9.Phylogenetic Diversity - Communities

Lauren Albert; Z620: Quantitative Biodiversity, Indiana University

28 February, 2023

OVERVIEW

Complementing taxonomic measures of α- and β-diversity with evolutionary information yields insight into a broad range of biodiversity issues including conservation, biogeography, and community assembly. In this worksheet, you will be introduced to some commonly used methods in phylogenetic community ecology.

After completing this assignment you will know how to:

1. incorporate an evolutionary perspective into your understanding of community ecology
2. quantify and interpret phylogenetic α- and β-diversity
3. evaluate the contribution of phylogeny to spatial 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 '9.PhyloCom' 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 *9.PhyloCom_Worksheet.Rmd* and the PDF output of `Knitr` (*9.PhyloCom_Worksheet.pdf*).

The completed exercise is due on **Wednesday, March 1st, 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 `/9.PhyloCom` folder,
4. load all of the required R packages (be sure to install if needed), and
5. load the required R source file.

```
#set up environment and directory
rm(list = ls())
getwd()
```

```
## [1] "/Users/laurenalbert/GitHub/QB2023_Albert/2.Worksheets/9.PhyloCom"
```

```
#install and load packages
package.list <- c('picante', 'ape', 'seqinr', 'vegan','fossil','reshape','devtools','BiocManager','ineq
for(package in package.list){
  if(!require(package, character.only = TRUE, quietly = TRUE)){
    install.packages(package, repos = 'http://cran.us.r-project.org')
    library(package, character.only = TRUE)
  }
}
```

```
## This is vegan 2.6-4
```

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

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

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

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

```
##
## Attaching package: 'shapefiles'
```

```
## The following objects are masked from 'package:foreign':
##
##     read.dbf, write.dbf
```

```
##
## Attaching package: 'devtools'
```

```
## The following object is masked from 'package:permute':
##
##     check
```

```
##
## Attaching package: 'BiocManager'

## The following object is masked from 'package:devtools':
##
##     install

## This is mgcv 1.8-41. For overview type 'help("mgcv-package")'.

## Registered S3 method overwritten by 'labdsv':
##   method       from
##   summary.dist ade4

## This is labdsv 2.0-1
## convert existing ordinations with as.dsvord()

##
## Attaching package: 'labdsv'

## The following object is masked from 'package:stats':
##
##     density

##
## Attaching package: 'matrixStats'

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

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#load source code
source("./bin/MothurTools.R")
```

2) DESCRIPTION OF DATA

need to discuss data set from spatial ecology!

We sampled >50 forested ponds in Brown County State Park, Yellowood State Park, and Hoosier National Forest in southern Indiana. In addition to measuring a suite of geographic and environmental variables, we characterized the diversity of bacteria in the ponds using molecular-based approaches. Specifically, we amplified the 16S rRNA gene (i.e., the DNA sequence) and 16S rRNA transcripts (i.e., the RNA transcript of the gene) of bacteria. We used a program called `mothur` to quality-trim our data set and assign sequences to operational taxonomic units (OTUs), which resulted in a site-by-OTU matrix.
In this module we will focus on taxa that were present (i.e., DNA), but there will be a few steps where we need to parse out the transcript (i.e., RNA) samples. See the handout for a further description of this week's dataset.

3) LOAD THE DATA

In the R code chunk below, do the following:
1. load the environmental data for the Brown County ponds (*20130801_PondDataMod.csv*),
2. load the site-by-species matrix using the `read.otu()` function,
3. subset the data to include only DNA-based identifications of bacteria,
4. rename the sites by removing extra characters,
5. remove unnecessary OTUs in the site-by-species, and
6. load the taxonomic data using the `read.tax()` function from the source-code file.

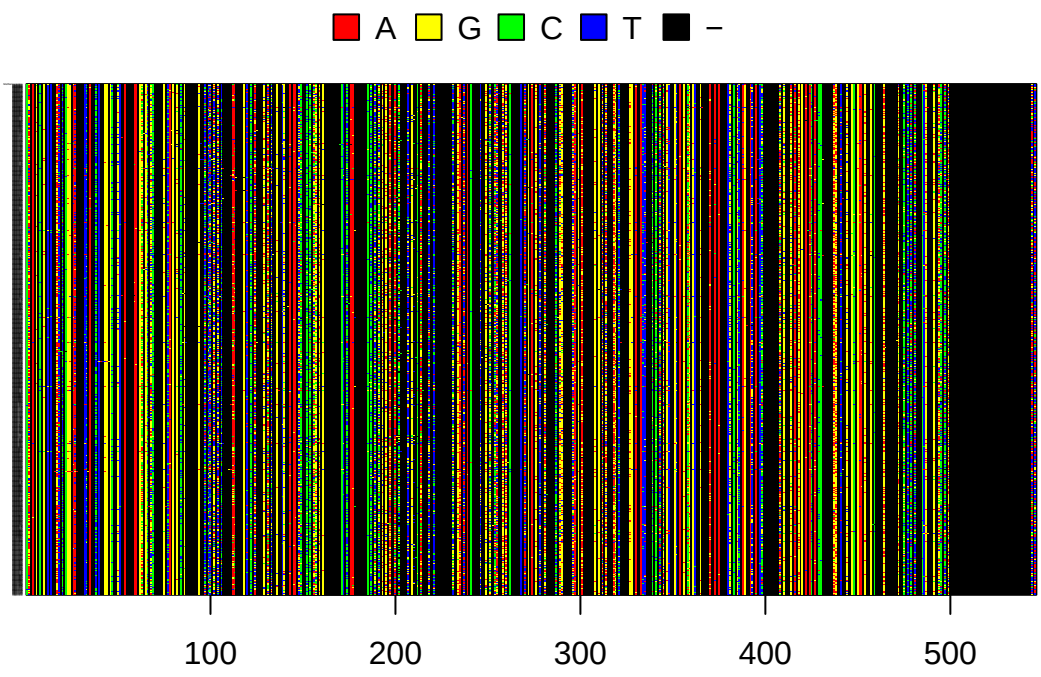
```
env <- read.table("data/20130801_PondDataMod.csv", sep = ",", header = TRUE)
env <- na.omit(env)

#load site-by-species matrix
comm <- read.otu(shared = "./data/INPonds.final.rdp.shared", cutoff = "1")

#select DNA data using grep()
comm <- comm[grep("*DNA", rownames(comm)), ]

#perform replacement of all matches with gsub()
rownames(comm) <- gsub("\\-DNA", "", rownames(comm))
rownames(comm) <- gsub("\\_", "", rownames(comm))

#remove sites not in the environmental data set
comm <- comm[rownames(comm) %in% env$Sample_ID, ]
#remove zero-abundance OTUs from data set
comm <- comm[ , colSums(comm) > 0]

#import taxonomic data
tax <- read.tax(taxonomy = "./data/INPonds.final.rdp.1.cons.taxonomy")
```

```
## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE

## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE

## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE

## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE

## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE

## Warning in type.convert.default(as.character(x)): 'as.is' should be specified by
## the caller; using TRUE
```

Next, in the R code chunk below, do the following:
1. load the FASTA alignment for the bacterial operational taxonomic units (OTUs),
2. rename the OTUs by removing everything before the tab (\t) and after the bar (|),
3. import the *Methanosarcina* outgroup FASTA file,

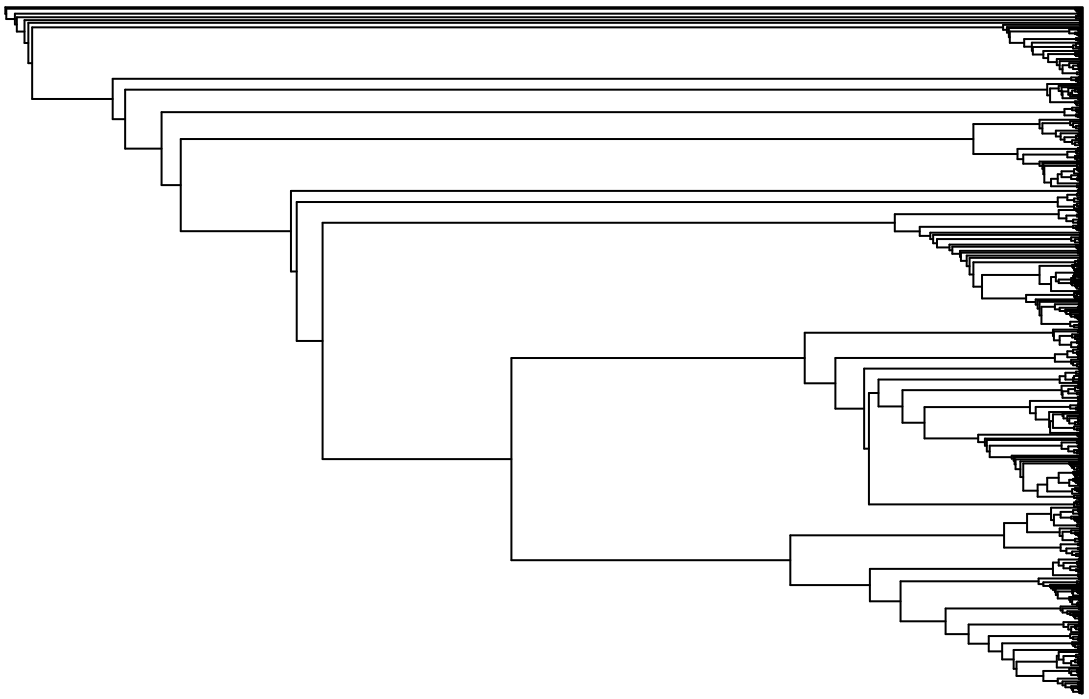
4. convert both FASTA files into the DNAbin format and combine using `rbind()`,
5. visualize the sequence alignment,
6. using the alignment (with outgroup), pick a DNA substitution model, and create a phylogenetic distance matrix,
7. using the distance matrix above, make a neighbor joining tree,
8. remove any tips (OTUs) that are not in the community data set,
9. plot the rooted tree.

```
#read the alignment file (seqinr)
ponds.cons <- read.alignment(file = "./data/INPonds.final.rdp.1.rep.fasta", format = "fasta")

#rename OTUs in FASTA file
#ponds.cons$nam <- gsub("\\|.*$", "", gsub("^.*", "", ponds.cons$nam))
ponds.cons$nam <- gsub(".*\t", "", ponds.cons$nam)
ponds.cons$nam <- gsub("\\|.*", "", ponds.cons$nam)
#import outgroup sequence
outgroup <- read.alignment(file = "./data/methanosarcina.fasta", format = "fasta")
#convert alignment file to DNAbin
DNAbin <- rbind(as.DNAbin(outgroup), as.DNAbin(ponds.cons))

#visualize alignment
image.DNAbin(DNAbin, show.labels = T, cex.lab = 0.05, las = 1)
```




```
#make distance matrix (ape)
seq.dist.jc <- dist.dna(DNAbin, model = "JC", pairwise.deletion = FALSE)
```

```r
#make a neighbor-joining tree file (ape)
phy.all <- bionj(seq.dist.jc)

#drop tips of zero-occurrence OTUs
phy <- drop.tip(phy.all, phy.all$tip.label[!phy.all$tiplabel %in%
                                             c(colnames(comm), "Methanosarcina")])

#identify outgroup sequence
outgroup <- match("Methanosarcina", phy$tip.label)

#root the tree
phy <- root(phy, outgroup, resolve.root = TRUE)

#plot the rooted tree
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(phy, main = "Neighbor Joining Tree", "phylogram",
           show.tip.label = FALSE, use.edge.length = FALSE,
           direction = "right", cex = 0.6, label.offset = 1)
```



4) PHYLOGENETIC ALPHA DIVERSITY

A. Faith's Phylogenetic Diversity (PD)

In the R code chunk below, do the following:
1. calculate Faith's D using the `pd()` function.

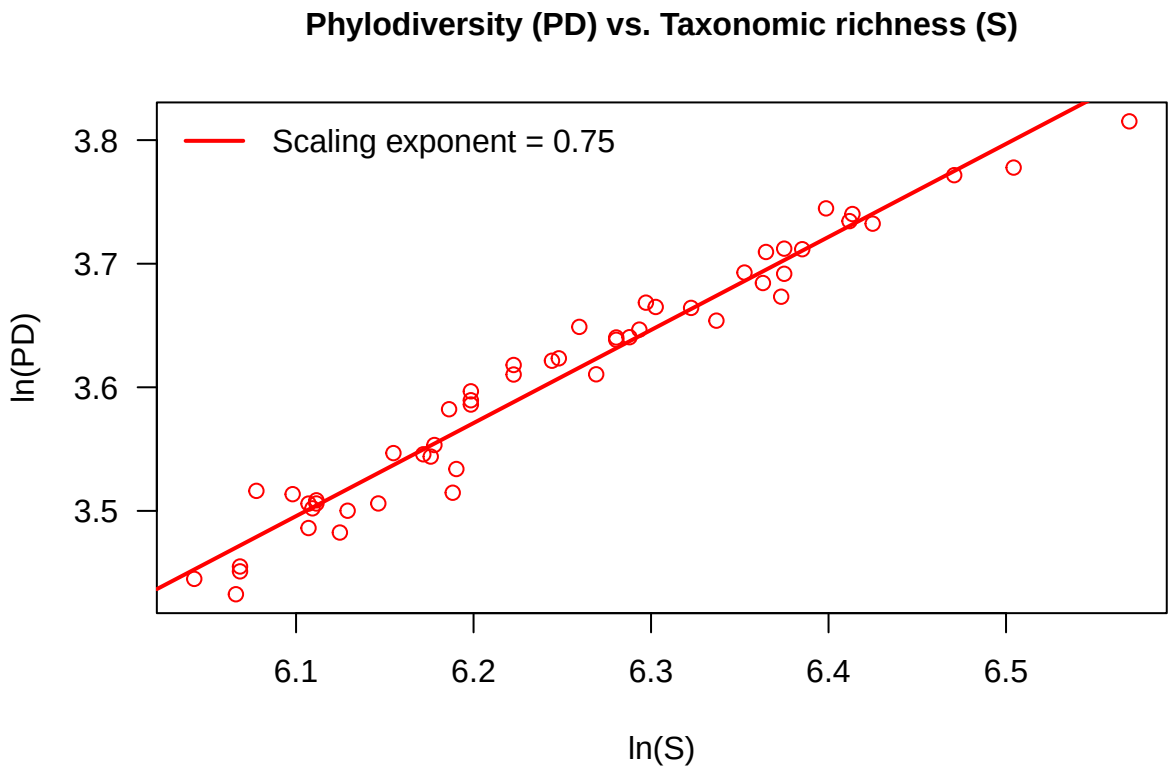
```
#calculate PD and S
pd <- pd(comm, phy, include.root = FALSE)
```

In the R code chunk below, do the following:
1. plot species richness (S) versus phylogenetic diversity (PD),
2. add the trend line, and
3. calculate the scaling exponent.

```
#Biplot of S and PD
par(mar = c(5, 5, 4, 1) + 0.1)

plot(log(pd$S), log(pd$PD),
     rch = 20, col = "red", las = 1,
     xlab = "ln(S)", ylab = "ln(PD)", cex.main = 1,
     main = "Phylodiversity (PD) vs. Taxonomic richness (S)")
```

```
## Warning in plot.window(...): "rch" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "rch" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "rch" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "rch" is not a
## graphical parameter

## Warning in box(...): "rch" is not a graphical parameter

## Warning in title(...): "rch" is not a graphical parameter
```

```
#test of power-law relationship
fit <- lm('log(pd$PD) ~ log(pd$S)')
abline(fit, col = "red", lw = 2)
exponent <- round(coefficients(fit)[2], 2)
legend("topleft", legend = paste("Scaling exponent = ", exponent, sep =""),
       bty = "n", lw = 2, col = "red")
```



Question 1: Answer the following questions about the PD-S pattern.
a. Based on how PD is calculated, why should this metric be related to taxonmic richness? b. Describe the relationship between taxonomic richness and phylodiversity. c. When would you expect these two estimates of diversity to deviate from one another? d. Interpret the significance of the scaling PD-S scaling exponent.

> ***Answer 1a***: The higher the PD value the more evolutionary taxa an assemblage contains, which suggest more richness ***Answer 1b***: Taxonomic richness and phylodiversity have a positive correlation – greater species richness equals more phylodiversity ***Answer 1c***: These two estimates might diverge from one another if there are many species that have a more constricted evolutionary history (low PD, high S) ***Answer 1d***: I interpret the scaling exponent as scaling the level of phyogenetic diversity data (gene) to number of species (individuals)

i. Randomizations and Null Models

In the R code chunk below, do the following:
1. estimate the standardized effect size of PD using the `richness` randomization method.

```
ses.pd <- ses.pd(comm[1:2,], phy, null.model = "richness", runs = 25, include.root = FALSE)

ses.pd2 <- ses.pd(comm[1:2,], phy, null.model = "frequency", runs = 25, include.root = FALSE)

ses.pd3 <- ses.pd(comm[1:2,], phy, null.model = "independentswap", runs = 25, include.root = FALSE)
```

Question 2: Using `help()` and the table above, run the `ses.pd()` function using two other null models and answer the following questions:

a. What are the null and alternative hypotheses you are testing via randomization when calculating `ses.pd`?
b. How did your choice of null model influence your observed ses.pd values? Explain why this choice affected or did not affect the output.

Answer 2a: The null hypotheses are testing the that the PD values are greater than 0 by randomizing the method of "pulling" from the community matrix. ***Answer 2b***: Each null model provided slighlty different means, ranks, and standard deviations of PD

B. Phylogenetic Dispersion Within a Sample

Another way to assess phylogenetic α-diversity is to look at dispersion within a sample.

i. Phylogenetic Resemblance Matrix

In the R code chunk below, do the following:
1. calculate the phylogenetic resemblance matrix for taxa in the Indiana ponds data set.

```
phydist <- cophenetic.phylo(phy)
```

ii. Net Relatedness Index (NRI)

In the R code chunk below, do the following:
1. Calculate the NRI for each site in the Indiana ponds data set.

```
ses.mpd <- ses.mpd(comm, phydist, null.model = "taxa.labels",
                   abundance.weighted = FALSE, runs = 25)

NRI <- as.matrix(-1 * ((ses.mpd[,2] - ses.mpd[,3]) / ses.mpd[,4]))
rownames(NRI) <- row.names(ses.mpd)
colnames(NRI) <- "NRI"


ses.mpd2 <- ses.mpd(comm, phydist, null.model = "taxa.labels",
                   abundance.weighted = TRUE, runs = 25)

NRI2 <- as.matrix(-1 * ((ses.mpd2[,2] - ses.mpd2[,3]) / ses.mpd2[,4]))
rownames(NRI2) <- row.names(ses.mpd2)
colnames(NRI2) <- "NRI"
```

iii. Nearest Taxon Index (NTI)

In the R code chunk below, do the following: 1. Calculate the NTI for each site in the Indiana ponds data set.

```
ses.mntd <- ses.mntd(comm, phydist, null.model = "taxa.labels",
                     abundance.weighted = FALSE, runs = 25)

NTI <- as.matrix(-1 * ((ses.mntd[,2] - ses.mntd[,3]) / ses.mntd[,4]))
rownames(NTI) <- row.names(ses.mntd)
colnames(NTI) <- "NTI"


ses.mntd2 <- ses.mntd(comm, phydist, null.model = "taxa.labels",
```

```
                    abundance.weighted = TRUE, runs = 25)

NTI2 <- as.matrix(-1 * ((ses.mntd2[,2] - ses.mntd2[,3]) / ses.mntd2[,4]))
rownames(NTI2) <- row.names(ses.mntd2)
colnames(NTI2) <- "NTI"
```

Question 3:

a. In your own words describe what you are doing when you calculate the NRI.
b. In your own words describe what you are doing when you calculate the NTI.
c. Interpret the NRI and NTI values you observed for this dataset.
d. In the NRI and NTI examples above, the arguments "abundance.weighted = FALSE" means that the indices were calculated using presence-absence data. Modify and rerun the code so that NRI and NTI are calculated using abundance data. How does this affect the interpretation of NRI and NTI?

Answer 3a: NRI is the inverse of the difference between the observed mean phylogenetic difference and the mean values of the phylogenetic difference generated via randomization, divided by the standard deviation of the mean phylogenetic differences from randomization null models. This calculates the relatedness of taxa that fall closely to one another on a phylogeny, comparing to the null to determine dispersion. ***Answer 3b***: NTI is the inverse of the difference between the observed mean nearest phylogenetic neighbor distance and the mean, divided by the standard deciation. Similarly, this measures the level of dispersion in the phylogeny and whether it is more or less than expected, however this uses all taxa ***Answer 3c***: Most NTI and NRI values are negative, suggesting overdispersion where nearer taxa are more distantly related than expected. ***Answer 3d***: When calculated using abundance order, there are significantly more positive NTI values – indicating phylogenetic clustering. The NRI values are still mostly negative, but are closer to zero. This suggests that including abundance data in the calculation can drastically change how the clustering or dispersion of a sample is considered.

5) PHYLOGENETIC BETA DIVERSITY

A. Phylogenetically Based Community Resemblance Matrix

In the R code chunk below, do the following:
1. calculate the phylogenetically based community resemblance matrix using Mean Pair Distance, and
2. calculate the phylogenetically based community resemblance matrix using UniFrac distance.

```
dist.mp <- comdist(comm, phydist)
```

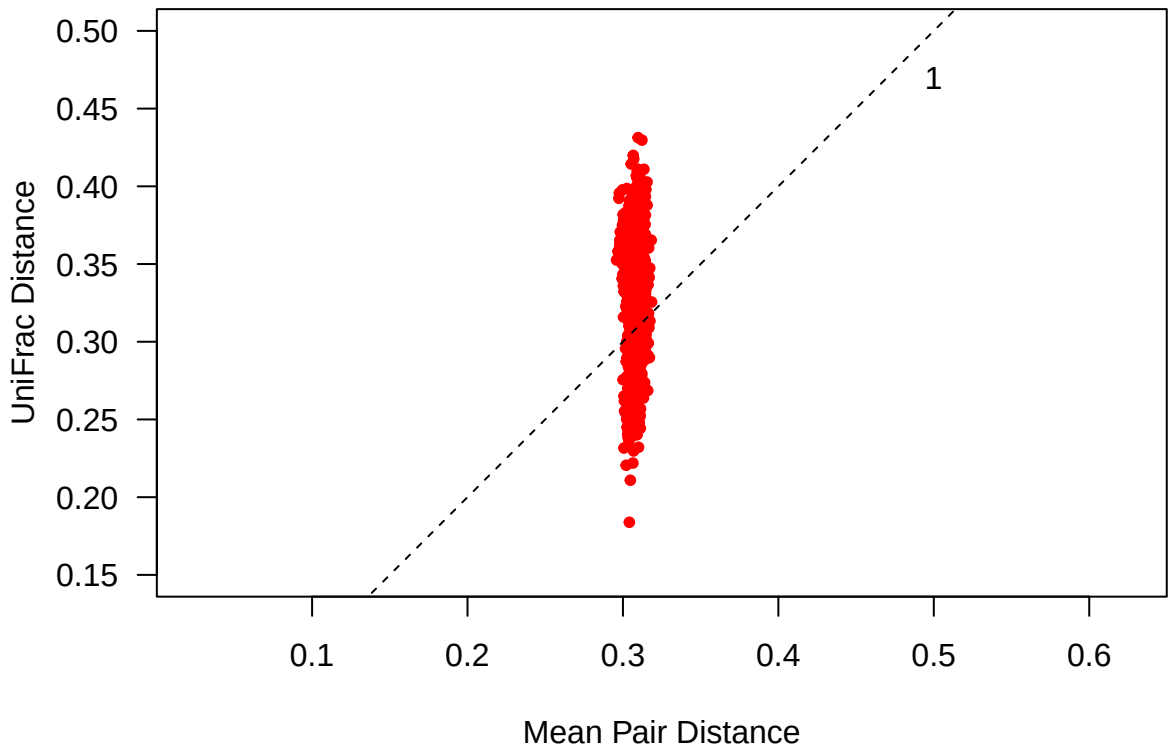
```
## [1] "Dropping taxa from the distance matrix because they are not present in the community data:"
##  [1] "Methanosarcina" "Otu0881"        "Otu0963"        "Otu0969"
##  [5] "Otu0984"        "Otu0990"        "Otu0991"        "Otu0997"
##  [9] "Otu0998"        "Otu1002"        "Otu1004"        "Otu1007"
## [13] "Otu1011"        "Otu1013"        "Otu1019"        "Otu1022"
## [17] "Otu1023"        "Otu1025"        "Otu1029"        "Otu1030"
## [21] "Otu1034"        "Otu1039"        "Otu1049"        "Otu1050"
## [25] "Otu1052"        "Otu1057"        "Otu1058"        "Otu1059"
## [29] "Otu1060"        "Otu1061"        "Otu1062"        "Otu1069"
## [33] "Otu1072"        "Otu1073"        "Otu1074"        "Otu1079"
## [37] "Otu1083"        "Otu1084"        "Otu1085"        "Otu1089"
## [41] "Otu1090"        "Otu1091"        "Otu1093"        "Otu1094"
```

```
## [45] "Otu1096"        "Otu1097"        "Otu1098"        "Otu1112"
## [49] "Otu1113"        "Otu1115"        "Otu1116"        "Otu1119"
## [53] "Otu1120"        "Otu1123"
```

```
dist.uf <- unifrac(comm,phy)
```

In the R code chunk below, do the following:
1. plot Mean Pair Distance versus UniFrac distance and compare.

```
par(mar = c(5, 5, 2 ,1) + 0.1)

plot(dist.mp, dist.uf,
     pch = 20, col = "red", las = 1, asp = 1, xlim = c(0.15, 0.5), ylim = c(0.15, 0.5),
     xlab = "Mean Pair Distance", ylab = "UniFrac Distance")
abline(b = 1, a = 0, lty = 2)
text(0.5, 0.47, 1:1)
```



Question 4:

a. In your own words describe Mean Pair Distance, UniFrac distance, and the difference between them.
b. Using the plot above, describe the relationship between Mean Pair Distance and UniFrac distance. Note: we are calculating unweighted phylogenetic distances (similar to incidence based measures). That means that we are not taking into account the abundance of each taxon in each site.
c. Why might MPD show less variation than UniFrac?

Answer 4a: Mean pairwise distance is the mean distance between the two taxa. The UniFrac is the sum of unshared branches divided by the total branches in the entire tree. Mean pairwise would then be a more "direct" calculation of the distance whereas UniFrac considers all routes in tree to determine proximity. ***Answer 4b***: I think this plot suggests that mean pair distance is more precise, or at least less varied because it only focuses on the distance between the taxa of interest. UniFrac values are more varied because of all of the branches in the tree being considered to locate the taxa of interest. Therefore, the relationship between the two measures does not fall of the 1:1 line (is actually perpendicular to it). ***Answer 4c***: ^

B. Visualizing Phylogenetic Beta-Diversity

Now that we have our phylogenetically based community resemblance matrix, we can visualize phylogenetic diversity among samples using the same techniques that we used in the β-diversity module from earlier in the course.

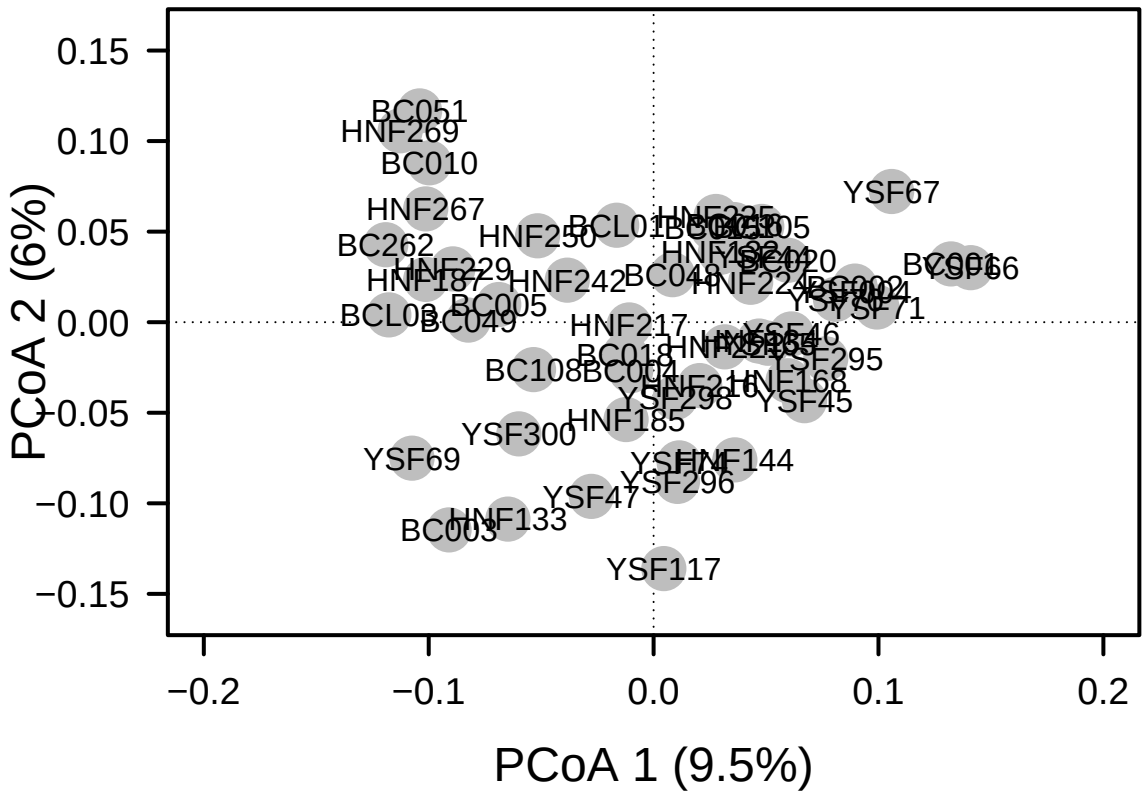
In the R code chunk below, do the following:
1. perform a PCoA based on the UniFrac distances, and
2. calculate the explained variation for the first three PCoA axes.

```
pond.pcoa <- cmdscale(dist.uf, eig = T, k = 3)

explainvar1 <- round(pond.pcoa$eig[1] / sum(pond.pcoa$eig), 3) * 100
explainvar2 <- round(pond.pcoa$eig[2] / sum(pond.pcoa$eig), 3) * 100
explainvar3 <- round(pond.pcoa$eig[3] / sum(pond.pcoa$eig), 3) * 100

sum.eig <- sum(explainvar1, explainvar2, explainvar3)
```

Now that we have calculated our PCoA, we can plot the results.

In the R code chunk below, do the following:
1. plot the PCoA results using either the R base package or the `ggplot` package,
2. include the appropriate axes,
3. add and label the points, and
4. customize the plot.

```
par(mar = c(5,5,1,2) + 0.1)

plot(pond.pcoa$points[, 1], pond.pcoa$points[,2],
     xlim = c(-0.2, 0.2), ylim = c(-.16,0.16),
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

points(pond.pcoa$points[,1], pond.pcoa$points[,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(pond.pcoa$points[,1], pond.pcoa$points[,2],
     labels = row.names(pond.pcoa$points))
```



In the following R code chunk: 1. perform another PCoA on taxonomic data using an appropriate measure of dissimilarity, and 2. calculate the explained variation on the first three PCoA axes.

Question 5: Using a combination of visualization tools and percent variation explained, how does the phylogenetically based ordination compare or contrast with the taxonomic ordination? What does this tell you about the importance of phylogenetic information in this system?

> ***Answer 5***: There is much less variation explained in this PCoA (max 9.5%) and the clustering is weak. In taxonomic PCoAs the percent variation was usually much higher and more clearly clustered. Phylogenetic information may not explain variation as well because mostly more conserved?

C. Hypothesis Testing

i. Categorical Approach

In the R code chunk below, do the following:
1. test the hypothesis that watershed has an effect on the phylogenetic diversity of bacterial communities.

```
watershed <- env$Location

phylo.adonis <- adonis2(dist.uf ~ watershed, permutations = 999)

tax.adonis <- adonis2(vegdist(
  decostand(comm, method = "log"),
  method = "bray") ~ watershed,
  permutations = 999)
```

ii. Continuous Approach

In the R code chunk below, do the following: 1. from the environmental data matrix, subset the variables related to physical and chemical properties of the ponds, and
2. calculate environmental distance between ponds based on the Euclidean distance between sites in the environmental data matrix (after transforming and centering using `scale()`).

```
envs <- env[,5:19]

envs <- envs[, -which(names(envs) %in% c("TDS", "Salinity", "Cal_Volume"))]

env.dist <- vegdist(scale(envs), method = "euclid")
```

In the R code chunk below, do the following:
1. conduct a Mantel test to evaluate whether or not UniFrac distance is correlated with environmental variation.

```
mantel(dist.uf, env.dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = dist.uf, ydis = env.dist)
##
## Mantel statistic r: 0.1604
##       Significance: 0.062
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.129 0.171 0.197 0.251
## Permutation: free
## Number of permutations: 999
```

Last, conduct a distance-based Redundancy Analysis (dbRDA).

In the R code chunk below, do the following:
1. conduct a dbRDA to test the hypothesis that environmental variation effects the phylogenetic diversity of bacterial communities,
2. use a permutation test to determine significance, and 3. plot the dbRDA results

```
ponds.dbrda <- vegan::dbrda(dist.uf ~ ., data = as.data.frame(scale(envs)))

anova(ponds.dbrda, by = "axis")
```

```
## Permutation test for dbrda under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: vegan::dbrda(formula = dist.uf ~ Elevation + Diameter + Depth + ORP + Temp + SpC + DO + pH + C
##          Df SumOfSqs      F Pr(>F)
## dbRDA1    1  0.10566 2.0152  0.449
```

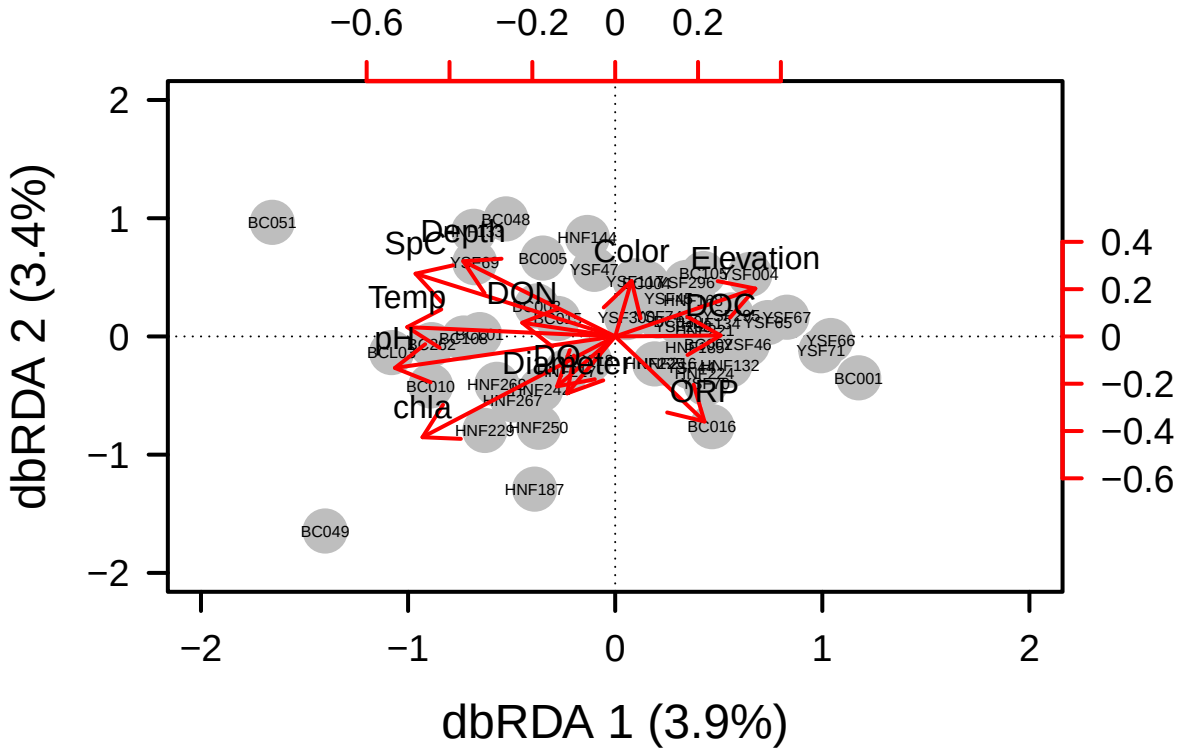
```
## dbRDA2    1  0.09258 1.7658  0.651
## dbRDA3    1  0.07555 1.4409  0.981
## dbRDA4    1  0.06677 1.2735  0.998
## dbRDA5    1  0.05666 1.0807  1.000
## dbRDA6    1  0.05293 1.0095  1.000
## dbRDA7    1  0.04750 0.9059  1.000
## dbRDA8    1  0.03941 0.7517  1.000
## dbRDA9    1  0.03775 0.7201  1.000
## dbRDA10   1  0.03280 0.6256  1.000
## dbRDA11   1  0.02876 0.5485  1.000
## dbRDA12   1  0.02501 0.4770  0.998
## Residual 39  2.04482
```

```
ponds.fit <- envfit(ponds.dbrda, envs, perm = 999)
ponds.fit
```

```
##
## ***VECTORS
##
##             dbRDA1   dbRDA2     r2 Pr(>r)
## Elevation  0.77670  0.62986 0.0959  0.075 .
## Diameter  -0.27972 -0.96008 0.0541  0.261
## Depth     -0.63137  0.77548 0.1756  0.011 *
## ORP        0.41879 -0.90808 0.1437  0.026 *
## Temp      -0.98250  0.18628 0.1523  0.022 *
## SpC       -0.77101  0.63682 0.2087  0.005 **
## DO        -0.39318 -0.91946 0.0464  0.337
## pH        -0.96210 -0.27270 0.1756  0.011 *
## Color      0.06353  0.99798 0.0464  0.334
## chla      -0.60392 -0.79704 0.2626  0.010 **
## DOC        0.99847 -0.05526 0.0382  0.390
## DON       -0.91633  0.40042 0.0339  0.417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
dbrda.explainvar1 <- round(ponds.dbrda$CCA$eig[1]/
                           sum(c(ponds.dbrda$CCA$eig, ponds.dbrda$CA$eig)), 3) * 100
dbrda.explainvar2 <- round(ponds.dbrda$CCA$eig[2]/
                           sum(c(ponds.dbrda$CCA$eig, ponds.dbrda$CA$eig)), 3) * 100

par(mar = c(5,5,4,4) + 0.1)

plot(scores(ponds.dbrda, display = "wa"), xlim = c(-2,2), ylim = c(-2,2),
     xlab = paste("dbRDA 1 (", dbrda.explainvar1, "%)", sep = ""),
     ylab = paste("dbRDA 2 (", dbrda.explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)
```

```
points(scores(ponds.dbrda, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(scores(ponds.dbrda, display = "wa"),
     labels = row.names(scores(ponds.dbrda, display = "wa")), cex = 0.5)

vectors <- scores(ponds.dbrda, display = "bp")
arrows(0,0, vectors[,1] * 2, vectors[,2]*2,
       lwd = 2, lty = 1, length = 0.2, col = "red")
text(vectors[,1] * 2, vectors[,2] * 2, pos = 3,
     labels = row.names(vectors))
axis(side = 3, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[,1])) * 2, labels = pretty(range(vectors[,1])))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[,2])) * 2, labels = pretty(range(vectors[,2])))
```



Question 6: Based on the multivariate procedures conducted above, describe the phylogenetic patterns of β-diversity for bacterial communities in the Indiana ponds.

> ***Answer 6***: Both the Mantel test and the dbRDA suggest there is little correlation between phylogenetic patterns of beta diveristy and the environmental characteristics of Indiana ponds. The r and p values of the Mantel test are not significant. The dbRDA does not appear to have any strong patterns.

6) SPATIAL PHYLOGENETIC COMMUNITY ECOLOGY

A. Phylogenetic Distance-Decay (PDD)

A distance decay (DD) relationship reflects the spatial autocorrelation of community similarity. That is, communities located near one another should be more similar to one another in taxonomic composition than distant communities. (This is analagous to the isolation by distance (IBD) pattern that is commonly found when examining genetic similarity of a populations as a function of space.) Historically, the two most common explanations for the taxonomic DD are that it reflects spatially autocorrelated environmental variables and the influence of dispersal limitation. However, if phylogenetic diversity is also spatially autocorrelated, then evolutionary history may also explain some of the taxonomic DD pattern. Here, we will construct the phylogenetic distance-decay (PDD) relationship

First, calculate distances for geographic data, taxonomic data, and phylogenetic data among all unique pair-wise combinations of ponds.

In the R code chunk below, do the following:
1. calculate the geographic distances among ponds,
2. calculate the taxonomic similarity among ponds,
3. calculate the phylogenetic similarity among ponds, and
4. create a dataframe that includes all of the above information.

```
long.lat <- as.matrix(cbind(env$long, env$lat))
coord.dist <- earth.dist(long.lat, dist = TRUE)

bray.curtis.dist <- 1 - vegdist(comm)

unifrac.dist <- 1 - dist.uf

unifrac.dist.mlt <- melt(as.matrix(unifrac.dist))[melt(upper.tri(as.matrix(unifrac.dist)))$value,]
```

```
## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE

## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE
```

```
bray.curtis.dist.mlt <- melt(as.matrix(bray.curtis.dist))[melt(upper.tri(as.matrix(bray.curtis.dist)))$
```

```
## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE

## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE
```

```
coord.dist.mlt <- melt(as.matrix(coord.dist))[melt(upper.tri(as.matrix(coord.dist)))$value,]
```

```
## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE

## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE
```

```
env.dist.mlt <- melt(as.matrix(env.dist))[melt(upper.tri(as.matrix(env.dist)))$value,]
```

```
## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE

## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by the
## caller; using TRUE
```

```
df <- data.frame(coord.dist.mlt, bray.curtis.dist.mlt[,3], unifrac.dist.mlt[,3], env.dist.mlt[,3])
names(df)[3:6] <- c("geo.dist", "bray.curtis", "unifrac", "env.dist")
```

Now, let's plot the DD relationships:
In the R code chunk below, do the following:
1. plot the taxonomic distance decay relationship,
2. plot the phylogenetic distance decay relationship, and
3. add trend lines to each.

```
par(mfrow = c(2,1), mar = c(1,5,2,1) + 0.1, oma = c(2,0,0,0))

plot(df$geo.dist, df$bray.curtis, xlab = "", xaxt = "n", las = 1, ylim = c(0.1,0.9),
     ylab = "Bray-Curtis Similarity",
     main = "Distance Decay", col = "SteelBlue")

DD.reg.bc <- lm(df$bray.curtis ~ df$geo.dist)
summary(DD.reg.bc)
```

```
##
## Call:
## lm(formula = df$bray.curtis ~ df$geo.dist)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31151 -0.08843  0.00315  0.09121  0.43817
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4463453  0.0066883  66.735   <2e-16 ***
## df$geo.dist -0.0013051  0.0005864  -2.226   0.0262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1303 on 1324 degrees of freedom
## Multiple R-squared:  0.003728,   Adjusted R-squared:  0.002975
## F-statistic: 4.954 on 1 and 1324 DF,  p-value: 0.0262
```

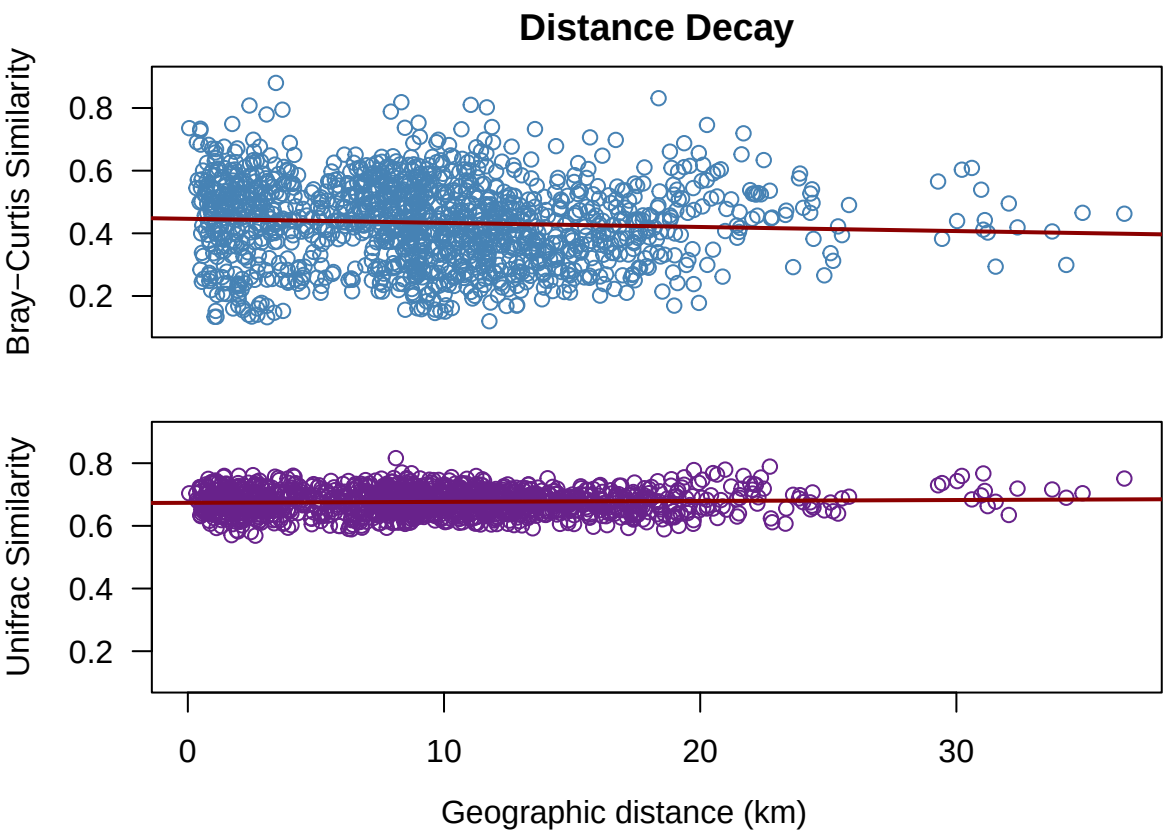
```
abline(DD.reg.bc, col = "red4", lwd = 2)

par(mar = c(2,5,1,1) + 0.1)

plot(df$geo.dist, df$unifrac, xlabb = "", las = 1, ylim = c(0.1,0.9),
     ylab = "Unifrac Similarity", col = "darkorchid4")
```

```
## Warning in plot.window(...): "xlabb" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlabb" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlabb" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlabb" is not a
## graphical parameter

## Warning in box(...): "xlabb" is not a graphical parameter

## Warning in title(...): "xlabb" is not a graphical parameter
```

```
DD.reg.uni <- lm(df$unifrac ~ df$geo.dist)
summary(DD.reg.uni)
```

```
##
## Call:
## lm(formula = df$unifrac ~ df$geo.dist)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.105629 -0.027107 -0.000077  0.026761  0.140215
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.6735186  0.0019206 350.677   <2e-16 ***
## df$geo.dist 0.0002976  0.0001684   1.767   0.0774 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03741 on 1324 degrees of freedom
## Multiple R-squared:  0.002354,   Adjusted R-squared:  0.0016
## F-statistic: 3.124 on 1 and 1324 DF,  p-value: 0.07738
```

```
abline(DD.reg.uni, col = "red4", lwd = 2)

mtext("Geographic distance (km)", side = 1, adj = 0.55, line = 0.5, outer = TRUE)
```



In the R code chunk below, test if the trend lines in the above distance decay relationships are different from one another.

```
"diffslope" <-
  function(x1, y1, x2, y2, permutations=1000, ic=FALSE, resc.x=FALSE, resc.y=TRUE, trace=FALSE, ...) {
    if (resc.x) {
      maxS <- max(mean(x1), mean(x2)) #das hoehere der beiden means herausfinden
      x1 <- x1+(maxS-mean(x1)) #und auf beide datensaetze anwenden
      x2 <- x2+(maxS-mean(x2))
    }
    if (resc.y) {
      maxD <- max(mean(y1), mean(y2))
      y1 <- y1+(maxD-mean(y1))
      y2 <- y2+(maxD-mean(y2))
    }
    m1 <- data.frame(as.numeric(y1), as.numeric(x1))
    m2 <- data.frame(as.numeric(y2), as.numeric(x2))
    names(m1) <- c("x","y")
    names(m2) <- c("x","y")
    m1.lm <- lm(m1) #die beiden linearen modelle rechnen
    m2.lm <- lm(m2)
    ds0 <- as.numeric(m1.lm$coefficients[2]-m2.lm$coefficients[2]) #deren differenz ausrechnen
    if(ic){
      m12.lmcoeff <- matrix(data=NA, nrow=permutations, ncol=2)
      m21.lmcoeff <- matrix(data=NA, nrow=permutations, ncol=2)
      dic <- as.numeric(m1.lm$coefficients[1]-m2.lm$coefficients[1])
```

```
    if (trace) {cat(permutations, "perms: ")}
    for(i in 1:permutations) {
      tmp1 <- sample(nrow(m1), nrow(m1)/2)
      tmp2 <- sample(nrow(m2), nrow(m2)/2)
      m12 <- rbind(m1[tmp1,], m2[tmp2,])
      m21 <- rbind(m1[-tmp1,], m2[-tmp2,])
      m12.lmcoeff[i,] <- as.numeric(lm(m12)$coefficients)
      m21.lmcoeff[i,] <- as.numeric(lm(m21)$coefficients)
      if (trace) {cat(paste(i,""))}
    }
    perms <- m12.lmcoeff - m21.lmcoeff
    if (ds0 >= 0) {
      signif <- length(perms[perms[,2]>=ds0,2])/permutations
    }
    else {
      signif <- length(perms[perms[,2]<=ds0,2])/permutations
    }
    if (dic >= 0) {
      signific <- length(perms[perms[,1]>=ds0,1])/permutations
    }
    else {
      signific <- length(perms[perms[,1]<=ds0,1])/permutations
    }

    if (signif == 0) {
      signif <- 1/permutations
    }
    if (signific == 0) {
      signific <- 1/permutations
    }
  }
  else{
    perms <- vector("numeric", permutations)
    if (trace) {cat(permutations, "perms: ")}
    for(i in 1:permutations) {
      tmp1 <- sample(nrow(m1), nrow(m1)/2)
      tmp2 <- sample(nrow(m2), nrow(m2)/2)
      m12 <- rbind(m1[tmp1,], m2[tmp2,])
      m21 <- rbind(m1[-tmp1,], m2[-tmp2,])
      perms[i] <- as.numeric(lm(m12)$coefficients[2]-lm(m21)$coefficients[2])
      if (trace) {cat(paste(i,""))}
    }
    if (ds0 >= 0) {
      signif <- length(perms[perms>=ds0])/permutations
    }
    else {
      signif <- length(perms[perms<=ds0])/permutations
    }
    if (signif == 0) {
      signif <- 1/permutations
    }
    perms <- cbind(perms,perms)
  }
```

```
    res <- c(call=match.call())
    res$slope.diff <- as.numeric(ds0)
    res$signif <- signif
    res$permutations <- permutations
    res$perms <- perms[,2]
    class(res) <- "dsl"
    if(ic) {
      res$intercept <- as.numeric(dic)
      res$signific <- as.numeric(signific)
      res$permsic <- as.numeric(perms[,1])
      class(res) <- "dsl2"
    }
    res
  }


#diffslope(df$geo.dist, df$unifrac, df$geo.dist, df$bray.curtis)
```

Question 7: Interpret the slopes from the taxonomic and phylogenetic DD relationships. If there are differences, hypothesize why this might be.

Answer 7:

SYNTHESIS

Ignoring technical or methodological constraints, discuss how phylogenetic information could be useful in your own research. Specifically, what kinds of phylogenetic data would you need? How could you use it to answer important questions in your field? In your response, feel free to consider not only phylogenetic approaches related to phylogenetic community ecology, but also those we discussed last week in the PhyloTraits module, or any other concepts that we have not covered in this course.

> I would be interested in incorporating phylogenetic information into my own research, specifically to capture differences in the clonal genotypes of Daphnia dentifera that we work with, if possible/makes sense to. Furthermore, it would be interesting to capture for of the phylogenetic diversity that exists in the lake communities. My own questions are focused on size structure and size variation in daphnia, which may be represented in some phylogeny? An old labmate recently came up with a sequencing technique, which might be promising to play around with to test some of these thoughts. Still I am unfamiliar with phylogenetic information so I first need to consider how the communities and taxa I work with would be represented, and how useful it would be at capturing beta diversity across sites.. that might be very cool actually. – additional considerations for responses to environmental characterisics/niche breadth.