11. Phylogenetic Diversity - Communities

Z620: Quantitative Biodiversity, Indiana University

OVERVIEW

Last week, we examined the distribution of functional traits with respect to the shared evolutionary history (i.e., phylogeny) of organisms. This week, we will use phylogenetic information to aid our understanding of patterns and processes in community ecology. To accomplish this, we will introduce concepts linking evolutionary relatedness to community structure. We will quantify phylogenetic α- and β-diversity, and discuss key patterns that are encountered in phylogenetic community ecology, such as clustering and overdispersion.

After completing this exercise 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

BACKGROUND

Phylogenetic insight has played a key role in the historical development of community ecology. Darwin (1859) observed that closely related species (e.g., species in the same genus) are more ecologically similar than distantly related species. This observation led to the identification of a paradox. On the one hand, in order for a species to successfully invade a community it must be sufficiently *related* to resident taxa and thereby share ecologically relevant traits that allow it to contend with local conditions (e.g., pH). On the other hand, the species must be sufficiently *unrelated* to resident taxa and thereby possess some unique traits so that it can exploit unoccupied niches (Diez et al. 2008). These concepts are reflected in the contemporary views of community ecology. Species in a community must be ecologically similar enough to one another to have positive fitness in the environment (i.e., equalizing forces), but also be different enough (i.e., stabilizing niche differences) that they can coexist (e.g., see Leibold and McPeek 2006, Adler et al. 2007, HilleRisLambers et al. 2012). In theory, ecological similarity can be determined by measuring the functional traits of all the individuals for all species in a community. In practice, it is challenging to measure ecologically relevant traits of individuals in complex communities. Therefore, many scientists use phylogeny as a proxy for traits especially since phylogenetic information is becoming increasingly accessible. Ecologists have attempted to use phylogenetic information to gain insight into community assembly processes. For example, if a community is dominated by closely related taxa (i.e. clustering), one might infer that environmental filtering is acting on a set of shared functional traits. In contrast, if a community is comprised of more distantly related taxa (i.e., overdispersion), one might infer that competitive exclusion is an important assembly process. It is critical to note, however, that other mechanisms can give rise to patterns of phylogenetic clustering and overdispersion (e.g., see Mayfield and Levine 2010 for discussion).

1) SETUP

A. Retrieve and Set Your Working Directory

```r
rm(list = ls())
getwd()
setwd("~/GitHub/QB-2023/1.HandOuts/11.PhyloCom")
```

B. Load Packages

Several R packages have been developed for conducting phylogenetic analyses (http://goo.gl/DtU16j). In particular, we will rely heavily on the R package `picante`. The `picante` package has many of the functions that are contained in the software Phylocom, which was developed for the analysis of phylogenetic community structure and character evolution (http://phylodiversity.net/phylocom/). We will describe `picante` and other packages in greater detail in the sections below. In the chunk of code below, the `require()` function in `R` returns `TRUE` if the package was successfully loaded or `FALSE` if the package failed to load. This `for` loop loads each package and installs the package when `require()` returns `FALSE`.

```r
package.list <- c('picante', 'ape', 'seqinr', 'vegan', 'fossil',
                  'reshape', 'devtools', 'BiocManager', 'ineq',
                  'labdsv', 'matrixStats', 'pROC')
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)
  }
}
```

C. Load Source Code

In addition to relying on contributed packages, we will also be using a source code file. A source code file has user-defined functions that are required for certain analyses. The benefit of source files is that they contain “vetted” code that can be used across multiple projects. Here, we will use a source code file that includes functions for reading in the output files that contain OTU and taxonomic information generated from `mothur` (http://www.mothur.org/), a popular software platform for analyzing bacterial sequence data. Take a moment to open the file and examine the source code.

```r
source("./bin/MothurTools.R")
```

2) DESCRIPTION OF DATA

We will revisit the environmental and community data that we used in the Spatial Diversity module of Quantitative Biodiversity a few weeks ago. As a reminder, in 2013 we sampled more than 50 forested ponds located in Brown County State Park, Yellowwood 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 using barcoded primers on the Illumina MiSeq platform. The recovery of a 16S rRNA gene tells us that a certain taxon was present in a pond, while the recovery of a 16S rRNA transcript provides information about the metabolic activity of that taxon. We used a `mothur` pipeline 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.

3) LOAD AND PROCESS DATA

A. Environmental and Geographic Data

First, let's load the environmental and geographic data. There are a few ponds with missing environmental data. For simplicity, we are going to remove those ponds from our analyses.

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

Here is a description of the column headers in `env`:

- Location = Brown County State Park (BCSP), Hoosier National Forest (HNF), Yellowwood State Forest (YSF)
- Sample_ID = unique pond identifier (e.g., "BC001" = pond 1 from BCSP)
- lat = geographic coordinate, degrees latitude
- long = geographic coordinate, degrees longitude
- Elevation = meters above sea level
- Diameter = dimension of pond (m)
- Depth = average depth of pond (m)
- Cal_Volume = calculated volume of pond (m^3)
- ORP = oxidation reduction potential (mV)
- Temp = surface water temperature (C °)
- SpC = specific conductivity of water (μS/m)
- DO = dissolved oxygen of water (mg/L)
- TDS = total dissolved solids (mg/L)
- Salinity = parts per thousand (‰)
- pH = negative log of the hydrogen ion concentration
- Color = absorbance of light by water at 440 nm
- chla = chlorophyll *a* (μg/L)
- DOC = dissolved organic carbon (mg/L)
- DON = dissolved organic nitrogen (mg/L)
- canopy = open canopy above pond (%)
- TP = total phosphorus (μg/L)

B. Taxonomic Data

Now let's load the bacterial OTU data (i.e., site-by-species matrix). First, we will use the `read.OTU()` function from the source code. The rownames contain information about whether a site (pond) represents data obtained from 16S rRNA genes (i.e., DNA) or 16S rRNA transcripts, which were sequenced as copy DNA (cDNA). We want to retain only the DNA data for this module. Therefore, we are going to use the `grep()` function, which is designed for matching patterns in a string of characters, or "regular expressions". Here, we only want the rownames that end in "-DNA", not those that end in "-cDNA". Then we will use the `gsub()` function, which replaces a matched regular expression (the first argument) with another character string (provided in the second argument). Here, we will simply clean up the rownames so they are less cumbersome. Inspect the rownames of `comm` (e.g., `rownames(comm)`) after each grep command to see how it works.

```
# 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))
```

Before moving forward, we need to do some additional clean-up of the `comm` dataframe. First, we'll remove sites that are not contained in the `env` dataframe. Second, because we removed the RNA samples, there are likely to be some OTUs that are not found in the DNA samples. So, we'll remove any zero-abundance taxa (i.e., empty columns).

```
# 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]
```

Now import and take a look at the taxonomic information associated with the OTU data using the `read.tax()` function from source code.

```
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
```

C. Phylogenetic Data

The next step is to load and process the phylogenetic data. First, we need to read in a file containing the aligned sequences using the `read.alignment()` function in the `seqinr` package.

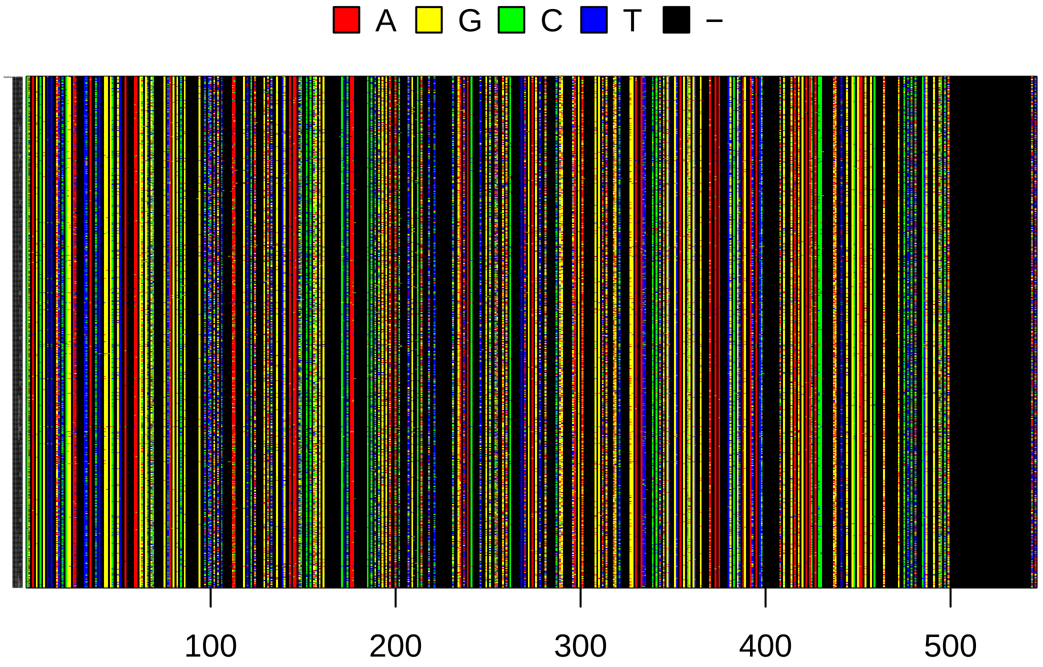
```
# Import the alignment file (`seqinr`)
ponds.cons <- read.alignment(file = "./data/INPonds.final.rdp.1.rep.fasta",
                             format = "fasta")
```

You can visualize the alignment in `ponds.cons`. In so doing, you may observe that the names associated with the FASTA entries are not very useful. So, again, we'll use `gsub()` to clean things up.

```
# Rename OTUs in the FASTA File
ponds.cons$nam <- gsub("\\|.*$", "", gsub("^.*?\t", "", ponds.cons$nam))
```

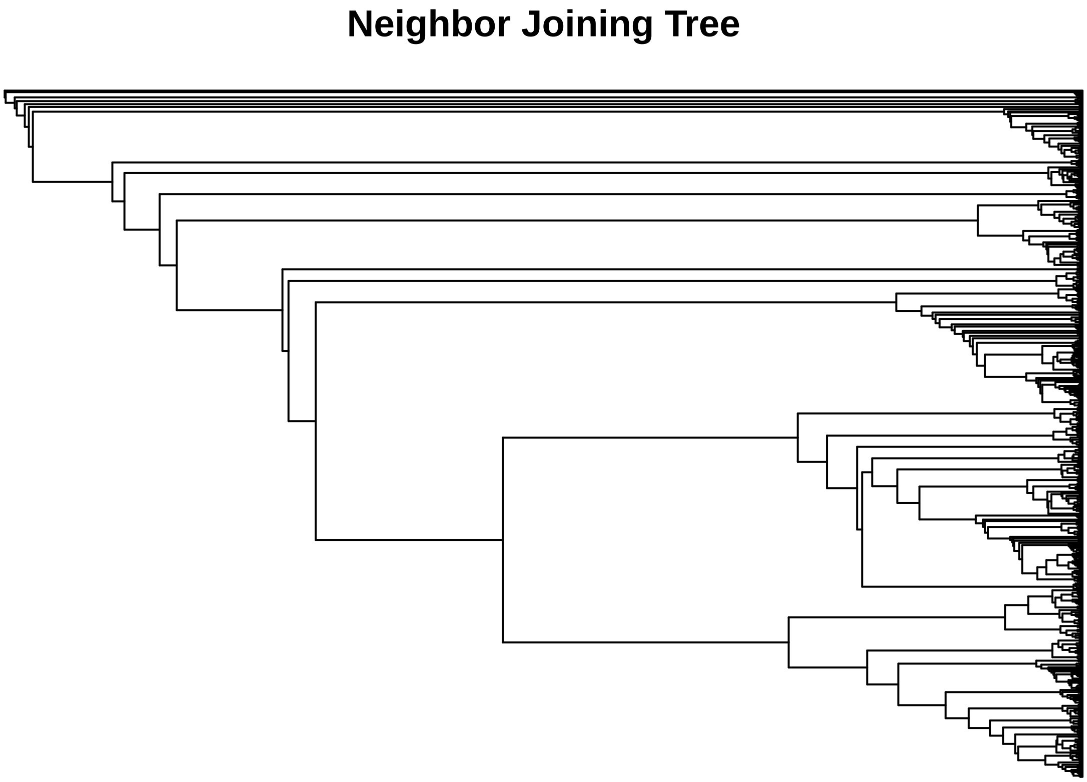
Next, we will import an archaeal sequence, specify this as our outgroup, and then convert the alignment file to a DNAbin object using the `ape` package. We can then visualize the alignment using the `image.DNAbin()` function.

```
# 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)
```



With the DNAbin object that we just made, we'll now create a distance matrix with the Jukes Cantor "JC" Model using the `ape` package. You will remember from the PhyloTraits module that one benefit of this approach is that it is computationally efficient. We will then use the neighbor joining algorithm to construct a tree file and we will remove any tips associated with zero-abundance taxa. Finally, we will specify the outgroup sequence, root the tree, and plot the tree.

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

# Make a neigbor-joining tree file (`ape`)
phy.all <- bionj(seq.dist.jc)

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

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

# Root the tree {ape}
phy <- root(phy, outgroup, resolve.root = TRUE)

# Plot the rooted tree {ape}
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

Now that we have loaded and processed all of our data, we are prepared to quantify phylogenetic diversity. Tucker et al. (2016) discuss several methods for quantifying phylogenetic diversity. We will start by assessing the within site or α-phylodiversity. In general, a sample consisting of species that are less closely related has higher phylogenetic diversity. Maintaining high phylogenetic diversity has been proposed as a more effective way to maintain ecosystem functionality and is often the focus of conservation biology. In the following sections, we will introduce a few ways to calculate α-phylodiversity.

A. Faith's Phylogenetic Diversity (PD)

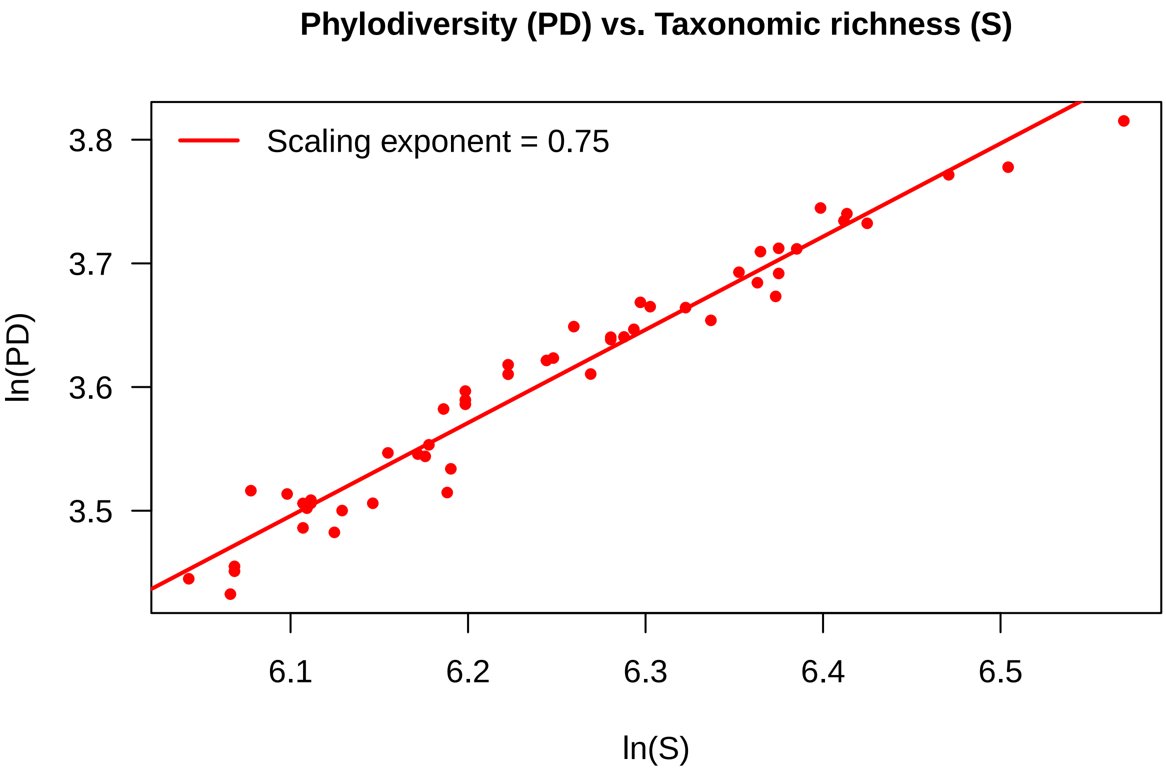
In 1992, Daniel Faith developed a diversity metric called Faith's PD (http://goo.gl/wM08Oy). The metric sums the branch lengths for each species found in a sample from the root to the tip of the phylogenetic tree. It is important to note that PD, as implemented here, is based on the presence-absence of an OTU; that is, it does not weight the branches of common species more than the branches of rare species. Higher PD values indicate that an assemblage contains more evolutionarily divergent taxa, while lower PD values indicate that an assemblage contains taxa with a more restricted evolutionary history.

Faith's PD can be implemented in R using the `pd()` function in the `picante` package. A phylogenetic tree containing the species pool is required. In addition to returning Faith's PD, the `pd()` function also returns species richness (S), which is identical to observed richness (S_{obs}) that we covered in the α diversity module.

```r
# Calculate PD and S {picante}
pd <- pd(comm, phy, include.root = FALSE)
```

Let's compare PD estimates with S of from our 52 pond samples. We'll transform our data by taking the natural logarithm of S and PD so that the slope of the relationship gives us a power-law exponent, which describes how PD scales with S.

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

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

# 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")
```



i. Randomizations and Null Models In order to draw strong conclusions about the phylogenetic diversity of a sample, it is useful to know something about how variable our metric is under null conditions. Randomization is powerful way to resample data to assess whether or not observed patterns are different from a null expectation. A number of the functions in the `picante` package allow us to specify different null models as an argument. These null models can control for features such as species richness, species occurrence frequency, and the diversity of the regional species pool. We will use some of these models for assessing the degree to which phylogenetic measures of α phylodiversity deviate from null expectations. The following table describes some of the null models that are available to us when using `picante`:

Null Model	Description
taxa.labels	Shuffles taxa labels across tips of phylogeny (across all taxa included in phylogeny)
richness	Randomizes community data matrix abundances within samples (maintains sample species richness)
frequency	Randomizes community data matrix abundances within species (maintains species occurrence frequency)
sample.pool	Randomizes community data matrix by drawing species from pool of species occurring in at least one community (sample pool) with equal probability
phylogeny.pool	Randomize community data matrix by drawing species from pool of species occurring in at least one community (sample pool) with equal probability
independentswap	Randomizes community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness
trialswap	Randomizes community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness

Now, we are going to use the `ses.pd()` function in `picante`. This function estimates the standardized effect size ("ses") using the following equation: `ses.pd` = (`pd.obs` - `pd.rand.mean`) / `pd.rand.sd`, where `pd.obs` is the observed PD, `pd.rand.mean` is the mean of the PD values generated via randomization under a null model, and `pd.rand.sd` is the standard deviation of the PD values generated via randomization under a null model (see table above). When the observed value is greater than the null expectation (i.e., ses.pd > 0), our sample is more phylogenetically diverse than expected under the null distribution. Given the size of both our site-by-species matrix and the phylogenetic tree, the randomization process is computationally intensive. Therefore, we are only going to run the `ses.pd` function for two ponds with a limited number of randomizations (i.e., "runs" argument).

```
# Estimate standardized effect size of PD via randomization (`picante`)
ses.pd <- ses.pd(comm[1:2,], phy, null.model = "richness", runs = 25,
                 include.root = FALSE)
```

B. Phylogenetic Dispersion Within a Sample

Earlier, we described how the relatedness of species in a sample can deviate from a random distribution. Clustered communities contain species that are more closely related than would be expected by chance. Overdispersed communities contain species that are more distantly related than would be expected by chance. In this section, we introduce two ways of quantifying α phylodiversity based on the **dispersion** of species in a sample: the Net Relatedness Index (NRI) and the Nearest Taxon Index (NTI). We will use randomization procedures to test whether species are phylogenetically clustered or overdispersed.

i. Phylogenetic Resemblance Matrix Before estimating dispersion metrics, we need to create a phylogenetic resemblance matrix. This type of matrix is nearly identical to the resemblance matrix introduced in the β-diversity module. The only difference is that the phylogenetic resemblance matrix contains distances between *taxa* in a tree, whereas the community resemblance matrix contains distances among *sites*. The elements in phylogenetic resemblance matrix are calculated as the pairwise branch-length distances between tips (i.e., taxa) on a phylogenetic tree. The phylogenetic resemblance matrix is sometimes referred to as the phylogenetic variance-covariance matrix. We will use the `cophenetic.phylo()` function in the `picante` package to calculate the phylogenetic resemblance matrix.

```r
# Create a Phylogenetic Distance Matrix (`picante`)
phydist <- cophenetic.phylo(phy)
```

ii. Net Relatedness Index (NRI) One common way to test for phylogenetic clustering and overdispersion is to use the Net Relatedness Index (NRI). NRI is based on the mean phylogenetic distance (MPD), which is calculated as the average pairwise branch length between the taxa in a sample, a subset of the phylogenetic resemblance matrix (the `phydist` object above). After calculating MPD, the NRI is expressed as: `-` (`mpd.obs` `-` `mpd.rand.mean`) / `mpd.rand.sd`, where `mpd.obs` is the observed MPD of a sample, `mpd.rand.mean` is the mean of the MPD values generated via randomization under a null model, and `mpd.rand.sd` is the standard deviation of the MPD values generated via randomization under a null model.

Negative NRI values indicate that a sample is phylogenetically overdispersed; that is, taxa are less related to one another than expected under the null model. Positive NRI values indicate that a sample is phylogenetically underdispersed, or clustered, such that taxa are more closely related to one another than expected under the null model.

As with Faith's PD, the randomization procedures are computationally intensive, so we are only going to perform a relatively small number of "runs".

```r
# Estimate standardized effect size of NRI via randomization (`picante`)
ses.mpd <- ses.mpd(comm, phydist, null.model = "taxa.labels",
                   abundance.weighted = FALSE, runs = 25)

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

iii. Nearest Taxon Index (NTI)

Another way to test for phylogenetic clustering and overdispersion in a sample is to use the Nearest Taxon Index (NTI). This index is mathematically similar to NRI, but uses the mean nearest phylogenetic neighbor distance (MNND) instead of MPD. MNND is the mean phylogenetic distance between all taxa in a sample and their phylogenetically closest neighbor. Effectively, this is the sum of the minimum values for each row or column (i.e., taxon) of the square, non-triangular, phylogenetic resemblance matrix. Because only the most closely related taxa are considered, NTI tends to emphasize terminal clustering (i.e., near the tips), independent of deep level clustering (Webb et al. 2002; http://goo.gl/WikgWE). Just like NRI, we perform randomizations and use this information to estimate the standardized effect size. The NTI is then calculated as -(`mntd.obs` `-` `mntd.mean`)/`mntd.sd`. Negative NTI values indicate phylogenetic overdispersion (because nearest taxa are more distantly related than expected) and positive NTI values indicate phylogenetic clustering (because nearest taxa are more closely related than expected).

```r
# Estimate Standardized Effect Size of NRI via Randomization {picante}
ses.mntd <- ses.mntd(comm, phydist, null.model = "taxa.labels",
                     abundance.weighted = FALSE, runs = 25)

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

5) PHYLOGENETIC BETA DIVERSITY

In the β-diversity module earlier this semester, we learned about approaches to quantify how diversity changes among sites and how to link this to underlying environmental gradients. Here, we will expand upon the concepts of taxonomic β-diversity by incorporating phylogenetic information. By doing so, we can gain insight into landscape patterns of biodiversity and the eco-evolutionary processes that may be shaping them.

A. Phylogenetically Based Community Resemblance Matrix

Recall that a resemblance matrix is needed to quantify β-diversity for more than two samples. When quantifying taxonomic β-diversity, we calculated the pairwise **similarity** or **dissimilarity** for all samples in a site-by-species matrix using metrics such as the Sørensen index or the Bray-Curtis index. We need a similar process for quantifying phylogenetic dissimilarity between sites. Instead of making the resemblance matrix based on incidence or abundance of taxa, we are going to incorporate information about the phylogenetic relationships among taxa. Similar to other measures of β-diversity, there are numerous ways to calculate the phylogenetic distances in the community resemblance matrix. Here, we will explore two: Mean Pairwise Distance and UniFrac distance.

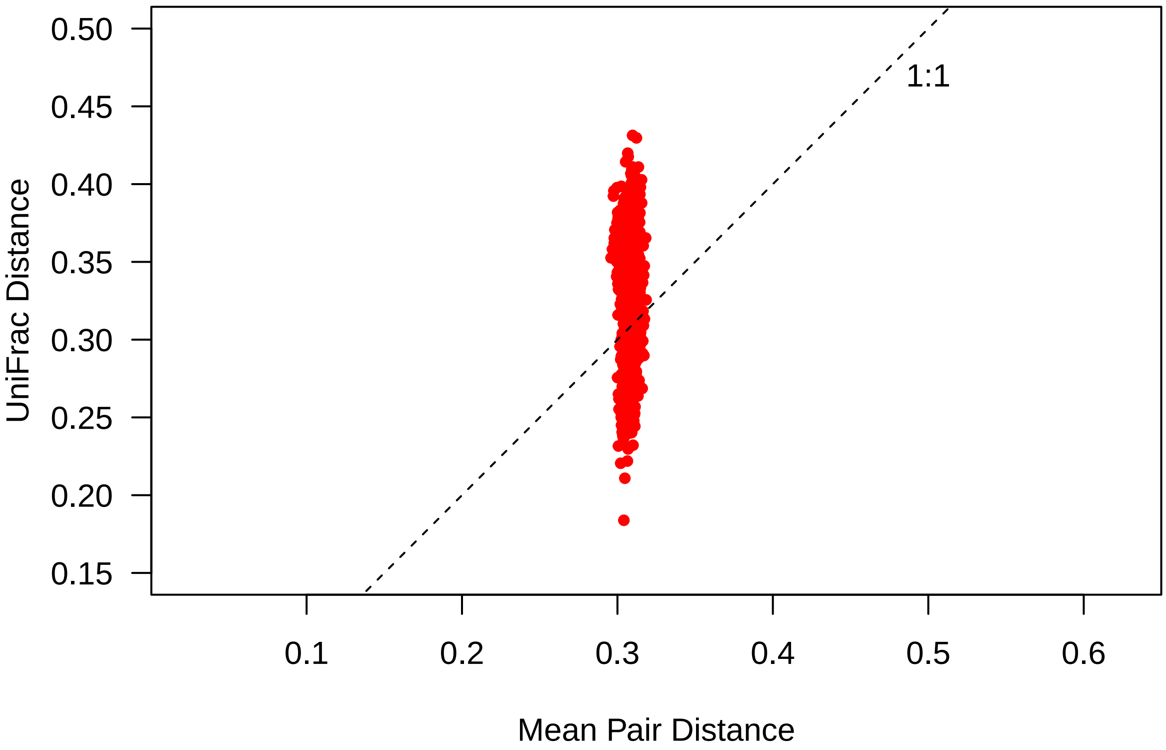
Index	Description
Mean Pairwise Distance	Distance between two samples calculated as the mean phylogenetic distance between pairs of taxa
UniFrac	Distance between two samples calculated as $\Sigma_{unshared}$ / Σ_{total}, where $\Sigma_{unshared}$ is the sum of unshared branch lengths between samples and Σ_{total} is the total (shared and unshared) branch lengths in a rooted tree

Let us calculate the phylogenetic community resemblance matrices for our pond data set using Mean Pairwise Distance and UniFrac distance.

```
# Mean Pairwise Distance
dist.mp <- comdist(comm, phydist)

# UniFrac Distance (Note: this takes a few minutes; be patient)
dist.uf <- unifrac(comm, phy)
```

Now, let us compare the Mean Pair Distance and UniFrac distance matrices.

```
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")
```



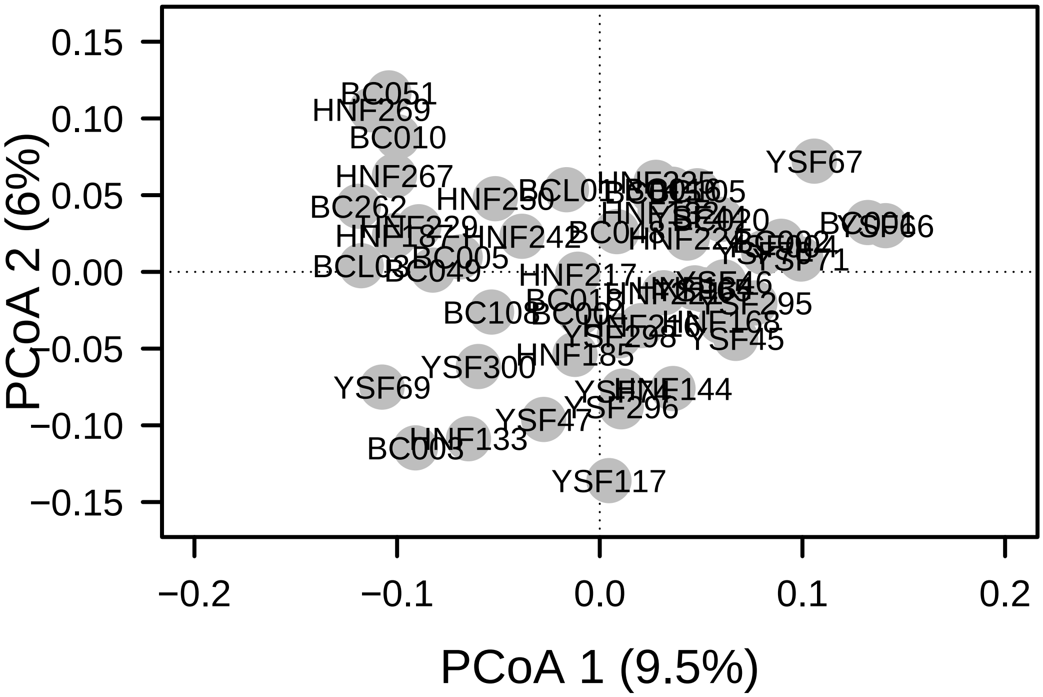
B. Visualizing Phylogenetic Beta Diversity

Now that we can generate a phylogenetically based community resemblance matrix, we are now ready to visualize phylogenetic diversity among samples using the same techniques that we used in the taxonomic β-diversity module. As an example, we will use ordination, but because we have a distance matrix (e.g., of UniFrac distances) any of the other β-diversity visualization techniques that operate on a distance matrix would also be appropriate (e.g., heatmaps and cluster analysis). Specifically, we will use the `cmdscale()` function to conduct a Principal Coordinates Analysis (PCoA) using the UniFrac distance matrix. Additionally, we will calculate the amount of explained variation for each phylogenetically informed PCoA axis.

```
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. Remember, to check the eigenvalues to determine your confidence in the data reduction approach.

```
# Define Plot Parameters
par(mar = c(5, 5, 1, 2) + 0.1)

# Initiate Plot
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)

# Add Axes
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)

# Add Points & Labels
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))
```



C. Hypothesis Testing

i. Categorical Designs The ponds that we sampled in southern Indiana were located in three distinct watersheds (BCSP, HNF, and YSF). For many organisms, watershed boundaries represent major dispersal barriers, which may influence the phylogenetic distribution of species. Here, we will test for this watershed affect using the Permutational Multivariate Analysis of Variance (PERMANOVA) test that we learned about in the β-diversity module.

```
# Define Environmental Category
watershed <- env$Location

# Run PERMANOVA with `adonis()` Function {vegan}
phylo.adonis <- adonis2(dist.uf ~ watershed, permutations = 999)

# We can compare to PERMANOVA results based on taxonomy
tax.adonis <- adonis2(
  vegdist(                             # create a distance matrix on
    decostand(comm, method = "log"),   # log-transformed relative abundances
    method = "bray") ~ watershed,      # using Bray-Curtis dissimilarity metric
  permutations = 999)
```

ii. Continuous Designs In the Spatial Diversity module of Quantitative Biodiversity, we demonstrated that there was substantial variation in environmental variables that are known to influence the structure and function of microbial communities. In the following section, we will revisit two methods that are used to test for relationships among multivariate environmental and biological data.

Before we do so, we need to define the environmental variables from the pond dataset so that we can create an environmental distance matrix. Some of the environmental variables are almost perfectly correlated with one another since they are derived from the same information collected by a suite of field probes. We will remove some of these variables (TDS, Salinity, and Cal_Volume) so as not to overfit the model.

```
# Define environmental variables
envs <- env[, 5:19]

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

# Create distance matrix for environmental variables
env.dist <- vegdist(scale(envs), method = "euclid")
```

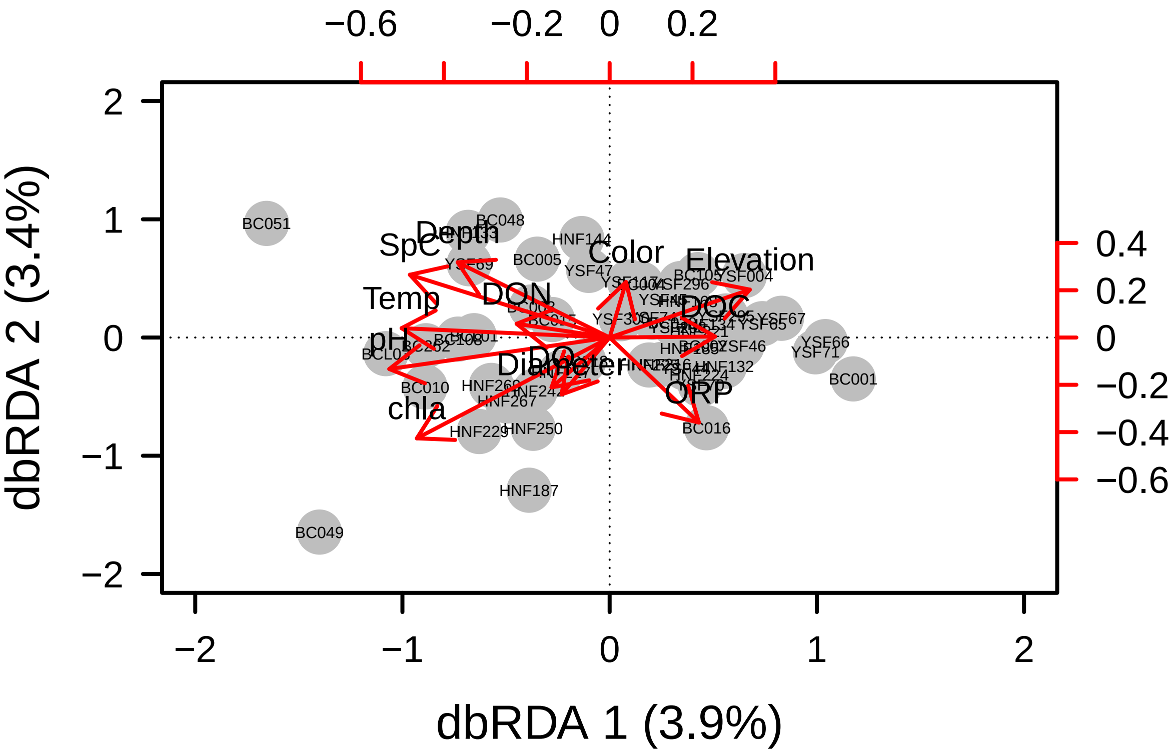
First, let us conduct a Mantel test to examine the potential correlation between UniFrac distance and environmental variation:

```
# Conduct Mantel Test (`vegan`)
mantel(dist.uf, env.dist)
```

Second, let us conduct a distance-based Redundancy Analysis (dbRDA). You will recall that this constrained ordination technique allows one to test for the effects of an explanatory matrix (e.g., environmental data) on a response matrix (e.g., phylogenetic distance matrix).

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

# Permutation tests: axes and environmental variables
anova(ponds.dbrda, by = "axis")
ponds.fit <- envfit(ponds.dbrda, envs, perm = 999)
ponds.fit

# Calculate explained variation
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
# Make dbRDA plot

# Define plot parameters
par(mar = c(5, 5, 4, 4) + 0.1)
```

```r
# Initiate plot
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)

# Add axes
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)

# Add points & labels
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)

# Add environmental vectors
vectors <- scores(ponds.dbrda, display = "bp")
#row.names(vectors) <- c("Temp", "DO", "chla", "DON")
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])))
```



15

6) SPATIAL PHYLOGENETIC COMMUNITY ECOLOGY

In the Spatial Diversity module, you learned that taxonomic diversity can be affected by geography for different reasons (e.g., dispersal and environmental conditions). For example, while population densities are often aggregated due to the spatial distribution of environmental variables, aggregated patterns of diversity can also arise from dispersal limitation and the stochastic distribution of individuals and species across space. You learned that factors such as these could lead to spatial autocorrelation in patterns of diversity, i.e., the tendency for near things to be more similar than distant things. Knowing whether your data is spatially autocorrelated is central to understanding the forces influencing diversity and to knowing the spatial scale at which comparisons should be made. You were also introduced to two primary spatial patterns of taxonomic diversity, i.e., the distance decay (DD) relationship and the species-area relationship (SAR).

Here, we will add a phylogenetic dimension to the study of spatial diversity. This section will combine the study of community ecology with one of the earliest scientific notions for why biological diversity varies across space. Specifically, evolutionary events like adaptive radiation, speciation, and extinction always have a geographic context (Lomolino et al. 2006). Consequently, spatial patterns of taxonomic diversity like the DD and SAR might reflect the influence of evolutionary processes and, hence, have a phylogenetic signal.

Here, you will learn to construct and analyze two primary spatial patterns of phylogenetic biodiversity, i.e., the phylogenetic distance decay (PDD) relationship and the phylogenetic diversity area relationship (PDAR).

A. Phylogenetic Distance-Decay (PDD)

Recall that the 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. 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 PDD relationship in a way that is similar to how we constructed the taxonomic DD. First, let's calculate pairwise geographic, taxonomic, and phylogenetic distances between sites of our Indiana ponds data set.

```r
# Geographic distances (kilometers) among ponds
long.lat <- as.matrix(cbind(env$long, env$lat))
coord.dist <- earth.dist(long.lat, dist = TRUE)

# Taxonomic similarity among ponds (Bray-Curits distance)
bray.curtis.dist <- 1 - vegdist(comm)

# Phylogenetic similarity among ponds (UniFrac)
unifrac.dist <- 1 - dist.uf

# Transform all distances into pairwise long format with the melt function from {reshape}:
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
```

```r
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,]
```

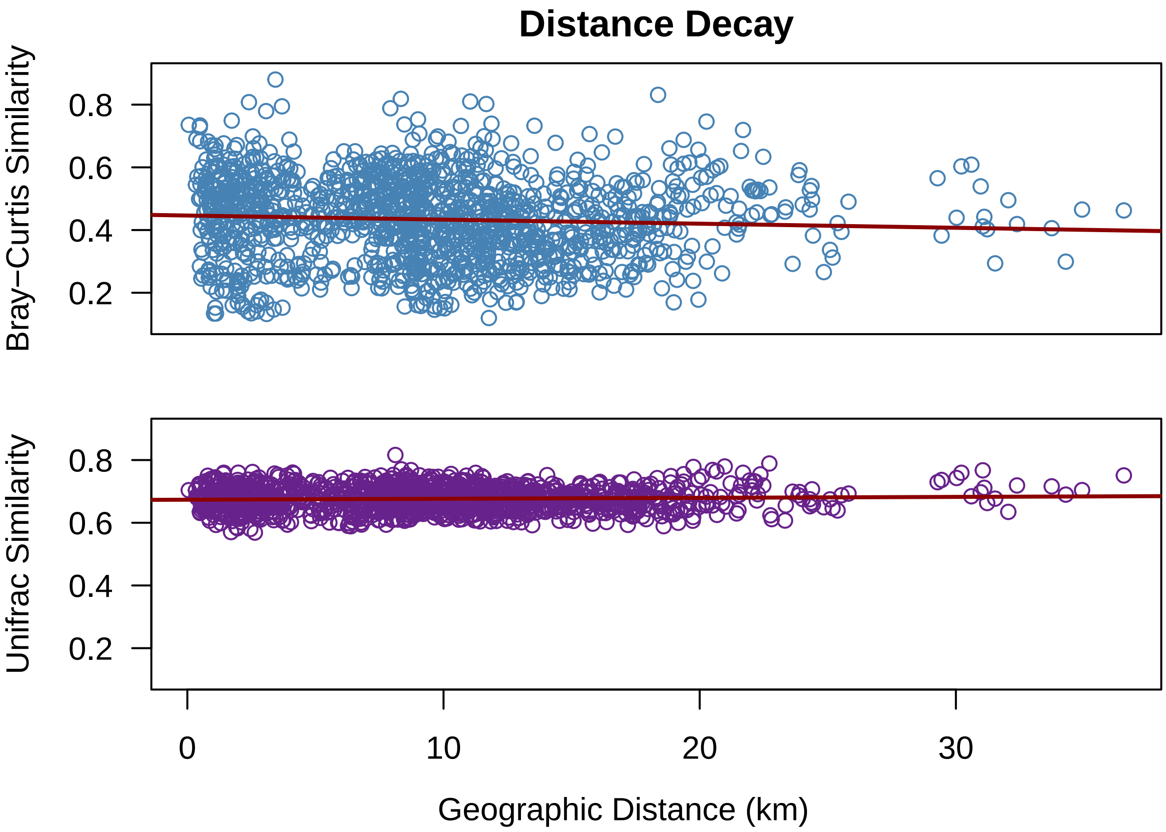
```
## Warning in type.convert.default(X[[i]], ...): 'as.is' should be specified by
## the caller; using TRUE

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

```
# Create a data frame from the lists of distances
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 us plot the DD relationships:

```
# Set initial plot parameters
par(mfrow=c(2, 1), mar = c(1, 5, 2, 1) + 0.1, oma = c(2, 0, 0, 0))

# Make plot for taxonomic DD
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")

# Regression for taxonomic DD
DD.reg.bc <- lm(df$bray.curtis ~ df$geo.dist)
summary(DD.reg.bc)
abline(DD.reg.bc , col = "red4", lwd = 2)

# New plot parameters
par(mar = c(2, 5, 1, 1) + 0.1)

# Make plot for phylogenetic DD
plot(df$geo.dist, df$unifrac, xlab = "", las = 1, ylim = c(0.1, 0.9),
     ylab = "Unifrac Similarity", col = "darkorchid4")

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

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

# Add x-axis label to plot
mtext("Geographic Distance (km)", side = 1, adj = 0.55,
      line = 0.5, outer = TRUE)
```



Finally, let's test whether the slopes for taxonomic and phylogenetic DD are significantly different from one another using the permutation-based `diffslope()` function in the `simba` package:

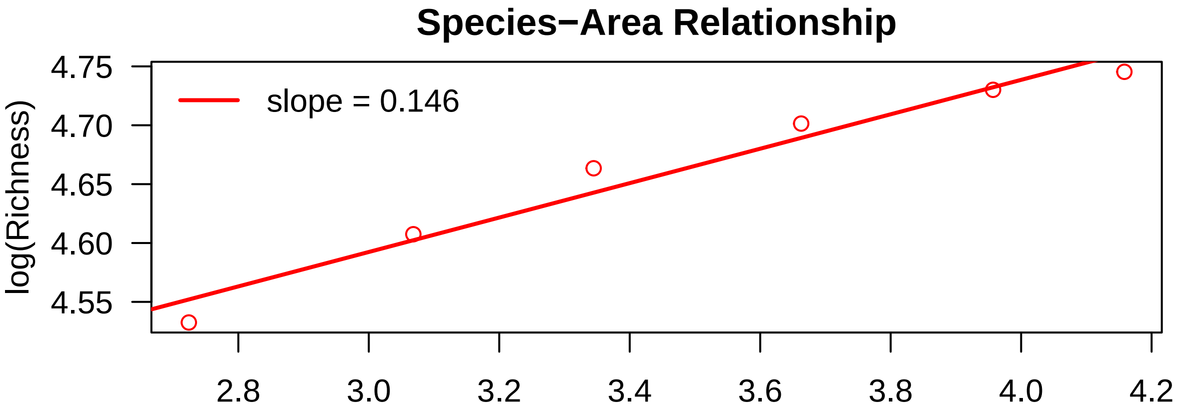
```
source("./bin/diffslope.R")
diffslope(df$geo.dist, df$unifrac, df$geo.dist, df$bray.curtis)
```

A. Phylogenetic diversity-area relationship (PDAR)

In the Spatial Diversity module you were introduced to one of ecology's most intensively studied patterns of biodiversity, i.e., the species-area relationship (SAR). The SAR reveals the rate at which species are discovered with increasing area and demonstrates the importance of considering spatial scale (i.e., extent, grain) and aggregation, the effects of random placement, and even the way in which area in the landscape is accumulated (e.g., nested design, accumulating contiguous or non-contiguous plots). You will recall that the SAR is typically constructed by accumulating area and in this way, the slope of the SAR cannot be negative. The slope of the SAR can however be 0, meaning that all species are found in all samples.

Recently, scientists have begun to explore the influence of evolutionary history on area relationships, specifically in regards to how phylogenetic diversity changes with area. Helmus and Ives (2012) developed methods to study how phylogenetic diversity changes with increasing area. They invoked macroevolutionary mechanisms such as *in situ* speciation (i.e., diversification within an area) and eco-evolutionary mechanisms such as phylogenetic repulsion (i.e., the tendency for closely related species to be found in similar habitats but not in the same area) to explain why phylogenetic diversity may increase with area. However, unlike the SAR, where newly observed species are simply added to the list of previously observed species, the PDAR can be negative because accumulating new taxa does not necessarily translate to accumulating greater evolutionary distance. In fact, phylogenetic repulsion would imply that sampling a new area could add species (increasing the SAR) that are phylogenetically closer than species in the same area (decreasing the PDAR).

In the following sections, we examine the microbial SAR and PDAR for our Indiana ponds data set. We will accumulate ponds at random across the landscape, as in Helmus and Ives (2012). Let's begin by constructing the SAR.



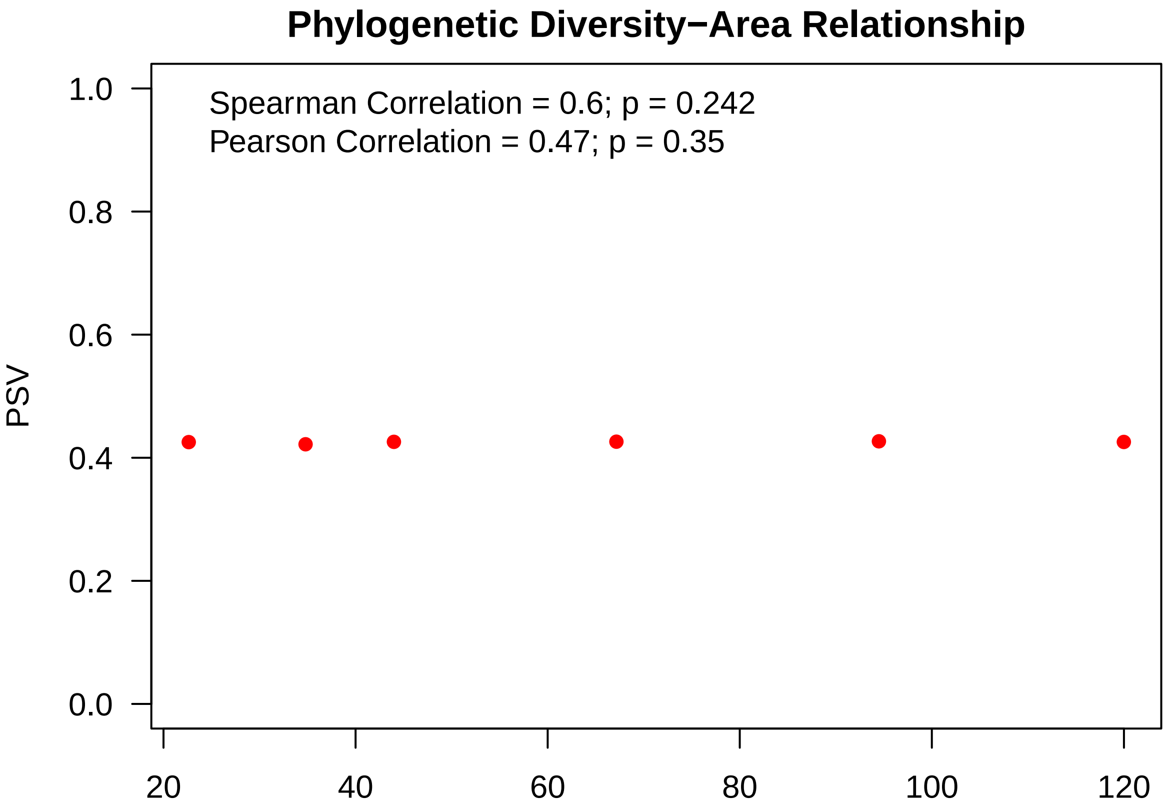
i. Constructing the PDAR Helmus and Ives (2012) used the phylogenetic species variability (PSV) metric to quantify phylogenetic diversity. PSV quantifies how phylogenetic relatedness decreases the variance of a hypothetical neutral trait shared by all species in a community. Type 'help(psv)' to learn more.

Let's construct the PDAR using the same approach to aggregating area. We will begin by writing a function to generate the PDAR.

```
PDAR <- function(comm, tree){

  # Create Objects to hold areas and diversity
  areas <- c()
  diversity <- c()

  # Create vector increasing number of plots by 2x
  num.plots <- c(2, 4, 8, 16, 32, 51)

  for (i in num.plots){
    # Create vectors to hold areas and diversity form iterations, used for means
    areas.iter <- c()
    diversity.iter <- c()

    # Iterate 10 times per sample size
    for (j in 1:10){
      # Sample w/o replacement
      pond.sample <- sample(51, replace = FALSE, size = i)

      # Create variable and vector to hold accumulating area and taxa
      area <- 0
      sites <- c()
      for (k in pond.sample) {       # Loop through each randomly drawn pond
        area <- area + pond.areas[k] # Aggregating area (roughly doubling)
        sites <- rbind(sites, comm[k, ])    # And sites
        }

      # Concatenate the area to areas.iter
      areas.iter <- c(areas.iter, area)
      # Calculate PSV or other phylogenetic alpha-diversity metric
      psv.vals <- psv(sites, tree, compute.var = FALSE)
      psv <- psv.vals$PSVs[1]
      diversity.iter <- c(diversity.iter, as.numeric(psv))
      }

    diversity <- c(diversity, mean(diversity.iter)) # Let Diversity be the Mean PSV
    areas <- c(areas, mean(areas.iter))             # Let areas be the Average Area
    print(c(i, mean(diversity.iter), mean(areas.iter))) # Print As We Go
    }
  # Return vectors of areas (x) and diversity (y)
  return(cbind(areas, diversity))
  }
```

ii. Evaluating the PDAR Let's examine the relationship between phylogenetic diversity and area using both Spearman's correlation coefficient (ρ) and Pearson's correlation coefficient (P). It is informative to use both because while ρ is computed on ranks and depicts monotonic relationships (the degree to which the relationship is continually increasing or decreasing), P is computed on the observed values and therefore depicts linear relationships.

```
# Calculate areas for ponds: find areas of all ponds
pond.areas <- as.vector(pi * (env$Diameter/2)^2)

# Compute the PDAR
pdar <- PDAR(comm, phy)
pdar <- as.data.frame(pdar)
pdar$areas <- sqrt(pdar$areas)

# Calculate Pearson's correlation coefficient
Pearson <- cor.test(pdar$areas, pdar$diversity, method = "pearson")
P <- round(Pearson$estimate, 2)
P.pval <- round(Pearson$p.value, 3)

# Calculate Spearman's correlation coefficient
Spearman <- cor.test(pdar$areas, pdar$diversity, method = "spearman")
rho <- round(Spearman$estimate, 2)
rho.pval <- round(Spearman$p.value, 3)

# Plot the PDAR
plot.new()
par(mfrow=c(1, 1), mar = c(1, 5, 2, 1) + 0.1, oma = c(2, 0, 0, 0))
plot(pdar[, 1], pdar[, 2], xlab = "Area", ylab = "PSV", ylim = c(0, 1),
     main = "Phylogenetic Diversity-Area Relationship",
     col = "red", pch = 16, las = 1)

legend("topleft", legend= c(paste("Spearman Correlation = ", rho, "; p = ", rho.pval, sep = ""),
                            paste("Pearson Correlation = ", P, "; p = ", P.pval, sep = "")),
                            bty = "n", col = "red")
```



8) REFERENCES

Adler, P. B., J. HilleRislambers, and J. M. Levine. 2007. A niche for neutrality. Ecology Letters 10:95-104.

Cadotte, M. W., and T. J. Davies. 2016. Phylogenies in Ecology: A Guide to Concepts and Methods, Princeton University Press.

Cavender-Bares, J., K. H. Kozak, P. V. A. Fine, and S. W. Kembel. 2009. The merging of community ecology and phylogenetic biology. Ecology Letters 12:693-715.

Chao, A., C.-H. Chiu, and L. Jost. 2014. Unifying species diversity, phylogenetic diversity, functional diversity, and related similarity and differentiation measures through Hill numbers. Annual Review of Ecology, Evolution, and Systematics 45:297-324.

Darwin, C. 1859, On the origin of species by means of natural selection. London, John Murray.

Diez, J. M., J. J. Sullivan, P. E. Hulme, G. Edwards, and R. P. Duncan. 2008. Darwin's naturalization conundrum: dissecting taxonomic patterns of species invasions. Ecology Letters 11:674-681.

Faith, D. P. 1992. Conservation evaluation and phylogenetic diversity. Biological Conservation 61:1-10.

Gotelli, N. J. 2000. Null model analysis of species co-occurrence patterns. Ecology 81:2606-2621.

Graham, C. H., and P. V. A. Fine. 2008. Phylogenetic beta diversity: linking ecological and evolutionary processes across space in time. Ecology Letters 11:1265-1277.

Helmus, M. R., and A. R. Ives. 2012. Phylogenetic diversity–area curves. Ecology 93:S31-S43.

HilleRisLambers, J., P. B. Adler, W. S. Harpole, J. M. Levine, and M. M. Mayfield. 2012. Rethinking community assembly through the lens of coexistence theory. Annual Review of Ecology, Evolution, and Systematics 43:227-248.

Ives, A. R., and M. R. Helmus. 2010. Phylogenetic metrics of community similarity. The American Naturalist 176:E128-E142.

Leibold, M. A., and M. A. McPeek. 2006. Coexistence of the niche and neutral perspectives in community ecology. Ecology 87:1399-1410.

Lomolino, M. V., Riddle, B. R., and J. H. Brown. 2006. Biogeography. 3rd Edition. Sunderland, MA. Sinauer Associates.

Mayfield, M. M., and J. M. Levine. 2010. Opposing effects of competitive exclusion on the phylogenetic structure of communities. Ecology Letters 13:1085-1093.

Miklós, I., and J. Podani. 2004. Randomization of presence-absence matrices: comments and new algorithms. Ecology 85:86-92.

Morlon, H., D. W. Schwilk, J. A. Bryant, P. A. Marquet, A. G. Rebelo, C. Tauss, B. J. M. Bohannan et al. 2011. Spatial patterns of phylogenetic diversity. Ecology Letters 14:141-149.

Mouquet, N., V. Devictor, C. N. Meynard, F. Munoz, L.-F. Bersier, J. Chave, P. Couteron et al. 2012. Ecophylogenetics: advances and perspectives. Biological Reviews 87:769-785.

Pavoine, S. 2016. A guide through a family of phylogenetic dissimilarity measures among sites. Oikos 125:1719-1732.

Swenson, N. G. 2014, Functional and Phylogenetic Ecology in R, Springer.

Tucker, C. M., M. W. Cadotte, S. B. Carvalho, T. J. Davies, S. Ferrier, S. A. Fritz, R. Grenyer et al. 2016. A guide to phylogenetic metrics for conservation, community ecology and macroecology. Biological Reviews:n/a-n/a.