

# Week 3 Exercise: Between Site Diversity

*Z620: Quantitative Biodiversity, Indiana University*

*January 30, 2015*

## Overview

In this exercise, we will explore between site biodiversity. We will start by using the methods we learned for measuring biodiversity within a single site. We will then expand this to learn how to properly measure and compare biodiversity between sites. We will also learn various graphics and hypothesis testing techniques.

## Initial Setup

### Retrieve and Set Your Working Directory

```
rm(list=ls())
getwd()
setwd("~/GitHub/QuantitativeBiodiversity/Assignments/Week3")
```

### Install Packages

Ecologists and other data scientists have developed numerous packages for various tasks in the R environment. This week we will be using a few different R packages. You can use the ‘help’ function to learn more about each package. Let’s go ahead and load a few of the packages that we will be using.

```
require("vegan") || install.packages("vegan"); require("vegan")
require("ade4") || install.packages("ade4"); require("ade4")
require("BiodiversityR") || install.packages("BiodiversityR"); require("BiodiversityR")
```

At this point you should be familiar with **vegan**. The other packages (**ade4** and **BiodiversityR**) contain some of the data sets that we will use this week.

### User Defined Functions

It helps to start scripts with any user defined functions that you know that you will need. For example, this week we will be using the **sem()** function we wrote in week 1. Let’s go ahead and define this function.

```
sem <- function(x){
  sd(x, na.rm = TRUE)/sqrt(length(na.omit(x)))
}
```

### Import Data

We will again be using the BCI data set (in **vegan**) for part of this week’s exercises. In addition, we will be using environmental data for the BCI plots. The **BiodiversityR** package has some BCI environmental data (BCI.env). There is additional data available for BCI and your Week3 folder has soil data for each plot (bci.soil.txt). We will go ahead and import all of these data now.

```
data(BCI) # BCI Tree Abundance (vegan)
data(BCI.env) # BCI Environmental Data (BiodiversityR)
BCI.soil <- read.delim("./bci.soil.txt") # BCI Soil Data
```

## A. Between Site Diversity

Last week we learned about alpha diversity. We calculated things like species richness, evenness, and diversity. However, as you will soon see, these tools are not always adequate when comparing communities.

Let's go back to the BCI dataset and calculate the tree richness across all sites.

```
bci.S <- specnumber(BCI)
mean(bci.S)
```

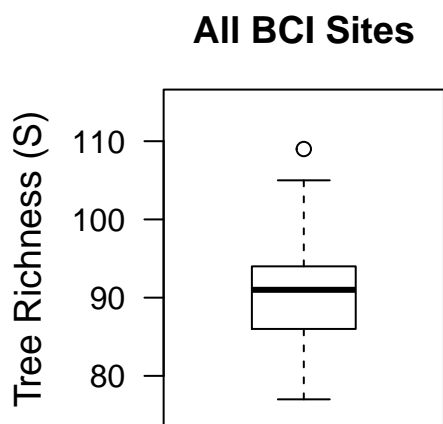
```
## [1] 90.78
```

```
sem(bci.S)
```

```
## [1] 0.9906872
```

We can visualize these data using a box and whisker plot. We are going to start our plot by changing the plotting environment using the `par()` function. Adjust the values in `mar` to see how it changes the plot. You can learn more about the diverse plotting options with the help files (`help(par)`)

```
par(mar = c(0.5, 4.5, 3, 0.5) + 0.1)
boxplot(bci.S, ylim = c(75, 115), las = 1, cex.lab = 1.2,
        main = "All BCI Sites", ylab = "Tree Richness (S)")
```



However, this doesn't allow us to compare between site variation. One way we can visualize this is by separating sites. Since there aren't any factors that separate BCI sites into groups, we can just plot them based on location (based on XY-coordinates). To make it easy to visualize differences, we can color code sites based on tree richness using a heatmap.

In R there are a few tools that allow you to make color palettes. Here we are going to use `terrain.colors` but you can learn about others by looking at the `Pallettes` help file: `help(Pallettes)`.

Let's start by making a color palette for both Richness (S) and Abundance (N)

```
BCI.S.color <- rev(terrain.colors(151))
BCI.N.color <- rev(terrain.colors(41))
```

Notice that we reversed ('rev()') the sequence of colors in the palette, this was done to make the colors more intuitive. You will have to make these decisions yourself when you make future figure. Also, notice that we had to define the number of colors in each palette. Here we used 151 for S and 41 for N. You can adjust these values if needed, but you will see soon why we used odd numbers.

## BCI Tree Richness

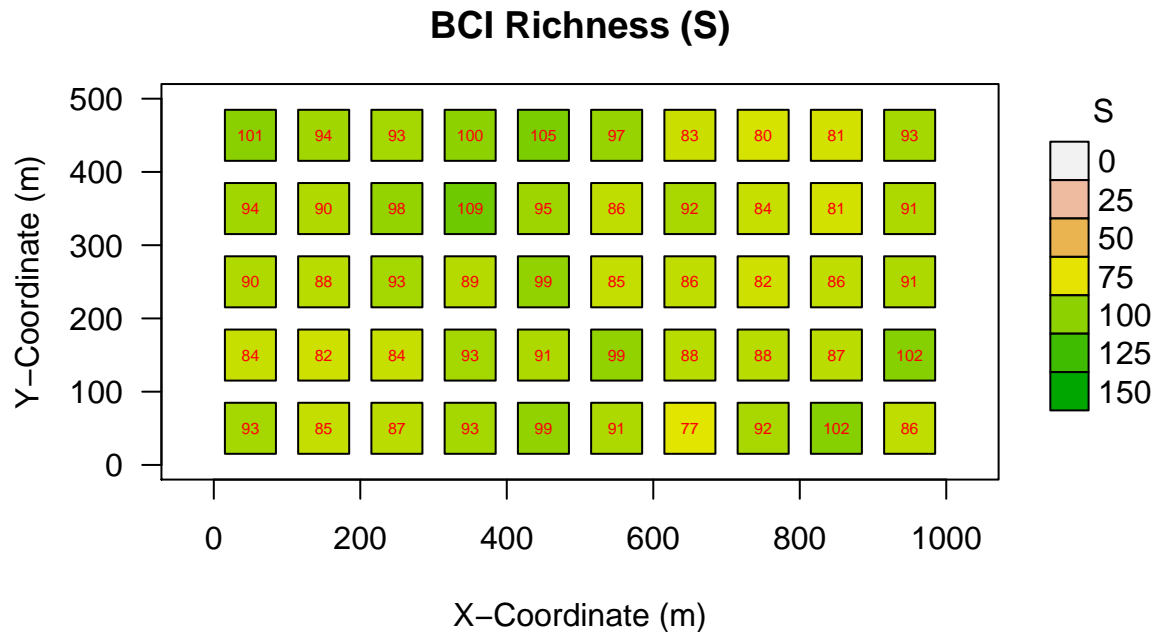
Let's start by plotting tree richness for each plot at BCI. But how do we know how these sites are arranged? Well lucky for us, the BCI soil dataset includes standardized XY-coordinates for each plot

```
head(BCI.soil, n = 2L)      # What does n = 2L do?
```

```
##      x      y      Al      B      Ca      Cu      Fe      K      Mg
## 1 50   50 901.0908 0.79448 1680.021 6.20312 135.287 141.8813 279.1291
## 2 50  150 954.2488 0.66968 1503.365 6.03148 141.808 137.2393 280.4524
##           Mn      P      Zn      N      N.min.      pH
## 1 266.9997 1.95248 2.96948 18.46500 -3.88544 4.32432
## 2 320.4786 2.24740 2.53208 21.59896  5.64388 4.37548
```

The BCI environmental dataset actually has the UTM coordinates for each plot, but the standardized version is easier to use for our purposes. Here we are going to start by plotting sites in XY-coordinate space using the same plotting symbol but we will use our heatmap color to display site richness. Notice that when you are using our heatmap color palette that we add 1 to each value. This is done to deal with any 0s in our data. This is also why we used 151 instead of 150. In addition, we will add a legend to the plot so that readers understand the heatmap scale. This plot also includes a bunch of new commands. You can change them to learn what they do, or you can use `help()` files.

```
# BCI Tree Richness
par(mar=c(4,4,3,4) + 0.1, xpd=TRUE) # What is xpd? It allows the legend to appear outside of the plot
plot(BCI.soil$x, BCI.soil$y, asp = 1, pch = 22, cex = 4, las = 1,
     bg = BCI.S.color[bci.S + 1],
     xlim = c(0, 1000), ylim = c(0, 500), main = "BCI Richness (S)",
     xlab = "X-Coordinate (m)", ylab = "Y-Coordinate (m)")
text(BCI.soil$x, BCI.soil$y, bci.S, cex = 0.5, col = "red")
legend("topright", inset = c(-0.2, 0), pch = 22, pt.cex = 3, title = "S",
     bty = 'n', legend = seq(0, 150, 25), pt.bg = BCI.S.color[seq(1, 151, 25)])
```

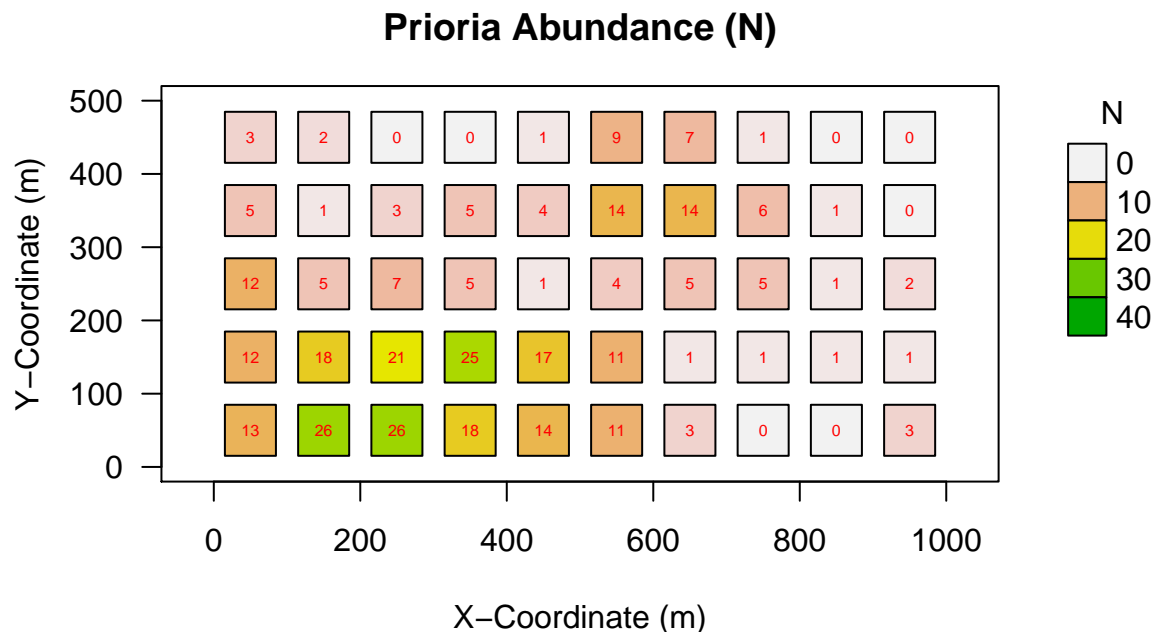


What do you notice when you compare all 50 sites? Is this a good way to compare the sites? What does this method of comparing sites miss?

### BCI Individual Tree Abundance

What would happen if we looked at only specific tree species? Let's use a similar plotting method and just focus on the abundance of a specific tree: the *Prioria* tree. We can use the abundance heatmap palette we created earlier. We will plot these data in the same way we did for richness.

```
# BCI Prioria Abundance
par(mar=c(4,4,3,4) + 0.1, xpd=TRUE)
plot(BCI.soil$x, BCI.soil$y, asp = 1, pch = 22, cex = 4, las = 1,
     bg = BCI.N.color[BCI$Prioria.copaifera + 1],
     xlim = c(0,1000), ylim = c(0, 500), main = "Prioria Abundance (N)",
     xlab = "X-Coordinate (m)", ylab = "Y-Coordinate (m)")
text(BCI.soil$x, BCI.soil$y, BCI$Prioria.copaifera, cex = 0.5, col="red")
legend("topright", inset=c(-0.2, 0), pch=22, pt.cex=3, bty='n',
     title="N", legend=seq(0, 40, 10), pt.bg = BCI.N.color[seq(1, 41, 10)])
```



What do you observe across BCI?

### Doubs River Fish Abundance

In contrast, some datasets might show variation in alpha diversity across samples or sites. The doubs data set in the `ade4` package has fish abundances, environmental variables, and spatial coordinates for 30 sites in the Doubs River (runs near the France-Switzerland boarder in the Jura Mountains). This data set has been used to show that fish communities can be a good indicator of ecological zones in rivers and streams. We will use this data for similar purposes throughout today's exercise.

First we need to import the data from the `ade4` package. We can also look at the dataset to see what it contains. We will use the `str()` function to see what information is in the dataset.

```
data(doubs)
str(doubs, max.level = 1)           # What does max.level do?
```

```
## List of 4
## $ env    : 'data.frame': 30 obs. of  11 variables:
## $ fish   : 'data.frame': 30 obs. of 27 variables:
## $ xy     : 'data.frame': 30 obs. of  2 variables:
## $ species: 'data.frame': 27 obs. of  4 variables:
```

Notice that the doubs data set is actually a list with 4 components (elements). Lists might be new data structures for you. In R, a list is an ordered collection of objects. The first component of the doubs list is the environmental data for each of the 30 sites (`doubs$env`). There are 11 environmental variables in this dataset. See the help file for more information on each (including units). The second component are the abundances at each site for 27 fish species. The third component has the xy spatial coordinates for each site. The last component contains the names of each fish species.

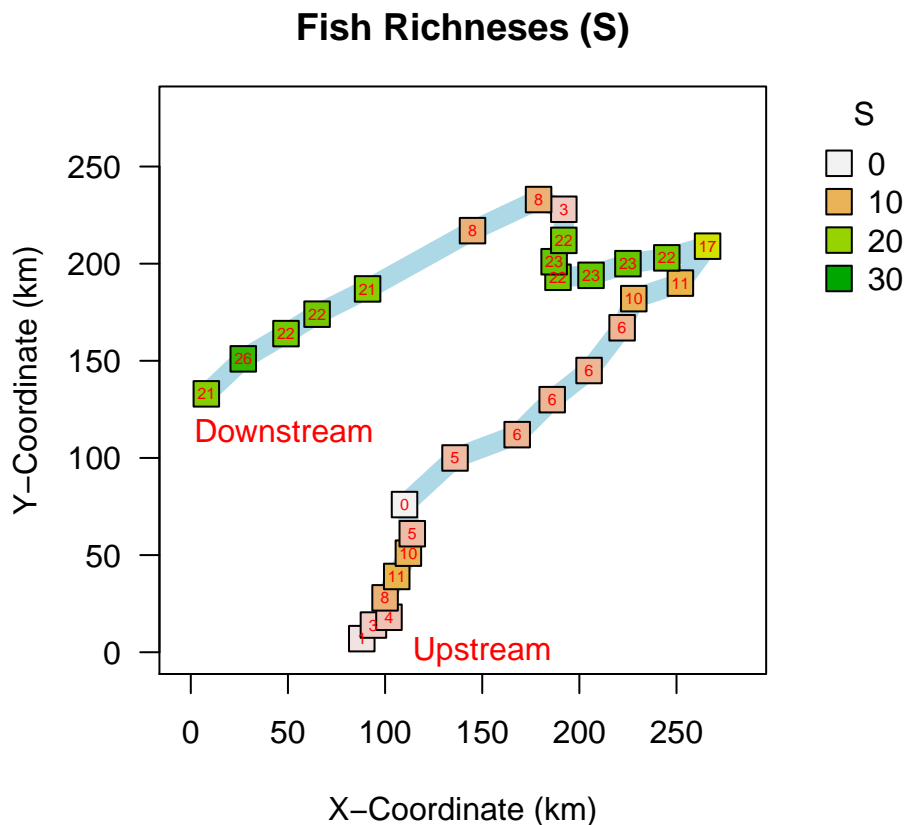
Let's plot fish richness in these stream communities. We will first need to calculate richness and define our color palettes. Once we have these items we can plot stream fish richness in a similar fashion as we did with BCI tree communities. Below are three plots similar to the BCI plots we created above. These plots show richness of fish at all sites in the Doubs River.

```

# Stream Fish
par(mar = c(4, 4, 3, 5) + 0.1, xpd = TRUE)      # Define Plot Parameters
spa.S <- specnumber(doubs$fish)                   # Calculate Richness
spa.S.color <- rev(terrain.colors(31))            # Define Richness Color Palette
spa.N.color <- rev(terrain.colors(7))             # Define Abundance Color Palette

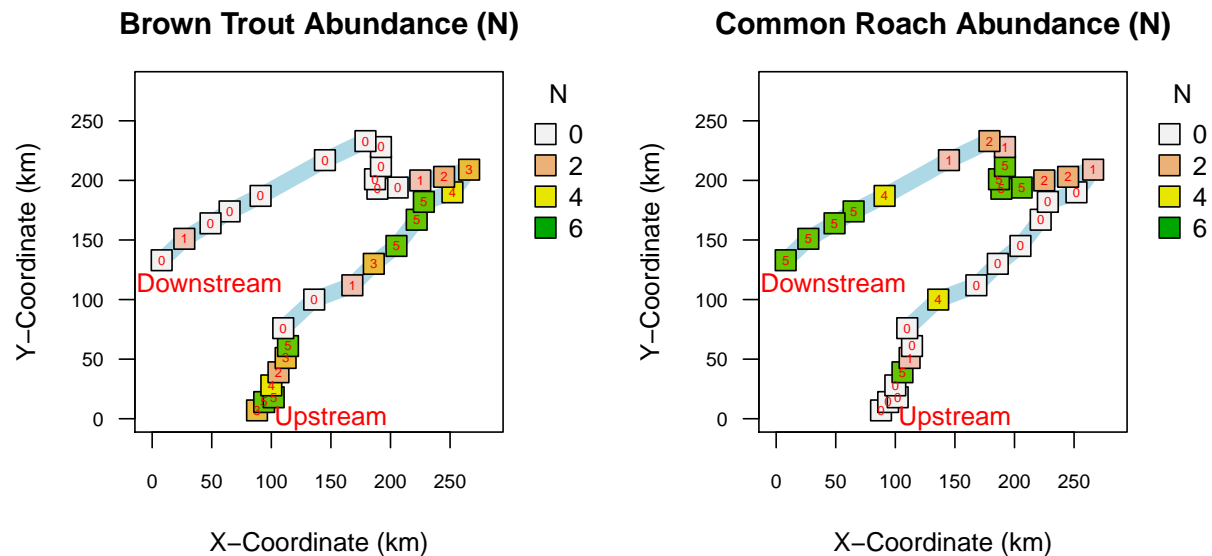
# Stream Fish Richness
plot(doubs$xy, asp = 1, type = 'l', col = "light blue", lwd = 10, las = 1,
     xlim = c(0,280), ylim = c(0,280), main = "Fish Richneses (S)",
     xlab = "X-Coordinate (km)", ylab = "Y-Coordinate (km)")
points(doubs$xy, pch = 22, cex=2, bg = spa.S.color[spa.S + 1])
text(doubs$xy, as.character(spa.S), cex = 0.5, col="red")
text(150, 0, "Upstream", cex = 1, col = "red")
text(48, 114, "Downstream", cex = 1, col = "red")
legend("topright", inset=c(-0.25, 0), pch = 22, pt.cex = 2, bty = 'n',
     title = "S", legend = seq(0, 30, 10), pt.bg = spa.S.color[seq(1, 31, 10)])

```



Notice that the code we used here is almost identical to that we used with the BCI plots. Though creating a single figure in R may seem like a lot of work, it is beneficial when making multiple plots because you can recycle code easily. However, sometimes there are slight variations (e.g., the blue line in the stream plot).

Below are the plots for two individual fish species in the Doubs River. You should be able to code this with ease using the Stream Fish Richness plot code and the BCI abundance plot code. You can try this on your own time, but here we are just going to provide the plots for you.



How does this dataset differ from the BCI data? Is richness the most appropriate tool to compare communities? What about other  $\alpha$ -diversity measures? What do they miss?

This week we are going to compare the diversity of communities across sites. We will start by comparing diversity across sites.

## B. Beta Diversity

Beta diversity is a measure of between-habitat diversity, or the difference in species composition between two or more localities.

### Turnover

One way to think about beta diversity is the change in communities over time or space. This concept is generally referred to as *turnover*. Turnover is defined as the rate or magnitude of change in species composition in time or along a predefined spatial or environmental gradient.

We can calculate turnover with the following equation:  $t = \frac{b+c}{S_1+S_2}$  where  $b$  = the number of species present in only the first census;  $c$  = the number of species present in only the second census;  $S_1$  = the total number of species in the first census; and  $S_2$  = the number of species in the second census.

In R, we could write this as a function as follows:

```
turnover <- function(site1 = " ", site2 = " "){
  site1 = subset(site1, select = site1 > 0)      # Removes absences
  site2 = subset(site2, select = site2 > 0)      # Removes absences
  b = length(setdiff(colnames(site1), colnames(site2))) # Number of unique in 1st
  c = length(setdiff(colnames(site2), colnames(site1))) # Number of unique in 2nd
  s1 = length(site1)                               # Number of species in 1st
  s2 = length(site2)                               # Number of species in 2nd
  t = round((b + c)/(s1 + s2), 3)                 # Calculats turnover to three decimals
  return(t)                                        # Returns turnover
}
```

We can now use this function to calculate turnover between any two points in time or space.

```
turnover(doubs$fish[1,], doubs$fish[2,])
```

```
## [1] 0.5
```

## $\beta$ -Diversity

Another classic measures of  $\beta$ -diversity was developed by Whittaker (1960):  $\beta_W = \frac{S}{\bar{\alpha}} - 1$  where  $S$  = the total number of species recorded in a system and  $\bar{\alpha}$  is the average sample richness. Subtracting 1 scales the results from 0 - 1.

Again, we can write this as a function in R:

```
beta.w <- function(site1 = "", site2 = ""){  
  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)  
  a.bar = mean(c(specnumber(site1), specnumber(site2)))  
  b.w   = s/a.bar - 1  
  return(b.w)  
}
```

$\beta_w$  ranges from 0 (minimum  $\beta$  diversity) to 1 (maximum  $\beta$  diversity). However, this measure works best for comparing two sites.

Using our new R function, we can now calculate  $\beta_w$  between two sites:

```
beta.w(doubs$fish[1, ], doubs$fish[2, ])
```

```
## [1] 0.5
```

These measures of  $\beta$ -diversity are useful when comparing two sites, but special precautions must be made before making multiple comparisons. These precautions involve things like corrections for time and for sample size, and you can find modifications to each. However, there are also better ways to compare multiple communities at the same time.

## Measures of compositional similarity

When comparing multiple communities, measures of resemblance (similarity) are the most appropriate way to compare sites. <- I think this should be more detailed These measures can either be based on incidence (presence-absence; qualitative) data, or abundance (absolute or total; quantitative) data. We will use common notations when referring to these measures. For reference, all notations will follow that of Legendre & Legendre 2012.



## 1. Incidence-Based

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 (global view)
Sørensen	$S_8 = \frac{2a}{(2a+b+c)}$	Compares the number of shared species to the mean number of species in a single assemblage (local view)

Where  $a$  = the number of shared species,  $b$  = the number of unique species in the first assemblage, and  $c$  = the number of unique species in the second assemblage. What are the differences between these two measures? Why might you be interested in using one over the other? We should note that there are other measures (including: Ochiai, Kulczynski-Cody, ...). See Table 6.1 in Magurran & McGill 2011, or Table 7.2 in Legendre & Legendre 2012 for more details. Also note, these equations calculate similarity. However, these values are often converted to dissimilarity, or distance, when calculated in R. In **vegan**, distances are calculated as  $D = 1 - S$ ; however, this conversion may not preserve Euclidean distance. See Table 7.2 in Legendre & Legendre 2012 for more information.

## 2. Abundance-Based

Index	Equation	Properties
Bray-Curtis Dissimilarity	$D_{14} = \frac{\sum_{j=1}^p  y_{1j} - y_{2j} }{\sum_{j=1}^p (y_{1j} + y_{2j})}$	A quantitative version of the Sørensen index. Semimetric. Generally used as an overall similarity measure. Also known as the percentage difference
Morisita-Horn	$S_{MH} = \frac{2 \sum_{j=1}^p y_{1j} y_{2j}}{\left( \sum_{j=1}^p y_{1j}^2 + \sum_{j=1}^p y_{2j}^2 \right)}$	Dominated by the most abundant species in each site. Resistant to undersampling (no equivalent in Legendre and Legendre)
Chord Distance	$D_3 = \sqrt{2 \left( 1 - \frac{\sum_{j=1}^p y_{1j} \cdot y_{2j}}{\sum_{j=1}^p y_{1j}^2 \cdot \sum_{j=1}^p y_{2j}^2} \right)}$	Range: 0 to $\sqrt{2}$ (two sites share the same species in the same proportions to no species in common)
Chi-Squared Distance	$D_{16} = \sqrt{\sum_{j=1}^p \frac{1}{y_{+j}/y_{++}} \left( \frac{y_{1j}}{y_{1+}} - \frac{y_{2j}}{y_{2+}} \right)^2}$	Used to compute the association between the rows and columns of a frequency table. Preserved in correspondence analysis.

Where  $y_{1j}$  is the abundance of each species (1:p) in community 1 and  $y_{2j}$  in community 2,  $y_{1+}$  is the sum of abundances in community 1 and  $y_{2+}$  in community 2,  $y_{++}$  is the sum of all abundances at all sites,  $y_{+j}$  is the sum of abundances of the  $j^{th}$  species across communities, and  $y_{ij}/y_{i+}$  is the relative abundances at site  $i$ .

It should be noted that many of these distances are not Euclidean. In **vegan**, distances are calculated as  $D = 1 - S$ ; however, this conversion may not preserve Euclidean distance. See Table 7.3 in Legendre & Legendre 2012 for more information.

### 3. Other Measures (may not be appropriate for species abundances)

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

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

#### Calculate Sample Resemblance

Many of these similarity metrics are in the vegan `vegdist()` function. We can use these to measure resemblance in each of our fish communities in the Doubs River. However, remember from our plots above that site 8 had no observations. We will need to remove that site from the data set.

```
spe <- doubs$fish           # Assign the site-by-species matrix to 'spe'

# It is always good to check your sites
rowSums(spe)               # Notice site 8 is empty

##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  3 12 16 21 34 21 16  0 14 14 11 18 19 28 33 40 44 42 46 56 62 72  4 15 11
## 26 27 28 29 30
## 43 63 70 87 89
```

```
spe <- spe[-8, ]

# Calculate Jaccard Dissimilarity
spe.dj <- vegdist(spe, method="jaccard", binary=TRUE) # Binary?

# Calculate Bray-Curtis Dissimilarity between Doubs River sites
spe.db <- vegdist(spe, method="bray")
```

We have now measured the resemblance off all Doubs River sites. How do you visualize these data? Well, you could print the resemblance matrix:

```
spe.db
```

If you run this command in your console, you will see a large diagonal matrix. Resemblance matrices usually just show either the upper or lower triangle of values. This is because the two triangles will have the same information. However, if you wanted a square resemblance matrix, you could use the following commands:

```
# Calculate Bray-Curtis Dissimilarity between Doubs River sites
spe.db <- vegdist(spe, method="bray", upper=TRUE, diag=TRUE)
```

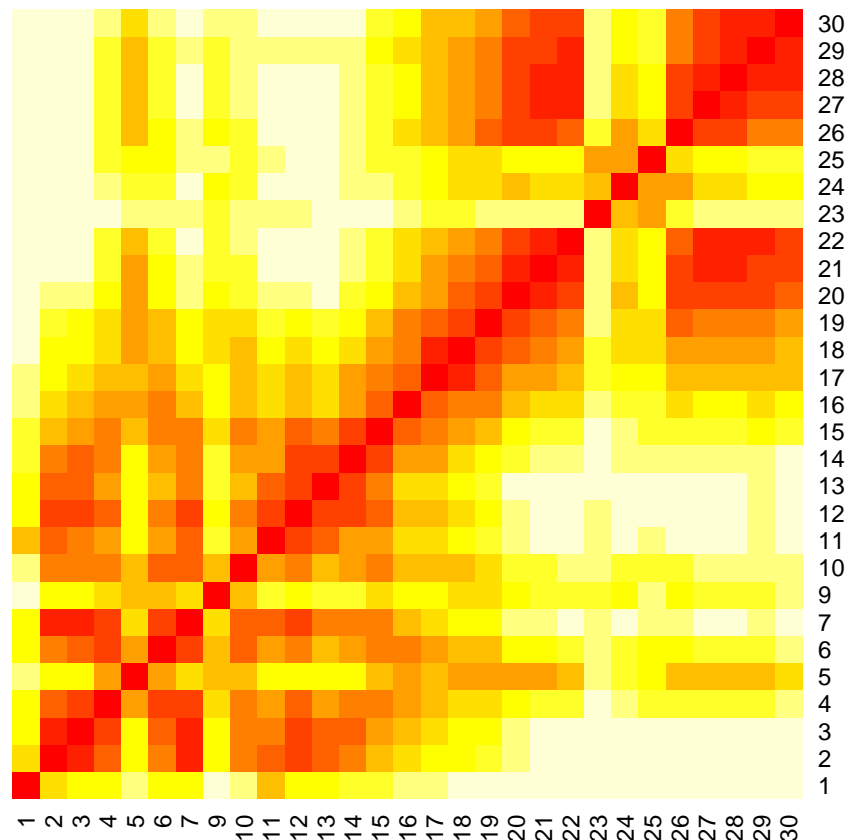
Again, you could look at this resemblance matrix by printing the values, but that is not easy to interpret and does not show relations between sites. In the next part, we will learn more elaborate ways to visualize the ecological resemblance of communities.

## D. Visualizaton

### Heatmaps

One way to visualize  $\beta$ -diversity is to plot the data in our resemblance matrix using a heatmap. R has a nice heatmap function that makes this fairly easy

```
heatmap(as.matrix(spe.db), Rowv=NA, Colv=NA, symm=TRUE, margins=c(2,2))
```

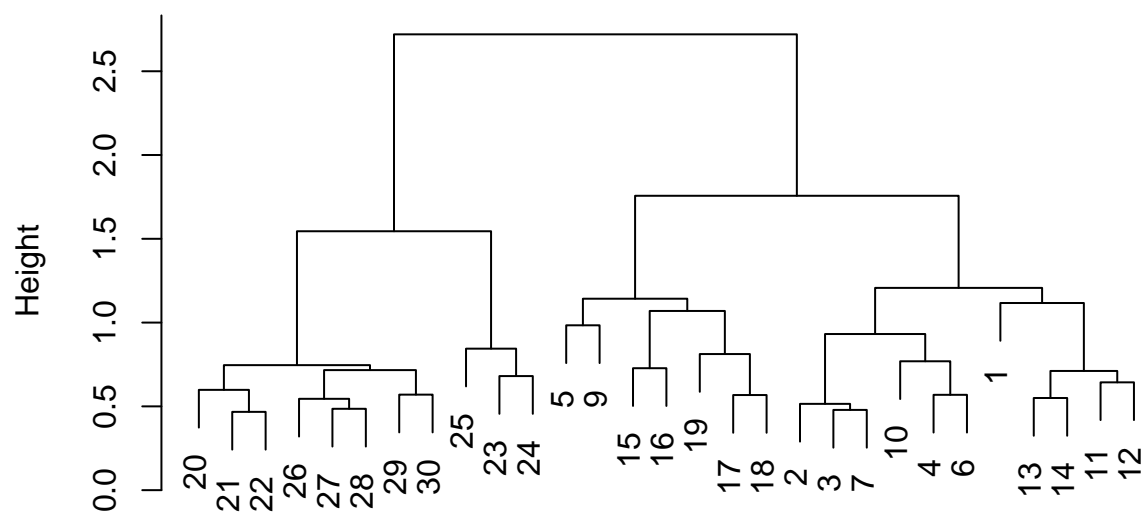


### Cluster Analysis

#### Hierarchical Clustering (Ward Clustering)

```
spe.db <- vegdist(spe, method="bray")
spe.norm <- decostand(spe, "normalize")
spe.ch <- vegdist(spe.norm, "euc")
spe.ch.ward <- hclust(spe.ch, method="ward.D")
spe.ch.ward$height <- sqrt(spe.ch.ward$height)
plot(spe.ch.ward)
```

## Cluster Dendrogram

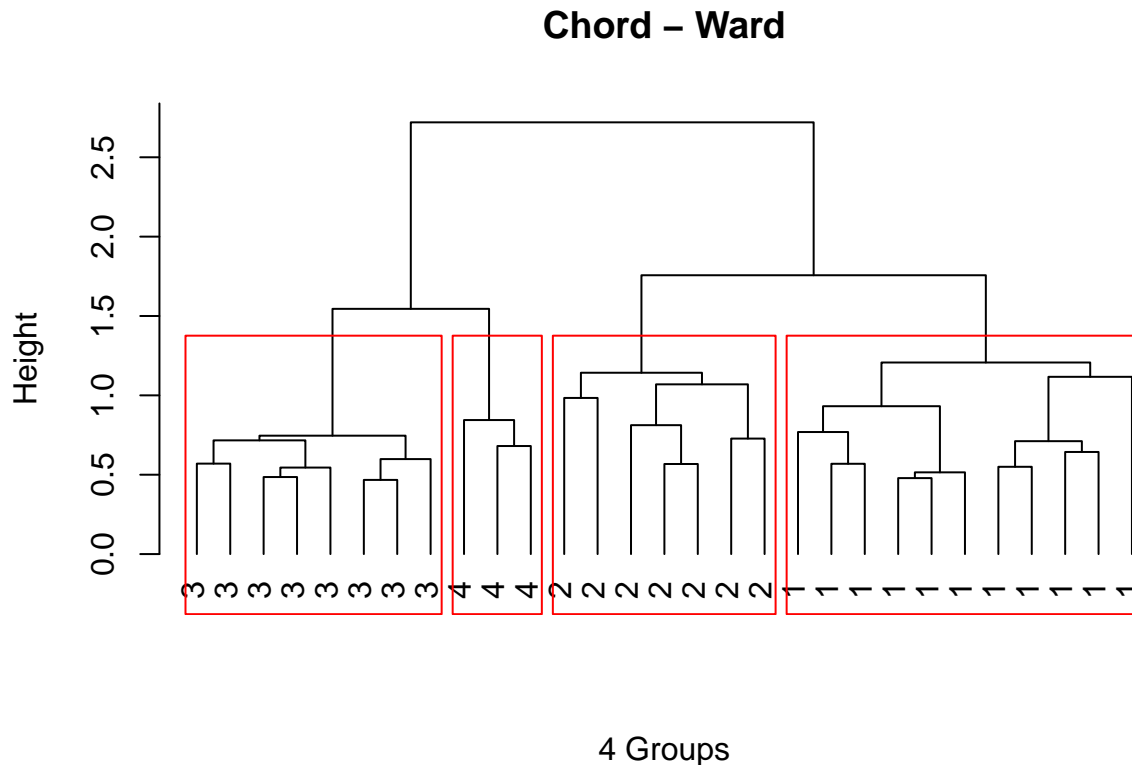


spe.ch  
hclust (\*, "ward.D")

```
# Groups
require(gclus)
```

```
## Loading required package: gclus
## Loading required package: cluster
```

```
spe.chwo <- reorder.hclust(spe.ch.ward, spe.ch)
plot(spe.chwo, hang=-1, xlab="4 Groups", sub="", ylab="Height", main="Chord - Ward", labels=cutree(spe.chwo, k=4))
rect.hclust(spe.chwo, k=4)
```



### Visualization using Ordination

Various techniques | Method | Distance Preserved | Variables | | Principapl component analysis (PCA) | Euclidean distance | Quantiative data, linear relationships | | Correspondence analysis (CA) |  $\chi^2$  distance | Non-negative, dimensionally homogeneous quantita6tive or binary data; species frequencies or presence/absence data | | Principal coordinate analysis (PCoA), metric (multidimensional) scalling, classical scaling | Any distance measure | Quantitative, semiquantitative, qualitative, or mixed | |Nonmetric multidimensional scalling (nMDS) | Any distance measure | Quantitative, semiquantitative, qualitative, or mixed |

(REF: L&L , Table 9.1)

How does ordination work?

### NMDS

```
spe.nmnds <- metaMDS(spe, distance="bray")
```

```
## Run 0 stress 0.07477822
## Run 1 stress 0.0750682
## ... procrustes: rmse 0.01492291 max resid 0.06352924
## Run 2 stress 0.07383921
## ... New best solution
## ... procrustes: rmse 0.01959818 max resid 0.09447098
## Run 3 stress 0.07376402
```

```
## ... New best solution
## ... procrustes: rmse 0.00380125  max resid 0.01474355
## Run 4 stress 0.1104334
## Run 5 stress 0.114184
## Run 6 stress 0.07506875
## Run 7 stress 0.114824
## Run 8 stress 0.1133739
## Run 9 stress 0.07478486
## Run 10 stress 0.1140208
## Run 11 stress 0.07376349
## ... New best solution
## ... procrustes: rmse 0.0001450475  max resid 0.0006169966
## *** Solution reached
```

```
spe.nmds
```

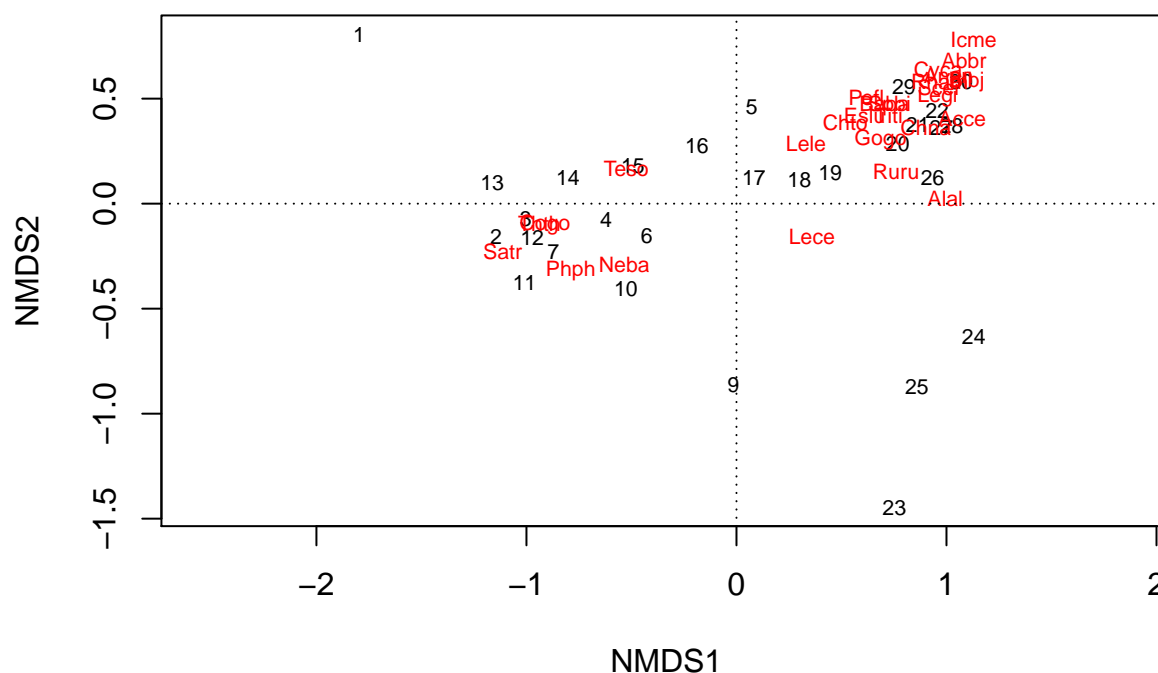
```
##
## Call:
## metaMDS(comm = spe, distance = "bray")
##
## global Multidimensional Scaling using monoMDS
##
## Data:      spe
## Distance: bray
##
## Dimensions: 2
## Stress:      0.07376349
## Stress type 1, weak ties
## Two convergent solutions found after 11 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'spe'
```

```
spe.nmds$stress
```

```
## [1] 0.07376349
```

```
plot(spe.nmds, type="t", main=paste("nMDS/Bray - Stress =", round(spe.nmds$stress, 3)))
abline(h=0, lty=3)
abline(v=0, lty=3)
```

## nMDS/Bray – Stress = 0.074



What is the nMDS stress? How is this used to judge the quality of the ordination

## Classic Multidimensional Scalling (PCoA)

```
spe.bray <- vegdist(spe, method="bray")
spe.b.pcoa <- cmdscale(spe.bray, eig=TRUE)

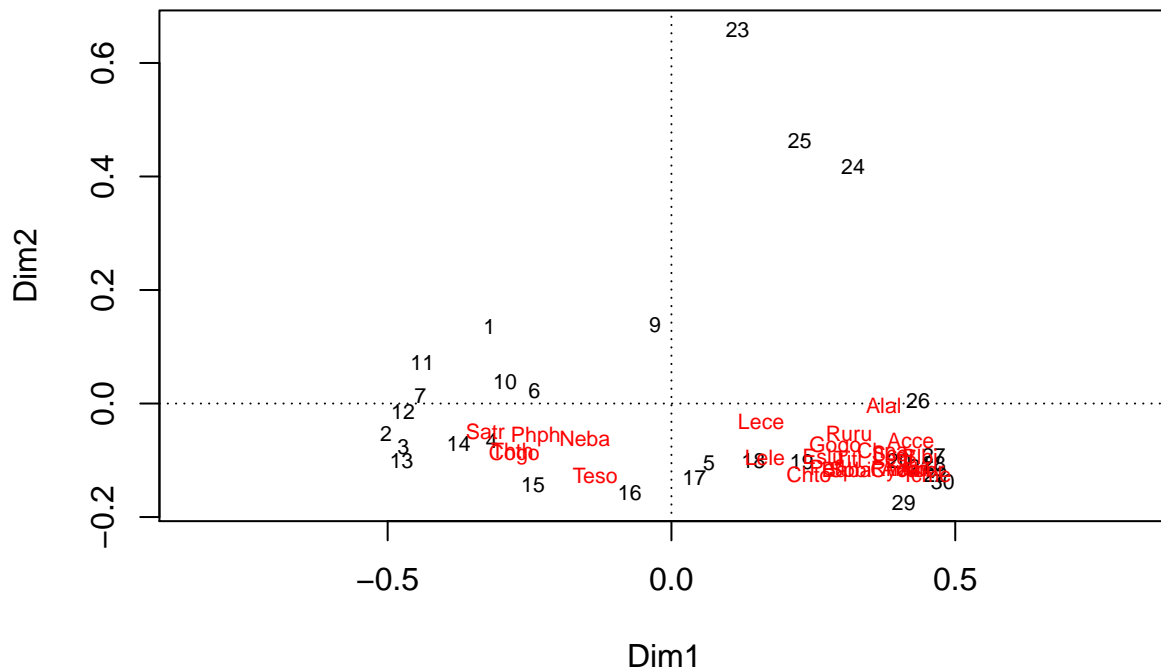
ordiplot(scores(spe.b.pcoa)[,c(1,2)], type='t', main="PCoA with species")

## Warning in ordiplot(scores(spe.b.pcoa)[, c(1, 2)], type = "t", main =
## "PCoA with species"): Species scores not available

abline(h=0, lty=3)
abline(v=0, lty=3)

# Add Species
spe.wa <- wascores(spe.b.pcoa$points[,1:2], spe)
text(spe.wa, rownames(spe.wa), cex=0.7, col="red")
```

## PCoA with species



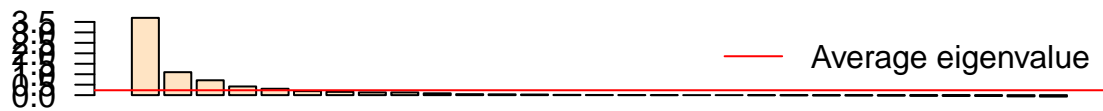
```
evplot <- function(ev)
{
  # Broken stick model (MacArthur 1957)
  n <- length(ev)
  bsm <- data.frame(j=seq(1:n), p=0)
  bsm$p[1] <- 1/n
  for (i in 2:n) bsm$p[i] <- bsm$p[i-1] + (1/(n + 1 - i))
  bsm$p <- 100*bsm$p/n
  # Plot eigenvalues and % of variation for each axis
  par(mfrow=c(2,1))
  barplot(ev, main="Eigenvalues", col="bisque", las=2)
  abline(h=mean(ev), col="red")
  legend("topright", "Average eigenvalue", lwd=1, col=2, bty="n")
  barplot(t(cbind(100*ev/sum(ev), bsm$p[n:1])), beside=TRUE,
    main="% variation", col=c("bisque",2), las=2)
  legend("topright", c("% eigenvalue", "Broken stick model"),
    pch=15, col=c("bisque",2), bty="n")
}

evplot(spe.b.pcoa$eig)

env <- doubs$env
env <- env[,-8, ]
spe.pcoa.env <- envfit(spe.b.pcoa, env)
evplot(spe.b.pcoa$eig)
```



## Eigenvalues



## % variation



Why not PCA?

## E. Hypothesis Testing

### Constrained Ordination

```
# subset explanatory variables
```

```
envdas <- env[,1]
envtopo <- env[,c(2:4)]
envchem <- env[,c(5:11)]
spe.hel <- decostand(spe, method="hellinger")

spe.rda <- rda(spe.hel ~ ., envchem)
coef(spe.rda)
```

```
##           RDA1           RDA2           RDA3           RDA4           RDA5
## pH    0.0020384098 -0.0031031275 -0.0190886957 -0.016823415  0.004127536
## har -0.0011912221  0.0007238813 -0.0066409901 -0.010576449  0.002295200
```

```
## pho 0.0008030375 0.0006560255 0.0002170545 0.002050325 0.007613774
## nit -0.0011933640 0.0011698470 -0.0005397298 0.001754538 -0.001273360
## amm 0.0024457337 -0.0049289180 0.0006951098 -0.004139305 -0.004732773
## oxy 0.0066262263 -0.0008245053 -0.0127449854 0.006086134 -0.005810343
## bdo -0.0002256933 -0.0042878090 -0.0058318080 -0.000140698 -0.011305550
##          RDA6          RDA7
## pH -0.0455461425 0.0996685848
## har 0.0074031347 -0.0009897574
## pho -0.0045440954 -0.0038977199
## nit -0.0008977745 -0.0002185856
## amm 0.0193330610 0.0115315915
## oxy -0.0015043312 -0.0046358191
## bdo -0.0080555706 -0.0033015665
```

```
spechem.physio <- rda(spe.hel, envchem, envtopo)
```

```
# Permutatoin Test
anova.cca(spe.rda, step=1000)
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = spe.hel ~ pH + har + pho + nit + amm + oxy + bdo, data = envchem)
##          Df Variance      F Pr(>F)
## Model      7 0.30442 4.6102 0.001 ***
## Residual  21 0.19809
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Variance Partitioning

```
spe.part.all <- varpart(spe.hel, envchem, envtopo)
```

## Multivariate Models

Mantel Test

PERMANOVA

## Homework

Microbes in an Agricultural System

History of the system Define hypotheses

Richness (rarefied) for each treatment (barplot, box-n-whisker plot) Bray Curtis PCoA PERMANOVA  
Indicator groups

BCI Data - which physical property underlies variation in Tree Composition