Handout: Among Site (Beta) Diversity

Z620: Quantitative Biodiversity, Indiana University

January 27, 2016

OVERVIEW

In this exercise, we move beyond the investigation of within-site α-diversity. We will explore β-diversity, which is defined as the diversity that occurs among sites. To analyze how diversity varies across sites, we must examine the compositional similarity of assemblages that vary in space or time.

After completing this exercise you will know how to:

1. formally quantify β-diversity
2. visualize β-diversity with heatmaps, cluster analysis, and ordination
3. test hypotheses about β-diversity using multivariate statistics

1) SETUP

A. Retrieve and Set Your Working Directory

```
rm(list = ls())
getwd()
setwd("~/GitHub/QB-2017/Week3-Beta/")
#/QuantitativeBiodiversity
```

B. Load Packages

We will be using the `vegan` package again; let's load it now.

```
#install.packages("vegan")
require("vegan")
```

2) LOADING DATA

A. Description of Data Set

To date, we have analyzed biodiversity datasets for freshwater zooplankton, tropical trees, and soil bacteria. In this exercise, we introduce a new dataset containing information on stream fish assemblages from the Doubs river, which runs near the France-Switzerland boarder in the Jura Mountains. The data set (`doubs`) includes fish abundances, environmental variables, and spatial coordinates for 30 sites. The data set has previously been used to demonstrate that fish communities can be good indicators of ecological zones in rivers and streams.

Let's load the `ade` package, which contains the `doubs` data set.

```
#install.packages("ade4")
require("ade4")
data(doubs)
```

B. Introduction to a R object class: Lists

While working in R this semester, we have learned about vectors, matrices, and data frames. Here we introduce another object class in R: a **list**. In R, a list is an object that contains a collection of other objects of similar or different types. To access objects within a list, place a dollar sign (`$`) between the name of the list and the object you want to call (e.g., `list$object`).

C. Exploring the Doubs River Dataset

We can use the `str()` function to describe the attributes of `doubs`, which is a list. Because this dataset is somewhat complex, we can pass the "max.level = 1" argument to minimize the `str()` output. Use the the dollar sign (`$`) after the list name (`doubs`) to access objects within the list (e.g., `env`). In this example, we introduce a new data structure that is similar in ways to the site-by-species matrix. For each site where we have species information (i.e., presence-absence or abundance), we also have environmental data. We will call this the **site-by-environment** matrix, and we will use it later to explain patterns of β-diversity. To access the site-by-environment matrix in this case, type `doubs$env` in the R console. Last, recall that you can use `help()` to learn more about a dataset contained in a package.

```
str(doubs, max.level = 1)
head(doubs$env)
```

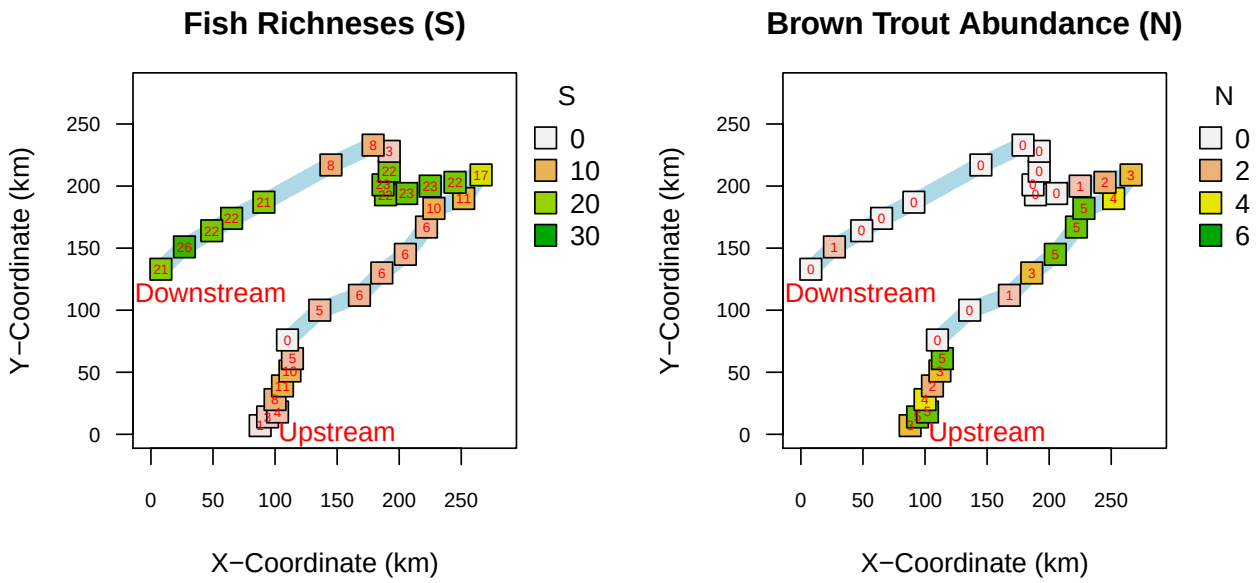
Question 1: Describe some of the attributes of the `doubs` dataset.

a. How many objects are in `doubs`?
b. What types of data structures are contained in `doubs`?
c. What are the units of nitrate ("nit") in the stream water?
d. How many fish species are there in the `doubs` data set?

Answer 1a:
Answer 1b:
Answer 1c:
Answer 1d:

D. Visualizing the Doubs River Dataset

There is a wealth of information in the `doubs` dataset that can be used to address various issues related to β-diversity. For example, we might use the environmental or spatial data to develop or test a hypothesis. Below we have generated two plots from the `doubs` fish data. The first plot shows fish richness at each site in the stream. The second plot shows the abundance of a particular fish species, Brown Trout (*Salmo trutta*), at each site in the stream.



Question 2: Answer the following questions based on the spatial patterns of richness (i.e., α-diversity) and Brown Trout (*Salmo trutta*) abundance in the Doubs River.

a. How does fish richness vary along the sampled reach of the Doubs River?
b. How does Brown Trout (*Salmo trutta*) abundance vary along the sampled reach of the Doubs River?
c. What do these patterns say about the limitations of using richness when examining patterns of biodiversity?

Answer 2a:
Answer 2b:
Answer 2c:

3) QUANTIFYING BETA-DIVERSITY

There are various ways to quantify β-diversity. Metrics of β-diversity can be broadly grouped into two categories: *variation* (non-directional) and *turnover* (directional). One of the simplest metrics is **Whittaker's β-Diversity**, developed by Robert Whittaker (1960). This classic index quantifies how many times more diverse the regional species pool (i.e., the richness across all sites, or γ-diversity) is than the average richness at each site within the region (i.e., $\bar{\alpha}$). Whittaker proposed that the relationship between local (α) and regional (γ) diversity is multiplicative. Thus, the equation for Whittaker's β-Diversity is: $\beta_W = \frac{\gamma}{\bar{\alpha}}$.

We can write β_W as a function in R as follows:

```
beta.w <- function(site.by.species = ""){
  SbyS.pa <- decostand(site.by.species, method = "pa") # convert to presence-absence
  S <- ncol(SbyS.pa[,which(colSums(SbyS.pa) > 0)])     # number of species in the region
  a.bar <- mean(specnumber(SbyS.pa))                   # average richness at each site
  b.w <- round(S/a.bar, 3)                             # round to 3 decimal places
  return(b.w)
}
```

Whittaker also used his metric to measure turnover between two sites (often called Whittaker's species turnover). Subtracting 1 scales β_w to produce values ranging from 0 (minimum β-diversity) to 1 (maximum β-diversity). Thus, this metric becomes $\beta_W = \frac{\gamma}{\bar{\alpha}} - 1$, where γ is now the combined richness of the two sites being compared.

We can modify our function to also calculate pairwise β_W for turnover as follows:

```
beta.w <- function(site.by.species = "", sitenum1 = "", sitenum2 = "", pairwise = FALSE){

  # ONLY if we specify pairwise as TRUE, do this:
    if (pairwise == TRUE){

      # As a check, let's print an error if we do not provide needed arguments
      if (sitenum1 == "" | sitenum2 == "") {
        print("Error: please specify sites to compare")
        return(NA)}

      # If our function made it this far, let's calculate pairwise beta diversity
      site1 = site.by.species[sitenum1,]                      # Select site 1
      site2 = site.by.species[sitenum2,]                      # Select site 2
      site1 = subset(site1, select = site1 > 0)               # Removes absences
      site2 = subset(site2, select = site2 > 0)               # Removes absences
      gamma = union(colnames(site1), colnames(site2))         # Gamma species pool
      s     = length(gamma)                                   # Gamma richness
      a.bar = mean(c(specnumber(site1), specnumber(site2)))   # Mean sample richness
      b.w   = round(s/a.bar - 1, 3)
      return(b.w)
    }

  # OTHERWISE pairwise defaults to FALSE, so do this, like before:
  else{
      SbyS.pa <- decostand(site.by.species, method = "pa") # convert to presence-absence
      S <- ncol(SbyS.pa[,which(colSums(SbyS.pa) > 0)])     # how many species are in the region
      a.bar <- mean(specnumber(SbyS.pa))                   # average richness at each site
      b.w <- round(S/a.bar, 3)
      return(b.w)
  }
}
```

We now have a robust function called `beta.w()` that can handle two ways of calculating beta diversity. In essence, this is what is going on behind the scenes with many of the functions from `vegan` that we will encounter.

Question 3: Using the `beta.w()` function above, answer the following questions:

a. Describe how local richness and turnover contribute to regional fish diversity in the Doubs.
b. What is the β-diversity for fish assemblages sampled from site 1 and site 2 of the `doubs` data set?
c. Based on the formula for β_w, what does this value represent?

Answer 3a:
Answer 3b:
Answer 3c:

See APPENDIX for more information on calculating β-diversity and on partitioning γ-diversity into its α and β components.

Often, we often want to compare the diversity of more than just a pair of samples. For example, it would be nice to be able to compare fish assemblages for *all* pairs of sites in the Doubs River. In the following sections, we will estimate β-diversity for multiple samples. During this process, you will learn how to generate similarity and dissimilarity matrices for different datasets that will be needed for visualizing and quantifying β-diversity.

A. Introducing the Resemblance Matrix

In order to quantify β-diversity for more than two samples, we need to introduce another primary ecological data structure: the **Resemblance Matrix**. In the context of biodiversity, a resemblance matrix is a data structure that calculates the pairwise **similarity** or **dissimilarity** for all samples in a site-by-species matrix. The resemblance matrix can be generated from a site-by-species matrix containing incidence (presence-absence) data or abundance data. In the sections below, we describe some of the similarity and dissimilarity metrics that are commonly used for constructing a resemblance matrix. Throughout this handout, we adopt the notations of Table 7.2 in Legendre & Legendre (2012); this book can be electronically accessed via the IU library (see the course syllabus [http://goo.gl/y4oK7c]).

B. Incidence-Based Measures of Similarity and Dissimilarity

When you are working with presence-absence data, you can use the following metrics for generating a similarity or dissimilarity matrix.

Index	Equation	Properties
Jaccard	$S_7 = \frac{a}{a+b+c}$	Compares the number of shared species to the number of species in the combined assemblages placing more emphasis on taxa not shared between sites
Sørensen	$S_8 = \frac{2a}{(2a+b+c)}$	Compares the number of shared species to the mean number of species in a single assemblage placing more emphasis on similarity of samples owing to shared species

In the above table, a = the number of species shared between assemblages, b = the number of unique species in the first assemblage, and c = the number of unique species in the second assemblage.

See APPENDIX for additional information on incidence-based similarity metrics

C. Abundance-Based Measures of Similarity and Dissimilarity

When you are working with abundance data, you can use the following metrics for generating a similarity or dissimilarity matrix.

Index	Equation	Properties
Bray-Curtis Dissimilarity	$D_{14} = \frac{\sum_{j=1}^{p} \lvert y_{1j} - y_{2j} \rvert}{\sum_{j=1}^{p} (y_{1j} + y_{2j})}$	A quantitative version of the Sørensen index. Commonly used measure of similarity. Also known as the *percentage difference*.
Morisita-Horn	$S_{MH} = \frac{2 \sum_{j=1}^{p} y_{1j} \cdot y_{2j}}{\left(\sum_{j=1}^{p} y_{1j}^2 + \sum_{j=1}^{p} y_{2j}^2 \right)}$	A measure of *compositional overlap*. Uses squared differences in relative abundance and thus is influenced by abundant species. Resistant to undersampling.

In the above table y_{1j} is the abundance of each species $(1:p)$ in site 1 and y_{2j} is the abundance of each species $(1:p)$ in site 2. As with incidence-based measures, there are many other options for calculating similarity or dissimilarity between communities, including Mean Character Difference, Canberra, Coefficient of Divergence, and Gower.

See APPENDIX for a cautionary note on other measures of distance

Question 4: Answer the following questions about measures of (dis)similarity:

a. What are the key differences between incidence- and abundance-based metrics?
b. When might you use one instead of the other?

Answer 4a:
Answer 4b:

D. Constructing the Resemblance Matrix

Conveniently, `vegan` includes many of the similarity metrics used to construct a resemblance matrix. These metrics can be calculated using the `vegdist()` function. Let's use `vegdist` to create a resemblance matrix for the fish assemblages in the Doubs River. Before that, we'll need to remove site 8 from `doubs` because for some reason it has no observations.

```
fish <- doubs$fish
fish <- fish[-8, ] # Remove site 8 from data

# Calculate Jaccard
fish.dj <- vegdist(fish, method = "jaccard", binary = TRUE)

# Calculate Bray-Curtis
fish.db <- vegdist(fish, method = "bray")

# Calculate Sørensen
fish.ds <- vegdist(fish, method = "bray", binary = TRUE)
```

Now that we've created a resemblance matrix, it would be nice to visualize the fish assemblages of the Doubs River. As a start, we can print the Bray-Curtis-based resemblance matrix in the console:

```
fish.db
```

From this, you will see a large diagonal matrix. Typically, resemblance matrices just show the upper or lower triangle of values. This is because the two triangles have the same information. Also notice that the diagonal (i.e., the resemblance of each site to itself) is missing from the resemblance matrix by default. However, you can generate a square resemblance matrix with the following command:

```
fish.db <- vegdist(fish, method = "bray", upper = TRUE, diag = TRUE)
```

Question 5: Does the resemblance matrix (`fish.db`) represent similarity or dissimilarity? What information in the resemblance matrix led you to arrive at your answer?

Answer 5:

4) VISUALIZING BETA-DIVERSITY

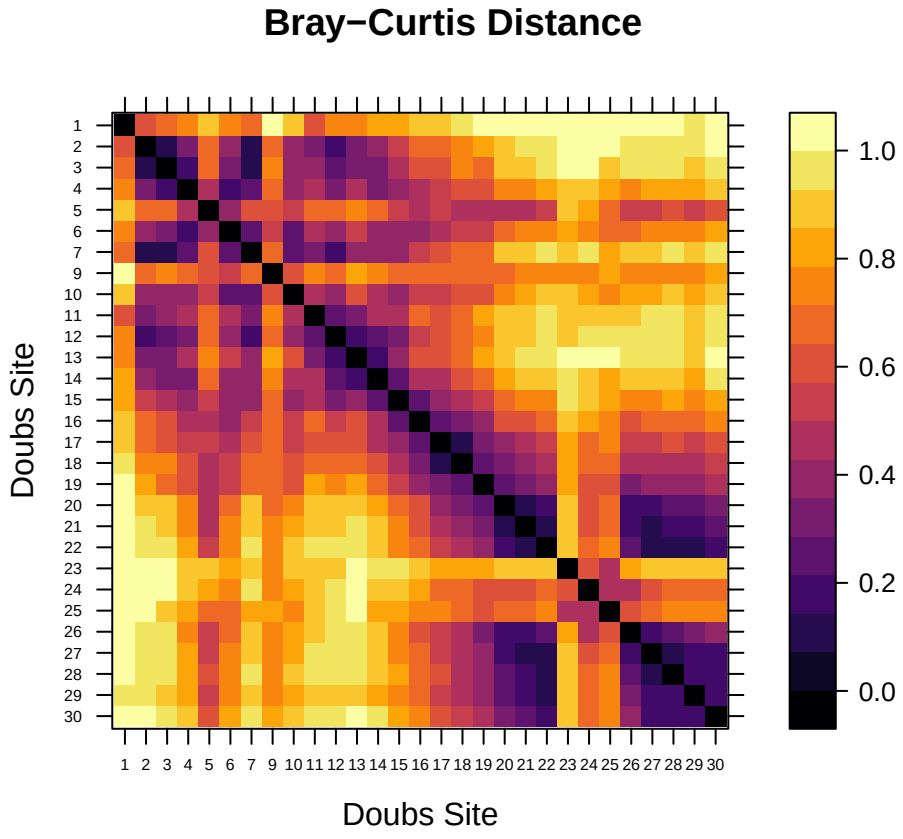
A. Heatmaps

One way to visualize β-diversity is to plot the resemblance matrix using a **heatmap**. A heatmap is a two-dimensional, color representation of a data matrix. Here we are going to use the `levelplot()` function in the `lattice` package of R. This function will allow us to make a basic heatmap of our resemblance matrix. First, there are a few things we need to do:

1. Define a color palette. R includes many predefined color palettes; however, we are going to use another color palette. The package viridis includes four color palettes, each of which are perceptually uniform and both colorblind- and black-and-white-friendly.
2. Ensure that our resemblance matrix is plotted correctly. In particular, we need specify the order in which we want our sites to be plotted in the heatmap.

```
# Get a Color Palette
# install.packages("viridis")
require("viridis")
```

```
# Define Order of Sites
order <- rev(attr(fish.db, "Labels"))

# Plot Heatmap
levelplot(as.matrix(fish.db)[, order], aspect = "iso", col.regions = inferno,
          xlab = "Doubs Site", ylab = "Doubs Site", scales = list(cex = 0.5),
          main = "Bray-Curtis Distance")
```

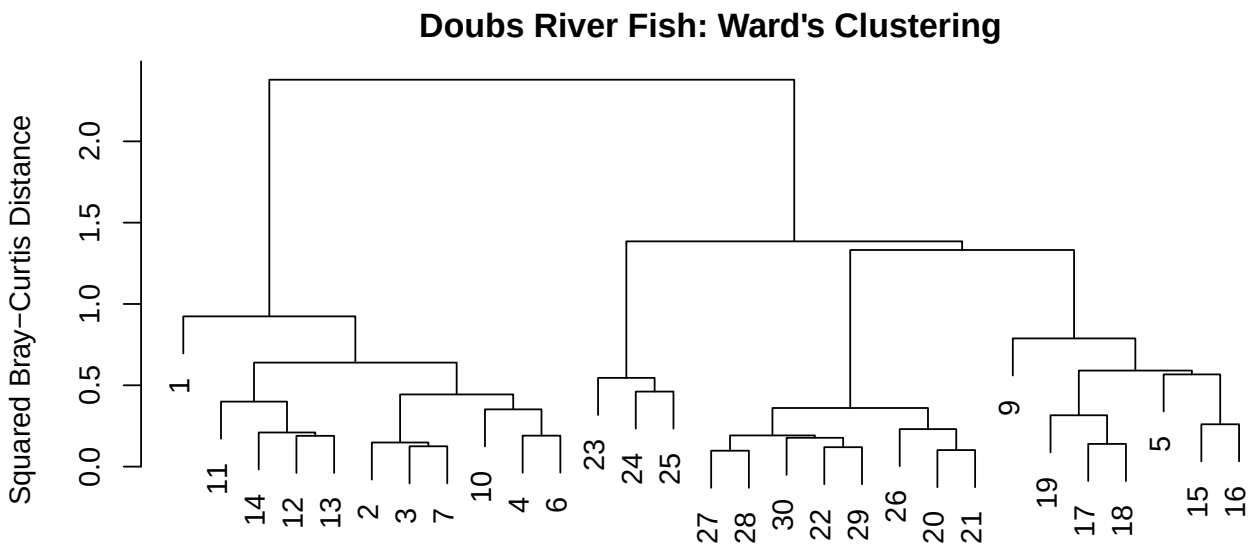


B. Cluster Analysis

Another common way to visualize β-diversity is through cluster analysis. Cluster analysis is an exploratory technique that assigns objects to groups based on their similarity to one another. In this exercise, we will use hierarchical clustering, specifically **Ward's Clustering**. Ward's Clustering (a.k.a., Ward's minimum variance method) is an agglomerative (i.e., pairs are grouped together based on their similarity) clustering technique based on the linear model criterion of least squares. The method minimizes within-cluster sums-of-squared distances between sites. However, there are numerous methods for clustering (e.g., Single Linkage, UPGMA, UPGMC), which can influence the conclusions that you draw from your analysis. See chapter 8 of Legendre

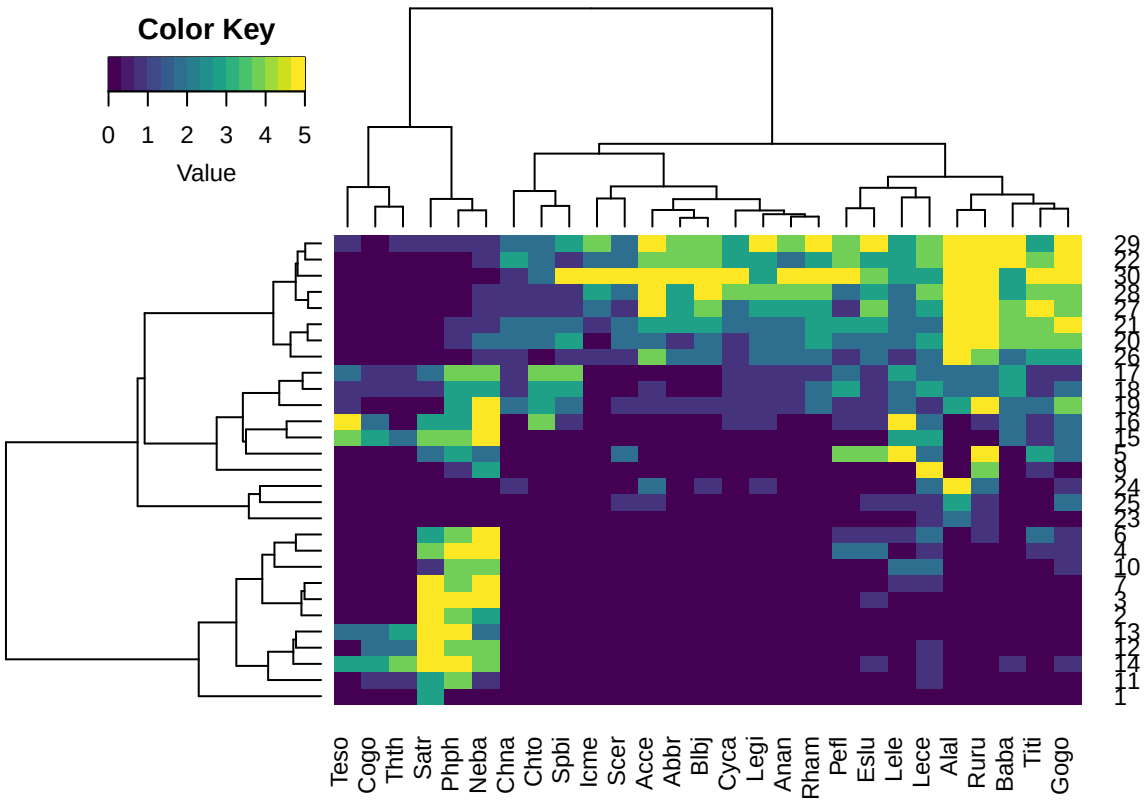
and Legendre (2012) for more information on the various clustering methods that can be used.

```
# Perform Cluster Analysis
fish.ward <- hclust(fish.db, method = "ward.D2")

# Plot Cluster
par(mar = c(1, 5, 2, 2) + 0.1)
plot(fish.ward, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```



Another clustering tool that aids in visualization for exploratory purposes is `heatmap.2()`, which is a function in the `gplots` package. This tool generates a cluster diagram that allows one to examine the abundance of different fish species in different sites.

```
# If we just need one function from a package, we do not have to load the whole package
# Instead, just type the package name followed by two colons before the function name
gplots::heatmap.2(as.matrix(fish), distfun = function(x) vegdist(x, method = "bray"),
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          col = viridis, trace = "none", density.info = "none")
```



Question 6: Based on cluster analyses and the introductory plots that we generated after loading the data, develop a hypothesis about the distribution of fish assemblages in the Doubs River?

Answer 6:

C. Ordination

The primary aim of ordination is to represent multiple objects in a reduced number of orthogonal (i.e., independent) axes. Often, the objects we want to visualize are samples in "species space", but with a simple transpose, one can look at species plotted in "sample space". The first axis of an ordination explains the most variation in the data set, followed by the second axis, then the third, and so on, where the total number of axes is less than or equal to the number of objects. Ordination plots are particularly useful for visualizing the similarity among objects. For example, in the context of β diversity, sites that are closer in ordination space have species assemblages that are more similar to one another than sites that are further apart in ordination space.

There are various ordination techniques that can be applied to multivariate biodiversity data. Common methods include: Principal Components Analysis (PCA), Correspondence Analysis (CA), Principal Coordinates Analysis (PCoA), Factor Analysis (FA), and Nonmetric Multidimensional Scaling (NMDS). When choosing an ordination technique, careful consideration should be given to the data type (continuous vs. categorical), model assumptions (i.e., which distances are preserved), and the underlying mathematical procedures that are involved.

i. An Overview of Principal Coordinates Analysis (PCoA)

In this exercise, we focus on Principal Coordinates Analysis (PCoA), which is sometimes referred to as metric multidimensional scaling (MDS). PCoA starts with creating a matrix, $\boldsymbol{A}$, which is a transformed and centered version of distance matrix, $\boldsymbol{D}$. Because these steps preserve all distances, PCoA is a flexible ordination

technique that allows us to use virtually any distance metric (e.g., Jaccard, Bray-Curtis, Gower, Euclidean, etc.). The dimensionality of $\boldsymbol{A}$ is then reduced by determining each eigenvector, $\boldsymbol{u}_i$, and eigenvalue, $\boldsymbol{\lambda}_i$, that solve the following equation: $\boldsymbol{A}\boldsymbol{u}_i = \boldsymbol{\lambda}_i\boldsymbol{u}_i$. Finally, each eigenvector, $\boldsymbol{u}_i$, is scaled to length $\sqrt{\boldsymbol{\lambda}_i}$ to obtain the principal coordinates. In essence, we are representing distances between objects in $\boldsymbol{D}$ as distances in fewer dimensions, calculated from each objects coordinates along the PCoA axes.

To conduct a PCoA, we will use the `cmdscale()` function from the `stats` package, which performs PCoA. The input for this function is our resemblance matrix (Bray-Curtis distances for the `doubs` dataset). In addition, we are going to set $k = 3$ (number of dimensions we want returned) and eig = TRUE (which saves the eigenvalues).

```
fish.pcoa <- cmdscale(fish.db, eig = TRUE, k = 3)
```

ii. Interpreting PCoA Output

The `cmdscale` function produces a list of output. The first item *points* contains the coordinates for each site in each reduced dimension. The second item *eig* contains the eigenvalues. The last three items pertain to other options of the analysis that we will not cover here.

First, we want to examine the eigenvalues. The eigenvalues are the scaling factors that allowed us to reduce the dimensionality of a data set. The eigenvalues can also be used to calculate the amount of variation that is explained by each orthogonal axis. To do this, we divide the eigenvalue of each axis by the sum of all eigenvalues. In the following chunk of R code, we quantify the percent variation in the `doubs` data set that is explained by the first three axes of the PCoA.

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

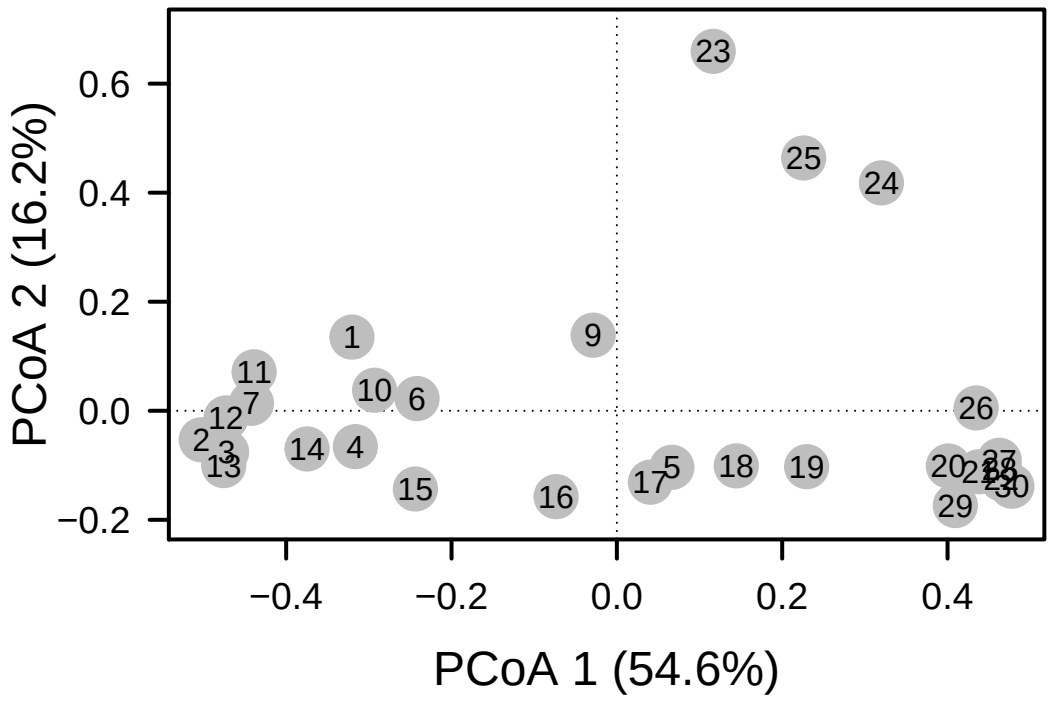
See APPENDIX for other ways to asses how well PCoA explains variation in your data

iii. Creating a PCoA Ordination Plot

Having evaluated the PCoA output, now we will create an ordination plot. We will plot all of the fish assemblages of the Doubs River for the first two PCoA axes.

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

# Initiate Plot
plot(fish.pcoa$points[ ,1], fish.pcoa$points[ ,2], ylim = c(-0.2, 0.7),
     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(fish.pcoa$points[ ,1], fish.pcoa$points[ ,2],
```

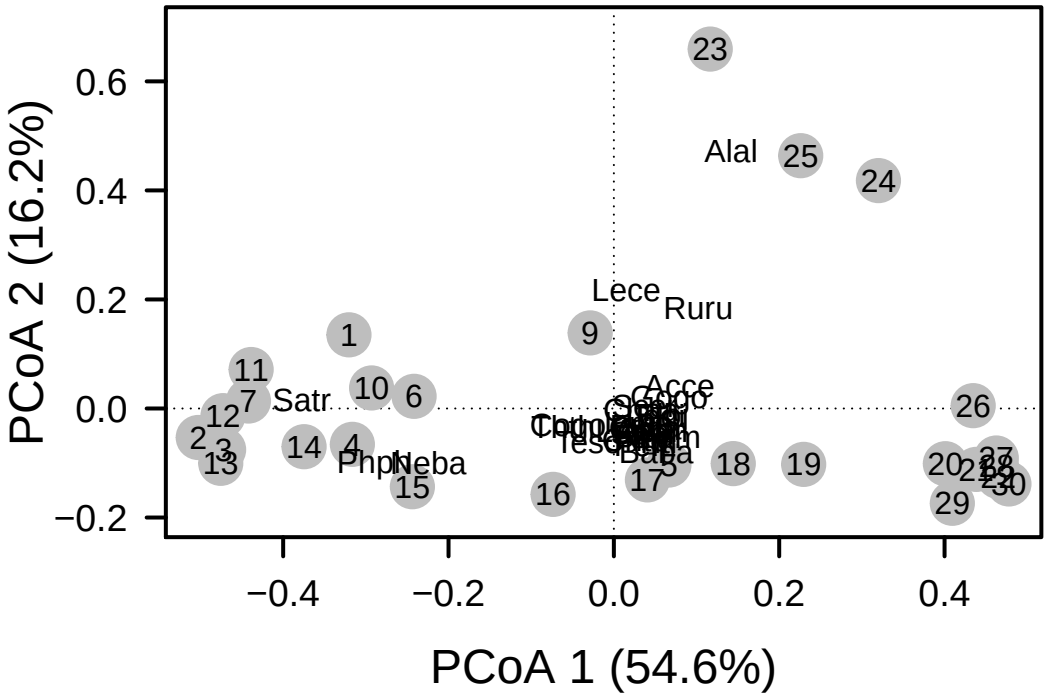
```
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(fish.pcoa$points[ ,1], fish.pcoa$points[ ,2],
     labels = row.names(fish.pcoa$points))
```



iv. Identifying and Visualizing Influential Species in PCoA

Basic ordination plots allow us to see how samples separate from one another. A logical follow-up is to ask what features of the dataset are driving the observed divergence among points. In the Doubs River example, sites are separating along the PCoA axes owing to variation in the abundance of different fish species. We can get a better sense of "who" is contributing to this trend by plotting explanatory vectors (i.e., species coordinates) in ordination space. We can obtain this information using the `add.spec.scores()` function in the `BiodiversityR` package. These coordinates reflect the strength and direction that each species has on the ordination of the different sites.

```
require("BiodiversityR")

# First we calculate the relative abundances of each species at each site
fishREL <- fish
  for(i in 1:nrow(fish)){
    fishREL[i, ] = fish[i, ] / sum(fish[i, ])
    }

# Now, we use this information to calculate and add species scores
fish.pcoa <- add.spec.scores(fish.pcoa,fishREL,method = "pcoa.scores")
text(fish.pcoa$cproj[ ,1], fish.pcoa$cproj[ ,2],
     labels = row.names(fish.pcoa$cproj), col = "black")
```



A more quantitative way of identifying influential species involves determining the correlation coefficient of each species along the PCoA axes. To do this, we will use the `add.spec.scores` function again. Then we can identify a correlation-coefficient cutoff (e.g. $r = 0.70$) to pull out important species. Finally, we will use the `envfit()` function from the `vegan` package, to conduct a permutation test on these correlations.

```
spe.corr <- add.spec.scores(fish.pcoa, fishREL, method = "cor.scores")$cproj
corrcut  <- 0.7       # user defined cutoff
imp.spp  <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]

# Permutation Test for Species Abundances Across Axes
fit <- envfit(fish.pcoa, fishREL, perm = 999)
```

Question 7: Address the following questions about the ordination results of the `doubs` data set:

a. Generate a hypothesis about the grouping of sites in the Doubs River based on fish community composition.
b. Generate a hypothesis about which fish species are potential indicators of river quality.
c. Do the different approaches described in the ordination section agree or disagree? Explain.

Answer 7a:
Answer 7b:
Answer 7c:

5) HYPOTHESIS TESTING

The visualization tools that we just learned about (i.e., heatmaps, cluster analysis, and ordination) are powerful for exploratory analysis and for *generating* hypotheses. In this section we introduce some methods that are better suited for *testing* hypotheses and predictions related to β-diversity.

A. Multivariate Procedures for Categorical Designs

i. PERMANOVA

PERMANOVA stands for permutational multivariate analysis of variance (Anderson 2001). It is a multivariate analog to univariate ANOVA and has fewer restrictions than parametric multivariate analysis of variance (MANOVA). As the name suggests, it uses permutation tests to evaluate the statistical significance and amount of variation explained by categorical predictor variables. We will implement PERMANOVA with the `adonis()` function in the `vegan` package.

To run `adonis`, you first need a factor vector or matrix that specifies your treatments and replicates. You can import a more complex design vector/matrix, but we will simply create one from scratch.

Earlier work done in the Doubs River suggested that the river has four distinct regions of habitat quality: the first region (sites 1-14) of "high quality"; the second (sites 15 - 19) and fourth (sites 26 - 30) of "moderate quality"; and the third (sites 20 - 25) of "low quality".

Let's test the hypothesis that fish community composition varies with river quality.

```
# Create "Factors"" vector
quality <- c(rep("HQ", 13), rep("MQ", 5), rep("LQ", 6), rep("MQ", 5))

# Run PERMANOVA with adonis function
adonis(fish ~ quality, method = "bray", permutations = 999)
```

ii. Species-Site Group Associations

Often we wish to know how individual species relate to groups of sites. For example, the presence of certain invertebrate taxa often indicates stream water quality. To determine whether any fish species in our dataset can be used as indicators of habitat quality, we can calculate an Indicator Value (IndVal) for each species in each habitat group, and evaluate its significance using a permutation test. Significant IndVal scores close to 1 suggest that the species is a strong indicator of that site group.

```
#install.packages("indicspecies")
require(indicspecies)
indval <- multipatt(fish, cluster = quality, func = "IndVal.g", control = how(nperm=999))
summary(indval)
```

Alternatively, we may ask about the habitat preferences of each species. In this case, we can calculate the phi coefficient of association (r_ϕ). This is the Pearson correlation between a vector of sites indicating species presence (1) or absence (0), and a vector of sites indicating in (1) or out (0) of the focal group. This approach can also be extended to abundance data by replacing the 1s of the species vector with species abundances. Values of the phi coefficient range from -1 (suggesting strong avoidance of a group) to 0 (suggesting no preference) to $+1$ (suggesting strong preference for a group).

```
fish.rel <- decostand(fish, method = "total")
phi <- multipatt(fish.rel, cluster = quality, func = "r.g", control = how(nperm=999))
summary(phi)
```

To learn more about species-site group associations, see Dufrêne and Legendre (1997) and De Cáceres and Legendre (2009), and references within them.

Question 8: Based on the PERMANOVA, IndVal, and phi coefficient analyses, what did you learn about the relationship between habitat quality and the fish species composition?

Answer 8:

B. Multivariate Procedures for Continuous Designs

i. Mantel Test

A Mantel test is essentially a multivariate correlation analysis that looks for monotonic correlations between elements of two matrices. It produces an r value that is analogous to the Pearson's correlation coefficient. In addition, it produces a p-value that is derived from the deviation of observed correlation to that of correlations derived from randomizations of the two matrices.

In the following section, we will perform a Mantel test using the `mantel()` function in `vegan`. This requires that we first have two distance matrices to compare. Here, we will compare the Bray-Curtis distance matrix we created earlier from the site-by-species matrix with the site-by-environment matrix (`doubs$env`). After creating the distance matrices, we will test the hypothesis that fish assemblages are correlated with stream environmental variables.

```
# Define Matrices
fish.dist <- vegdist(doubs$fish[-8, ], method = "bray")
env.dist <- vegdist(scale(doubs$env[-8,]),method = "euclid")

#Mantel Test
mantel(fish.dist,env.dist)
```

Question 9: What do the results from our Mantel test suggest about fish diversity and stream environmental conditions? How might this relate to your hypothesis about stream quality influencing fish communities?

Answer 9:

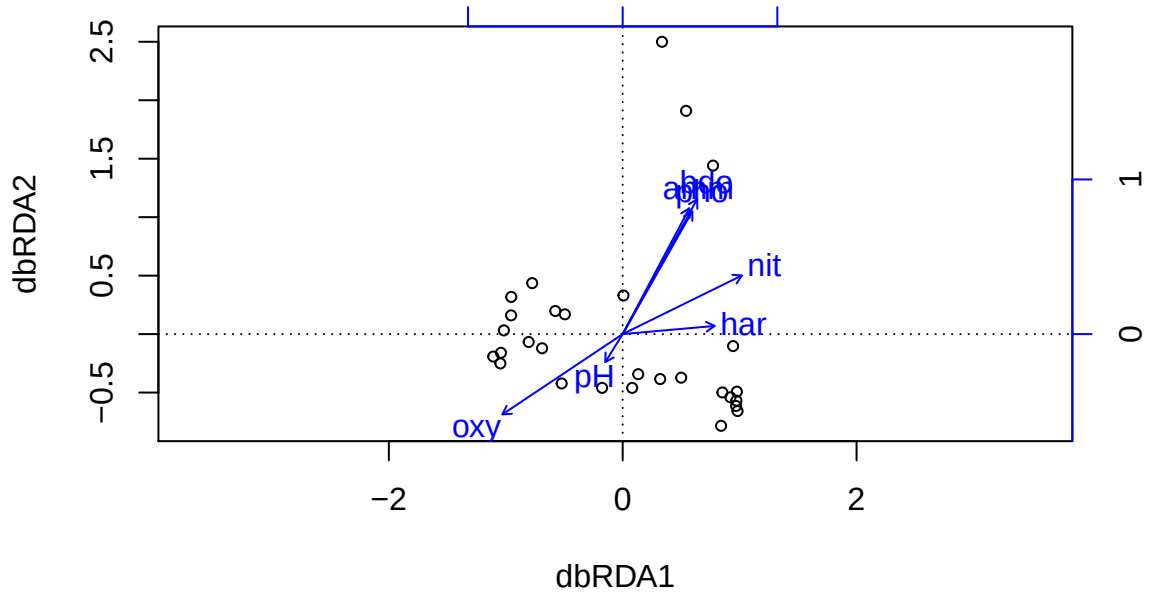
ii. Constrained Ordination

Another way we can test hypotheses with continuous data is to use **constrained ordination**, which is sometimes referred to as canonical ordination. Constrained ordination explores the relationships between two matrices: an **explanatory matrix** and a **response matrix**. Canonical correspondence analysis (CCA) and redundancy analysis (RDA) are two common types of constrained ordination.

See APPENDIX for more information on these constrained ordination families, and for an example of how to perform a CCA

Here, we will use environmental data to conduct a redundancy analysis on the fish assemblages of the Doubs River. We will start by creating an explanatory matrix that contains water chemistry data. Note that RDA preserves Euclidean distances (and thus requires an appropriate transformation, e.g., Hellinger, see APPENDIX). Analogous to PCoA, a method of constrained ordination called distance-based Redundancy Analysis (dbRDA) has been implemented that operates on a resemblance matrix, allowing the use of alternative metrics, such as Bray-Curtis distance. dbRDA uses all the principal coordinates of a PCoA performed on the resemblance matrix as the response variable in constrained ordination. We will use the `dbrda()` function from the `vegan` package to perform it. Then we will use permutation tests to evaluate the significance of our model. Finally, we will test the influence of each environmental variable on the constrained axes.

```
# Define environmental matrix
env.chem <- as.matrix(doubs$env[-8 , 5:11])

# Perform dbRDA
doubs.dbrda <- dbrda(fish.db ~ ., as.data.frame(env.chem))
ordiplot(doubs.dbrda)
```



It looks like many of our environmental variables are highly correlated with one another (try `psych::corr.test(env.chem)` to check for pairwise correlations) and could lead to model overfitting. Let's use model selection to let R add/remove explanatory variables for us until it finds the model with the lowest AIC value (indicating a good model that uses fewer parameters).

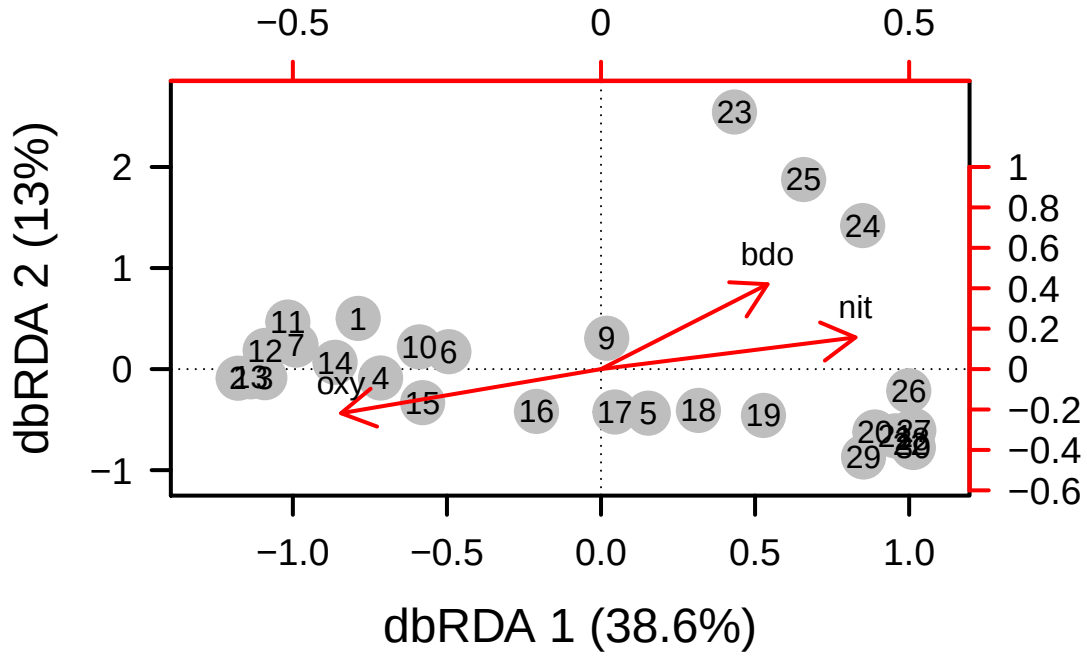
```
# First, we will model only the intercept
doubs.dbrda.mod0 <- dbrda(fish.db ~ 1, as.data.frame(env.chem))

# Note there are no vectors here (we didn't constrain anything)
# so the axes suggest this is a simple MDS (i.e., PCoA)
ordiplot(doubs.dbrda.mod0)
```

```
# Next, we will model the full model, with all explanatory variables
doubs.dbrda.mod1 <- dbrda(fish.db ~ ., as.data.frame(env.chem))

# Now we step through all combinations of explanatory variables in our model
# The function returns the model with the lowest AIC value
doubs.dbrda <- ordiR2step(doubs.dbrda.mod0, doubs.dbrda.mod1, perm.max = 200)

# Lets look at the model that was selected
doubs.dbrda$call
doubs.dbrda$anova
ordiplot(doubs.dbrda)
```

```
# Permutation tests to evaluate significance
permutest(doubs.dbrda, permutations = 999)
envfit(doubs.dbrda, env.chem[,c(4,6,7)], perm = 999)

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

Now, let's plot the ordination for the selected model.

```r
# Define Plot Parameters
par(mar = c(5, 5, 4, 4) + 0.1)

# Initiate Plot
plot(scores(doubs.dbrda, display = "wa"), xlim = c(-1.3, 1.1), ylim = c(-1.1, 2.7),
     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(doubs.dbrda, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(scores(doubs.dbrda, display = "wa"),
     labels = row.names(scores(doubs.dbrda, display = "wa")))

# Add Environmental Vectors
vectors <- scores(doubs.dbrda, display = "bp")
#row.names(vectors) <- rownames(vectors)
arrows(0, 0, vectors[,1], vectors[, 2],
       lwd = 2, lty = 1, length = 0.2, col = "red")
text(vectors[,1], vectors[, 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 10: Based on the constrained ordination, what are the environmental variables (or groups of correlated variables) that seem to be contributing to variation in fish community structure?

Answer 10:

iii. Variation Partitioning

We have shown that spatial variation in fish community structure in the Doubs River can be explained by environmental variables. However, the environmental variables also vary across space. A major question in the study of β-diversity is whether variation in community structure is more strongly driven by changes in underlying environmental variables (after controlling for spatial variation in those variables), by changes in spatial position alone (after controlling for environmental variation), or by spatially structured environmental variation. A common approach to answering these types of questions is to use variation partitioning, which determines the amount of variation in a response variable that can be explained by two or more explanatory matrices, jointly and independently. For more detailed explanation of the application of variation partition in ecology, see Borcard et al. (1992), Legendre et al. (2005), and Peres-Neto et al. (2006).

Here, we will perform variation partitioning using two constrained ordinations (by space and by environment) and two partial constrained ordinations (by space controlling for environment and by environment controlling for space).

Above, we already performed dbRDA using the environmental data as an explanatory matrix.

```
# Remember, our environmental model uses oxy, bdo, and nit and has R2 of 0.53
doubs.dbrda$anova

# Let's create a matrix model for our environmental data
env.mod <- model.matrix(~ oxy + bdo + nit, as.data.frame(env.chem))[,-1]
```

Next, we will create the spatial model. The spatial model is a bit more involved, but just follow along for now. We will perform Principal Coordinates of Neighbor Matrices (Borcard and Legendre 2002), a special form of distance based Moran's Eigenvector Maps (dbMEMs, Dray et al. 2006). In general, we will create a spatial distance matrix between our sites, truncate sites that are far away from each other (hence the "Neighbor Matrices"), then create spatial eigenvectors from this truncated matrix (hence the "Principal Coordinates"). Our goal is to represent "space" as some combination of these eigenvectors, so that we can identify spatial structure in species abundances on broad, intermediate, and fine spatial scales. The resulting eigenvectors model space from broad scale to fine scale with increasing eigenvector number.

```
# First, we will weight each site by its relative abundance
rs <- rowSums(fish)/sum(fish)

# Next, we will perform PCNM
doubs.pcnmw <- pcnm(dist(doubs$xy[-8,]), w = rs, dist.ret = T)

# PCNM can return negative eigenvalues, but only the
# eigenvectors associated with the positive eigenvalues are meaningful
doubs.pcnmw$values > 0
```

This analysis returns many eigenvalues, but some of them are redundant. So we will again perform model selection (forward and backward) to determine which eigenvalues create the most informative model with the fewest parameters (Blanchet et al. 2008).

```
doubs.space <- as.data.frame(scores(doubs.pcnmw))
doubs.pcnm.mod0 <- dbrda(fish.db ~ 1, doubs.space)
doubs.pcnm.mod1 <- dbrda(fish.db ~ ., doubs.space)
step.pcnm <- ordiR2step(doubs.pcnm.mod0, doubs.pcnm.mod1, perm.max = 200)
```

```
# Because this is another dbRDA, we could visualize the biplot
# showing how each vector explains variation across sites
plot(step.pcnm)
```

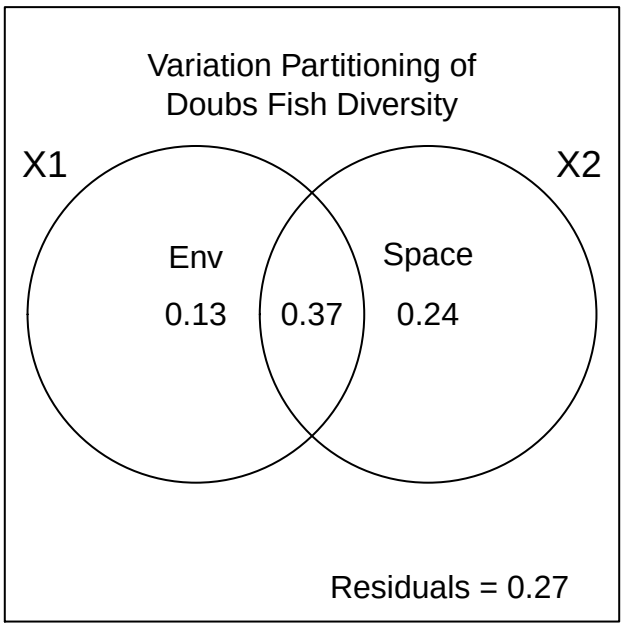
```
# The object `step.pcnm` now contains the selected model.
step.pcnm$anova

# We can now construct a spatial model using only the selected PCNM axes.
space.mod <- model.matrix(~ PCNM2 + PCNM3 + PCNM5 + PCNM1 +
                            PCNM13 + PCNM16 + PCNM6, doubs.space)[,-1]
```

Now we wish to determine the independent and joint explanatory power of the environmental and spatial datasets using variation partitioning. To do so, we will perform **partial constrained ordination**, which requires a second explanatory matrix. We will control for the effects of one explanatory matrix on the other using `Condition()` in the model formula.

```
# First conduct constrained ordinations
doubs.total.env <- dbrda(fish.db ~ env.mod)
doubs.total.space <- dbrda(fish.db ~ space.mod)

# Next construct partial constrained ordinations
doubs.env.cond.space <- dbrda(fish.db ~ env.mod + Condition(space.mod))
doubs.space.cond.env <- dbrda(fish.db ~ space.mod + Condition(env.mod))

# Next test for significance of the dbRDA fractions.
permutest(doubs.env.cond.space, permutations = 999)
permutest(doubs.space.cond.env, permutations = 999)
permutest(doubs.total.env, permutations = 999)
permutest(doubs.total.space, permutations = 999)
```

Now, we need to calculate the fractions of variation explained by space alone, environment alone, both space and environment, and neither space nor environment. Here, we will use the `varpart()` function from `vegan`. A more detailed explanation of how the constrained and partial constrained ordinations are being used to calculate each component of explained variation can be found in the APPENDIX.

```
# Using the built-in varpart() function
doubs.varpart <- varpart(fish.db, env.mod, space.mod)
doubs.varpart

par(mar = c(2,2,2,2))
plot(doubs.varpart)
text(1, 0.25, "Space")
text(0, 0.25, "Env")
mtext("Variation Partitioning of\nDoubs Fish Diversity", side = 3, line = -3)
```



APPENDIX

Diversity partitioning and beta diversity

The history of β-diversity, and how it relates α to γ diversity has enjoyed considerable debate over the past half century. While Whittaker (1960, 1972) advocated a multiplicative approach, others have suggested additive partitioning: $\gamma = \alpha + \beta$ (see Lande (1996) and Veech et al. (2002) for reviews). Additive partitioning changes the interpretation of β from a measure of *how many times* more diverse the region is than local sites, to a measure of *how many more species* exist in the regional pool than in local sites. There has been a rejuvenation of research on diversity partition over the past decade or so, particularly with the embrace of "numbers equivalents" or "Hill numbers" reflecting M. O. Hill's (1973) demonstration of the mathematical relationships between richness, Shannon's index, and Simpson's index. More discussion of the usefulness of the numbers equivalents approach can be found in Jost (2006, 2007) and more recent debates that have emerged since their publication.

Notes on incidence-based similarity:

Jacard and Sørensen are perhaps the two most commonly used incidence-based measures of similarity. Note: Sørensen's coefficient of similarity (S_8) is related to pairwise $\beta_W = S/\bar{a} - 1$ by $\beta_W = 1 - S_8$. Others include Ochiai, Kulczynski-Cody, and Lennon, which can be found in Table 6.1 of Magurran & McGill (2011). The differences in these measures include how means are calculated (Sørensen = harmonic mean, Ochiai = geometric mean, and Kulczynski-Cody = arithmetic mean), and how unique species are dealt with if only one sample has unique species (Lennon). Also, it is important to note that these metrics calculate similarity, but can (and in many cases should) be converted to dissimilarity (D). In `vegan`, dissimilarities (D) are usually returned from functions instead of similarities (S). The conversion between similarity and dissimilarity is calculated as `D = 1 - S`.

A Cautionary Note on Other Measures of Distance:

There are other distance measures that you are likely to encounter because they are widely used. It is important to note that some of these should *not* be used for abundance-based data in biodiversity analyses.

This is because they lead to the **Species Abundance Paradox**, which is a phenomenon that occurs when the distance between two sites that have no species in common is smaller than the distance between sites with shared species. In particular, the paradox described above arises when Euclidean Distance and Manhattan Distance are used:

Euclidean Distance: $D_1 = \sqrt{\sum_{j=1}^{p} (y_{1j} - y_{2j})^2}$

Manhattan Distance: $D_7 = \sum_{j=1}^{p} |y_{ij} - y_{2j}|$

Methods exist to transform species abundance data prior to implementing metrics like Euclidean Distance and Manhattan Distance (e.g., Chord transformation and Chi-Square transformation). It is recommended that you read more about these distance metrics if you are interested in implementing them to address your research questions. One highly recommended transformation that is appropriate for multivariate analyses is the Hellinger transformation:

Hellinger Distance: $D_{17} = \sqrt{\sum_{j=1}^{p} \left[\sqrt{\frac{y_{1j}}{y_{1+}}} - \sqrt{\frac{y_{2j}}{y_{2+}}}\right]^2}$

The Hellinger distance can be calculated on your dataset and used in multivariate techniques that preserve the Euclidean distance (e.g., PCA and RDA), or equivalently, in methods that accept a distance matrix (e.g., PCoA and dbRDA):

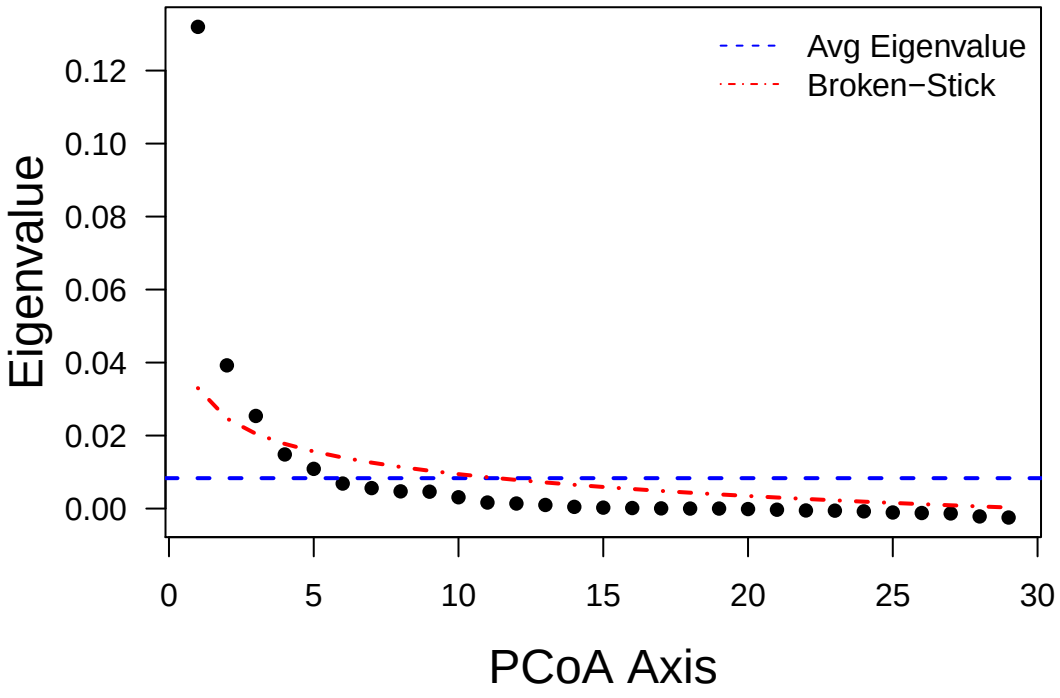
```
fish <- doubs$fish[-8,]
fish.hel <- decostand(fish, method = "hellinger") # hellinger transform
fish.dh <- vegdist(fish.hel, method = "euclidean") # distances
```

Other ways to assess how well PCoA explains variation in your data

Another way to evaluate our analysis is to assess whether or not the first few PCoA axes capture a disproportionately large amount of the total explained variation. First, the eigenvalues associated with the first few axes should be larger than the average of all the eigenvalues (*Kaiser-Guttman criterion*). Second, we can compare the eigenvalues associated with the first few axes to the expectations of the *broken-stick model*, which we introduced in the α-diversity exercise when discussing species abundance distributions (SAD). In the current context, the broken stick model assumes that the total sum of eigenvalues decreases sequentially with ordered PCoA axes.

We will evaluate these two criteria with the following plots:

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

# Plot Eigenvalues
plot(fish.pcoa$eig, xlab = "PCoA Axis", ylab = "Eigenvalue",
     las = 1, cex.lab = 1.5, pch = 16)

# Add Expectation based on Kaiser-Guttman criterion and Broken Stick Model
abline(h = mean(fish.pcoa$eig), lty = 2, lwd = 2, col = "blue")
b.stick <- bstick(29, sum(fish.pcoa$eig))
lines(1:29, b.stick, type = "l", lty = 4, lwd = 2, col = "red")

# Add Legend
legend("topright", legend = c("Avg Eigenvalue", "Broken-Stick"),
       lty = c(2, 4), bty = "n", col = c("blue", "red"))
```



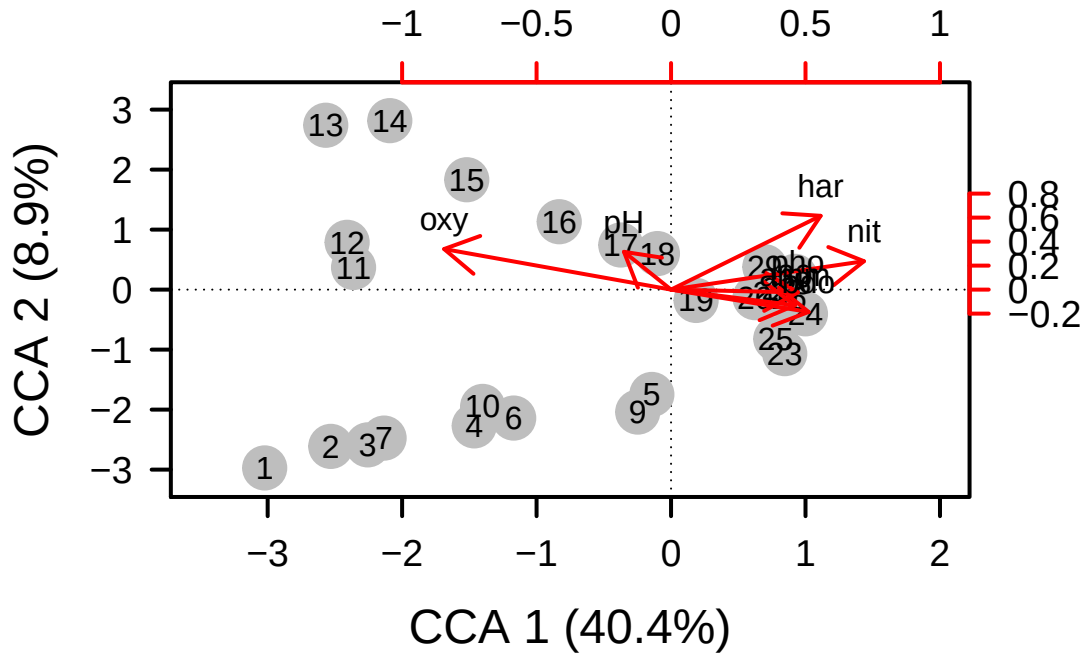
More on constrained ordination

Constrained ordination techniques are based on the linear model framework and thus can be used to formally test hypotheses. Constrained ordination works by first conducting multivariate multiple linear regression followed either by correspondence analysis (CA) with CCA or Principal Components Analysis (PCA) with RDA, while using the matrix of fitted values to obtain a constrained ordination. Thus, CCA preserves chi-squared distances (analogous to CA), while RDA preserves Euclidean distances (analogous to PCA). A permutation test can then be used to test for overall significance.

Example of how to perform a CCA

```
# Define Environmental Matrix
env.chem <- as.matrix(doubs$env[-8 , 5:11])

# Conduct CCA
doubs.cca <- vegan::cca(fish ~ env.chem)

# Permutation Tests
anova(doubs.cca, by = "axis")
cca.fit <- envfit(doubs.cca, env.chem, perm = 999)
cca.fit

# Calculate Explained Variation
cca.explainvar1 <- round(doubs.cca$CCA$eig[1] /
                         sum(c(doubs.cca$CCA$eig, doubs.cca$CA$eig)), 3) * 100
cca.explainvar2 <- round(doubs.cca$CCA$eig[2] /
                         sum(c(doubs.cca$CCA$eig, doubs.cca$CA$eig)), 3) * 100

# Define Plot Parameters
par(mar = c(5, 5, 4, 4) + 0.1)
```

```r
# Initiate Plot
plot(scores(doubs.cca, display = "wa"), xlim = c(-3.5, 2), ylim = c(-3.2, 3.2),
     xlab = paste("CCA 1 (", cca.explainvar1, "%)", sep = ""),
     ylab = paste("CCA 2 (", cca.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(doubs.cca, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(scores(doubs.cca, display = "wa"),
     labels = row.names(scores(doubs.cca, display = "wa")))

# Add Environmental Vectors
vectors <- scores(doubs.cca, display = "bp")
row.names(vectors) <- c("pH", "har", "pho", "nit", "amm", "oxy", "bdo")
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])))
```



Step-by-step demonstration of how to do variation partitioning

```
# First, calculate the total variation in the dataset
doubs.ca <- dbrda(fish.db ~ 1)
total.var <- sum(doubs.ca$CA$eig)

# Calculate fractions of explained variation:
step.1 <- sum(doubs.total.env$CCA$eig) * 100 / total.var
step.2 <- sum(doubs.total.space$CCA$eig) * 100 / total.var
step.3 <- sum(doubs.env.cond.space$CCA$eig) * 100 / total.var
step.4 <- sum(doubs.space.cond.env$CCA$eig) * 100 / total.var

# Total explained variation
# These should be the same because they are non-overlapping fractions

sum(step.1 + step.4)
```

```
## [1] 82.91657
```

```
sum(step.2 + step.3)
```

```
## [1] 82.91657
```

```
# Partition the variation explained
# [a] Pure Environment
pure.env <- step.3
# [b] Spatially structured env
env.space <- step.1 - step.3
space.env <- step.2 - step.4
# [c] Pure Space
pure.space <- step.4
# [d] Unexplained
resid.var <- 100 - sum(step.1 + step.4)

sum(c(pure.env, env.space, pure.space, resid.var))
```

```
## [1] 100
```

```
# Note that the individual fractions line up with the fractions from varpart()
# however, varpart() applies a correction the partial fractions
```