#converting alignment file to DNAbin object
p.DNAbin<-as.DNAbin(seqs.afa)
#finding region of the gene to visualize
vis.window<-p.DNAbin[,1:1500]
#plotting alignment
image.DNAbin(vis.window,cex.lab=0.5)
```



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: The reference sequence from the Archaea outgroup seems to be approximately 1500 nucleotides long. The seqeunces for the bacterial isolates vary quite a bit in their lengths and are generally pretty patchy (lots of open spaces with no sequence data). The longest of these appear to fall between bp 75 and bp 950 on the reference sequence (roughly 800 bp in length), but most fall between the bp 100 and bp 800 marks (about 700 bp long).
Answer 2b: The patchiness of the bacterial isolates would make it difficult to compare sequences effectively for phylogenetic inference, but there is a chunk that is relatively unbroken across all samples between bp 500 and bp 700. This region would likely be the most useful for phylogenetic inference because sequence differences can be compared for every sample. There is also a small region between bp 400 and bp 475 which might be too small to effectively compare (smaller sequence regions may not show much of a difference simply due to looking at fewer nucleotides), but this could potentially be used to expand the area used for phylogenetic inference if the patchy areas (seem to be 25-50 bp in length) are not considered in the comparison.

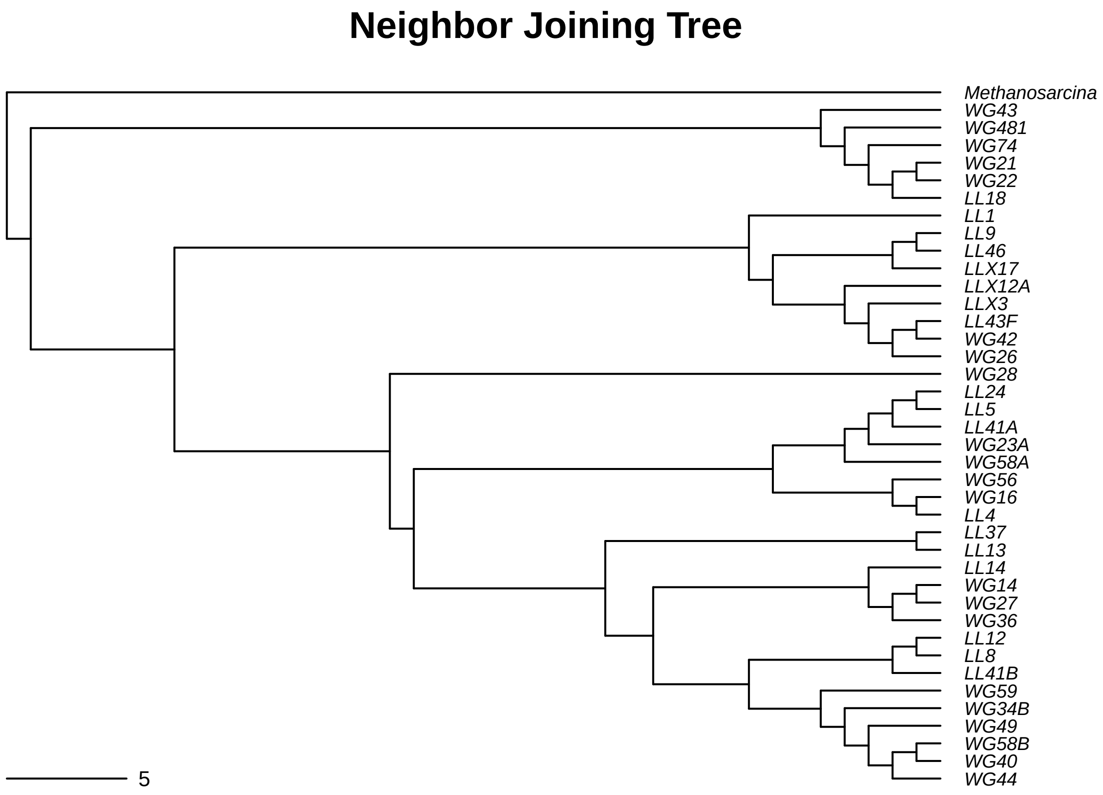
4) MAKING A PHYLOGENETIC TREE

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
#creating distance matrix
seq.dist.raw<-dist.dna(p.DNAbin,model="raw",pairwise.deletion=FALSE)
#creating tree based on distance matrix
neighbor.tree<-bionj(seq.dist.raw)
#specifying Methanosarcina as outgroup
outgroup<-match("Methanosarcina",neighbor.tree$tip.label)
#rooting tree
neighbor.root<-root(neighbor.tree,outgroup,resolve.root=TRUE)
#plotting tree
par(mar=c(1,1,2,1)+0.1)
plot.phylo(neighbor.root,main="Neighbor Joining Tree","phylogram",
           use.edge.length=FALSE,direction="right",cex=0.6,label.offset=1)
add.scale.bar(cex=0.7)
```



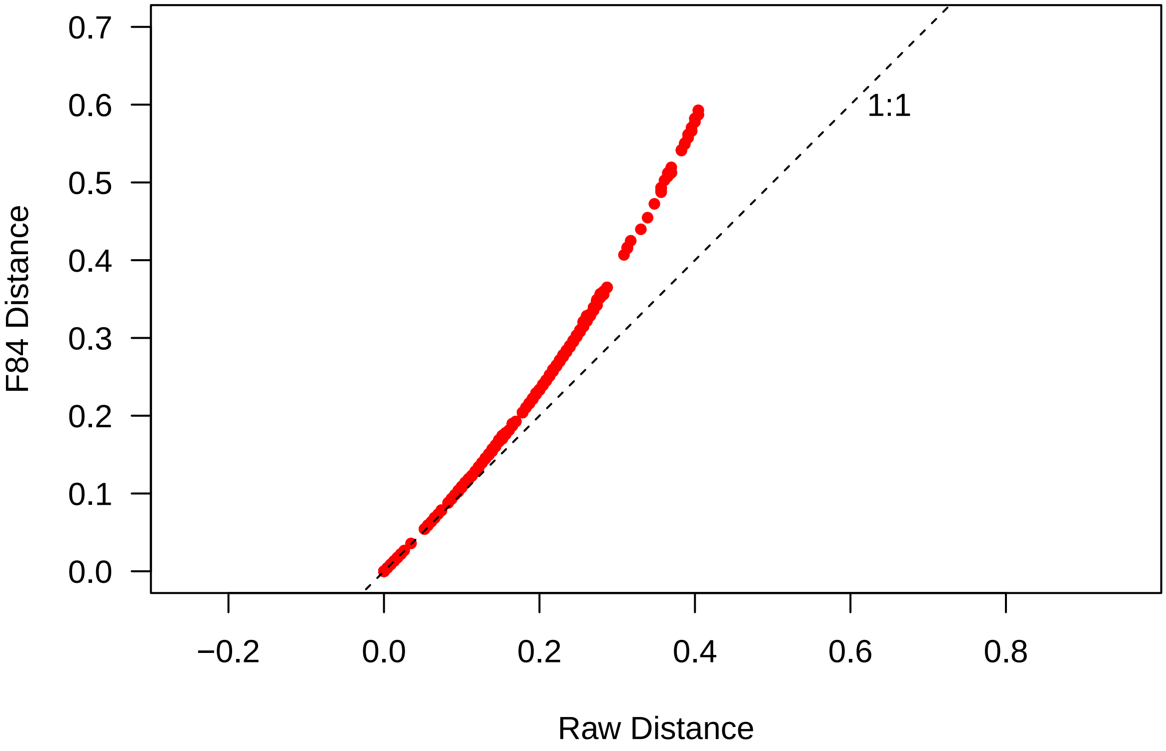
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

Answer 3: A major advantage of a neighbor joining tree is that it is a relatively simple starting point for constructing phylogenies. It also can be built using fundamental data structures because it is based on a distance matrix of the taxa being analyzed. The main downside of neighbor joining trees is that they don't reflect some nuances in sequence evolution. For example, the distance matrix used to build the tree has no information about whether multiple substitutions have happened in one spot or how some nucleotide substitutions are more likely than others. Because these features aren't accounted for, neighbor joining trees may not be the most accurate representation of evolutionary relationships among the species in question.

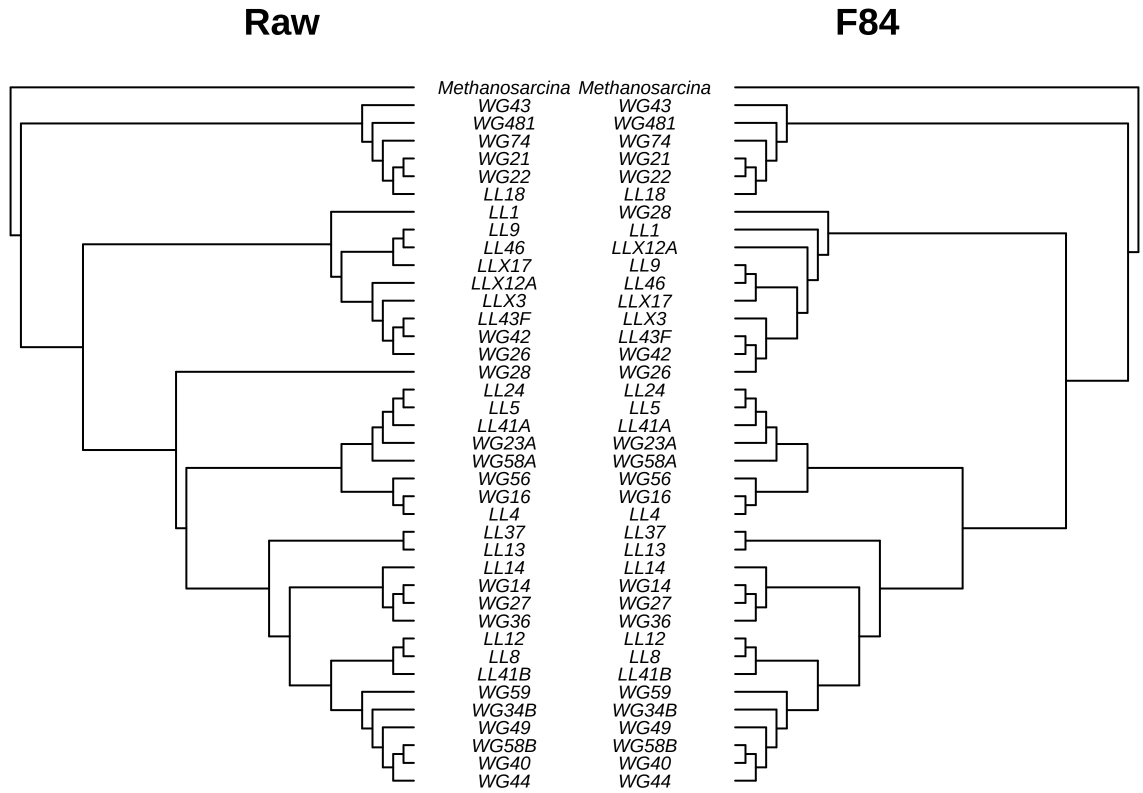
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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 F84 model distance matrix
seq.dist.f84<-dist.dna(p.DNAbin,model="F84",pairwise.deletion=FALSE)
#making saturation plot to compare raw and F84 distance matrices
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")
```

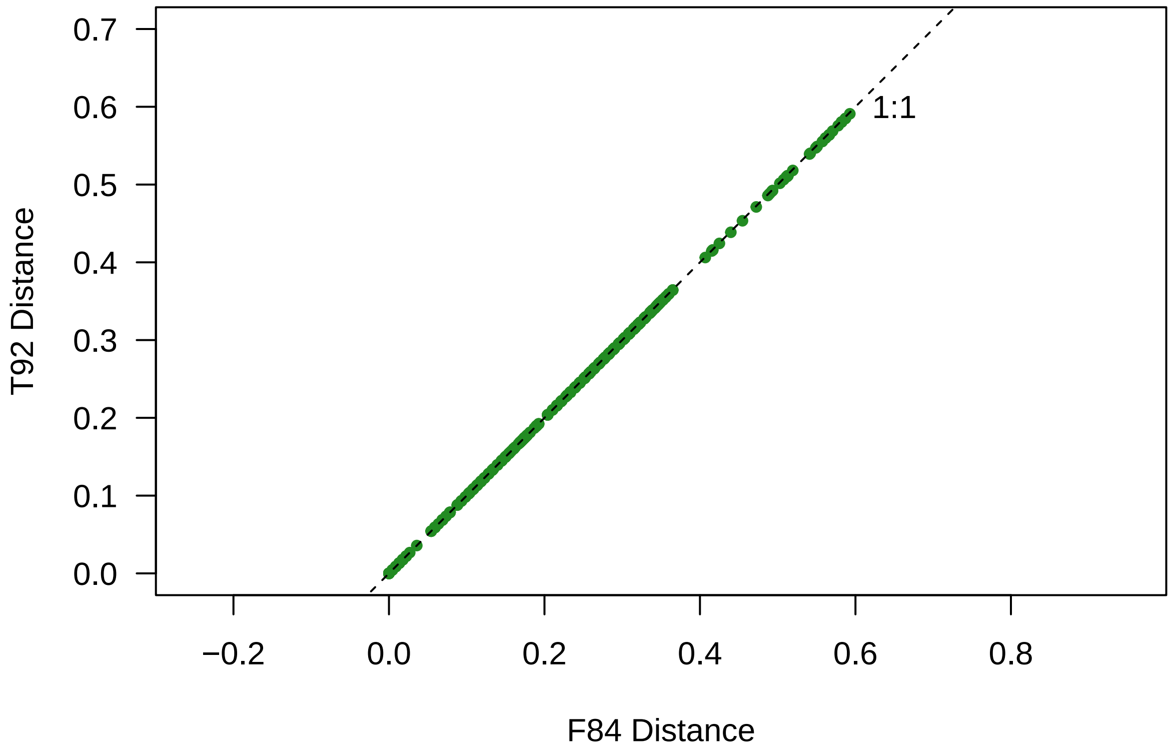


```
#making neighbor joining trees for both models
rawtree<-bionj(seq.dist.raw)
f84tree<-bionj(seq.dist.f84)
#specifying outgroups for each tree
raw.og<-match("Methanosarcina",rawtree$tip.label)
f84.og<-match("Methanosarcina",f84tree$tip.label)
#rooting trees
rawroot<-root(rawtree,raw.og,resolve.root=TRUE)
f84root<-root(f84tree,f84.og,resolve.root=TRUE)
#building cophylogenetic plot
layout(matrix(c(1,2),1,2),width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(rawroot,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(f84root,type="phylogram",direction="left",show.tip.label=TRUE,
           use.edge.length=FALSE,adj=0.5,cex=0.6,label.offset=2,main="F84")
```

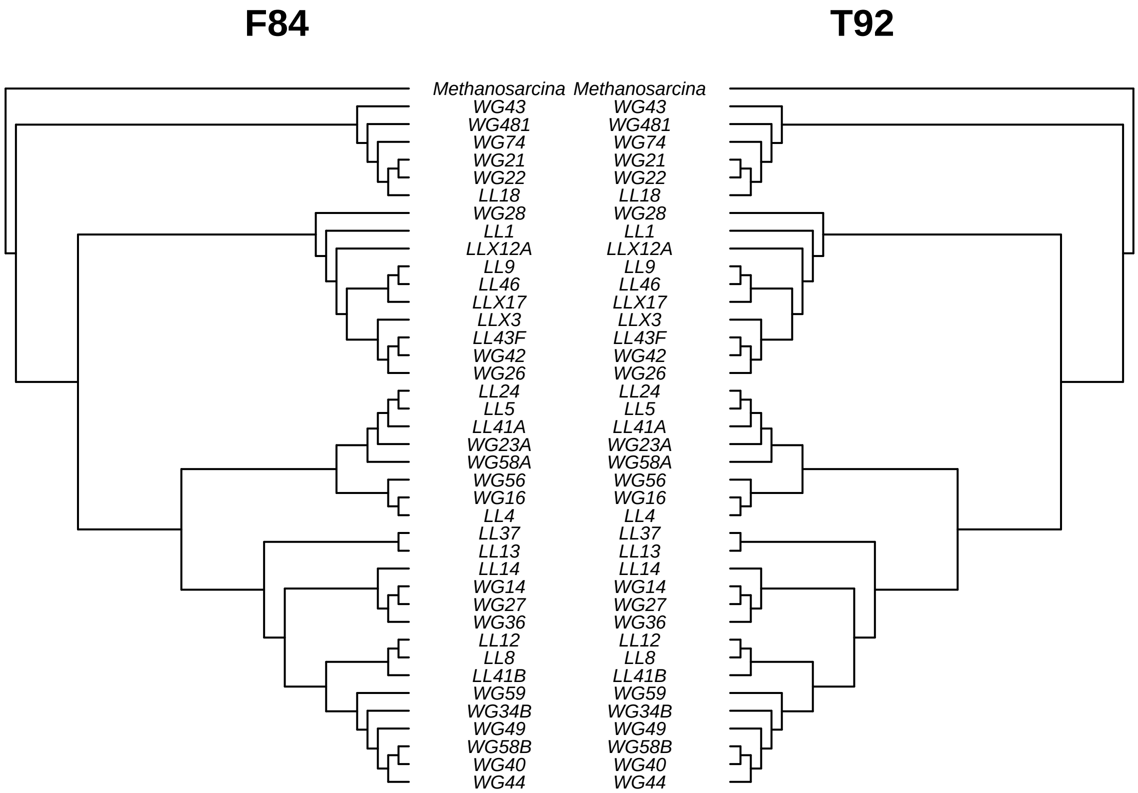


In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5.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

```
#will choose the Tamura model (T92) for comparison
#making distance matrix for T92 model
seq.dist.t92<-dist.dna(p.DNAbin,model="T92",pairwise.deletion=FALSE)
#making saturation plot to compare T92 and F84
par(mar=c(5,5,2,1)+0.1)
plot(seq.dist.f84,seq.dist.t92,pch=20,col="forest green",las=1,asp=1,
     xlim=c(0,0.7),ylim=c(0,0.7),xlab="F84 Distance",ylab="T92 Distance")
abline(b=1,a=0,lty=2)
text(0.65,0.6,"1:1")
```



```
#making neighbor joining tree for T92 model
t92tree<-bionj(seq.dist.t92)
#specifying outgroup for T92 model
t92.og<-match("Methanosarcina",t92tree$tip.label)
#rooting T92 tree
t92root<-root(t92tree,t92.og,resolve.root=TRUE)
#making cophylogenetic plot to compare F84 and T92 models
layout(matrix(c(1,2),1,2),width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(f84root,type="phylogram",direction="right",show.tip.label=TRUE,
           use.edge.length=FALSE,adj=0.5,cex=0.6,label.offset=2,main="F84")
par(mar=c(1,0,2,1))
plot.phylo(t92root,type="phylogram",direction="left",show.tip.label=TRUE,
           use.edge.length=FALSE,adj=0.5,cex=0.6,label.offset=2,main="T92")
```



Question 4:

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

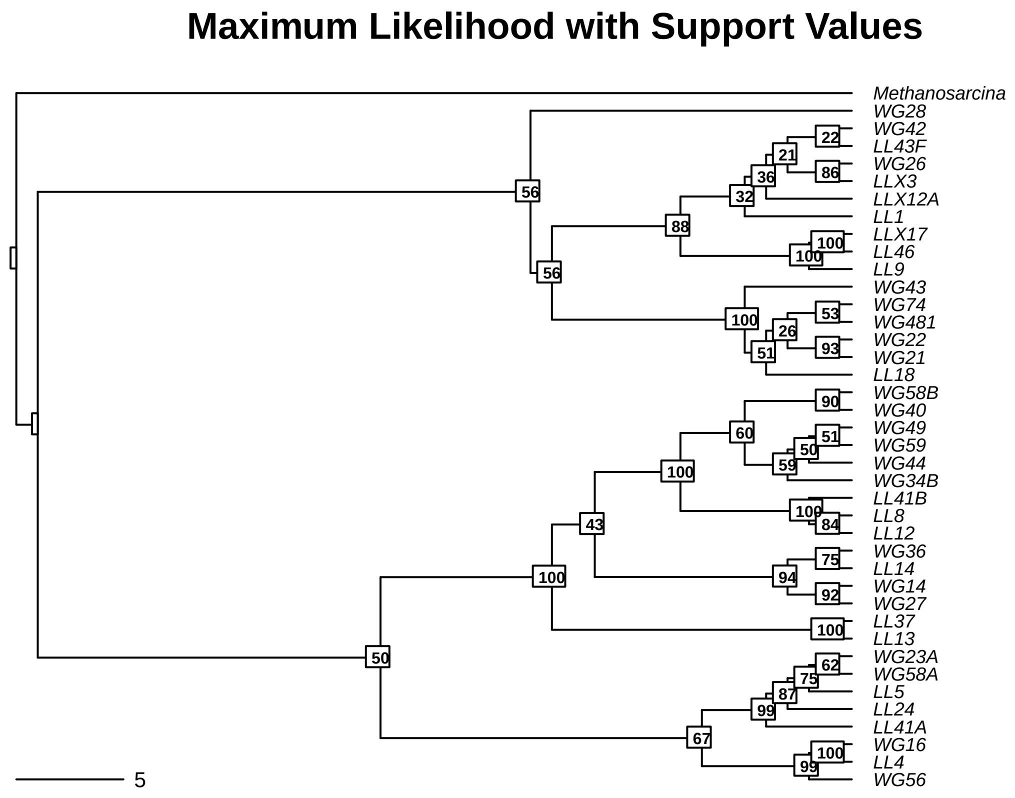
Answer 4a: I chose the Tamura 92 model (T92) to compare to the F84 model. This model shares the assumption of the Kimura K80 model that transition mutations are more likely to occur than transversions, but does not assume that nucleotides are all in equal frequencies. This model also assumes that substitution rates are the same across all sites. The F84 model has somewhat similar assumptions and allows for both different rates of transition and transversion substitutions and different nucleotide frequencies.
Answer 4b: My specific choice of using the Tamura model didn't affect how the phylogeny was built. I examined each of the relationships in the tree and found them to be the same in both models. I would assume that these phylogenies are similar because many of their assumptions about substitution models are roughly the same. I would expect that a model not accounting for different likelihoods of transition v. transversion mutations or assuming all nucleotides to exist in equal proportions might find some taxa more different from one another in a distance matrix than they might actually be in their evolutionary histories. This would likely result in a phylogeny with somewhat different branches.
Answer 4c: As I explained above, the T92 model I chose produces an identical tree to that made using the F84 model. Since F84 just assumes different substitution rates in transition and

transversion mutations and produces the same tree as T92 (which holds that transitions occur more frequently than transversions), I would take this as confirmation that transition mutation rates are higher than transversions. If transversion rates were higher, I would expect the tree built via the F84 model would look differently than it currently does (it may have taxa with transversion differences more closely related, under the assumption that these changes could have happened more recently).

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 ML tree
ml.boottree<-read.tree("./data/ml_tree/RAxML_bipartitions.T1")
#plotting bootstrap values
par(mar=c(1,1,2,1)+0.1)
plot.phylo(ml.boottree,type="phylogram",direction="right",show.tip.label=TRUE,
           use.edge.length=FALSE,cex=0.6,label.offset=1,
           main="Maximum Likelihood with Support Values")
add.scale.bar(cex=0.7)
nodelabels(ml.boottree$node.label,font=2,bg="white",frame="r",cex=0.5)
```



Question 5:

a) How does the maximum likelihood tree compare 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 maximum likelihood tree does not match the neighbor-joining tree in the handout. Following the root, the neighbor-joining tree immediately splits into one smaller clade and one large clade. In contrast, the maximum likelihood tree splits into two roughly equal branches following the root. Overall, the maximum likelihood tree seems organized into these two main groupings, while the neighbor-joining tree does not have groupings that are as distinct (more of a nested pattern). The differences between these phylogenies is likely the result of how they were created; maximum likelihood considers which specific nucleotides are in the sequence, while neighbor-joining only considers how un-alike the sequences are (via distance matrix). By making these considerations, maximum likelihood may find some taxa more or less genetically similar than neighbor-joining, so the relationships in the phylogeny would be different.
Answer 5b: We use bootstrapping when constructing phylogenetic trees because it is really difficult to know the true evolutionary relationships between groups of species. The algorithms used to build the trees could potentially make them differently each time, so bootstrapping uses resampling to see how common particular phylogenetic relationships are across many different attempts at creating a tree.
Answer 5c: The bootstrap values on the phylogeny indicate what percentage of trees created throughout the bootstrapping process have that same relationship between taxa. Branches with highest support have values above 95, medium tend to be around 70, and those at or below 50 are not highly supported.
Answer 5d: Some branches on the tree above have values below 50 indicating very low support. These include the node for WG42 and LL43F (22), the branch joining those two with WG26 and LLX3 (21), the branch joining that whole group to LLX12A (36), and the branch joining that entire set to LL1 (32). The branch containing the clade of Wg74, WG481, WG22, and WG21 had low support as well (26), as did the clade in the lower group on the phylogeny that is marked at its fork with a value of 43 (includes 13 taxa).
Answer 5e: I would not be inclined to trust these branches. I'm not quite sure what would be needed to resolve them more completely, but think it could be helpful to do the analysis with data from more taxa. There could be some evolutionary relationships with taxa not present in the dataset that would help to make a more complete and accurate phylogeny.

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 phosphorus data
p.data<-read.table("./data/p.isolates.raw.growth.txt",sep="\t",header=TRUE,
                   row.names=1)
#standardizing growth rates across strains
p.growth.std<-p.data/(apply(p.data,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.

```
#finding maximum growth rate for each isolate
umax<-(apply(p.data,1,max))
#making niche breadth function
levins.nb<-function(p_xi=""){
  p=0
  for(i in p_xi){
    p=p+i^2
  }
  nb=1/(length(p_xi)*p)
  return(nb)
}
#finding niche breadth for each isolate
p.nb<-as.matrix(levins.nb(p.growth.std))
#adding row names to matrix
p.nb<-setNames(as.vector(p.nb),as.matrix(row.names(p.data)))
```

C. Visualizing Traits on Trees

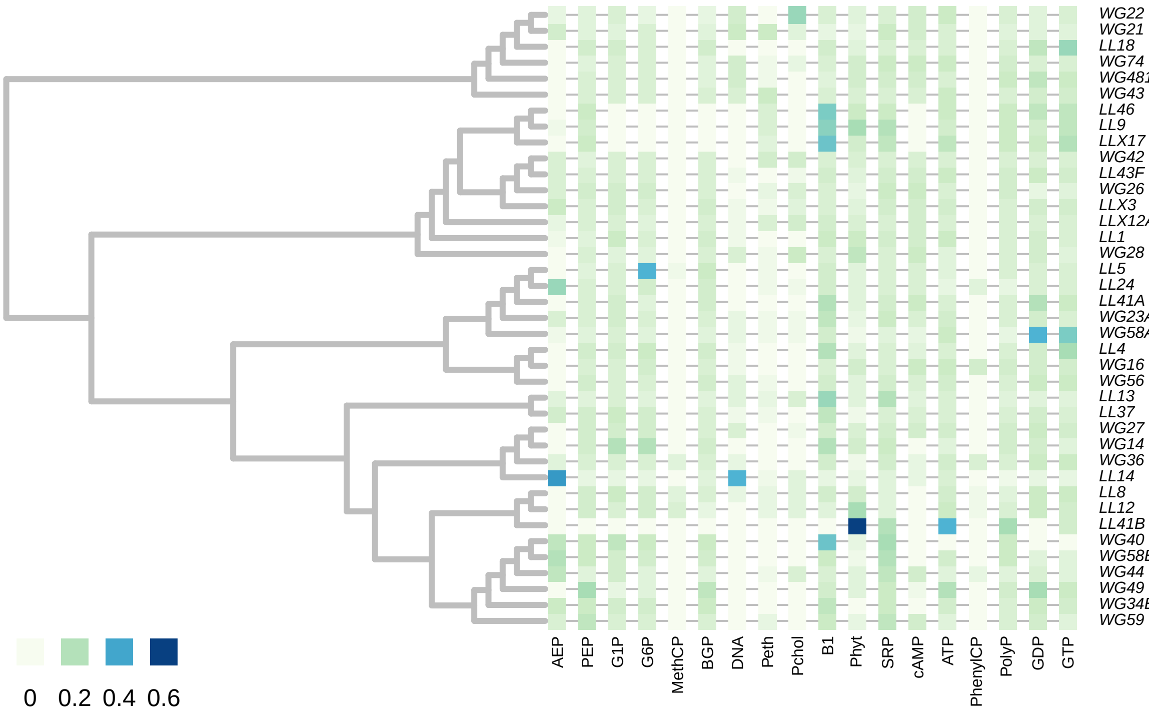
In the R code chunk below, do the following:
1. pick your favorite substitution model and make a Neighbor Joining tree,
2. define your outgroup and root the tree, and
3. remove the outgroup branch.

```
#making neighbor joining tree, will use t92 as substitution model
njtree<-bionj(seq.dist.t92)
#defining outgroup
outgroup<-match("Methanosarcina",njtree$tip.label)
#making rooted tree
njroot<-root(njtree,outgroup,resolve.root=TRUE)
#removing outgroup branch
njroot<-drop.tip(njroot,"Methanosarcina")
```

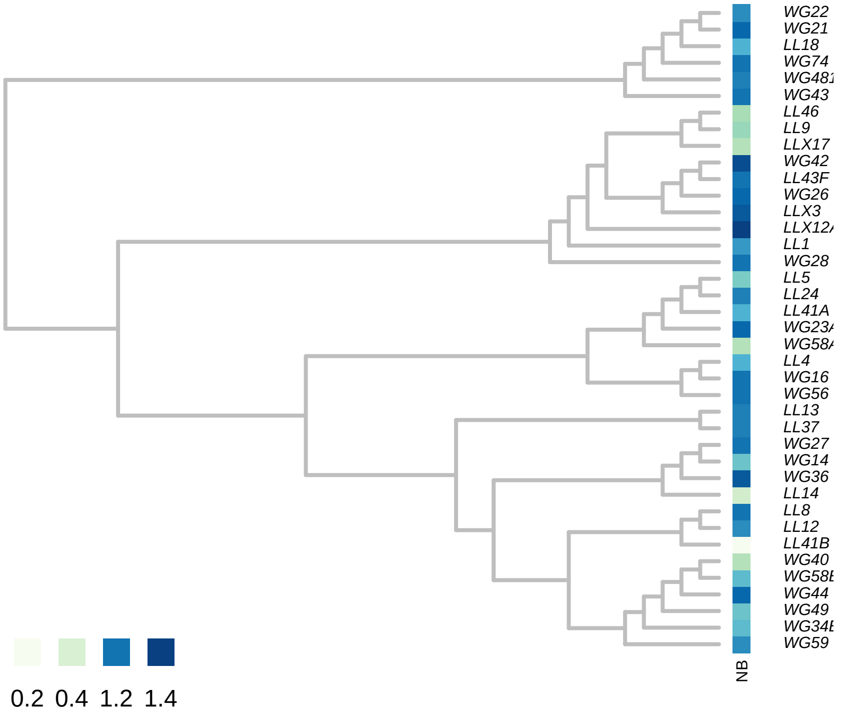
In the R code chunk below, do the following:
1. define a color palette (use something other than “YlOrRd”),
2. map the phosphorus traits onto your phylogeny,
3. map the *nb* trait on to your phylogeny, and
4. customize the plots as desired (use `help(table.phylo4d)` to learn about the options).

```
#making color palette
palette<-colorRampPalette(brewer.pal(9,"GnBu"))
#need to correct for zero branch lengths on tree
njplot<-njroot
njplot$edge.length<-njplot$edge.length+10^-1
#mapping phosphorus traits on phylogeny
par(mar=c(1,1,1,1)+0.1)
x<-phylo4d(njplot,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="gray",edge.width=3,
              box=FALSE,col=palette(25),pch=15,cex.symbol=1.25,ratio.tree=0.5,
              cex.legend=1.5,center=FALSE)
```



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



Question 6:

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

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

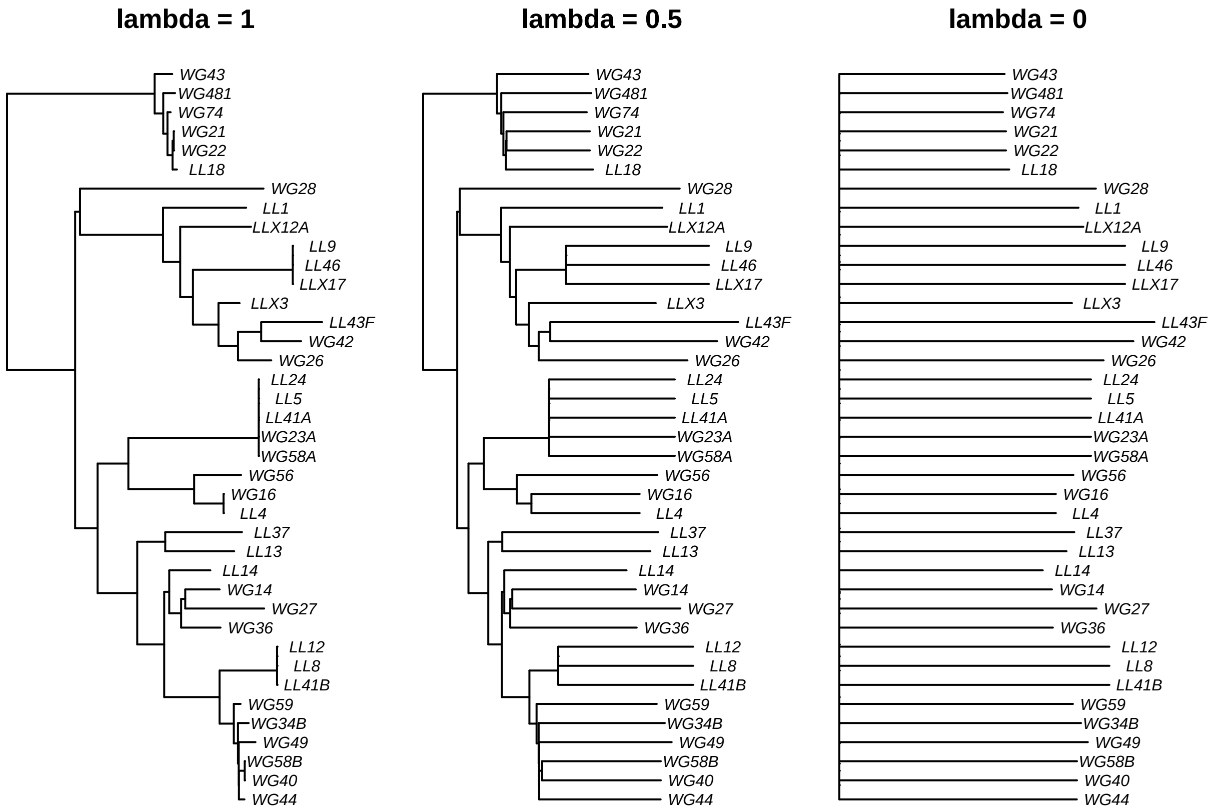
> ***Answer 6a***: I would hypothesize that species in the above phylogenies with narrower niches would have higher growth rates when grown with specific sources of phosphorus than species with wide niches would.
> ***Answer 6b***: If this hypothesis were supported, I would expect taxa with the smallest niche breadth values to have the largest values of growth rate, but these would only be on one or two sources of phosphorus. In addition to this, I would expect taxa with larger values for niche breadth to have similar values for growth rate across many phosphorus sources, but none of these would be large values like those seen in the specialist taxa.

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.

```
#making rescaled phylogenies
njlambda.5<-geiger::rescale(njroot,"lambda",0.5)
njlambda.0<-geiger::rescale(njroot,"lambda",0)
#plotting rescaled trees with original
layout(matrix(c(1,2,3),1,3),width=c(1,1,1))
par(mar=c(1,0.5,2,0.5)+0.1)
plot(njroot,main="lambda = 1",cex=0.7,adj=0.5)
plot(njlambda.5,main="lambda = 0.5",cex=0.7,adj=0.5)
plot(njlambda.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.

```
#comparing phylogenetic signal in trees
fitContinuous(njroot,p.nb,model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.020682
##  sigsq = 0.106809
##  z0 = 0.661298
##
##  model summary:
##  log-likelihood = 21.656475
##  AIC = -37.312951
##  AICc = -36.627236
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 47
##  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(njlambda.0,p.nb,model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.106713
##  z0 = 0.657740
##
##  model summary:
##  log-likelihood = 21.647816
##  AIC = -37.295632
##  AICc = -36.609918
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 83
##  frequency of best fit = 0.83
##
##  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 value for the untransformed tree (lambda = 1) is 0.020683, while it is 0.0 for the transformed tree (lambda = 0). ***Answer 7b***: The AIC score for the untransformed model (lambda = 1) is -37.312951, and the AIC score for the transformed model (lambda = 0) is -37.295632. I would choose the untransformed model because it has the more negative AIC score (lower AIC indicates better model); however, the AIC scores of the models should be greater than 2 for the models to be considered different, so these test statistics do not provide evidence

for difference between the two models.
Answer 7c: Because there is no evidence of a difference between the untransformed model and the transformed model (which lacks a phylogenetic signal), I would conclude that this result does not support the presence of a phylogenetic signal in the trait data.

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.

```
#correcting branch lengths for any zeros
njroot$edge.length<-njroot$edge.length+10^-7
#calculating Blomberg K for phosphorus resources
#first making 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")
#for loop to calculate blomberg K for every resource
for (i in 1:18){
  x<-setNames(as.vector(p.growth.std[,i]),row.names(p.data))
  out<-phylosignal(x,njroot)
  p.phylosignal[1:5,i]<-round(t(out),3)
}
#BH correction for p values
p.phylosignal[6,]<-round(p.adjust(p.phylosignal[4,],method="BH"),3)
#phylosignal matrix results
print(p.phylosignal)
```

```
##                    AEP      PEP      G1P      G6P  MethCP      BGP      DNA
## K                0.000    0.000    0.000    0.000   0.000    0.000    0.000
## PIC.var.obs   4373.157  664.095  948.941 5924.730 350.894  536.104  259.084
## PIC.var.mean  8246.925 1502.425 1860.581 3855.474 511.002 1767.463 5214.873
## PIC.var.P        0.256    0.070    0.121    0.738   0.339    0.023    0.001
## PIC.var.z       -0.836   -1.327   -1.149    0.809  -0.479   -1.760   -1.328
## PIC.P.BH         0.616    0.315    0.436    0.781   0.616    0.138    0.018
##                   Peth    Pchol       B1     Phyt      SRP     cAMP      ATP
## K                0.000    0.000    0.000    0.000    0.000    0.000    0.000
## PIC.var.obs   1446.463 2368.391 3517.018 9240.368 1307.025  690.723 4040.137
## PIC.var.mean  1827.993 3316.188 5351.943 9257.212 1614.082 3021.676 3182.192
## PIC.var.P        0.342    0.398    0.219    0.564    0.315    0.005    0.602
## PIC.var.z       -0.470   -0.549   -0.789   -0.002   -0.543   -2.528    0.373
## PIC.P.BH         0.616    0.651    0.616    0.722    0.616    0.045    0.722
##               PhenylCP    PolyP      GDP      GTP
## K                0.000    0.000    0.000    0.000
## PIC.var.obs   1224.017 1126.345 4473.878 2721.766
## PIC.var.mean   781.074 1213.096 3672.525 3004.671
## PIC.var.P        0.810    0.498    0.645    0.473
## PIC.var.z        0.973   -0.156    0.362   -0.203
## PIC.P.BH         0.810    0.690    0.726    0.690
```

```
#finding blomberg k for niche breadth
signal.nb<-phylosignal(p.nb,njroot)
signal.nb
```

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 3.435768e-06         49966.78              49356.48          0.558
##   PIC.variance.Z
## 1     0.02913382
```

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***: After considering the p-values that have been corrected for false discovery rate, it looks like there is a significant phylogenetic signal for growth on DNA (K = 0.0, p = 0.018) and cAMP (K = 0.0, p = 0.018). Neither niche breadth or any other resources showed a significant phylogenetic signal.
> ***Answer 8b***: The K values for growth on both DNA and cAMP are 0.0, which indicates overdispersion. This means that related species have growth associated with these resources that is less similar than one might expect by chance.

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.

```
#converting continuous data to categorical
p.growth.pa<-as.data.frame((p.data>0.01)*1)
#finding phosphorus use for every resource
apply(p.growth.pa,2,sum)
```

```
##      AEP      PEP      G1P      G6P   MethCP      BGP      DNA     Peth
##       20       38       35       34        3       35       19       21
##    Pchol       B1     Phyt      SRP     cAMP      ATP PhenylCP    PolyP
##       18       38       36       39       29       38        6       39
##      GDP      GTP
##       37       38
```

```
#adding column with isolate names
p.growth.pa$name<-rownames(p.growth.pa)
#combining tree and trait data
p.traits<-comparative.data(njroot,p.growth.pa,"name")
#calculating D for three phosphorus traits
phylo.d(p.traits,binvar=DNA,permut=10000)
```

```
##
## 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 :  njroot
##   Number of permutations :  10000
##
## Estimated D :  0.603615
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0285
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0043
```

```
phylo.d(p.traits,binvar=ATP,permut=10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  ATP
##   Counts of states:  0 = 1
##                      1 = 38
##   Phylogeny :  njroot
##   Number of permutations :  10000
##
## Estimated D :  4.329797
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.8459
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.0698
```

```
phylo.d(p.traits,binvar=Phyt,permut=10000)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  Phyt
##   Counts of states:  0 = 3
##                      1 = 36
##   Phylogeny :  njroot
##   Number of permutations :  10000
##
## Estimated D :  0.2971527
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.0531
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.4094
```

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.

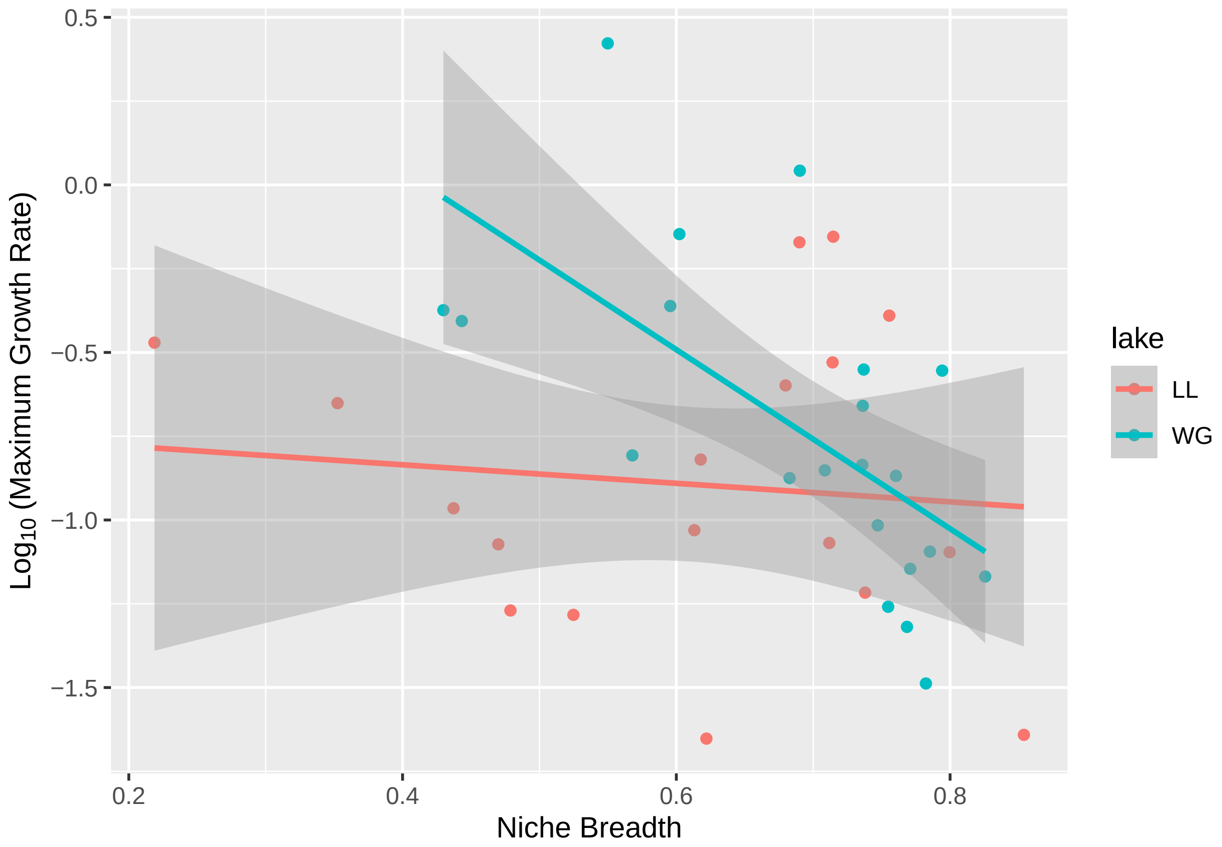
> ***Answer 9a***: I chose to test phosphorus growth traits for DNA, ATP, and Phyt. DNA has a D value of 0.6043. Using the Bonferroni-corrected p value of 0.025, the only significant probability for this was for trait dispersion being the result of a Brownian phylogenetic structure (p = 0.0044), indicating that the traits may be clustered (even though the D statistic is closer to 1, which would indicate the opposite). ATP has a D value of 4.6414. There was no significant probability for the trait dispersion here being either clustered or overdispersed. Phyt has a D value of 0.3011. Testing for growth traits on this resource also lacked evidence for significant probability of trait dispersion being clustered or overdispersed. ***Answer 9b***: These results matched those of the Blomberg K analysis in that growth on DNA as a phosphorus resource had a notable phylogenetic signal while the others (ATP, Phyt) did not. However, Blomberg K indicated potential evidence for overdispersion with growth traits on DNA, while trait dispersion analysis seems to have indicated that the traits may be more clustered than expected by chance. The D statistic for DNA is closer to 1 than to 0, which would indicate that the trait is more randomly dispersed than would be expected (as with overdispersion). Because of this, I'm wondering if I may just be interpreting the p-values incorrectly for the trait dispersion analysis.
> ***Answer 9c***: One factor that may result in differences in these metrics is the type of data they work with. The trait dispersion metric (D) considers categorical data, which here is whether or not the bacteria were able to grow using the given phosphorus source. The Blomberg K metric does not use categorical data and therefore may be able to capture variation in growth capabilities on specific phosphorus sources. These resource usage/growth traits may be polygenic and lead to continuous trait variation, which could be captured more accurately by the Blomberg K metric and lead to different conclusions about whether the traits are more clustered or randomly dispersed.

7) PHYLOGENETIC REGRESSION

In the R code chunk below, do the following:
1. Clean the resource use dataset to perform a linear regression to test for differences in maximum growth rate by niche breadth and lake environment, 2. Fit a linear model to the trait dataset, examining the relationship between maximum growth rate by niche breadth and lake environment, 3. Fit a phylogenetic regression to the trait dataset, taking into account the bacterial phylogeny

```r
#adding lake origin data to niche breadth data
nblake<-as.data.frame(as.matrix(p.nb))
nblake$lake<-rep("A")
for(i in 1:nrow(nblake)){
  ifelse(grepl("WG",row.names(nblake)[i]),nblake[i,2]<-"WG",nblake[i,2]<-"LL")
}
#adding column name for niche breadth values
colnames(nblake)[1]<-"NB"
#calculating maximum growth rate
umax<-as.matrix((apply(p.data,1,max)))
nblake<-cbind(nblake,umax)
#plotting linear model for trait data
ggplot(data=nblake,aes(x=NB,y=log10(umax),color=lake))+geom_point()+
  geom_smooth(method="lm")+xlab("Niche Breadth")+
  ylab(expression(Log[10]~"(Maximum Growth Rate)"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
#simple linear regression, for comparison purposes
fit.lm<-lm(log10(umax)~NB*lake,data=nblake)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = log10(umax) ~ NB * lake, data = nblake)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7557 -0.3108 -0.1077  0.3102  0.7800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7247     0.3852  -1.882   0.0682 .
## NB           -0.2763     0.6097  -0.453   0.6533
## lakeWG        1.8364     0.6909   2.658   0.0118 *
## NB:lakeWG    -2.3958     1.0234  -2.341   0.0251 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.418 on 35 degrees of freedom
## Multiple R-squared:  0.2595, Adjusted R-squared:  0.196
## F-statistic: 4.089 on 3 and 35 DF,  p-value: 0.01371
```

```r
#fitting phylogenetic regression
fit.plm<-phylolm(log10(umax)~NB*lake,data=nblake,njroot,model="lambda",boot=0)
summary(fit.plm)
```

```
##
## Call:
## phylolm(formula = log10(umax) ~ NB * lake, data = nblake, phy = njroot,
##     model = "lambda", boot = 0)
##
##    AIC logLik
##  40.42 -14.21
##
## Raw residuals:
##     Min      1Q  Median      3Q     Max
## -0.7539 -0.1858 -0.0711  0.3286  0.9615
##
## Mean tip height: 0.1833069
## Parameter estimate(s) using ML:
## lambda : 0.4990872
## sigma2: 0.9176995
##
## Coefficients:
##              Estimate    StdErr t.value p.value
## (Intercept) -0.891289  0.371109 -2.4017 0.02176 *
## NB          -0.011444  0.520678 -0.0220 0.98259
## lakeWG       1.435199  0.574466  2.4983 0.01732 *
## NB:lakeWG   -1.958340  0.844401 -2.3192 0.02634 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1954    Adjusted R-squared: 0.1264
##
## Note: p-values and R-squared are conditional on lambda=0.4990872.
```

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 while making a phylogenetic regression because while normal regressions assume that all of our data points are independent, this is not the case with species in a phylogeny. The traits being assessed by regression are present in species that are all (to varying degrees) evolutionarily related to one another, so these data are not independent.
Answer 10b: In a standard linear regression, the resitual errors are assumed to be independent and to follow a normal distribution. A phylogenetic regression on the other hand takes the phylogeny's branch lengths into consideration and the residuals follow a covariance matrix. ***Answer 10c***: In the standard linear model, the niche breadth term had a slope of -0.2763 (p = 0.6533), the lake term had a slope of 1.8364 (p = 0.0118), and the interaction term had a slope of -2.3958 (p = 0.0251). The overall model had an R^2 value of 0.2595 ($F_{3,35}$ = 4.089, p = 0.01371). The phylogenetic regression had a niche breadth term with a slope of -0.0114 (p = 0.98259), a lake

term with a slope of 1.4352 (p = 0.0173), and an interaction term with a slope of -1.9583 (p = 0.0263). The whole model had an R^2 value of 0.1954. Overall, it seems that the standard linear model provides a better fit for the data, but I would argue that the phylogenetic model is more appropriate. Since the standard regression does not account for phylogenetic relationships among the taxa and assumes they are all independent, it would likely show stronger relationships between the growth traits and niche breadth/lake location than there really are once those links are accounted for.

Answer 10d: If the phylogeny was accounted for, I might expect the relationship between photosynthetic output in a prairie plant community and subplot location within the prairie (assuming they all have the access to the same amount of light and nutrients, possibly in a controlled experimental plot) to disappear, mostly if not completely. Photosynthetic output (when given the same resources) would likely see variation based on whether a plant does C3 or C4 photosynthesis (which is a genetic trait), and within those groups other variation in photosynthetic output might depend on other traits such as root depth, stomata number, SLA, and many others.

7) SYNTHESIS

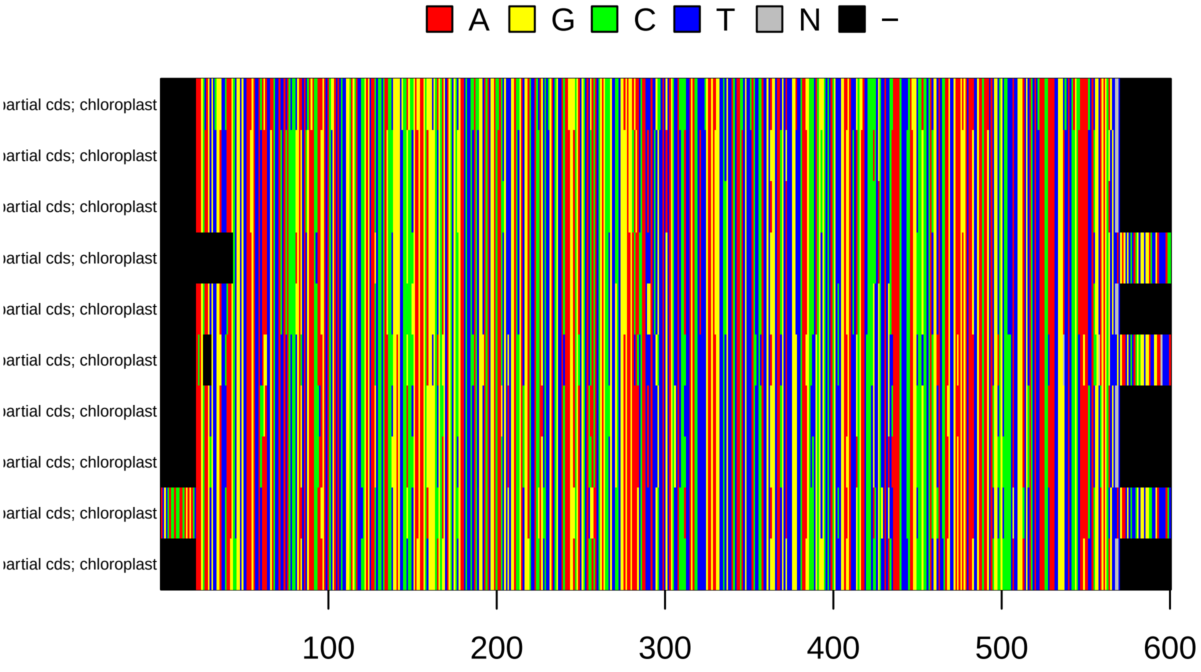
Work with members of your Team Project to obtain reference sequences for taxa in your study. Sequences for plants, animals, and microbes can found in a number of public repositories, but perhaps the most commonly visited site is the National Center for Biotechnology Information (NCBI) https://www.ncbi.nlm.nih.gov/. 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 the ribosomal RNA are good for molecular phylogenies. With your team members, do some research and identify a good candidate gene.

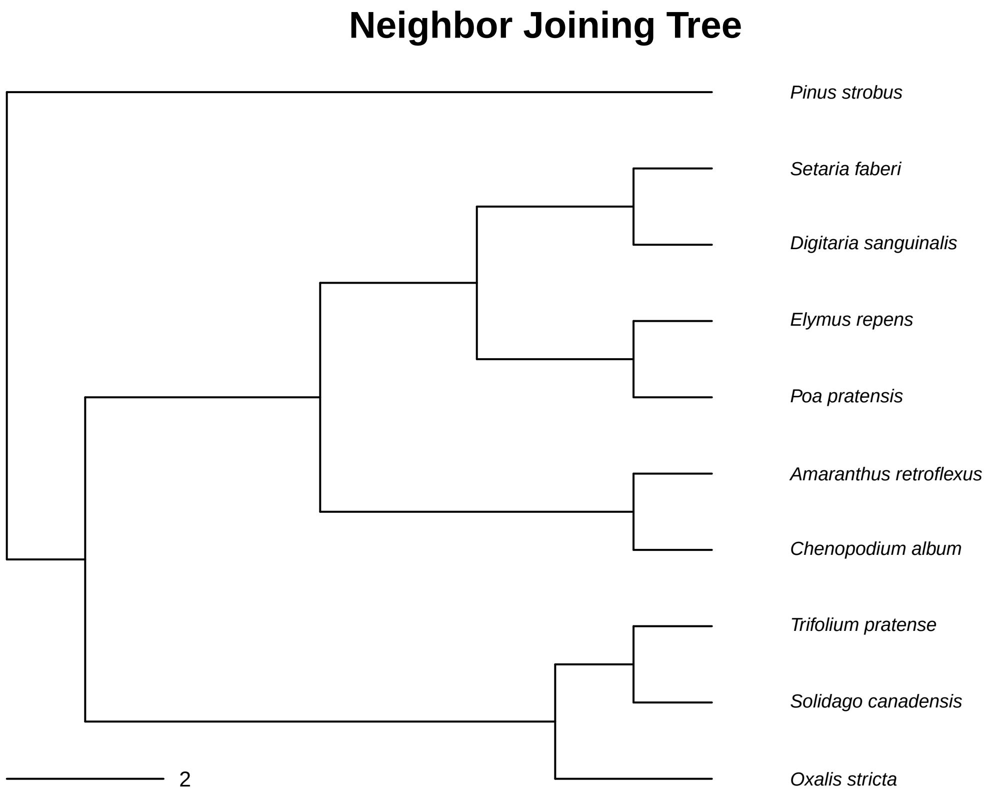
After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. Based on the decisions above and the output, does your tree jibe with 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
#importing plant sequences
plantseq<-readDNAStringSet("data/plant.sequence.data.fasta",format="fasta")
#aligning sequences
plant.read.aln<-msaMuscle(plantseq)
#saving/exporting alignment (might need later)
save.plant.aln<-msaConvert(plant.read.aln,type="bios2mds::align")
#converting alignment to DNAbin
plantDNAbin<-as.DNAbin(plant.read.aln)
```

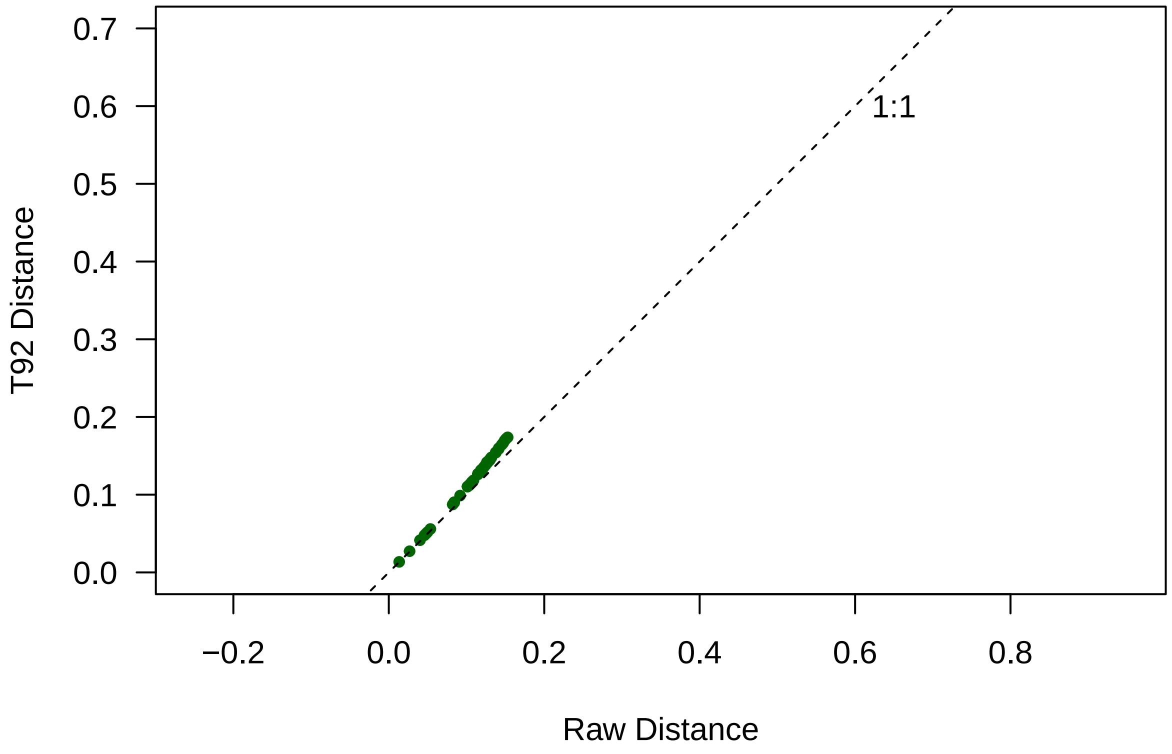
```
#specifying which range of base pairs to look at
plantwindow<-plantDNAbin[,1:600]
#visualizing sequence alignment
image.DNAbin(plantwindow,cex.lab=0.5)
```



```
#building neighbor joining tree
#making distance matrix of sequence data
distmtx.rawplant<-dist.dna(plantDNAbin,model="raw",pairwise.deletion=FALSE)
#making tree via neighbor joining
njtree.plant<-bionj(distmtx.rawplant)
#altering tip labels
njtree.plant$tip.label<-c("Pinus strobus","Setaria faberi",
                          "Digitaria sanguinalis","Elymus repens","Poa pratensis",
                          "Trifolium pratense","Amaranthus retroflexus",
                          "Chenopodium album","Solidago canadensis",
                          "Oxalis stricta")
#specifying outgroup
plantoutgroup<-match("Pinus strobus",njtree.plant$tip.label)
#rooting tree
njplantroot<-root(njtree.plant,plantoutgroup,resolve.root=TRUE)
#plotting neighbor joining tree
par(mar=c(1,1,2,1)+0.1)
plot.phylo(njplantroot,main="Neighbor Joining Tree","phylogram",
           use.edge.length=FALSE,direction="right",cex=0.6,label.offset=1)
add.scale.bar(cex=0.7)
```

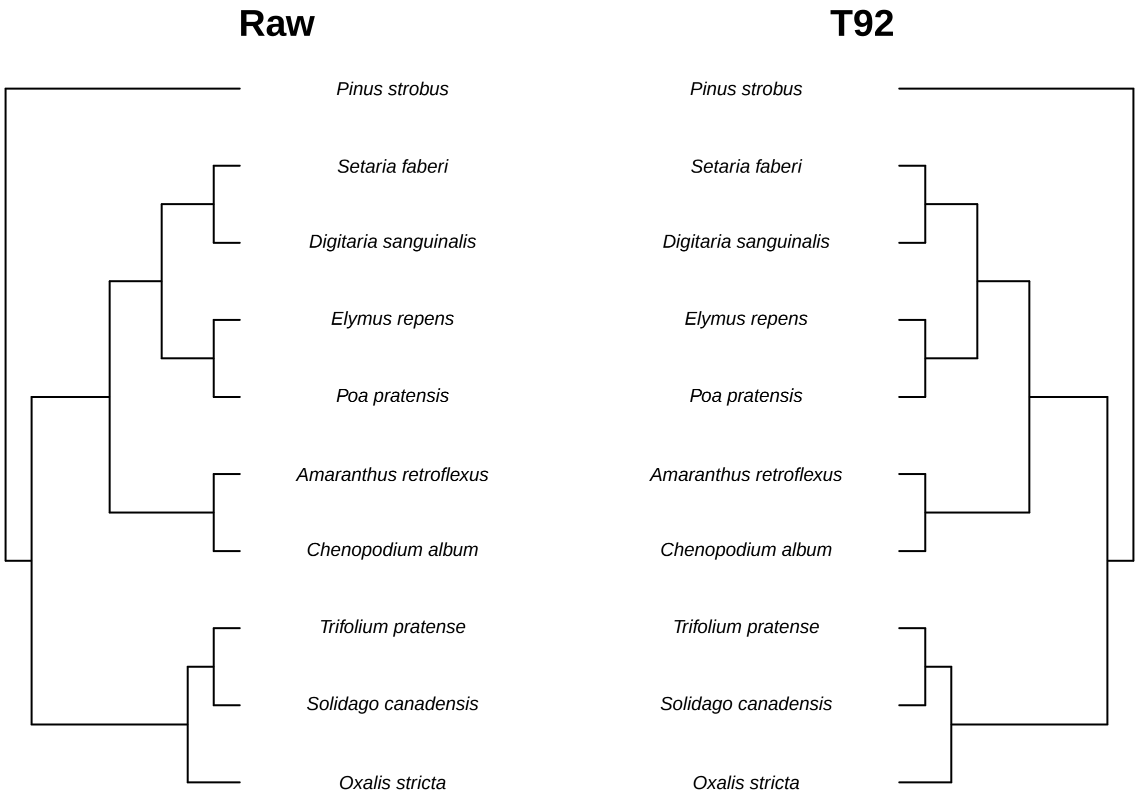


```
#Building T92 substitution model to plot/compare w/ raw model
#making T92 distance matrix
T92plantdist<-dist.dna(plantDNAbin,model="T92",pairwise.deletion=FALSE)
#plotting to compare distances in each substitution model
par(mar=c(5,5,2,1)+0.1)
plot(distmtx.rawplant,T92plantdist,pch=20,col="dark green",las=1,asp=1,
     xlim=c(0,0.7),ylim=c(0,0.7),xlab="Raw Distance",ylab="T92 Distance")
abline(b=1,a=0,lty=2)
text(0.65,0.6,"1:1")
```



```
#making neighbor joining tree for T92 model
T92njtree<-bionj(T92plantdist)
#altering tip labels
T92njtree$tip.label<-c("Pinus strobus","Setaria faberi",
                          "Digitaria sanguinalis","Elymus repens","Poa pratensis",
                          "Trifolium pratense","Amaranthus retroflexus",
                          "Chenopodium album","Solidago canadensis",
                          "Oxalis stricta")
#defining outgroup for model
t92outgroup<-match("Pinus strobus",T92njtree$tip.label)
#rooting t92 tree
t92root.plant<-root(T92njtree,t92outgroup,resolve.root=TRUE)

#plotting cophylogenetic tree for both models
layout(matrix(c(1,2),1,2),width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(njplantroot,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(t92root.plant,type="phylogram",direction="left",show.tip.label=TRUE,
           use.edge.length=FALSE,adj=0.5,cex=0.6,label.offset=2,main="T92")
```



> For this synthesis problem, we found sequences of the rbcl gene for ten plant species: nine from the KBS microplots, and one from white pine (*Pinus strobus*) to serve as an outgroup. I chose to make a phylogeny using neighbor joining and the Tamura model (T92), which does not assume equal nucleotide frequencies and accounts for the fact that transition mutations are more common than transversions. The resulting trees from both the T92 model and the raw model are actually identical. I believe this indicates that the rbcl sequeunces from the chloroplasts of these plants are not evolutionarily related enough for the changes in assumptions to cause a rearrangement of the phylogeny.

Overall, I think that this phylogeny is a good representation of the evolutionary relationships of these organisms. The white pine is a clear outgroup; it is a gymnosperm, while plants from our KBS community are all angiosperms. The groupings among the other plants also seem fairly consistent. All of the true grasses (*S. faberi*, *D. sanguinalis*, *E repens*, *P. pratensis*) are within their own clade, the amaranths (*A. retroflexus*, *C. album*) have their own group, and the other plants form their own grouping (*T. pratense*, *S. canadensis*, *O. stricta*). The main confusing point here is that the amaranths (dicots) share a branch with the true grasses (which are monocots) instead of the other dicots. I wouldn't be sure of the exact answer without further reading, but this could be the result of the ancestor of the amaranths developing a dicotyledon growth form independently of an ancestor of the other dicots after some prior evolutionary split.

The main things that I think could improve this tree would be to include either more species or to collect sequence data directly from the KBS communities. Including more species that may be more closely related to some of these current species could potentially result in a more "complete" phylogeny that is a better representation of the evolutionary relationships among the members of the community. Additionally, collecting sequence data from KBS for these plants could result in a more accurate picture of the evolutionary relationships in the actual community we're looking at. It's entirely possible that different local adaptations, introgressions from other species, or other phenomena have made relationships in the very local scale of the KBS prairie community distinct from what we might find in a reference sequence (although it's entirely possible that this information would be a better reflection of a particular abiotic environment than it would of different phylogenetic relationships).