11. Worksheet: Phylogenetic Diversity - Traits

Caroline Edwards; Z620: Quantitative Biodiversity, Indiana University

30 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 28th, 2021 before 12:00 PM (noon)**.

1) SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:
1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/11.PhyloTraits*" folder, and
4. load all of the required R packages (be sure to install if needed).

```
rm(list = ls())
getwd()
```

```
## [1] "/Users/carolineedwards/quant_bio/GitHub/QB2021_Edwards/2.Worksheets/11.PhyloTraits"
```

```
setwd("~/quant_bio/GitHub/QB2021_Edwards/2.Worksheets/11.PhyloTraits/")
package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo', 'geiger', 'picante', 'stats', 'RColorBrewer
for (package in package.list) {
  if (!require(package, character.only=TRUE, quietly=TRUE)) {
    install.packages(package)
    library(package, character.only=TRUE)
  }
}
```

```
## Warning: package 'seqinr' was built under R version 3.6.2
```

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

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

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

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

```
## Warning: package 'ade4' was built under R version 3.6.2
```

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

```
## Warning: package 'geiger' was built under R version 3.6.2
```

```
## Warning: package 'picante' was built under R version 3.6.2
```

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

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

```
## This is vegan 2.5-6

##
## Attaching package: 'nlme'

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

## Warning: package 'pmc' was built under R version 3.6.2

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

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

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'phangorn'

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

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.

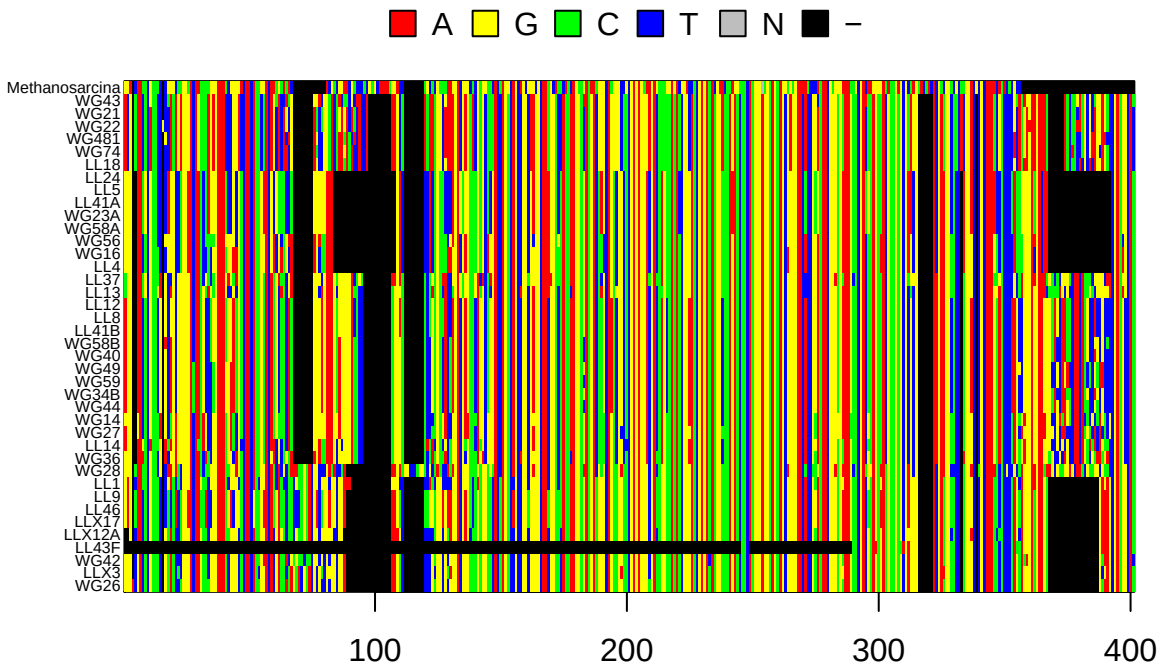
3) SEQUENCE ALIGNMENT

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 just has nucleotide sequences for each sample whereas .afa file has spaces within the sequence that align it with the rest of the sequences, so each site is homologous.

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.

```
read.aln<-read.alignment(file="./data/p.isolates.afa", format="fasta")
p.DNAbin<-as.DNAbin(read.aln)
window<-p.DNAbin[, 100:500]
image.DNAbin(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: ~400 bp ***Answer 2b***: 0-75 and 120-320 seem to have very few gaps, so they would be good for phylogenetic inferences, because there are the most comparisons that can be made at each site. If gaps exist at some site for some samples, inferences about relatedness cannot be made from those sites for those samples.

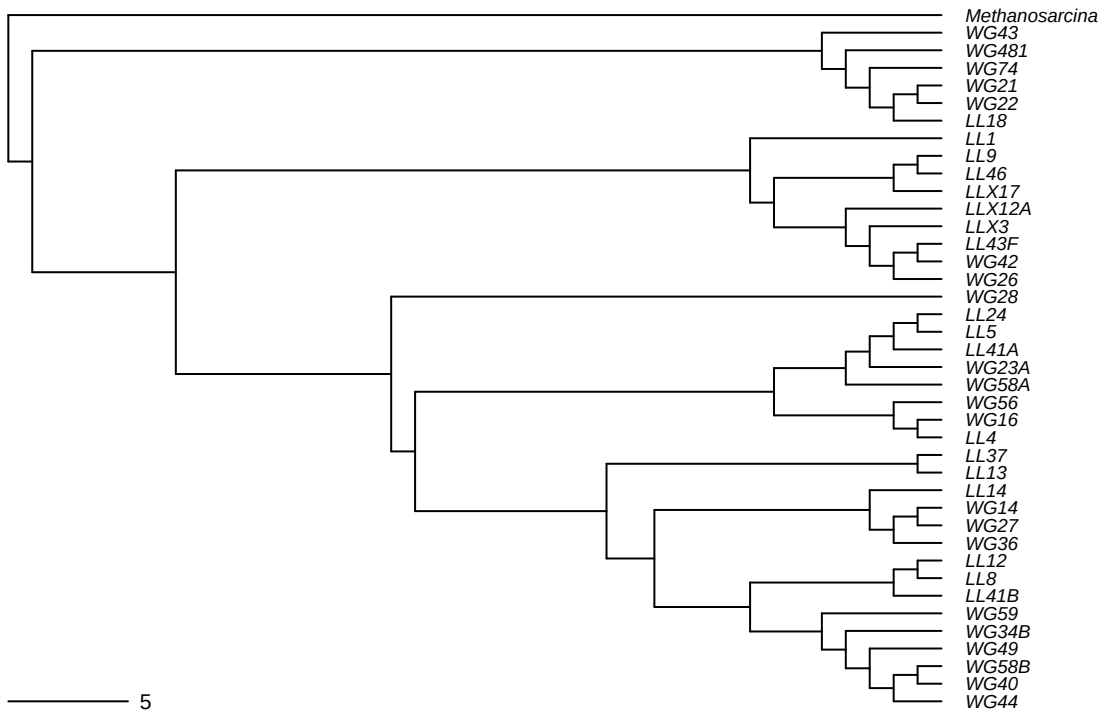
4) MAKING A PHYLOGENETIC TREE

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.

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



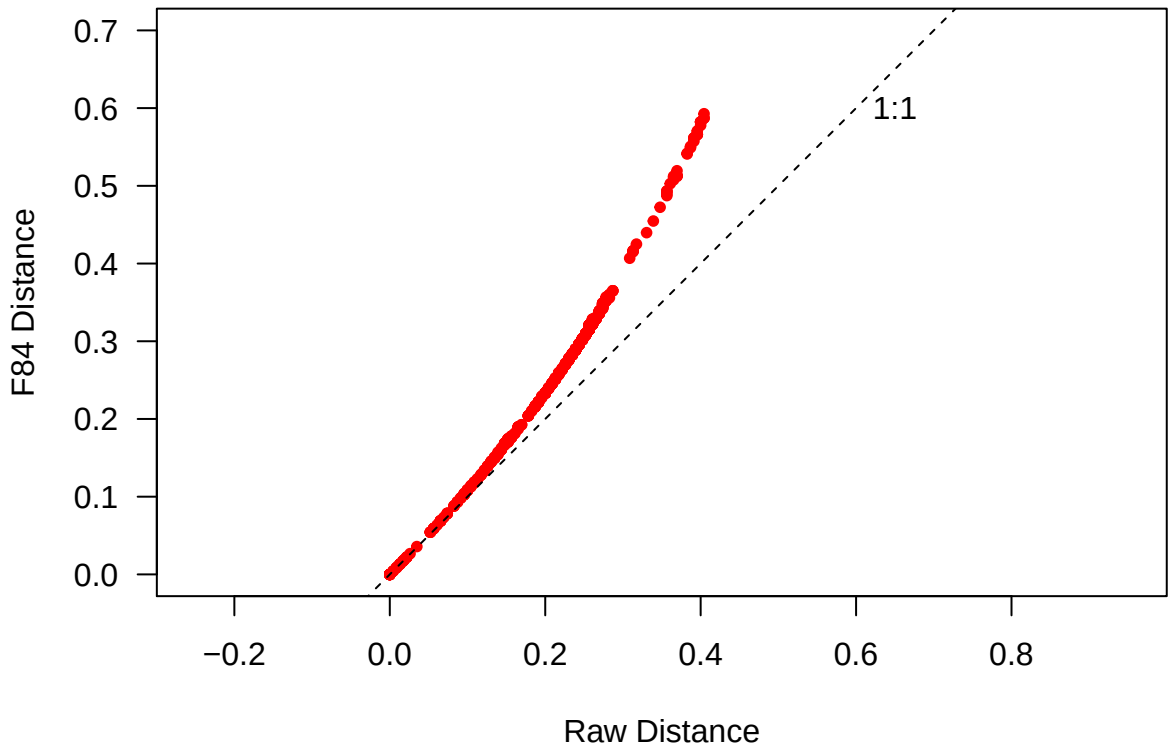
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: An advantage of neighbor joining trees are that they are relatively easy computationally, but a disadvantage is that it isn't as sophisticated as methods that use likelihood functions to determine the probability of the data given some model like in maximum likelihood and bayesian methods.

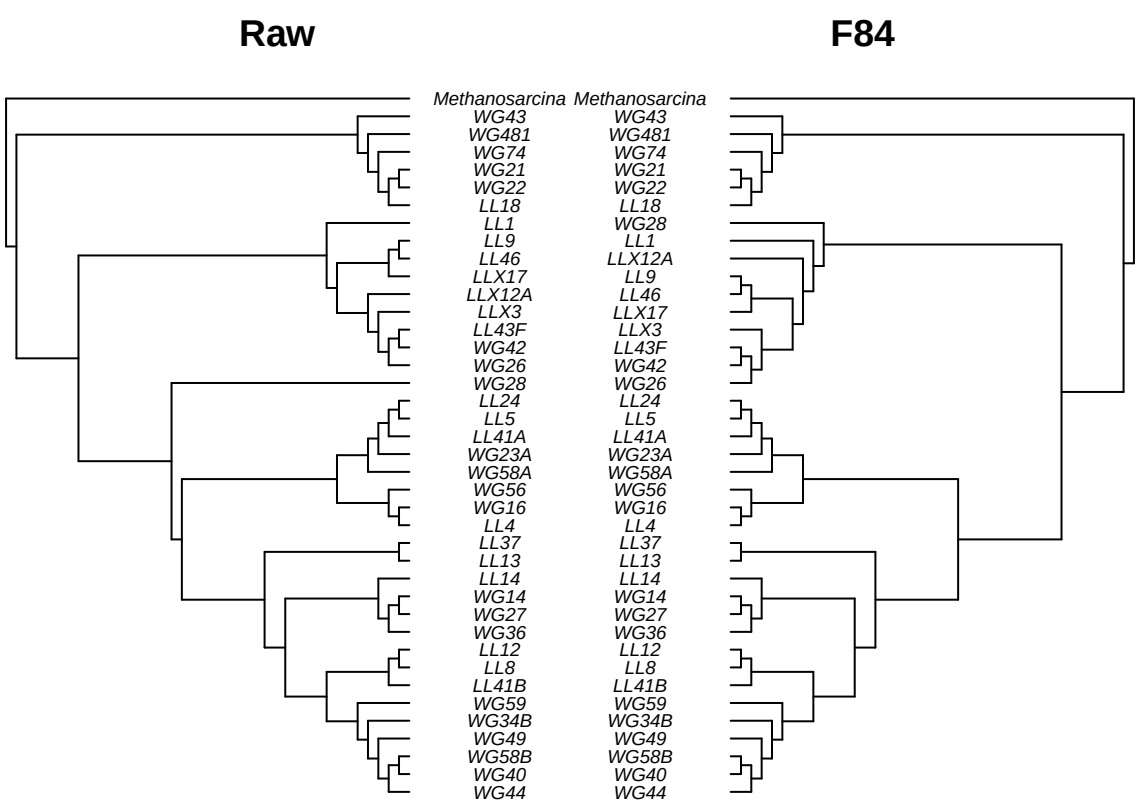
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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

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




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

layout(matrix(c(1,2),1,2), width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(raw.rooted, type="phylogram", direction="right", show.tip.label = TRUE,
           use.edge.length = FALSE, adj=0.5, cex=0.6, label.offset = 2, main="Raw")

par(mar=c(1,0,2,1))
plot.phylo(F84.rooted, type="phylogram", direction="left", show.tip.label = TRUE,
           use.edge.length = FALSE, adj=0.5, cex=0.6, label.offset = 2, main="F84")
```

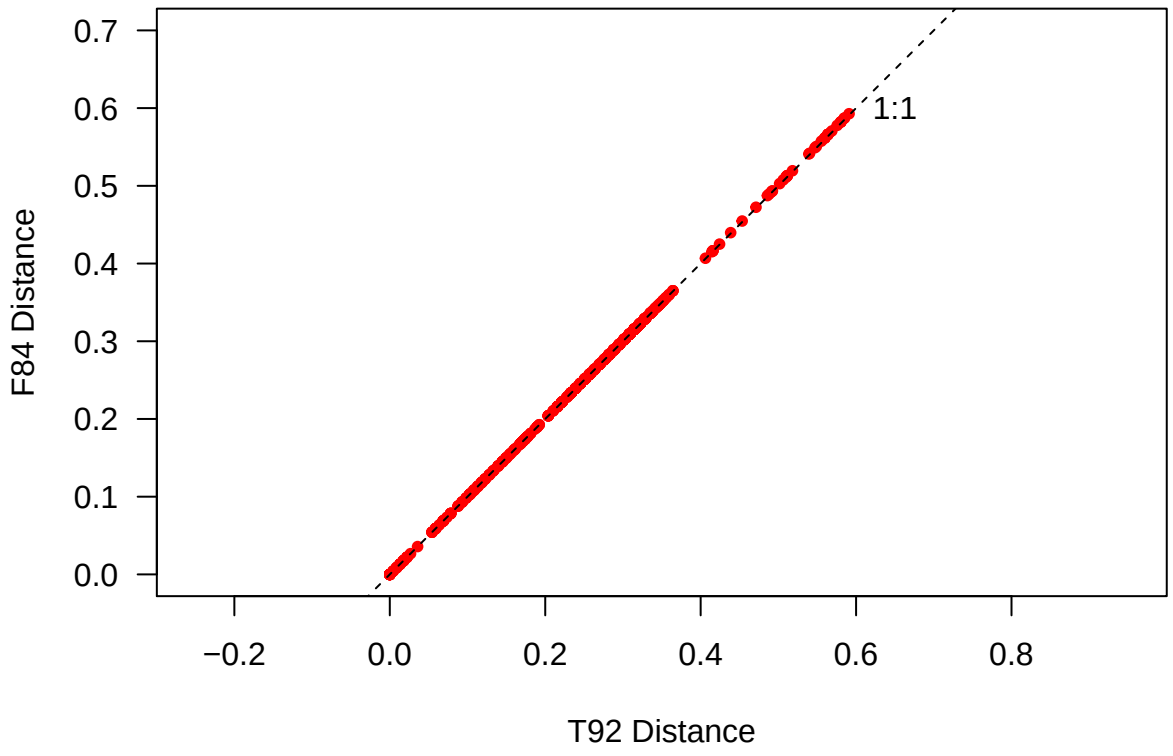



```
dist.topo(raw.rooted, F84.rooted, method="score")
```

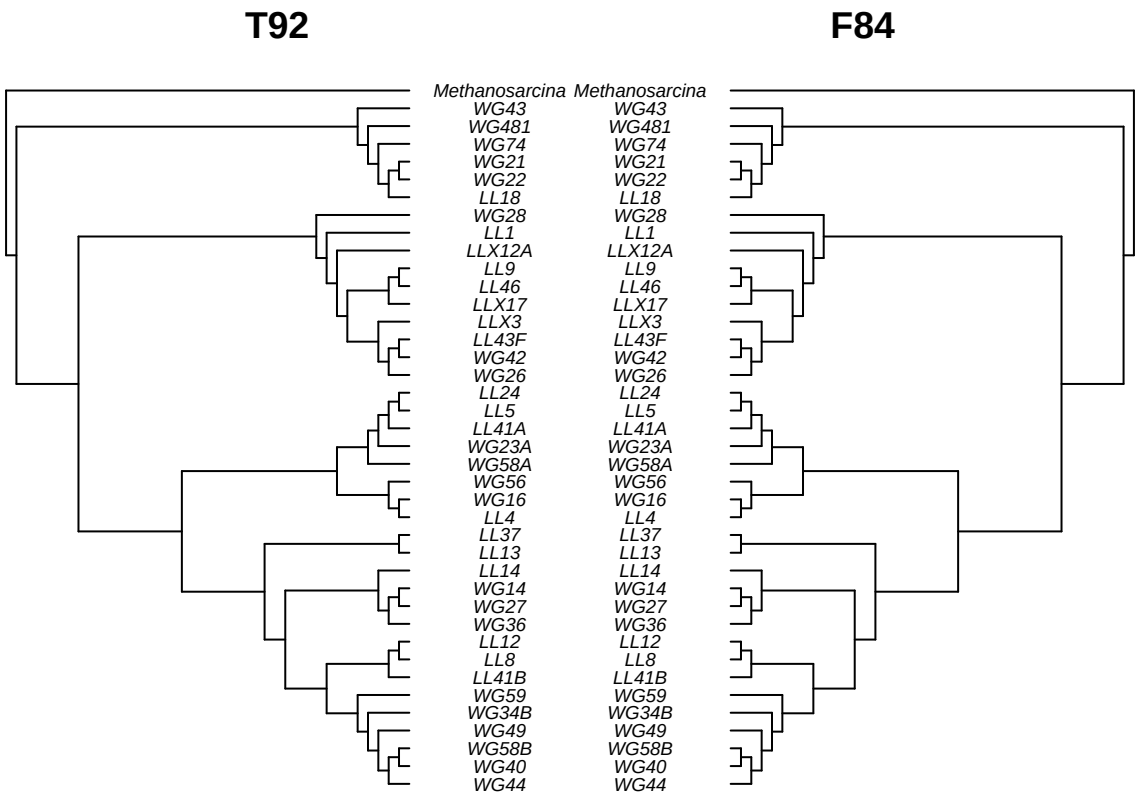
```
##            tree1
## tree2 0.04387426
```

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

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

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




```
T92.tree<-bionj(seq.dist.T92)
T92.outgroup<-match("Methanosarcina", T92.tree$tip.label)
T92.rooted<-root(T92.tree, T92.outgroup, resolve.root=TRUE)

layout(matrix(c(1,2),1,2), width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(T92.rooted, type="phylogram", direction="right", show.tip.label = TRUE,
           use.edge.length = FALSE, adj=0.5, cex=0.6, label.offset = 2, main="T92")

par(mar=c(1,0,2,1))
plot.phylo(F84.rooted, type="phylogram", direction="left", show.tip.label = TRUE,
           use.edge.length = FALSE, adj=0.5, cex=0.6, label.offset = 2, main="F84")
```




```
dist.topo(T92.rooted, F84.rooted, method="score")
```

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

Question 4:

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

Answer 4a: I chose the substitution model T92, which is a model that assumes equal frequencies of nucleotides, higher probabilities of transitions than transversions, and accounts for G+C content. The F84 model also assumes different rates of transitions and transversions, but different than T92, it allows for differences in base frequencies. ***Answer 4b***: Choosing the T92 model compared to F84 model does not change the phylogenetic reconstruction. ***Answer 4c***: n/a

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

```
#library(phangorn)
#p.DNAbin.phyDat<-read.phyDat("./data/p.isolates.afa", format="fasta", type="DNA")
#fit<-pml(nj.rooted, data=p.DNAbin.phyDat)
#fitJC<-optim.pml(fit,TRUE)
#fitGTR<-optim.pml(fitGTR, model="GTR", optInv=TRUE, optGamma=TRUE)
#anova(fitJC,fitGTR)
#AIC(fitJC)
#AIC(fitGTR)
#
#bs=bootstrap.pml(fitJC, bs=100, optNni=TRUE,
#                 control=pml.control(trace=0))
```

Question 5:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches?

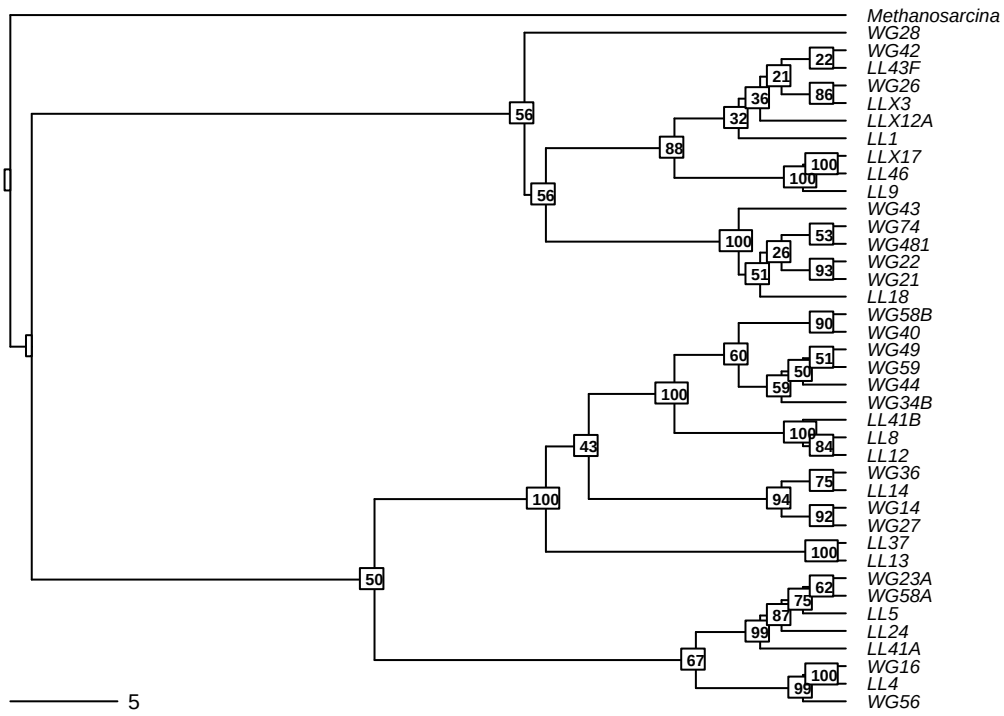
Answer 5a: The neighbor joining tree mostly grouped based on the lake they were sampled from, but the maximum likelihood tree recovered clades that were much more mixed with regard to location. ***Answer 5b***: We bootstrap trees to give some quantitative value that measures the how reliable the tree is by resampling your data and measuring how often each relationship is still recovered. ***Answer 5c***: The bootstrap values tell you how often the relationship/topology was found when your alignment data was resampled randomly and used to build a tree. It acts as a support metric. ***Answer 5d***: Branches that have been found to differ in topology in a majority of the topologies from the resampled data will have low support, proportional to the amount of times they have been recovered. ***Answer 5e***: Branches with low support should not be trusted, at least not as much as relationships with higher support, and if the support is really low it should probably be treated as equivocal.

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.

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



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.

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

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

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

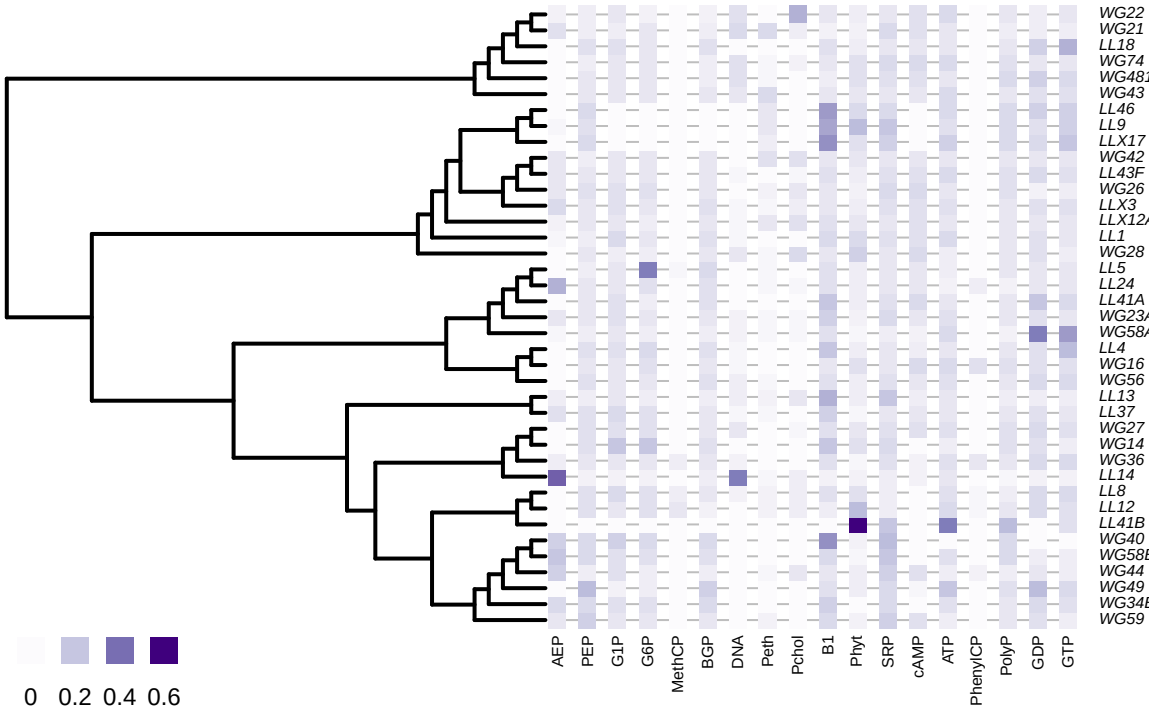
C. Visualizing Traits on Trees

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

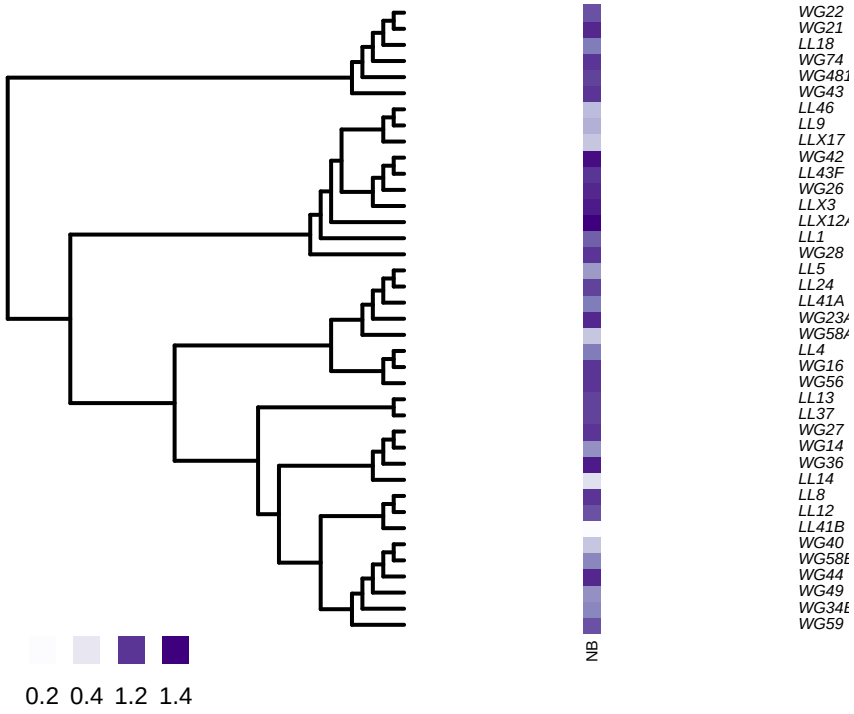
```
nj.tree<-bionj(seq.dist.F84)
outgroup<-match("Methanosarcina", nj.tree$tip.label)
nj.rooted<-root(nj.tree, outgroup, resolve.root = TRUE)
nj.rooted<-drop.tip(nj.rooted, "Methanosarcina")
```

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).

```
mypalette<-colorRampPalette(brewer.pal(9,"Purples"))

par(mar=c(1,1,1,1)+0.1)
x<-phylo4d(nj.rooted, p.growth.std)
table.phylo4d(x, treetype = "phylo", symbol="colors", show.node=TRUE,
              cex.label = 0.5, scale=FALSE, use.edge.length=FALSE,
              edge.color="black", edge.width=2, box=FALSE,
              col=mypalette(25), pch=15, cex.symbol = 1.25,
              ratio.tree = 0.5, cex.legend = 1.5, center=FALSE)
```




```
par(mar=c(1,5,1,5)+0.1)
x.nb<-phylo4d(nj.rooted, nb)
table.phylo4d(x.nb, treetype="phylo", symbol="colors", show.node=TRUE,
              cex.label = 0.5, scale=FALSE, use.edge.length=FALSE,
              edge.color="black", edge.width=2, box=FALSE,
              col=mypalette(25), pch=15, cex.symbol = 1.25, var.label=("     NB"),
              ratio.tree = 0.5, 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: If there is a gnerealist-specialist trade-off, then you would expect to see overdispersion because there might be some specialists related to a generalist where those specialists are going to be more likely to have divergent traits with closely related species to gain access to some specialized resource pool. ***Answer 6b***: You would see groups of specialist with overdispersion.

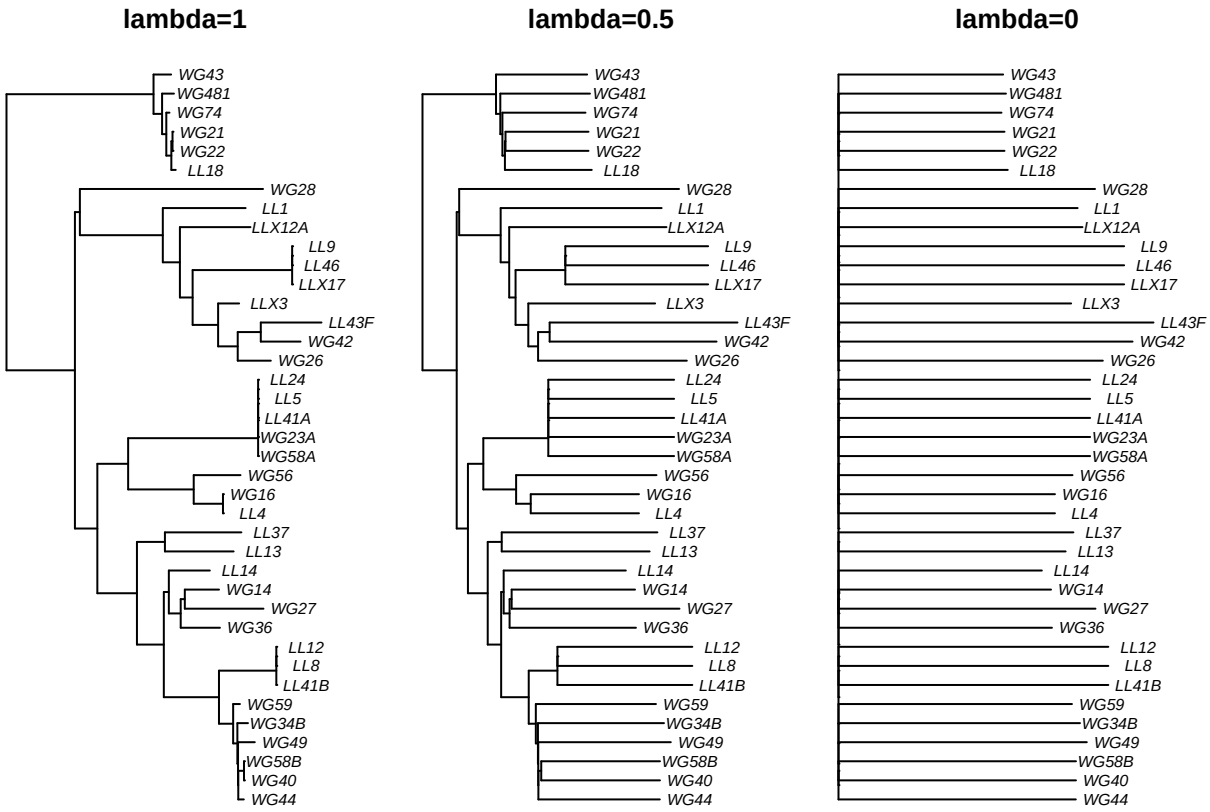
6) HYPOTHESIS TESTING

A) Phylogenetic Signal: Pagel's Lambda

In the R code chunk below, do the following:
1. create two rescaled phylogenetic trees using lambda values of 0.5 and 0,
2. plot your original tree and the two scaled trees, and
3. label and customize the trees as desired.

```
nj.lambda.5<-rescale(nj.rooted, "lambda", 0.5)
nj.lambda.0<-rescale(nj.rooted, "lambda", 0)
layout(matrix(c(1,2,3),1,3), width=c(1,1,1))
```

```
par(mar=c(1,0.5,2,0.5)+0.1)
plot(nj.rooted, main="lambda=1", cex=0.7, adj=0.5)
plot(nj.lambda.5, main="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.

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

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.106395
##  z0 = 0.657777
##
##  model summary:
##  log-likelihood = 21.652293
##  AIC = -37.304587
##  AICc = -36.618872
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 46
```

```
##  number of iterations with same best fit = NA
##  frequency of best fit = NA
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

```
fitContinuous(nj.lambda.0, nb, model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.000000
##  sigsq = 0.106395
##  z0 = 0.657777
##
##  model summary:
##  log-likelihood = 21.652293
##  AIC = -37.304587
##  AICc = -36.618872
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 84
##  frequency of best fit = 0.84
##
##  object summary:
##  'lik' -- likelihood function
##  'bnd' -- bounds for likelihood search
##  'res' -- optimization iteration summary
##  'opt' -- maximum likelihood parameter estimates
```

Question 7: There are two important outputs from the `fitContinuous()` function that can help you interpret the phylogenetic signal in trait data sets. a. Compare the lambda values of the untransformed tree to the transformed (lambda = 0). b. Compare the Akaike information criterion (AIC) scores of the two models. Which model would you choose based off of AIC score (remember the criteria that the difference in AIC values has to be at least 2)? c. Does this result suggest that there's phylogenetic signal?

> ***Answer 7a***: Lambda for the original tree is 0.020848, whereas the lambda for the transformed tree is 0. ***Answer 7b***: The AIC score for the originial tree is -37.322208 and the AIC score for the transformed tree is -37.304587. These AIC scores are so similar that it would be hard to decide which model is better based of off the this. ***Answer 7c***: This result suggests that there isn't a phylogenetic signal because when the tree is transformed to have a lambda of 0, all phylogenetic signal is removed, so if the AIC scores are not very different between the two models, then they probably both have almost no 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.

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

for (i in 1:18){
  x<-as.matrix(p.growth.std[ ,i, drop=FALSE])
  out<-phylosignal(x, nj.rooted)
  p.phylosignal[1:5,i]<-round(t(out),3)
}

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

```
##                                        AEP      PEP      G1P      G6P
## K                                    0.000    0.000    0.000    0.000
## PIC.var.obs                       4373.157  664.095  948.941 5924.730
## PIC.var.mean                      8400.625 1532.900 1848.492 3717.425
## \n                       PIC.var.P    0.244    0.081    0.119    0.765
## PIC.var.z                           -0.844   -1.312   -1.164    0.902
## PIC.P.BH                             0.617    0.364    0.428    0.810
##                                     MethCP      BGP      DNA     Peth
## K                                    0.000    0.000    0.000    0.000
## PIC.var.obs                        350.894  536.104  259.084 1446.463
## PIC.var.mean                       506.709 1769.492 5207.660 1854.811
## \n                       PIC.var.P   0.343    0.026    0.002    0.339
## PIC.var.z                           -0.458   -1.726   -1.292   -0.489
## PIC.P.BH                             0.617    0.156    0.036    0.617
##                                      Pchol       B1     Phyt      SRP
## K                                    0.000    0.000    0.000    0.000
## PIC.var.obs                       2368.391 3517.018 9240.368 1307.025
## PIC.var.mean                      3261.757 5489.071 8951.690 1588.177
## \n                       PIC.var.P    0.398    0.220    0.594    0.341
## PIC.var.z                           -0.509   -0.817    0.037   -0.482
## PIC.P.BH                             0.651    0.617    0.736    0.617
##                                       cAMP      ATP PhenylCP    PolyP
## K                                    0.000    0.000    0.000    0.000
## PIC.var.obs                        690.723 4040.137 1224.017 1126.345
## PIC.var.mean                      3009.299 3122.870  777.346 1220.681
## \n                       PIC.var.P    0.005    0.613    0.833    0.497
## PIC.var.z                           -2.519    0.400    0.947   -0.168
## PIC.P.BH                             0.045    0.736    0.833    0.688
##                                        GDP      GTP
## K                                    0.000    0.000
## PIC.var.obs                       4473.878 2721.766
## PIC.var.mean                      3573.385 2995.745
## \n                       PIC.var.P    0.658    0.483
## PIC.var.z                            0.414   -0.202
## PIC.P.BH                             0.740    0.688
```

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

```
##              K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 3.427719e-06         49966.78              50812.76          0.534
##   PIC.variance.Z
## 1    -0.04206929
```

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: Phylogenetic signal is low and niche breadth does not have significant phylogenetic signal. The phosphorous resources that are statistically significant are DNA, cAMP, and BGP, however all K values in the analysis were 0 (not sure if this is an error). ***Answer 8b***: The results are suggestive of overdispersion.

C. Calculate Dispersion of a Trait

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

```
p.growth.pa<-as.data.frame((p.growth>0.01)*1)
apply(p.growth.pa, 2, sum)
```

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

```
p.growth.pa$name<-rownames(p.growth.pa)
p.traits<-comparative.data(nj.rooted, p.growth.pa, "name")
phylo.d(p.traits, binvar=AEP)
```

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  AEP
##   Counts of states:  0 = 19
##                      1 = 20
```

```
##   Phylogeny :  nj.rooted
##   Number of permutations :  1000
##
## Estimated D :  0.4634094
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.005
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.023
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  PhenylCP
##   Counts of states:  0 = 33
##                      1 = 6
##   Phylogeny :  nj.rooted
##   Number of permutations :  1000
##
## Estimated D :  0.8721809
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.274
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.015
```

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

```
##
## 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 :  nj.rooted
##   Number of permutations :  1000
##
## Estimated D :  0.3050092
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.054
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.397
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  PEP
##   Counts of states:  0 = 1
##                      1 = 38
##   Phylogeny :  nj.rooted
##   Number of permutations :  1000
##
## Estimated D :  -0.5488131
```

```
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.3
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.536
```

Question 9: Using the estimates for D and the probabilities of each phylogenetic model, answer the following questions:

a. Choose three phosphorus growth traits and test whether they are significantly clustered or overdispersed?

b. How do these results compare the results from the Blomberg's K analysis?

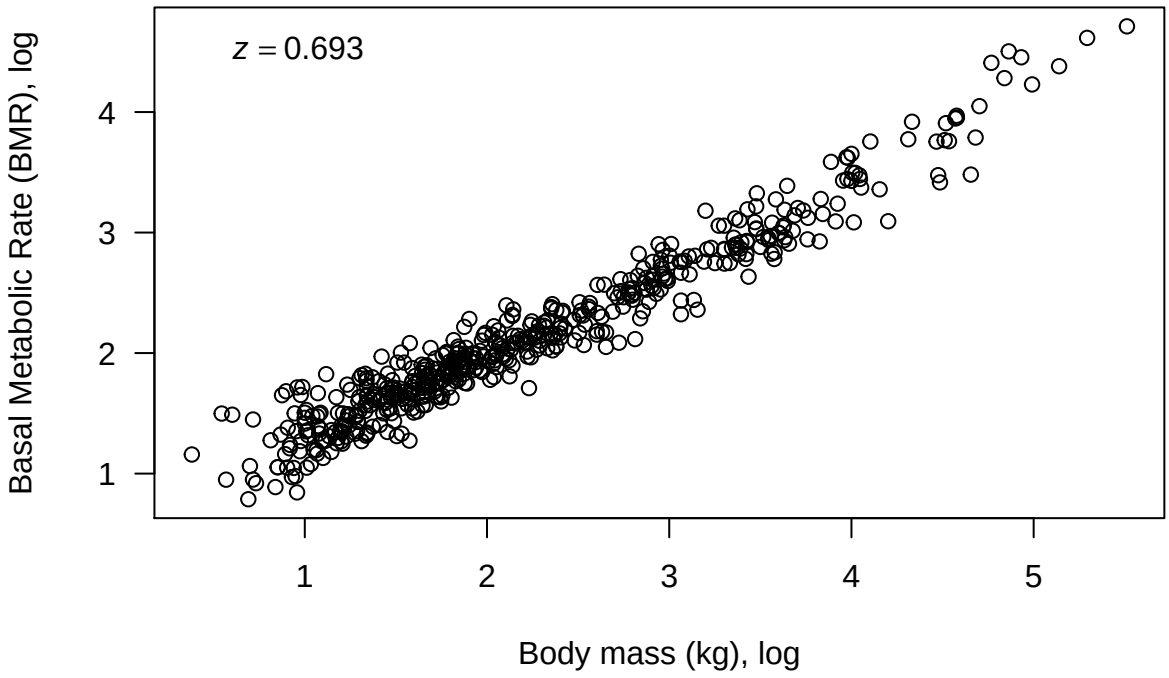
c. Discuss what factors might give rise to differences between the metrics.

Answer 9a: AEP, PhenylCP, Phyt, and PEP are all calculated as overdispersed traits, but it looks like maybe only AEP is significantly overdispersed. ***Answer 9b***: These results differ from those of Blomberg's K because those K values all came out to be zero, whereas these show that there is some significant varying amounts of overdispersion. ***Answer 9c***:

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

```
mammal.Tree<-read.tree("./data/mammal_best_super_tree_fritz2009.tre")
mammal.data<-read.table("./data/mammal_BMR.txt", sep="\t", header=TRUE)
mammal.data<-mammal.data[,c("Species","BMR_.mlO2.hour.","Body_mass_for_BMR_.gr.")]
mammal.species<-array(mammal.data$Species)
pruned.mammal.tree<-drop.tip(mammal.Tree, mammal.Tree$tip.label[-na.omit(match(mammal.species, mammal.T
pruned.mammal.data<-mammal.data[mammal.data$Species %in% pruned.mammal.tree$tip.label,]
rownames(pruned.mammal.data)<-pruned.mammal.data$Species

fit<-lm(log10(BMR_.mlO2.hour.)~log10(Body_mass_for_BMR_.gr.), data=pruned.mammal.data)
plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.),
     log10(pruned.mammal.data$BMR_.mlO2.hour.), las=1,
     xlab="Body mass (kg), log", ylab="Basal Metabolic Rate (BMR), log")
abline(a=fit$coefficientsp[1], b=fit$coefficients[2])
b1<-round(fit$coefficients[2],3)
eqn<-bquote(italic(z)==.(b1))
text(0.5,4.5,eqn,pos=4)
```




```
#fit.phy<-phylolm(log10(BMR_.mlO2.hour.)~log10(Body_mass_for_BMR_.gr.),
#                  data=pruned.mammal.data, pruned.mammal.tree, model = 'lambda', boot=0)
#plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.),
#     log10(pruned.mammal.data$BMR_.mlO2.hour.), las=1, xlab="Body mass (kg), log",
#     ylab="Basal Metabolic Rate (BMR), log")
#abline(a=fit.phy$coefficientsp[1], b=fit.phy$coefficients[2])
#b1.phy<-round(fit.phy$coefficients[2],3)
#eqn<-bquote(italic(z)==.(b1.phy))
#text(0.5,4.5,eqn,pos=4)
```

a. Why do we need to correct for shared evolutionary history?
b. How does a phylogenetic regression differ from a standard linear regression?
c. Interpret the slope and fit of each model. Did accounting for shared evolutionary history improve or worsen the fit?
d. Try to come up with a scenario where the relationship between two variables would completely disappear when the underlying phylogeny is accounted for.

Answer 10a: We need to correct for shared evolutionary hsitory because any traits that occur in multiple species can't be measured as independent occurrences but have a shared ancestry that makes them non-independent. If you compare a species from clade A to clade B and all clade A species all have red flowers and clade B have blue flowers, the each species pair from clade A and B will differ, but they are all measuring the same transition that only happened once and are not independent. ***Answer 10b***: In a linear regression, the residual errors are assumed to be independent and normally distributed with a mean of 0 and a normal variance, whereas in a phylogenetic regression, the variance of the residual errors are described by a covariance matrix

that account for branch lengths in the tree. ***Answer 10c***: Accounting for shared evolutionary history improved the fit of the model. ***Answer 10d***: If two variables are all completely ancestral and there are no derived differences, then all the relationship would be due to shared evolutionary history, so the relationship would disappear after accounting for phylogeny.

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?

```
#if (!requireNamespace("BiocManager", quietly=TRUE))
#install.packages("BiocManager")
#BiocManager::install("msa")
#library(msa)
#a list of accession numbers to download from NCBI
plant.acc.num<-c("JX305536.1","MF596572.1","MF596590.1","JF943527.1","MG227250.1","FJ548263.1",
                 "MK925150.1","MG226591.1","KX821229.1","KT280100.1","MK526140.1","KC483135.1",
                 "MK925521.1","JF942756.1","MF596911.1","MK526467.1","KC483885.1","KF293395.1",
                 "KC484096.1","EF646928.1","MW308596.1","MK925263.1", "KF163411.1","KF163413.1",
                 "MK925326.1","MG249728.1")

#download rbcL regions from plants species of interest from NCBI
plants.DNAbin<-read.GenBank(access.nb = plant.acc.num, species.names = TRUE, as.character = TRUE)

#write those data into a .fasta file
write.dna(plants.DNAbin, file ="plants_fasta.fasta", format = "fasta", append = FALSE, colw = 70, nbcol=
plants.seqs <- read.fasta(file = "plants_fasta.fasta", seqtype = "DNA", as.string = TRUE, forceDNAtolow
```

```
#align sequences
#plant.aln<-msa(plants.seqs)
#
#mySequenceFile <- system.file("seqs","plants_fasta.fasta", package="msa")
#mySequences <- readAAStringSet(mySequenceFile)
#
#plants.dist.raw<-dist.dna(plants.DNAbin, model="raw", pairwise.deletion = FALSE)
#nj.tree<-bionj(plants.dist.raw)
```