11. Worksheet: Phylogenetic Diversity - Traits

Ford Fishman; 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.pdf**).

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] "C:/Users/fordf/OneDrive/Documents/GitHub/QB2021_Fishman/2.Worksheets/11.PhyloTraits"
```

```
setwd("~/GitHub/QB2021_Fishman/2.Worksheets/11.PhyloTraits")
```

2) DESCRIPTION OF DATA

The maintenance of biodiversity is thought to be influenced by **trade-offs** among species in certain functional traits. One such trade-off involves the ability of a highly specialized species to perform exceptionally well on a particular resource compared to the performance of a generalist. In this exercise, we will take a phylogenetic approach to mapping phosphorus resource use onto a phylogenetic tree while testing for specialist-generalist trade-offs.

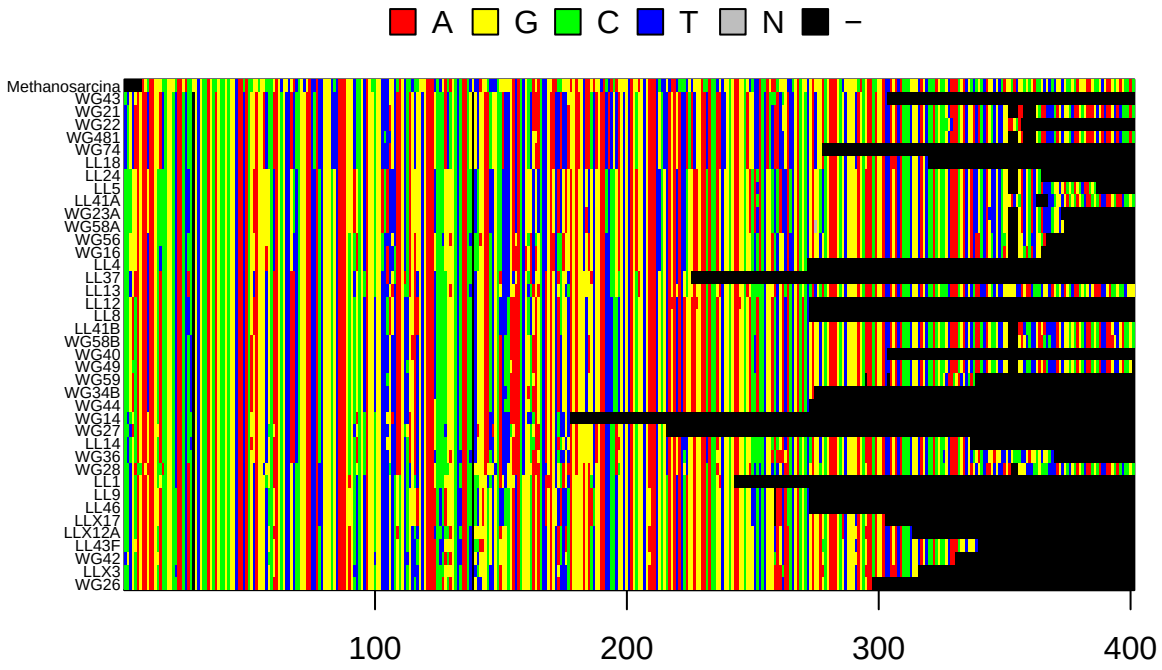
3) SEQUENCE ALIGNMENT

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

> ***Answer 1***: The primary difference between the files is that in the .fasta file, each sequence is not the same length, whereas in the alignment file, they all padded with gaps to be the same length.

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[,500:900]

image.DNAbin(window, cex.lab=0.5)
```




```
ncol(p.DNAbin)
```

```
## [1] 1500
```

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 length of the aligned reads is 1500, though a significant portion of the alignment is only represented by a single sequence.
Answer 2b: The middle regions are best for phylogenetic inferences, as most samples have sequence here.

4) MAKING A PHYLOGENETIC TREE

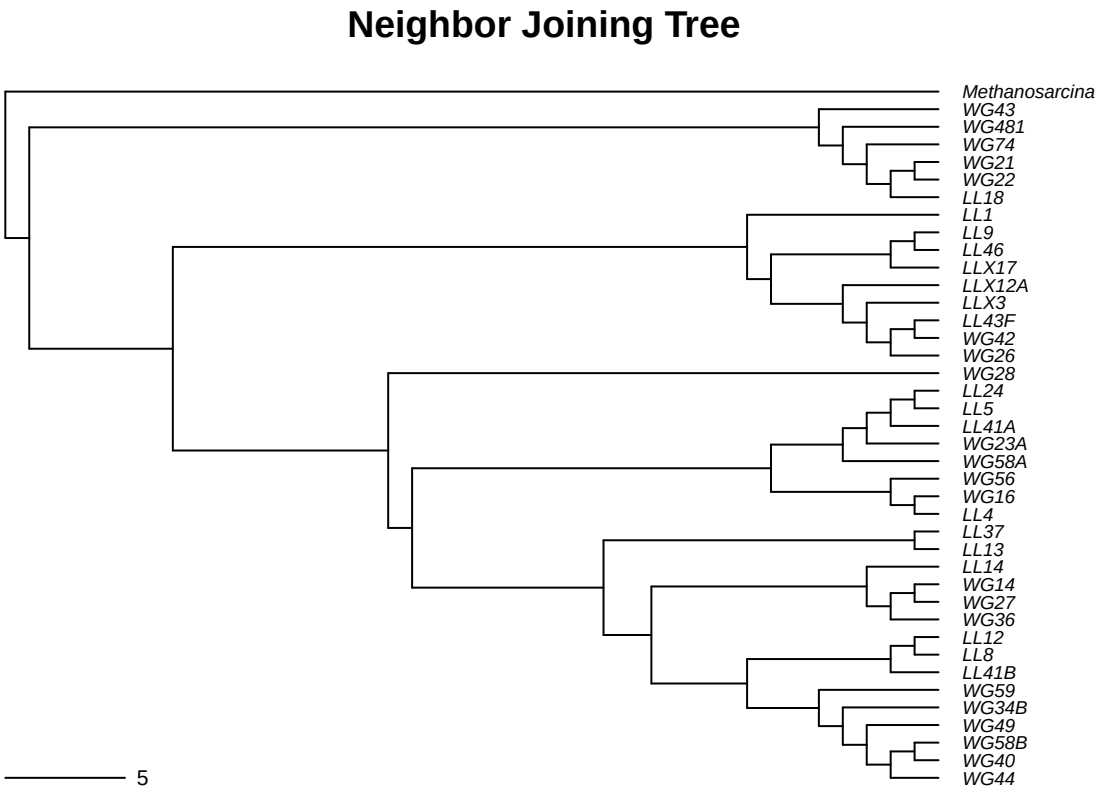
Once you have aligned your sequences, the next step is to construct a phylogenetic tree. Not only is a phylogenetic tree effective for visualizing the evolutionary relationship among taxa, but as you will see later, the information that goes into a phylogenetic tree is needed for downstream analysis.

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 = F)
nj.tree <- bionj(seq.dist.raw)

outgroup <- match( "Methanosarcina", nj.tree$tip.label)
nj.rooted <- root(nj.tree, outgroup, resolve.root = T)

par(mar=c(1,1,2,1)+0.1)
plot.phylo(nj.rooted, main="Neighbor Joining Tree", "phylogram", use.edge.length = F,
           direction = "right", cex=0.6, label.offset = 1)
add.scale.bar(cex=0.7)
```



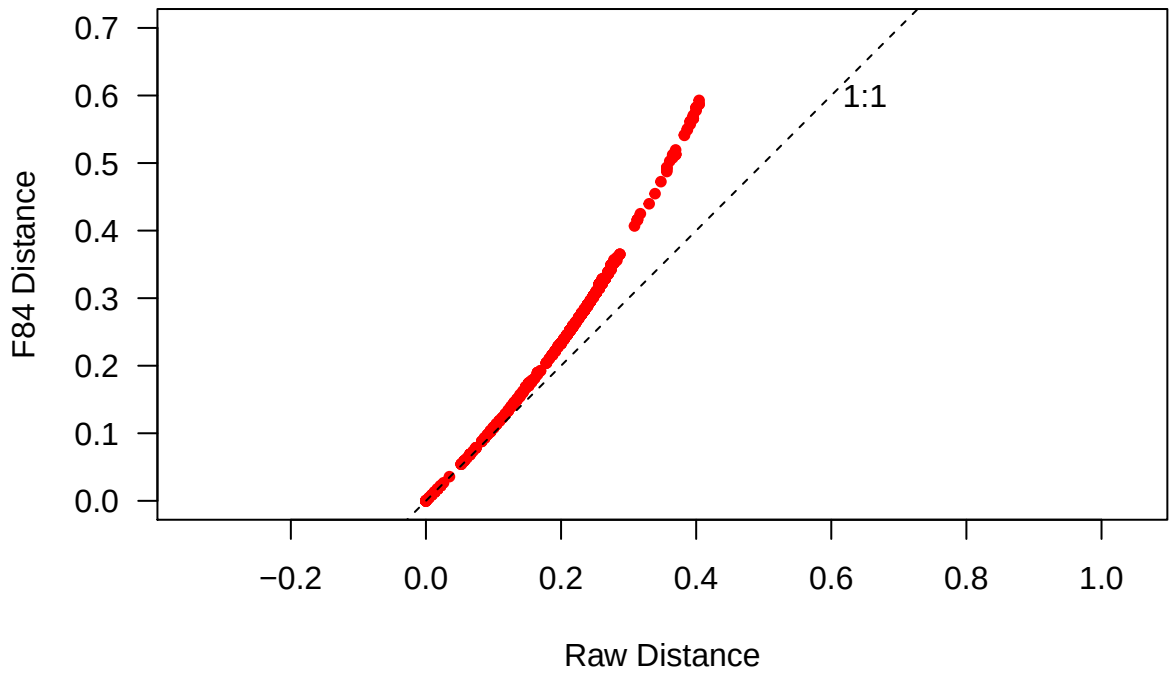
Question 3: What are the advantages and disadvantages of making a neighbor joining tree?

> ***Answer 3***: NJ trees are very fast to make, and will return the same tree every time for the same sequences. It permits lineages with very different branch lengths. However, it relies on distance as a metric, which will not always represent the true phylogeny. It is also dependent on the model evolution used to create the distance matrix.

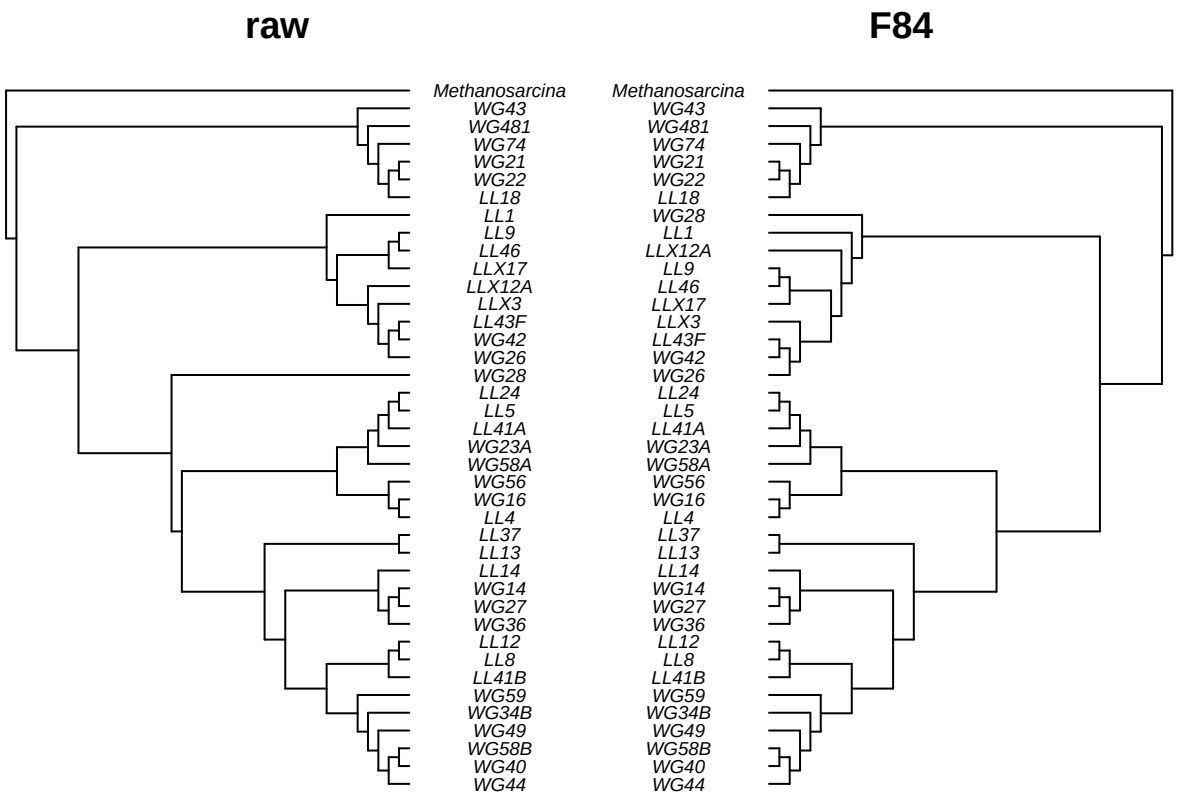
B) SUBSTITUTION MODELS OF DNA EVOLUTION

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 = F)

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")
```

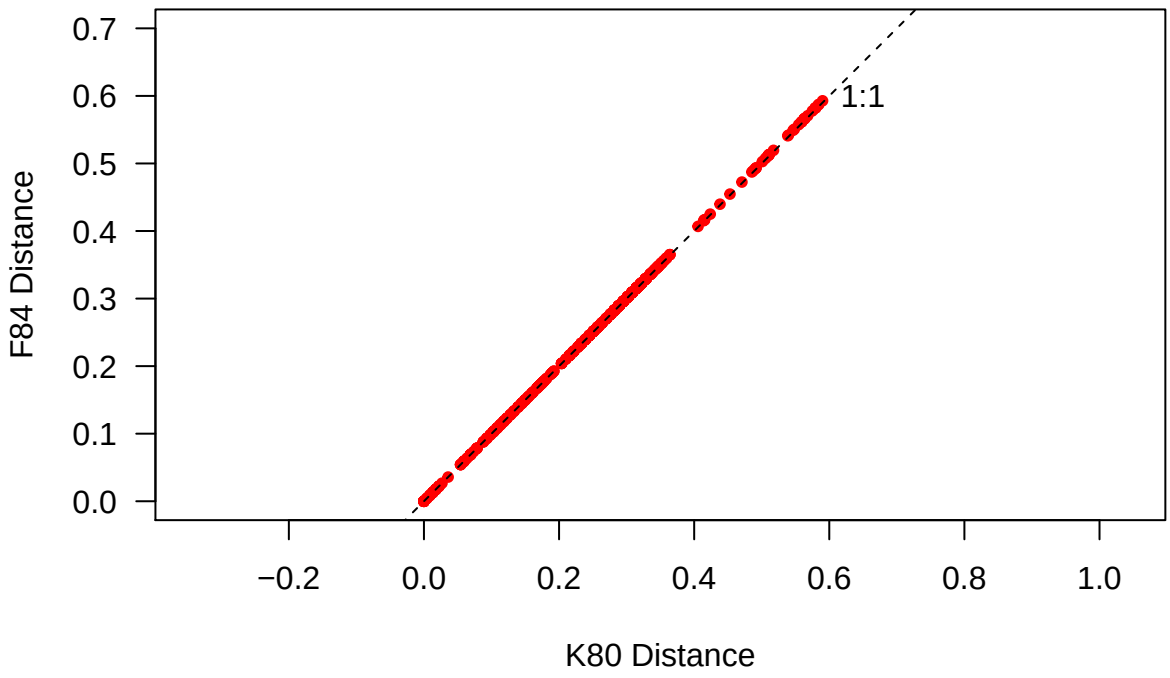



```
F84.tree <- bionj(seq.dist.F84)

F84.outgroup <- match( "Methanosarcina", F84.tree$tip.label)
F84.rooted <- root(F84.tree, outgroup, resolve.root = T)

layout(matrix(c(1,2),1,2), width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(nj.rooted, type="phylogram", show.tip.label = T, use.edge.length = F,
           direction = "right", adj=0.5, cex=0.6, label.offset = 2, main="raw")
plot.phylo(F84.rooted, type="phylogram", show.tip.label = T, use.edge.length = F,
           direction = "left", adj=0.5, cex=0.6, label.offset = 2, main="F84")
```

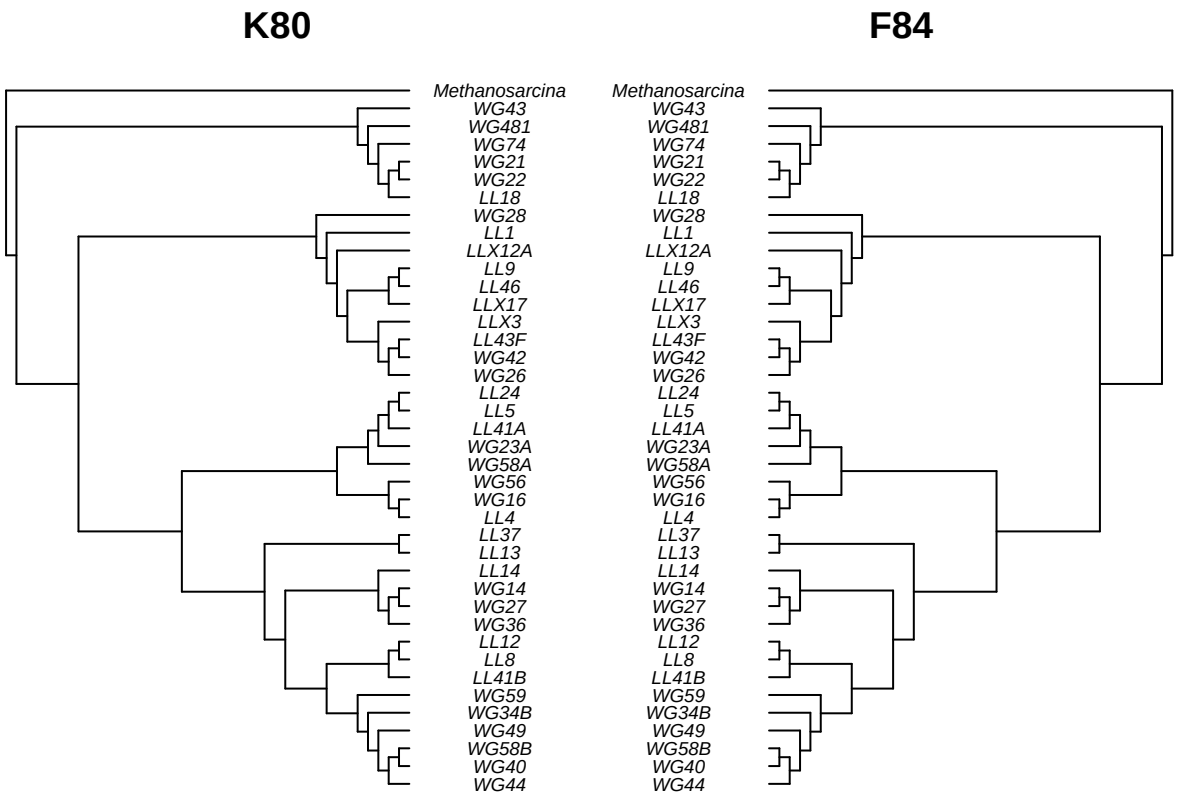


In the R code chunk below, do the following:
1. pick another substitution model,
2. create a distance matrix and tree for this model,
3. make a saturation plot that compares that model to the *Felsenstein (F84)* model,
4. make a cophylogenetic plot that compares the topologies of both models, and
5.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

```
seq.dist.K80 <- dist.dna(p.DNAbin, model="K80", pairwise.deletion = F)

plot(seq.dist.K80, seq.dist.F84,
     pch=20, col="red", las=1, asp=1, xlim=c(0,0.7), ylim=c(0,0.7),
     xlab="K80 Distance", ylab="F84 Distance")
abline(b=1,a=0,lty=2)
text(0.65,0.6, "1:1")
```




```
K80.tree <- bionj(seq.dist.K80)

K80.outgroup <- match( "Methanosarcina", K80.tree$tip.label)
K80.rooted <- root(K80.tree, outgroup, resolve.root = T)

layout(matrix(c(1,2),1,2), width=c(1,1))
par(mar=c(1,1,2,0))
plot.phylo(K80.rooted, type="phylogram", show.tip.label = T, use.edge.length = F,
           direction = "right", adj=0.5, cex=0.6, label.offset = 2, main="K80")
plot.phylo(F84.rooted, type="phylogram", show.tip.label = T, use.edge.length = F,
           direction = "left", adj=0.5, cex=0.6, label.offset = 2, main="F84")
```



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?

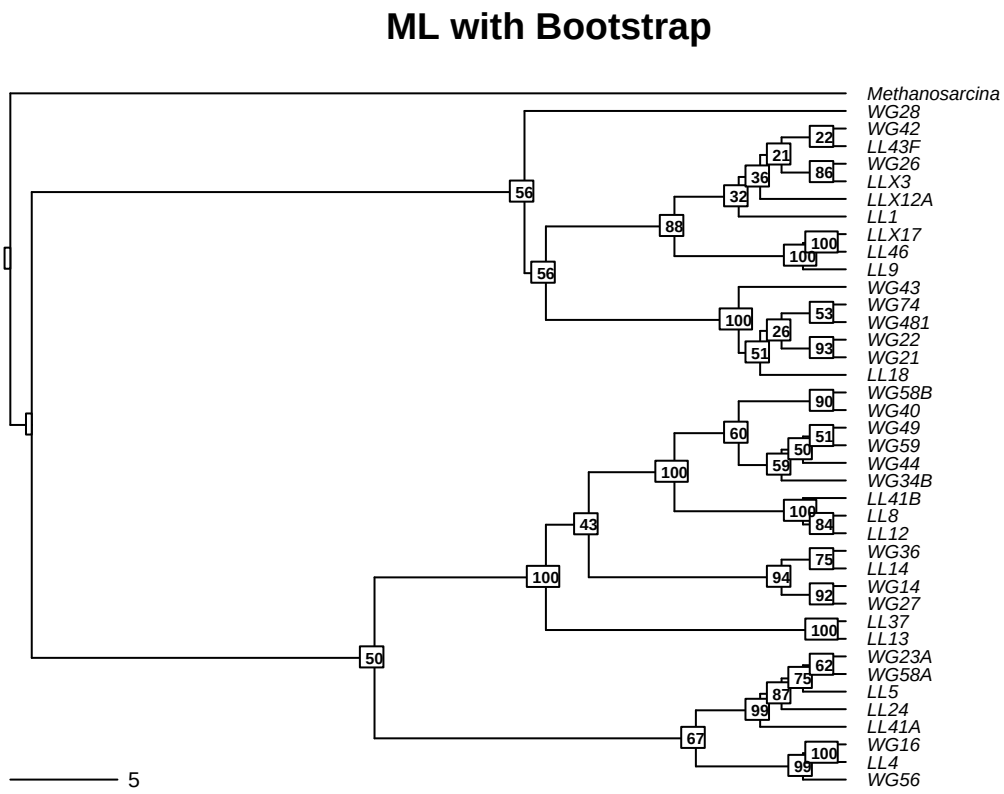
Answer 4a: K80 assumes equal frequencies of nucleotides but assumes that transitions are more common than transversions. F84 also allows transitions and transversions to occur at different rates, but does not assume equal frequencies of nucleotides. ***Answer 4b***: The plots are highly consistent with each other, returning a saturation plot that follows directly on the 1:1 line and also the exact same trees. ***Answer 4c***: Because both models also allows substitution rates to vary and both return the same tree, these differing rates are likely very important for obtaining these trees' topologies.

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

```
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 = T, use.edge.length = F
```

```
          main="ML with Bootstrap")
add.scale.bar(cex=0.7)
nodelabels(ml.bootstrap$node.label, font=2, bg="white", frame="r", cex=0.5)
```



Question 5:

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

b) Why do we bootstrap our tree?

c) What do the bootstrap values tell you?

d) Which branches have very low support?

e) Should we trust these branches?

Answer 5a: The trees are fairly inconsistent with each other. This may be due to the fact the ML is not as sensitive to the model of evolution as NJ.
Answer 5b: We bootstrap in order to obtain an quantative measure of accuracy of our tree.
Answer 5c: Bootstrapping tells you how many trees that were made by sampling from the alignment with replacement were able to obtain the same nodes.
Answer 5d: Many of the nodes have low bootstrap values, but the nodes at the top (besides the outgroup) have the lowest values. ***Answer 5e***: These values are far below 95%, so these branches should not be trusted.

5) INTEGRATING TRAITS AND PHYLOGENY

A. Loading Trait Database

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

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

B. Trait Manipulations

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

```
umax <- apply(p.growth, 1, max)
levins <- function(p_xi){
  p <- 0

  for (i in p_xi){  p <- p + i^2 }

  nb <- 1 / (length(p_xi) *p)
  return(nb)
}
nb <- as.matrix(levins(p.growth.std))

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

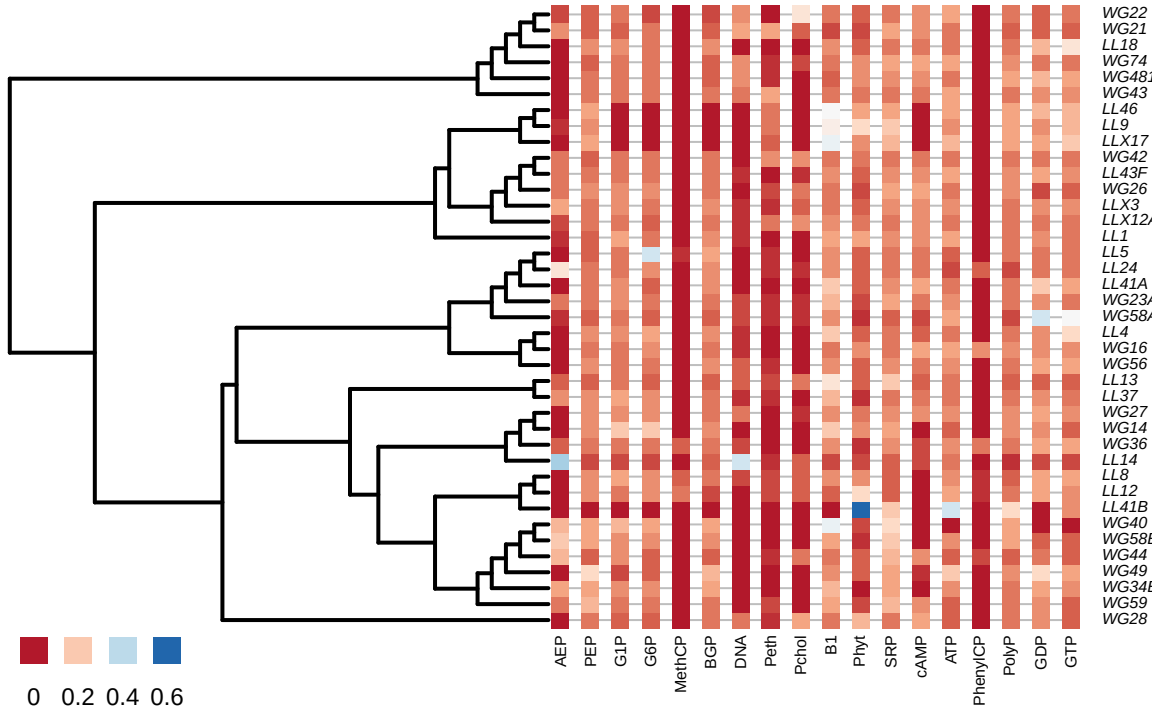
C. Visualizing Traits on Trees

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

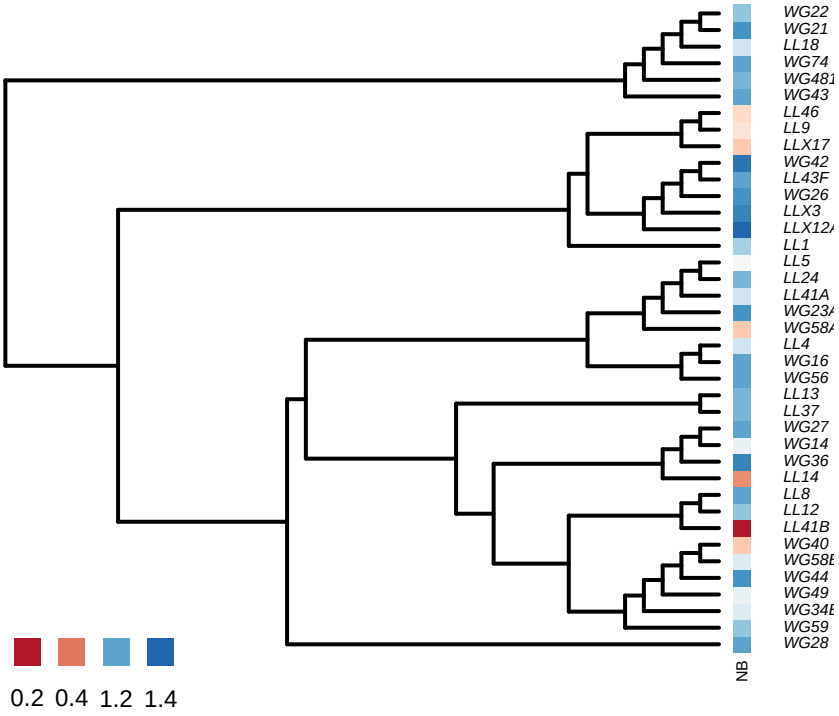
```
# tree previously made above
F84.rooted <- drop.tip(nj.rooted, "Methanosarcina")
```

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, "RdBu"))

par(mar=c(1,1,1,1)+0.1)
x <- phylo4d(F84.rooted, p.growth.std)
table.phylo4d(x, treetype="phylo", symbol = "colors", show.node=T, cex.label = 0.5, scale = F,
              use.edge.length=F, edge.color = "black", edge.width = 2, box = F, col=mypalette(25), pch=
              cex.symbol=1.25, ratio.tree = 0.5, cex.legend = 1.5, center=F)
```




```
par(mar=c(1,5,1,5)+0.1)
x.nb <- phylo4d(F84.rooted, nb)

table.phylo4d(x.nb, treetype="phylo", symbol = "colors", show.node=T, cex.label = 0.5, scale = F,
              use.edge.length=F, edge.color = "black", edge.width = 2, box = F, col=mypalette(25), pch=
              cex.symbol=1.25,var.label = "NB", ratio.tree = 0.9, cex.legend = 1.5, center=F)
```



Question 6:

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

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

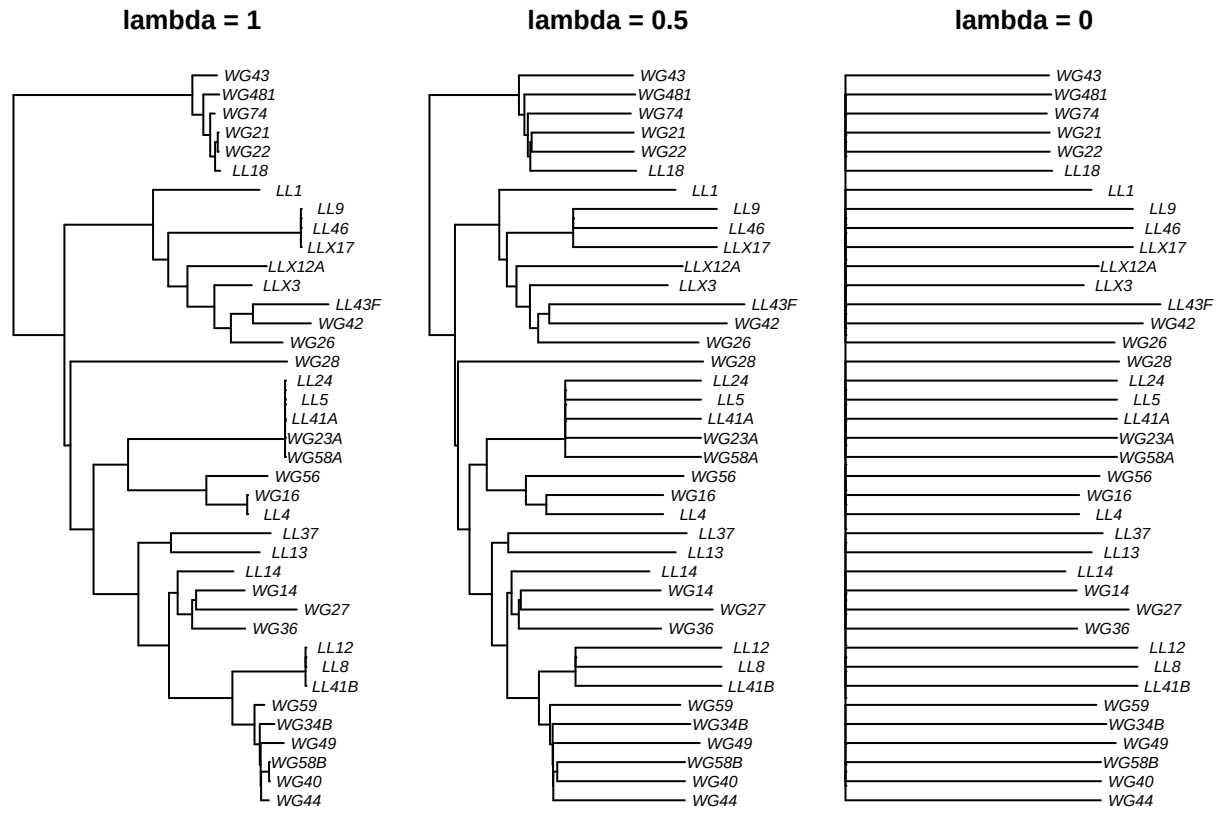
Answer 6a: Lineages that have high growth rates on specific nutrients should have low niche breadth, and vice versa. This tradeoff should have some phylogenetic basis. ***Answer 6b***: Clusters of lineages should cluster phylogenetically on high growth rate on resource usage, as well as niche breadth, and these clusters should not overlap.

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(F84.rooted, "lambda", 0.5)
nj.lambda.0 <- rescale(F84.rooted, "lambda", 0)

layout(matrix(c(1,2,3), 1, 3), widths = c(1,1,1))
par(mar=c(1,0.5,2,0.5)+0.1)
plot(F84.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(F84.rooted, nb, model="lambda")
```

```
## GEIGER-fitted comparative model of continuous data
##  fitted 'lambda' model parameters:
##  lambda = 0.061977
##  sigsq = 0.140021
##  z0 = 0.664039
##
##  model summary:
##  log-likelihood = 21.456738
##  AIC = -36.913476
##  AICc = -36.227761
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 53
##  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 = 0.139249
##  z0 = 0.656203
##
##  model summary:
##  log-likelihood = 21.399126
##  AIC = -36.798252
##  AICc = -36.112537
##  free parameters = 3
##
## Convergence diagnostics:
##  optimization iterations = 100
##  failed iterations = 0
##  number of iterations with same best fit = 86
##  frequency of best fit = 0.86
##
##  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 untransformed tree fits a lambda value of ~0.06, which is not too far from 0. ***Answer 7b***: The AIC values are very similar. I might pick the lambda = 0 model, as the value is a little lower, but the models are essentially equivalent. ***Answer 7c***: This model suggests that there is 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.

```
F84.rooted$edge.length <- F84.rooted$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=F])
  out <- phylosignal(x, F84.rooted)
  p.phylosignal[1:5,i] <- round(t(out), 3)
}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

signal.nb <- phylosignal(nb, F84.rooted)
```

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

```
p.phylosignal
```

```
##                 AEP    PEP    G1P    G6P MethCP    BGP    DNA   Peth  Pchol
## K             0.445  0.372  0.379  0.321  0.431  0.505  0.459  0.488  0.298
## PIC.var.obs   0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## PIC.var.mean  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
## PIC.var.P     0.055  0.049  0.009  0.060  0.022  0.002  0.011  0.286  0.221
## PIC.var.z    -1.690 -1.660 -2.453 -1.388 -2.944 -3.814 -2.446 -0.545 -0.768
## PIC.P.BH      0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
##                  B1   Phyt    SRP   cAMP    ATP PhenylCP  PolyP    GDP    GTP
## K             0.469  0.364  0.449  0.668  0.293    0.268  0.314  0.291  0.318
## PIC.var.obs   0.000  0.000  0.000  0.000  0.000    0.000  0.000  0.000  0.000
## PIC.var.mean  0.000  0.000  0.000  0.000  0.000    0.000  0.000  0.000  0.000
## PIC.var.P     0.007  0.006  0.010  0.002  0.218    0.785  0.094  0.209  0.104
## PIC.var.z    -2.965 -2.387 -2.562 -3.413 -0.745    0.819 -1.328 -0.748 -1.318
## PIC.P.BH      0.000  0.000  0.000  0.000  0.000    1.000  0.000  0.000  0.000
```

```
signal.nb
```

```
##           K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.3682751     1.092539e-09          1.257297e-09          0.147
##   PIC.variance.Z
## 1      -1.045542
```

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: Most of the phosphorus growth rates seem to have some phylogenetic signal. Niche breadth has not phylogenetic signal. ***Answer 8b***: The phosphorus growth rates are consistent with 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, "name")

p.traits <- comparative.data(F84.rooted, p.growth.pa, "name")
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 :  F84.rooted
##   Number of permutations :  1000
##
## Estimated D :  5.908594
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.888
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.034
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  Peth
##   Counts of states:  0 = 18
##                      1 = 21
##   Phylogeny :  F84.rooted
##   Number of permutations :  1000
##
## Estimated D :  0.5059825
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.038
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.114
```

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

```
##
## Calculation of D statistic for the phylogenetic structure of a binary variable
##
##   Data :  p.growth.pa
##   Binary variable :  DNA
##   Counts of states:  0 = 20
##                      1 = 19
##   Phylogeny :  F84.rooted
##   Number of permutations :  1000
##
## Estimated D :  0.4867322
```

```
## Probability of E(D) resulting from no (random) phylogenetic structure :  0.025
## Probability of E(D) resulting from Brownian phylogenetic structure    :  0.118
```

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?

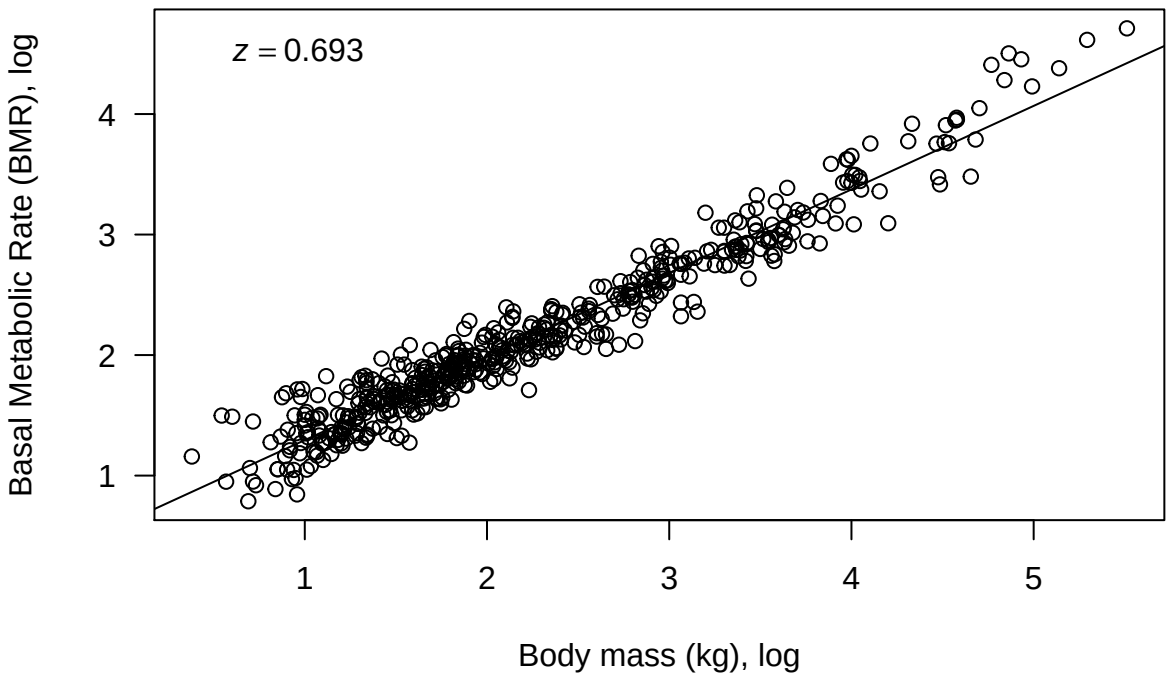
c. Discuss what factors might give rise to differences between the metrics.

Answer 9a: I chose PEP, Peth, and DNA. Peth and DNA are significantly overdispersed. ***Answer 9b***: These results agree with the Blomberg's K analysis, though PEP was found to be overdispersed in that case as well. ***Answer 9c***: Would the fact that you have to change the growth rates into categorical features have an impact?

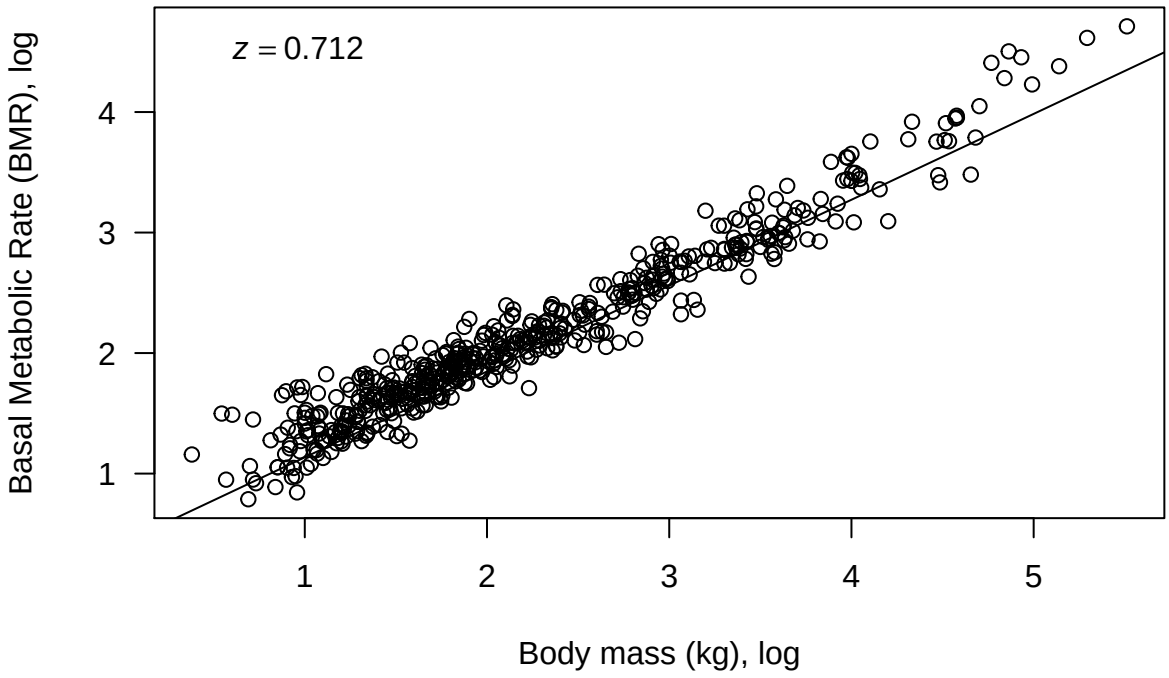
7) PHYLOGENETIC REGRESSION

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=T)

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

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

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

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




```
# phyl fit
fit.phy <- phylolm(log10(BMR_.mlO2.hour.)~log10(Body_mass_for_BMR_.gr.), data=pruned.mammal.data, prune
plot(log10(pruned.mammal.data$Body_mass_for_BMR_.gr.),
     log10(pruned.mammal.data$BMR_.mlO2.hour.), las=1,
     xlab="Body mass (kg), log", ylab="Basal Metabolic Rate (BMR), log")
abline(a = fit.phy$coefficients[1], b=fit.phy$coefficients[2])
b1 <- round(fit.phy$coefficients[2],3)
eqn <- bquote(italic(z)==.(b1))
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: Shared evolutionary history violatesthe assumption of indepedence in standard linear regression. ***Answer 10b***: Instead of assuming that the covariance matrix for the residuals is simply variance times the identity matrix (which would imply no covariance between samples), the covariance matrix is allowed to have entries off the diagonal (which allows for sample covariance). ***Answer 10c***: The phylogenetic model increased the value of z, implying a stronger relationship. ***Answer 10d***: I'm not sure if this correct, but would the relationship dissappear if all of the variance found is due to phylogenetic relationships?

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.

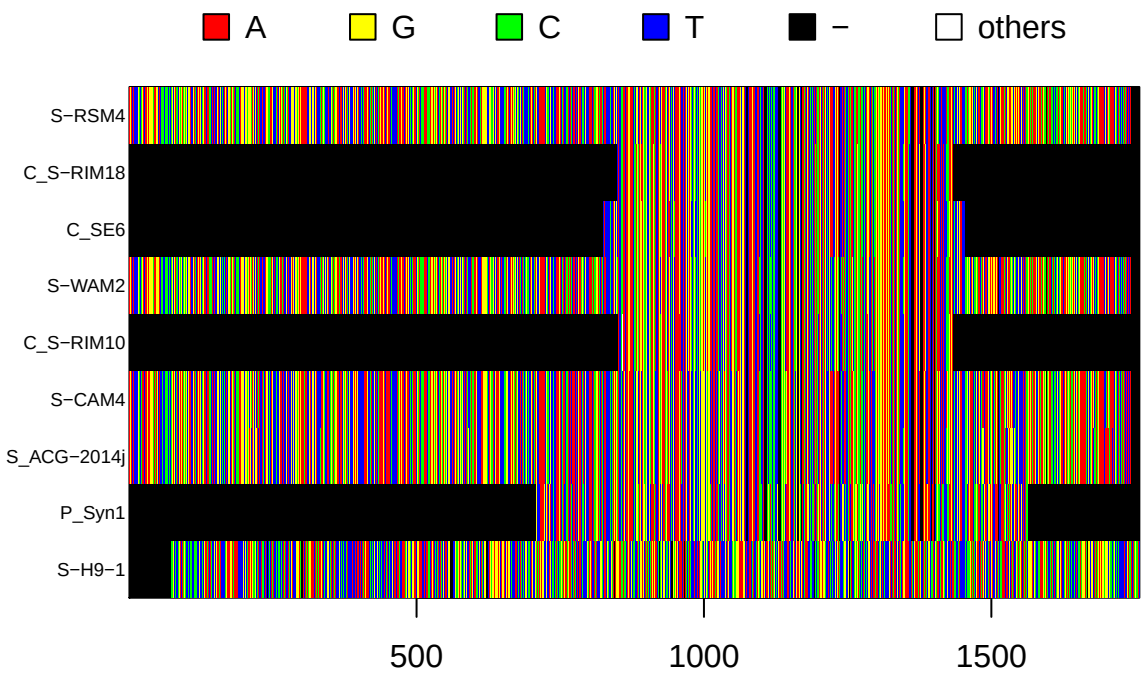
After you identify an appropriate gene, download sequences and create a properly formatted fasta file. Next, align the sequences and confirm that you have a good alignment. Choose a substitution model and make a tree of your choice. Based on the decisions above and the output, does your tree jibe with 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?

```
library("msa")
```

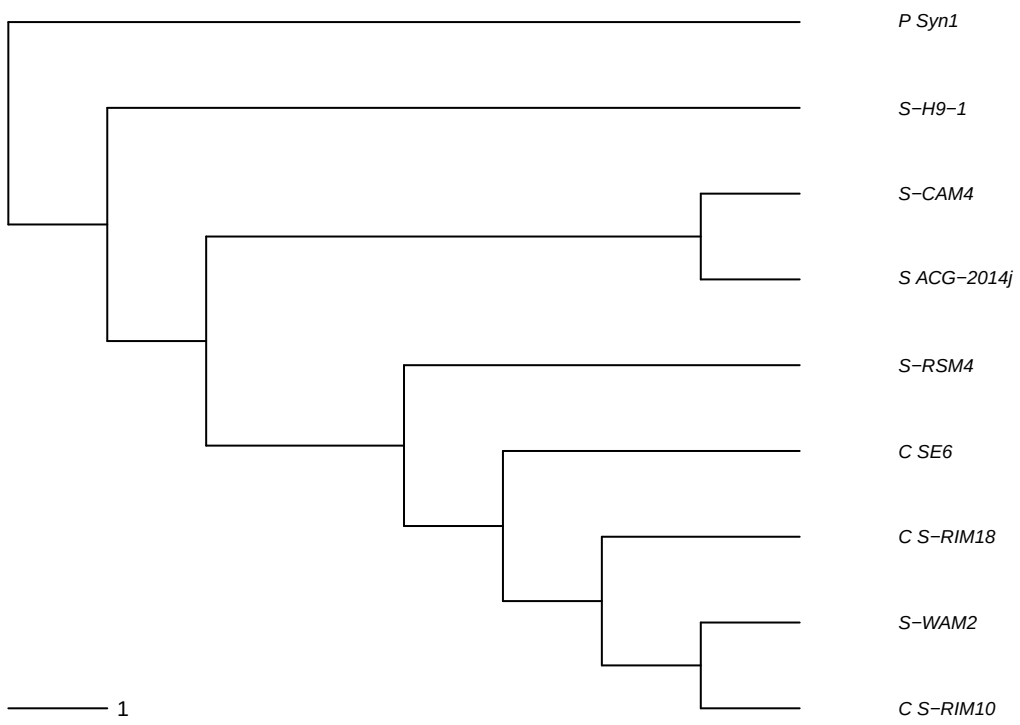
```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

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

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

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

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

## The following object is masked from 'package:ade4':
##
##     score

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

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

## Loading required package: S4Vectors

## Loading required package: stats4
```

```
##
## Attaching package: 'S4Vectors'
## The following objects are masked from 'package:dplyr':
##
##     first, rename
## The following object is masked from 'package:tidyr':
##
##     expand
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
## The following object is masked from 'package:nlme':
##
##     collapse
## The following object is masked from 'package:grDevices':
##
##     windows
## Loading required package: XVector
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
## Found more than one class "Annotated" in cache; using the first, from namespace 'RNeXML'
## Also defined by 'S4Vectors'
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:seqinr':
##
##     translate
```

```
## The following object is masked from 'package:ape':
##
##     complement
## The following object is masked from 'package:base':
##
##     strsplit
fasta <- readDNAStringSet("data/g20.fasta")
align <- msa(fasta)
## use default substitution matrix
ape_dna <- as.DNAbin(align)
image.DNAbin(ape_dna, cex.lab = 0.6)
```




```
g20.dist.F84 <- dist.dna(ape_dna, model="F84", pairwise.deletion = F)
g20.tree <- bionj(g20.dist.F84)

outgroup <- match( "P_Syn1", g20.tree$tip.label)
g20.rooted <- root(g20.tree, outgroup, resolve.root = T)

par(mar=c(1,1,2,1)+0.1)
plot.phylo(g20.rooted, main="Neighbor Joining Tree", "phylogram", use.edge.length = F,
           direction = "right", cex=0.6, label.offset = 1)
add.scale.bar(cex=0.7)
```



: This tree is of the g20 gene in marine bacteriophages. The tree somewhat makes sense. A Prochlorococcus gene is the outgroup. The S (Synechococcus) genes somewhat cluster together, though they are not monophyletic. They are mixed in the more broad C (cyanophage) grouping. Obtaining a larger sample might be useful, as well as using ML methods to produce the tree.