Assignment: Among Site (Beta) Diversity

Savannah Bennett; Z620: Quantitative Biodiversity, Indiana University

31 January, 2017

OVERVIEW

In this Assignment, 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. This requires that we 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

Directions:

1. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the exercise as possible during class; what you do not complete in class will need to be done on your own outside of class.
3. Use the Handout as a guide; it contains a more complete description of data sets along with the proper scripting needed to carry out the exercise.
4. Be sure to **answer the questions** in this exercise document; they also correspond to the Handout. Space for your answer is provided in this document and indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">".
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. When you are done wit the Assignment, **Knit** the text and code into a html file.
7. After Knitting, please submit the completed Assignment by creating a **pull request** via GitHub. Your pull request should include this file *beta_assignment.Rmd* and the html output of `Knitr` (*beta_assignment.html*).

1) R SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:

1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/Week3-Beta*" folder, and
4. load the `vegan` R package (be sure to install if needed).

```
rm(list = ls())
getwd()
```

```
## [1] "C:/Users/Savannah/GitHub/QB2017_Bennett/Week3-Beta"
```

```
setwd("C:/Users/Savannah/GitHub/QB2017_Bennett/Week3-Beta")
```

2) LOADING DATA

Load dataset

In the R code chunk below, do the following:

1. load the `doubs` dataset from the `ade4` package, and
2. explore the structure of the dataset.

```
package.list <- c('vegan', 'ade4', 'viridis', 'gplots', 'BiodiversityR', 'indicspecies')
for (package in package.list) {
if (!require(package,character.only=T,quietly=T)) {
install.packages(package)
library(package,character.only=T)
}
}
```

```
## This is vegan 2.4-2
```

```
##
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:vegan':
##
##     cca
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
## BiodiversityR 2.8-0: Use command BiodiversityRGUI() to launch the Graphical User Interface and to lea
```

```
data(doubs)
str(doubs,max.level =1)
```

```
## 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:
```

```
head(doubs$env)
```

```
##   dfs alt   slo flo pH har pho nit amm oxy bdo
## 1   3 934 6.176  84 79  45   1  20   0 122  27
## 2  22 932 3.434 100 80  40   2  20  10 103  19
## 3 102 914 3.638 180 83  52   5  22   5 105  35
## 4 185 854 3.497 253 80  72  10  21   0 110  13
## 5 215 849 3.178 264 81  84  38  52  20  80  62
## 6 324 846 3.497 286 79  60  20  15   0 102  53
```

```
help(doubs)
```

```
## starting httpd help server ...
```

```
##  done
```

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

a. How many objects are in `doubs`?
b. How many fish species are there in the `doubs` dataset?
c. How many sites are in the `doubs` dataset?

Answer 1a:There are 4 objects in the dataset. ***Answer 1b***:There are 27 fish species in the doubs dataset. ***Answer 1c***:There are 30 sites in this dataset.

Visualizing the Doubs River Dataset

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:Fish richness generally seems higher at the downstream site compared to the upstream site.
Answer 2b:Brown Trout abundance seeems higher at the upstream site compared to the downstream site. ***Answer 2c***:These patterns indicate that it is important to analyze species composition in addition to species richness. Richness alone should not be used to analyze diversity, because this metric may not be the best representation of the community or system (i.e. it may not accurately reflect community dynamics).

3) QUANTIFYING BETA-DIVERSITY

In the R code chunk below, do the following:

1. write a function (`beta.w()`) to calculate Whittaker's β-diversity (i.e., β_w) that accepts a site-by-species matrix with optional arguments to specify pairwise turnover between two sites, and
2. use this function to analyze various aspects of β-diversity in the Doubs River.

```
beta.w <-function(site.by.species =""){
SbyS.pa <- decostand(site.by.species,method ="pa")
S <- ncol(SbyS.pa[,which(colSums(SbyS.pa) >0)])
a.bar <- mean(specnumber(SbyS.pa))
b.w <- round(S/a.bar,3)
return(b.w)
}

beta.w2 = function(site.by.species = "", sitenum1 = "", sitenum2 = "", pairwise = FALSE){
  if (pairwise == TRUE){
    if (sitenum1 == "" | sitenum2 == "") {
      print("Error: please specify sites to compare")
      return(NA)}
    site1 = site.by.species[sitenum1,]
    site2 = site.by.species[sitenum2,]
    site1 = subset(site1, select = site1 > 0)
    site2 = subset(site2, select = site2 > 0)
    gamma = union(colnames(site1), colnames(site2))
    s = length(gamma)
    a.bar = mean(c(specnumber(site1), specnumber(site2)))
    b.w = round(s/a.bar - 1, 3)
    return(b.w)
```

```
  }
  else{
    SbyS.pa = decostand(site.by.species, method = "pa")
    S = ncol(SbyS.pa[,which(colSums(SbyS.pa) > 0)])
    a.bar = mean(specnumber(SbyS.pa))
    b.w = round(S/a.bar, 3)
    return(c(b.w,a.bar,S))
  }
}

beta.w2(doubs$fish)
```

```
## [1]  2.16 12.50 27.00
```

```
beta.w2(doubs$fish, 1,2, pairwise=TRUE)
```

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

```
beta.w2(doubs$fish, 1,10, pairwise=TRUE)
```

```
## [1] 0.714
```

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

a. Describe how local richness (α) and turnover (β) contribute to regional (γ) fish diversity in the Doubs.
b. Is the fish assemblage at site 1 more similar to the one at site 2 or site 10?
c. Using your understanding of the equation $\beta_w = \gamma/\alpha$, how would your interpretation of β change if we instead defined beta additively (i.e., $\beta = \gamma - \alpha$)?

Answer 3a:The turnover describes how many times higher the gamma diversity is than the alpha diversity. The relationship between these variables is multiplicative in that the gamma diversity is equal to the alpha diversity multiplied by the beta diversity. In this dataset, the beta diversity value indicates that the gamma diversity is about twice as much as the alpha diversity. Therefore, gamma diversity is 27.
Answer 3b:Site 1 is more similar to site 2 than site 10 because the beta diversity value for these two sites is lower.
Answer 3c:If beta was defined additively, it would no longer reflect how many times more diverse the regional species pool is than the local pool. The beta value for fish in this dataset would be about 14.5 instead of 12.5. Diversity among sites would therefore seem more similar.

The Resemblance Matrix

In order to quantify β-diversity for more than two samples, we need to introduce a new primary ecological data structure: the **Resemblance Matrix**.

Question 4: How do incidence- and abundance-based metrics differ in their treatment of rare species?

Answer 4:The incidence based tests account for rare species more so than the abundance based tests. Unlike the incidence based metric, the abundance based metric simply sums the number of each species at each site, which causes the rate species to have less weight in determining similarity among separate sites.

In the R code chunk below, do the following:

1. make a new object, `fish`, containing the fish abundance data for the Doubs River,
2. remove any sites where no fish were observed (i.e., rows with sum of zero),
3. construct a resemblance matrix based on SÃ¸rensen's Similarity ("fish.ds"), and
4. construct a resemblance matrix based on Bray-Curtis Distance ("fish.db").

```r
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)

fish.db
```

```
##             1          2          3          4          5          6
## 2  0.60000000
## 3  0.68421053 0.14285714
## 4  0.75000000 0.33333333 0.18918919
## 5  0.89189189 0.69565217 0.68000000 0.49090909
## 6  0.75000000 0.39393939 0.29729730 0.19047619 0.41818182
## 7  0.68421053 0.14285714 0.12500000 0.24324324 0.64000000 0.24324324
## 9  1.00000000 0.69230769 0.73333333 0.65714286 0.58333333 0.54285714
## 10 0.88235294 0.38461538 0.40000000 0.37142857 0.54166667 0.25714286
## 11 0.57142857 0.30434783 0.40740741 0.43750000 0.68888889 0.43750000
## 12 0.71428571 0.20000000 0.23529412 0.33333333 0.69230769 0.38461538
## 13 0.72727273 0.29032258 0.31428571 0.45000000 0.73584906 0.55000000
## 14 0.80645161 0.40000000 0.31818182 0.34693878 0.67741935 0.42857143
## 15 0.83333333 0.51111111 0.46938776 0.40740741 0.55223881 0.37037037
## 16 0.86046512 0.65384615 0.57142857 0.47540984 0.45945946 0.37704918
## 17 0.91489362 0.67857143 0.63333333 0.50769231 0.51282051 0.44615385
## 18 0.95555556 0.74074074 0.72413793 0.58730159 0.50000000 0.52380952
## 19 1.00000000 0.79310345 0.70967742 0.61194030 0.50000000 0.52238806
## 20 1.00000000 0.91176471 0.88888889 0.74025974 0.48888889 0.68831169
## 21 1.00000000 0.94594595 0.92307692 0.78313253 0.50000000 0.73493976
## 22 1.00000000 0.97619048 0.95454545 0.82795699 0.52830189 0.78494624
## 23 1.00000000 1.00000000 1.00000000 0.92000000 0.89473684 0.84000000
## 24 1.00000000 1.00000000 1.00000000 0.88888889 0.79591837 0.77777778
## 25 1.00000000 1.00000000 0.92592593 0.81250000 0.68888889 0.68750000
## 26 1.00000000 0.96363636 0.93220339 0.78125000 0.55844156 0.68750000
## 27 1.00000000 0.97333333 0.94936709 0.83333333 0.56701031 0.76190476
## 28 1.00000000 0.97560976 0.95348837 0.82417582 0.57692308 0.78021978
## 29 0.97777778 0.93939394 0.92233010 0.81481481 0.53719008 0.77777778
## 30 1.00000000 1.00000000 0.98095238 0.87272727 0.59349593 0.83636364
##             7          9         10         11         12         13
## 2
## 3
## 4
## 5
## 6
## 7
## 9  0.66666667
## 10 0.26666667 0.57142857
## 11 0.33333333 0.76000000 0.44000000
## 12 0.17647059 0.68750000 0.37500000 0.24137931
## 13 0.37142857 0.81818182 0.57575758 0.33333333 0.18918919
## 14 0.36363636 0.76190476 0.47619048 0.43589744 0.21739130 0.19148936
## 15 0.38775510 0.65957447 0.40425532 0.50000000 0.33333333 0.38461538
```

```
## 16 0.53571429 0.70370370 0.51851852 0.64705882 0.55172414 0.59322034
## 17 0.60000000 0.68965517 0.51724138 0.63636364 0.58064516 0.61904762
## 18 0.68965517 0.64285714 0.57142857 0.69811321 0.66666667 0.70491803
## 19 0.67741935 0.66666667 0.63333333 0.82456140 0.75000000 0.81538462
## 20 0.86111111 0.68571429 0.77142857 0.91044776 0.89189189 0.92000000
## 21 0.89743590 0.76315789 0.81578947 0.91780822 0.92500000 0.95061728
## 22 0.93181818 0.76744186 0.86046512 0.95180723 0.95555556 0.97802198
## 23 0.90000000 0.77777778 0.88888889 0.86666667 0.90909091 1.00000000
## 24 0.93548387 0.72413793 0.79310345 0.92307692 0.93939394 1.00000000
## 25 0.85185185 0.84000000 0.76000000 0.90909091 0.93103448 1.00000000
## 26 0.89830508 0.71929825 0.82456140 0.92592593 0.93442623 0.96774194
## 27 0.92405063 0.76623377 0.84415584 0.94594595 0.95061728 0.97560976
## 28 0.93023256 0.76190476 0.85714286 0.95061728 0.95454545 0.97752809
## 29 0.90291262 0.78217822 0.84158416 0.89795918 0.90476190 0.90566038
## 30 0.96190476 0.84466019 0.90291262 0.98000000 0.98130841 1.00000000
##            14         15         16         17         18         19
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15 0.24590164
## 16 0.44117647 0.26027397
## 17 0.50000000 0.40259740 0.26190476
## 18 0.60000000 0.46666667 0.34146341 0.13953488
## 19 0.67567568 0.56962025 0.39534884 0.31111111 0.25000000
## 20 0.83333333 0.70786517 0.58333333 0.42000000 0.32653061 0.23529412
## 21 0.86666667 0.76842105 0.62745098 0.49056604 0.40384615 0.29629630
## 22 0.90000000 0.77142857 0.66071429 0.55172414 0.47368421 0.38983051
## 23 0.93750000 0.94594595 0.90909091 0.83333333 0.82608696 0.84000000
## 24 0.90697674 0.87500000 0.81818182 0.69491525 0.64912281 0.63934426
## 25 0.84615385 0.81818182 0.76470588 0.74545455 0.66037736 0.61403509
## 26 0.85915493 0.76315789 0.63855422 0.54022989 0.45882353 0.32584270
## 27 0.89010989 0.77083333 0.66990291 0.57009346 0.48571429 0.37614679
## 28 0.89795918 0.78640777 0.69090909 0.57894737 0.50000000 0.41379310
## 29 0.84347826 0.73333333 0.65354331 0.51145038 0.44186047 0.41353383
## 30 0.93162393 0.81967213 0.72093023 0.57894737 0.52671756 0.48148148
##            20         21         22         23         24         25
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
```

```
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21 0.10169492
## 22 0.18750000 0.10447761
## 23 0.86666667 0.87878788 0.89473684
## 24 0.57746479 0.61038961 0.65517241 0.57894737
## 25 0.67164179 0.69863014 0.73493976 0.46666667 0.46153846
## 26 0.21212121 0.20000000 0.25217391 0.82978723 0.48275862 0.59259259
## 27 0.19327731 0.13600000 0.12592593 0.88059701 0.61538462 0.70270270
## 28 0.22222222 0.16666667 0.12676056 0.89189189 0.64705882 0.72839506
## 29 0.24475524 0.18120805 0.11949686 0.91208791 0.70588235 0.77551020
## 30 0.29655172 0.23178808 0.18012422 0.91397849 0.71153846 0.78000000
##            26         27         28         29
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27 0.18867925
## 28 0.23893805 0.09774436
## 29 0.33846154 0.18666667 0.14649682
## 30 0.36363636 0.19736842 0.15723270 0.14772727
```

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

Question 5: Using the distance matrices from above, answer the following questions:

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

b. Compare the resemblance matrices (`fish.db` or `fish.ds`) you just created. How does the choice of the SÃ¸rensen or Bray-Curtis distance influence your interpretation of site (dis)similarity?

Answer 5a:The resemblance matrix represents dissimilarity. One can see that the resemblance matrix represents dissimilarity because the values comparing sites 1 and 2, and sites 1 and 10 were equal to the Whittaker's beta diversity values (which measured dissimilarity) for these sites.
Answer 5b:The Sorensen index puts a greater emphasis on the number of shared species or the similarity of two sites compared to the Bray-Curtis index. In other words, the similariy could be inflated in the Sorensen index.

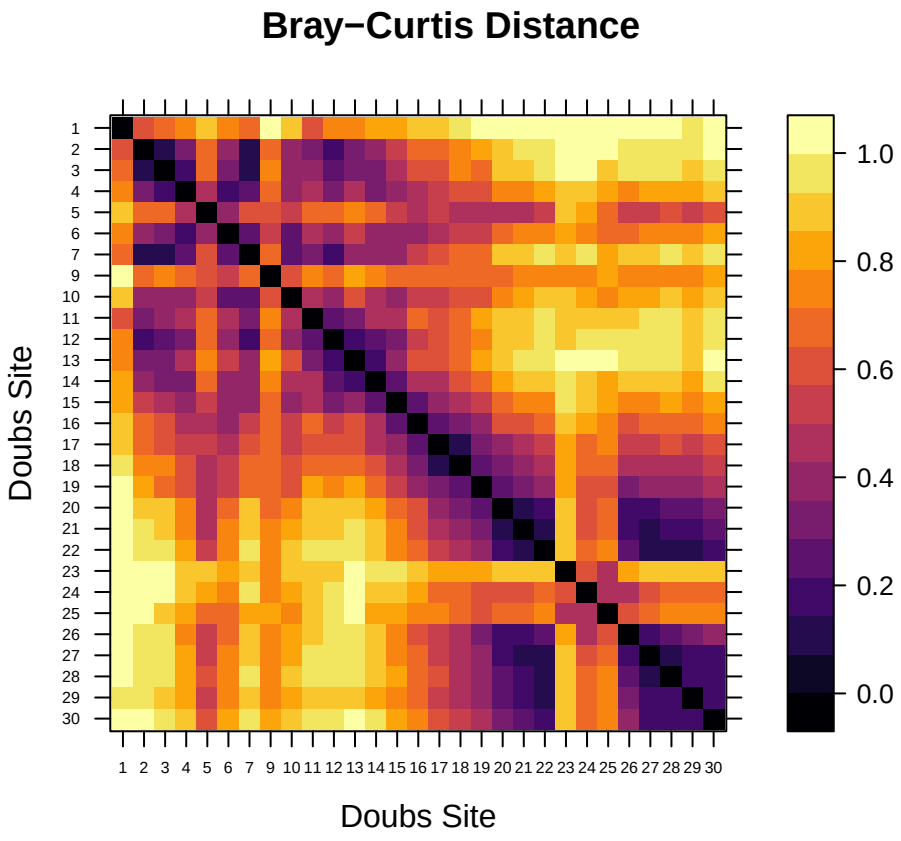
4) VISUALIZING BETA-DIVERSITY

A. Heatmaps

In the R code chunk below, do the following:

1. define a color palette,
2. define the order of sites in the Doubs River, and
3. use the `levelplot()` function to create a heatmap of fish abundances in the Doubs River.

```
# 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")
```

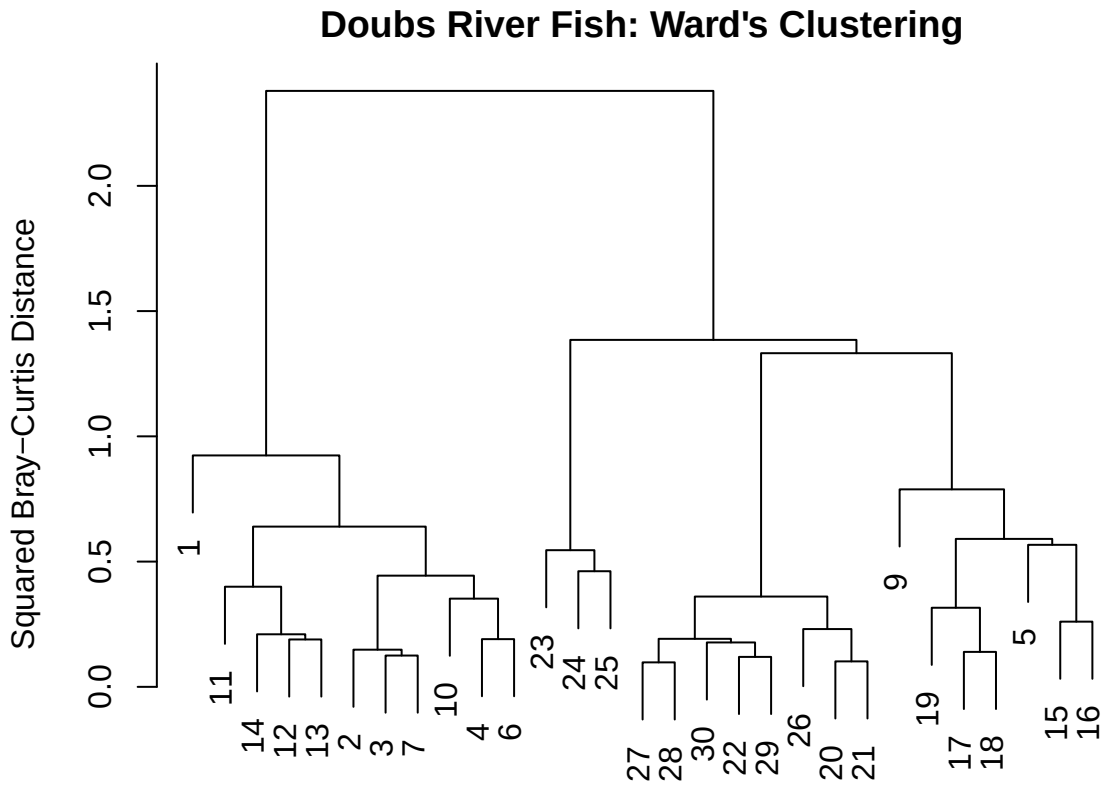


B. Cluster Analysis

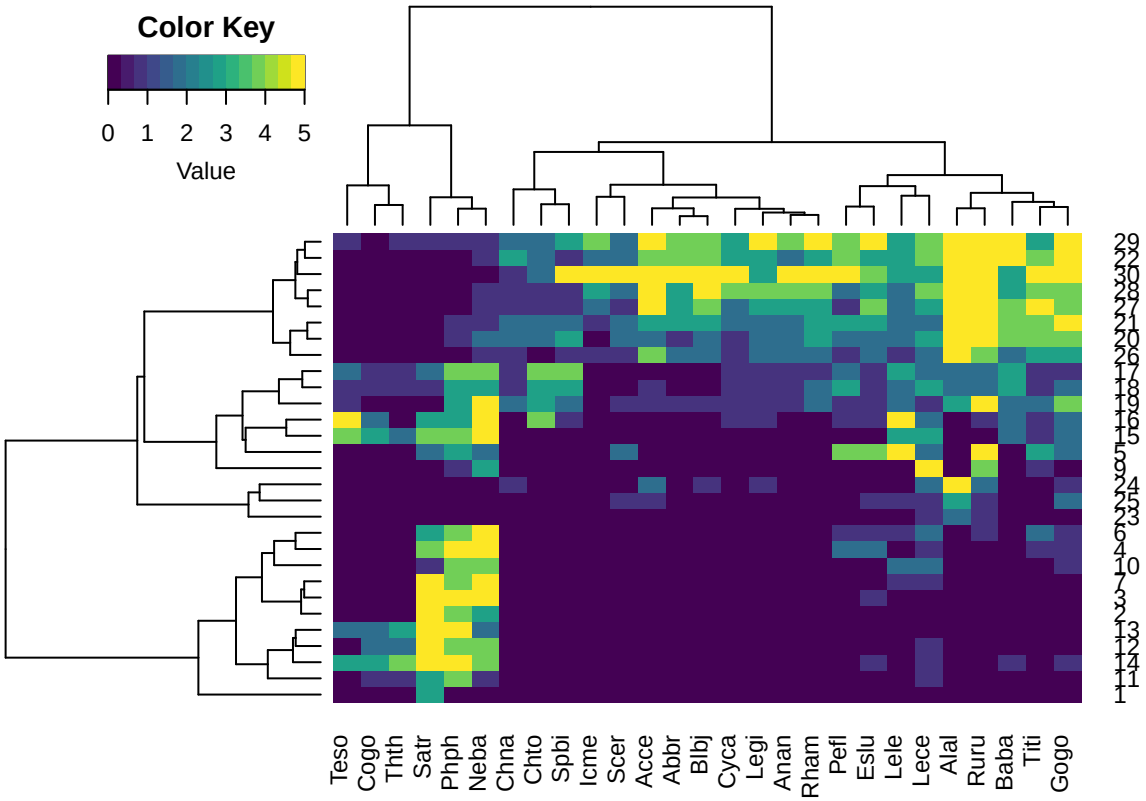
In the R code chunk below, do the following:

1. perform a cluster analysis using Ward's Clustering, and
2. plot your cluster analysis (use either `hclust` or `heatmap.2`).

```
# 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")
```




```
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 an ecological hypothesis for fish diversity the `doubs` data set?

> ***Answer 6***:The upstream sites have lower diversity and abundance compared to sites further downstream. Perhaps some environmental factor such as stream flow or the presence of a pollutant may have caused this difference. However, based on the data provided, one cannot discern the cause of this hypothesized difference in diversity and abundance.

C. Ordination

Principal Coordinates Analysis (PCoA)

In the R code chunk below, do the following:

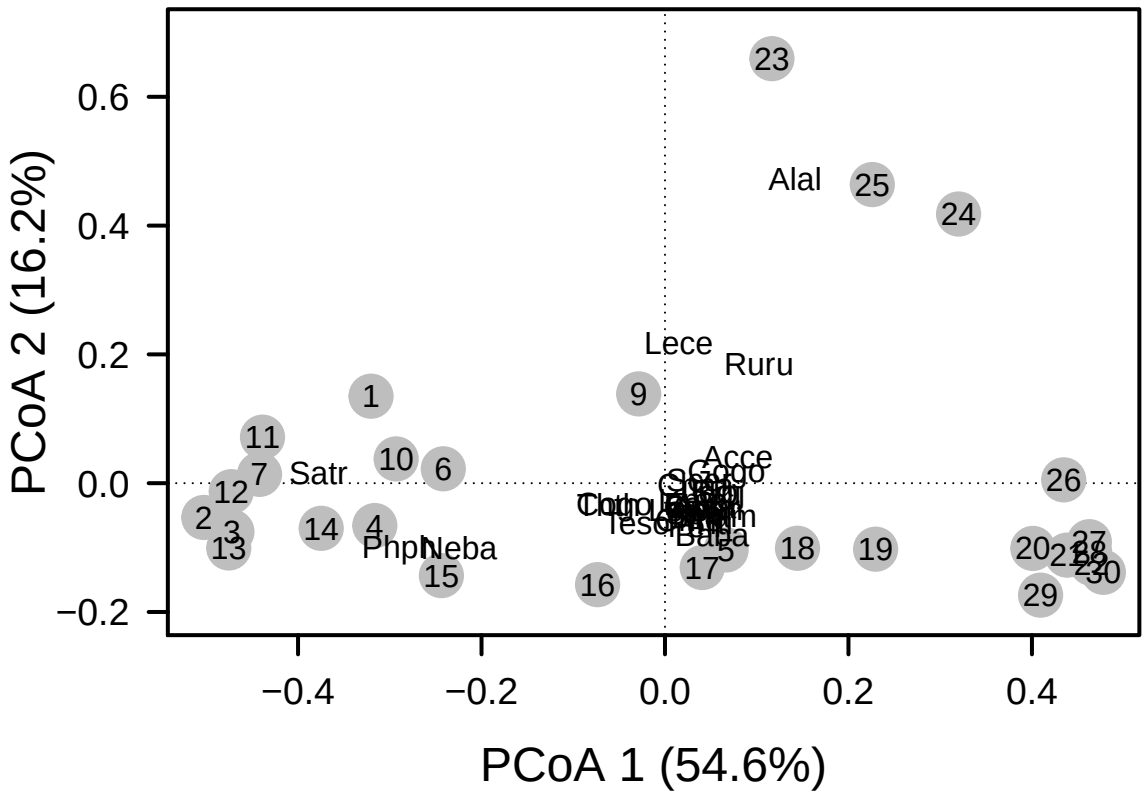
1. perform a Principal Coordinates Analysis to visualize beta-diversity
2. calculate the variation explained by the first three axes in your ordination
3. plot the PCoA ordination,
4. label the sites as points using the Doubs River site number, and
5. identify influential species and add species coordinates to PCoA plot.

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

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)
```

```
dev.off
```

```
## function (which = dev.cur())
## {
##     if (which == 1)
##         stop("cannot shut down device 1 (the null device)")
##     .External(C_devoff, as.integer(which))
##     dev.cur()
## }
## <bytecode: 0x000000000a2ffaa8>
## <environment: namespace:grDevices>
```

```
# 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))

# 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, ])
}

# 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")
```



In the R code chunk below, do the following:

1. identify influential species based on correlations along each PCoA axis (use a cutoff of 0.70), and
2. use a permutation test (999 permutations) to test the correlations of each species along each axis.

```
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, ]
print(imp.spp)
```

```
##            Dim1       Dim2        Dim3
## Phph -0.8674640 -0.1699316 -0.12463098
## Neba -0.7674114 -0.1855678 -0.36963830
## Rham  0.8088751 -0.4192567  0.14136301
## Legi  0.8201759 -0.1701803  0.12423941
## Cyca  0.7595122 -0.4442926  0.17313658
## Abbr  0.7704744 -0.3452714  0.29277803
## Acce  0.7635195  0.2155765  0.10288179
## Blbj  0.8118483 -0.1324698  0.25581178
## Alal  0.4471283  0.8119843 -0.05167131
## Anan  0.7974122 -0.3918972  0.20944968
```

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

```
##            Dim1       Dim2        Dim3
## Phph -0.8674640 -0.1699316 -0.12463098
```

```
## Neba -0.7674114 -0.1855678 -0.36963830
## Rham  0.8088751 -0.4192567  0.14136301
## Legi  0.8201759 -0.1701803  0.12423941
## Cyca  0.7595122 -0.4442926  0.17313658
## Abbr  0.7704744 -0.3452714  0.29277803
## Acce  0.7635195  0.2155765  0.10288179
## Blbj  0.8118483 -0.1324698  0.25581178
## Alal  0.4471283  0.8119843 -0.05167131
## Anan  0.7974122 -0.3918972  0.20944968
```

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

a. Describe 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.

Answer 7a:Based on the PCoA results, the downstream sites were clustered together, and the upstream sites were clustered together. ***Answer 7b***:Phph, Neba, Rham, Legi, Cyca, Abbr, Acce, Blbj, Alal, Anan are potential indicators of water quality.

5) HYPOTHESIS TESTING

A. Multivariate Procedures for Categorical Designs

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".

In the code chunk below, test the hypothesis that fish community composition varies with river quality.

1. create a factor vector that categorizes habitat quality in the Doubs River,
2. use the multivariate analyses for categorical predictors to describe how fish community structure relates to habitat 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)
```

```
##
## Call:
## adonis(formula = fish ~ quality, permutations = 999, method = "bray")
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## quality    2    3.0947 1.54733   10.97 0.45765  0.001 ***
## Residuals 26    3.6674 0.14105         0.54235
## Total     28    6.7621                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#IndVal Scores
indval <- multipatt(fish,cluster =quality,func ="IndVal.g",control = how(nperm=999))
summary(indval)
```

```
##
##  Multilevel pattern analysis
##  ---------------------------
##
##  Association function: IndVal.g
##  Significance level (alpha): 0.05
##
##  Total number of species: 27
##  Selected number of species: 23
##  Number of species associated to 1 group: 1
##  Number of species associated to 2 groups: 22
##
##  List of species associated to each combination:
##
##  Group MQ  #sps.  1
##       stat p.value
## Teso 0.686   0.016 *
##
##  Group HQ+MQ  #sps.  2
##       stat p.value
## Satr 0.860   0.004 **
## Phph 0.859   0.012 *
##
##  Group LQ+MQ  #sps.  20
##       stat p.value
## Alal 0.935   0.001 ***
## Gogo 0.933   0.001 ***
## Ruru 0.916   0.001 ***
## Legi 0.901   0.001 ***
## Baba 0.895   0.001 ***
## Chna 0.866   0.001 ***
## Spbi 0.866   0.001 ***
## Cyca 0.866   0.001 ***
## Acce 0.866   0.001 ***
## Lele 0.863   0.007 **
## Titi 0.853   0.005 **
## Chto 0.829   0.002 **
## Rham 0.829   0.001 ***
## Anan 0.829   0.001 ***
## Eslu 0.827   0.035 *
## Pefl 0.806   0.019 *
## Blbj 0.791   0.003 **
## Scer 0.766   0.009 **
## Abbr 0.750   0.007 **
## Icme 0.661   0.023 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Phi Coefficient of Association
fish.rel <- decostand(fish,method ="total")
```

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

```
##
##  Multilevel pattern analysis
##  ---------------------------
##
##  Association function: r.g
##  Significance level (alpha): 0.05
##
##  Total number of species: 27
##  Selected number of species: 18
##  Number of species associated to 1 group: 9
##  Number of species associated to 2 groups: 9
##
##  List of species associated to each combination:
##
##  Group HQ  #sps.  3
##       stat p.value
## Phph 0.802   0.001 ***
## Neba 0.734   0.001 ***
## Satr 0.650   0.001 ***
##
##  Group LQ  #sps.  2
##       stat p.value
## Alal 0.693   0.001 ***
## Ruru 0.473   0.031 *
##
##  Group MQ  #sps.  4
##       stat p.value
## Anan 0.571   0.006 **
## Spbi 0.557   0.012 *
## Chto 0.542   0.016 *
## Icme 0.475   0.036 *
##
##  Group LQ+MQ  #sps.  9
##       stat p.value
## Legi 0.658   0.003 **
## Baba 0.645   0.004 **
## Rham 0.600   0.007 **
## Acce 0.594   0.007 **
## Cyca 0.586   0.008 **
## Chna 0.571   0.003 **
## Blbj 0.571   0.007 **
## Gogo 0.523   0.016 *
## Abbr 0.499   0.030 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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***: The PERMANOVA indicates that community composition varies significantly with water quality. The IndVal test suggests that Teso is an indicator species for the medium quality region, and Satr and Phph are indicator species of the high and middle water quality regions.

Further, Alal, Gogo, Ruru, Legi, Baba, Chna, Spbi, Cyca, Acce, Lele, Titi, Chto, Rham, Anan, Eslu, Pefl, Blbj, Scer, Abbr, and Icme are indicators for the low and medium water quality regions. In other words, this analysis provided indicator species for each region. Similarly, the phi coefficient analysis indicates that Phph, Neba, and Satr have a strong preference for the high quality region. Alal and Ruru have a strong preference for the low quality region, and Anan, Spbi, Chto, and Icme have a strong preference for the medium quality region. Overall, these analyses are suggesting that fish species composition depends on water quality.

B. Multivariate Procedures for Continuous Designs

i. Mantel Test

In the R code chunk below, do the following:

1. create distance matrices for both fish communities and environmental factors, and
2. use a Mantel test to determine if these matrices are correlated, and 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)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = fish.dist, ydis = env.dist)
##
## Mantel statistic r: 0.604
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.104 0.137 0.164 0.192
## Permutation: free
## Number of permutations: 999
```

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

> ***Answer 9***:The Mantel test suggests that fish diversity is significantly correlated with stream environmental variables, which supports the hypothesis that stream quality influences fish communities.

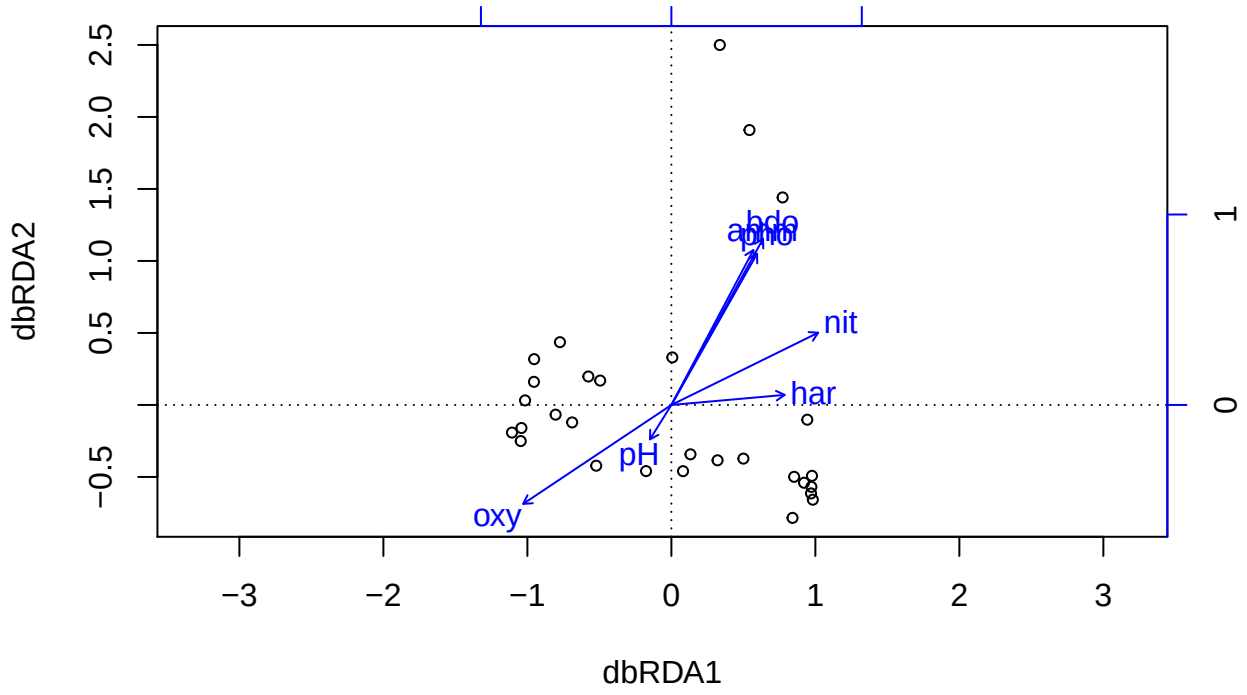
ii. Constrained Ordination

In the R code chunk below, do the following:

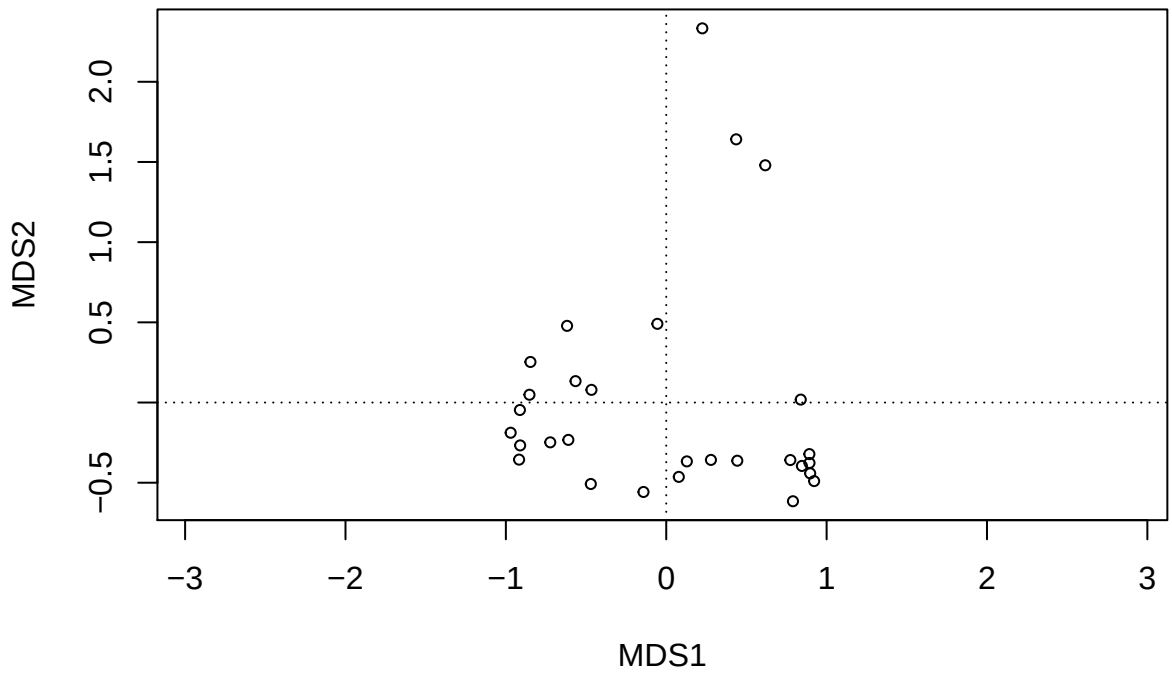
1. create an environmental matrix of the water chemistry data included in the `doubs` dataset using forward and reverse selection of variables,
2. conduct a redundancy analysis on the fish assemblages of the Doubs River,
3. use a permutation test to determine the significance of the constrained analysis,

4. use a permutation test to determine the correlation of each environmental factor on the constrained axes,
5. calculate the explained variation on the first and second constrained axes,
6. plot the constrained ordination results including labeled points for each site, and
7. add vectors that demonstrate the influence of each environmental factor the constrained ordination.

```
# 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)
```




```
# Model 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)
# Therefore, the axes suggest this is a simple MDS (i.e., PCoA)
ordiplot(doubs.dbrda.mod0)
```




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

# 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)
```

```
## Step: R2.adj= 0
## Call: fish.db ~ 1
##
##                  R2.adjusted
## <All variables>  0.53032584
## + oxy            0.27727176
## + nit            0.25755208
## + bdo            0.17477787
## + pho            0.14568614
## + har            0.14174915
## + amm            0.14142804
## <none>           0.00000000
## + pH            -0.01827054
##
##       Df    AIC      F Pr(>F)
## + oxy  1 47.939 11.742  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Step: R2.adj= 0.2772718
## Call: fish.db ~ oxy
##
##                 R2.adjusted
## <All variables>   0.5303258
## + bdo             0.4009000
## + amm             0.3474192
## + pho             0.3452702
## + har             0.3331357
## + nit             0.3316120
## <none>            0.2772718
## + pH              0.2586983
## - oxy             0.0000000
##
##       Df    AIC      F Pr(>F)
## + bdo  1 43.404 6.5716  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4009
## Call: fish.db ~ oxy + bdo
##
##                 R2.adjusted
## <All variables>   0.5303258
## + nit             0.4980793
## + har             0.4695121
## <none>            0.4009000
## + pho             0.3938042
## + amm             0.3869134
## + pH              0.3865240
## - bdo             0.2772718
## - oxy             0.1747779
##
##       Df    AIC     F Pr(>F)
## + nit  1 39.134 6.034  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4980793
## Call: fish.db ~ oxy + bdo + nit
##
##                 R2.adjusted
## + amm             0.5415705
## <All variables>   0.5303258
## + pho             0.5277128
## + har             0.5218852
## <none>            0.4980793
## + pH              0.4843267
## - nit             0.4009000
## - oxy             0.3420426
## - bdo             0.3316120
```

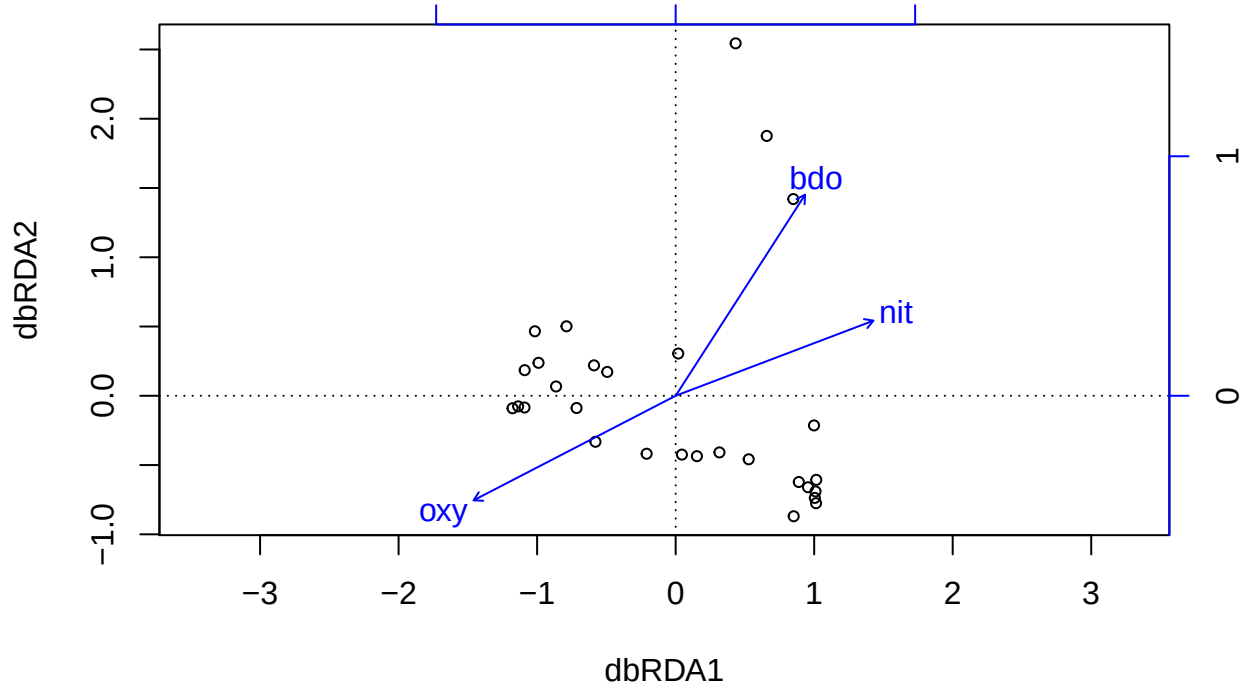
```
doubs.dbrda$call
```

```
## dbrda(formula = fish.db ~ oxy + bdo + nit, data = as.data.frame(env.chem))
```

```
doubs.dbrda$anova
```

```
##                  R2.adj Df    AIC       F Pr(>F)
## + oxy           0.27727  1 47.939 11.7421  0.002 **
## + bdo           0.40090  1 43.404  6.5716  0.002 **
## + nit           0.49808  1 39.134  6.0340  0.002 **
## <All variables> 0.53033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ordiplot(doubs.dbrda)
```




```
# Permutation tests to evaluate significance
permutest(doubs.dbrda,permutations =999)
```

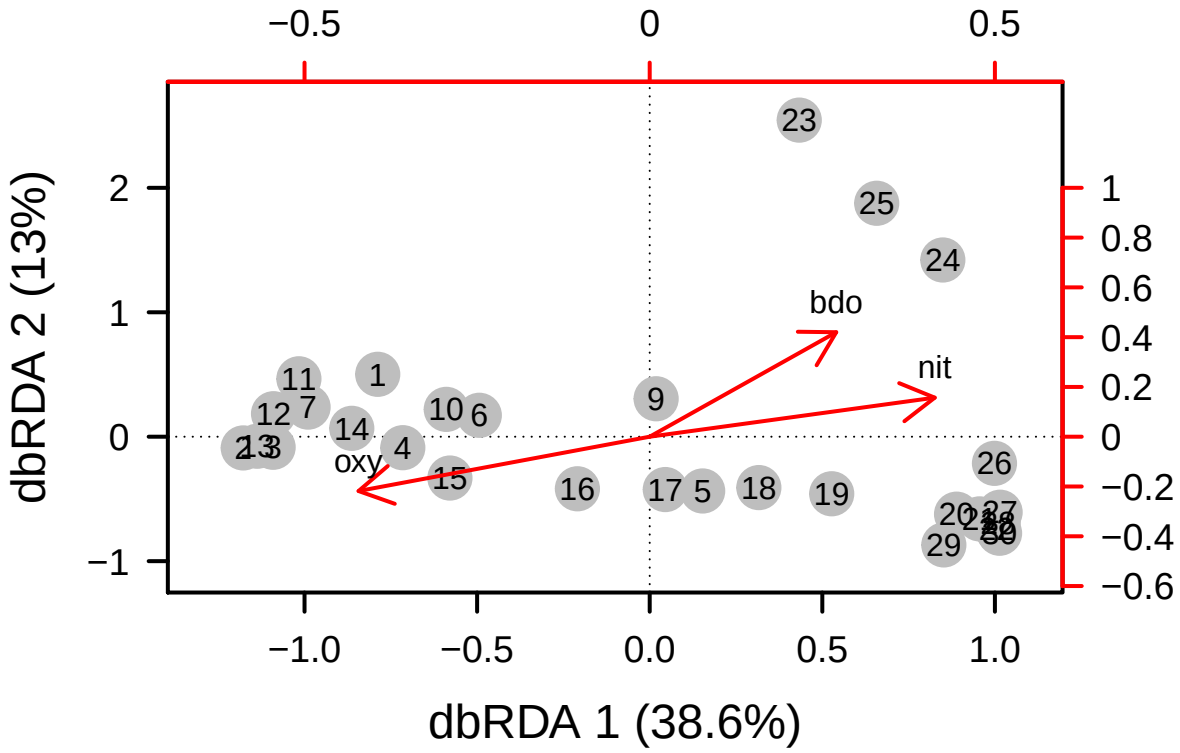
```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ oxy + bdo + nit, data =
## as.data.frame(env.chem))
## Permutation test for all constrained eigenvalues
## Pseudo-F:     10.2619 (with 3, 25 Degrees of Freedom)
## Significance:     0.001
```

```r
envfit(doubs.dbrda, env.chem[,c(4,6,7)],perm =999)
```

```
##
## ***VECTORS
##
##       dbRDA1   dbRDA2     r2 Pr(>r)
## nit  0.87724  0.48005 0.6431  0.001 ***
## oxy -0.82864 -0.55979 0.7656  0.001 ***
## bdo  0.55603  0.83116 0.8939  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```r
# 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

#Plot the ordination for the selected model.

# 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***: Oxygen, bdo, and nitrogen appear to contribute to variation in fish community structure according to the constrained ordination.

iii. Variation Partitioning

In the code chunk below,

1. Create a matrix model of the selected environmental variables,
2. Create a matrix model of the selected PCNM axes,
3. Perform constrained and partial constrained ordinations using the spatial and environmental models you just created,
4. Test the significance of each of your constrained ordinations using permutation tests,
5. Partition the variation among sites into the relative importance of space, environment, spatially structured environment, and residuals,
6. Plot the variation partitioning output to visualize it.

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

```
##                 R2.adj Df    AIC       F Pr(>F)
## + oxy          0.27727  1 47.939 11.7421  0.002 **
## + bdo          0.40090  1 43.404  6.5716  0.002 **
```

```
## + nit           0.49808  1 39.134  6.0340  0.002 **
## <All variables> 0.53033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

# Weight each site by its relative abundance
rs <- rowSums(fish)/sum(fish)

# 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
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## [23] FALSE FALSE FALSE FALSE
```

```
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)
```

```
## Step: R2.adj= 0
## Call: fish.db ~ 1
##
##                  R2.adjusted
## <All variables>  0.626011301
## + PCNM2          0.235370423
## + PCNM3          0.078394885
## + PCNM13         0.065305668
## + PCNM5          0.046185074
## + PCNM6          0.032809156
## + PCNM16         0.030486700
## + PCNM14         0.029680999
## + PCNM9          0.020357410
## + PCNM15         0.013632610
## + PCNM8          0.009411968
## + PCNM1          0.003986221
## + PCNM17         0.002415012
## + PCNM10         0.001326442
## <none>           0.000000000
## + PCNM7         -0.001861430
## + PCNM11        -0.006841522
## + PCNM4         -0.007089863
## + PCNM12        -0.014396973
##
##         Df    AIC     F Pr(>F)
## + PCNM2  1 49.574 9.619  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

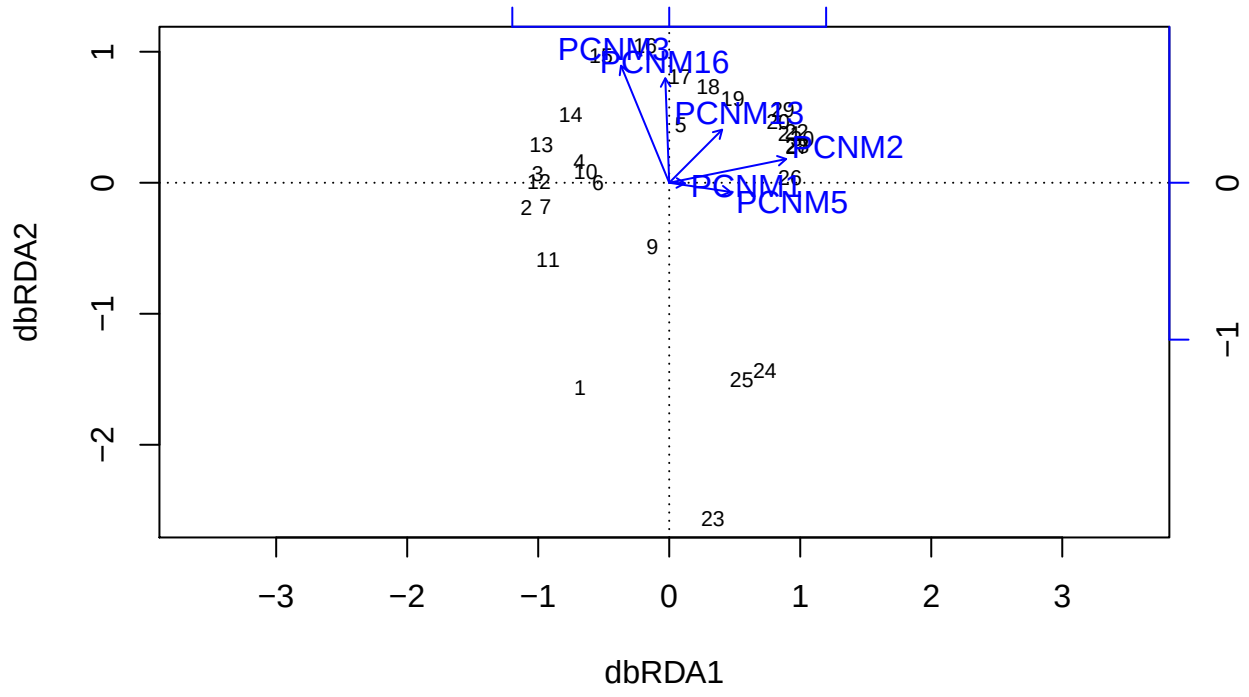
```
## Step: R2.adj= 0.2353704
## Call: fish.db ~ PCNM2
##
##                 R2.adjusted
## <All variables>   0.6260113
## + PCNM3           0.3429270
## + PCNM5           0.3057368
## + PCNM1           0.2885396
## + PCNM16          0.2786746
## + PCNM14          0.2744520
## + PCNM15          0.2692809
## + PCNM6           0.2659866
## + PCNM13          0.2636194
## + PCNM9           0.2517847
## + PCNM8           0.2496240
## + PCNM10          0.2434688
## + PCNM7           0.2431476
## + PCNM17          0.2404343
## + PCNM11          0.2366833
## <none>            0.2353704
## + PCNM12          0.2288789
## + PCNM4           0.2189522
## - PCNM2           0.0000000
##
##         Df    AIC      F Pr(>F)
## + PCNM3  1 46.083 5.4196  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.342927
## Call: fish.db ~ PCNM2 + PCNM3
##
##                 R2.adjusted
## <All variables>  0.62601130
## + PCNM5           0.40760197
## + PCNM1           0.39703000
## + PCNM16          0.38532100
## + PCNM15          0.38287481
## + PCNM14          0.37818268
## + PCNM13          0.37703761
## + PCNM6           0.35956442
## + PCNM8           0.35568849
## + PCNM7           0.35416308
## + PCNM10          0.35267745
## + PCNM17          0.35136832
## + PCNM9           0.34336720
## <none>            0.34292704
## + PCNM11          0.34163988
## + PCNM12          0.33965471
## + PCNM4           0.33115086
## - PCNM3           0.23537042
## - PCNM2           0.07839489
##
##         Df    AIC      F Pr(>F)
```

```
## + PCNM5  1 43.941 3.8385  0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.407602
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5
##
##                 R2.adjusted
## <All variables>   0.6260113
## + PCNM1           0.4721469
## + PCNM16          0.4631976
## + PCNM15          0.4589111
## + PCNM14          0.4535248
## + PCNM13          0.4511582
## + PCNM6           0.4305640
## + PCNM7           0.4261965
## + PCNM8           0.4224505
## + PCNM17          0.4181666
## + PCNM10          0.4154485
## + PCNM11          0.4112178
## + PCNM9           0.4111995
## + PCNM12          0.4087602
## <none>            0.4076020
## + PCNM4           0.3976526
## - PCNM5           0.3429270
## - PCNM3           0.3057368
## - PCNM2           0.1195237
##
##         Df    AIC     F Pr(>F)
## + PCNM1  1 41.411 4.057  0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.4721469
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1
##
##                 R2.adjusted
## <All variables>   0.6260113
## + PCNM13          0.5212427
## + PCNM16          0.5208668
## + PCNM15          0.5161770
## + PCNM14          0.5147355
## + PCNM6           0.4999020
## + PCNM7           0.4936559
## + PCNM8           0.4904113
## + PCNM17          0.4856884
## + PCNM10          0.4835952
## + PCNM11          0.4760087
## + PCNM9           0.4751424
## + PCNM12          0.4747221
## <none>            0.4721469
## + PCNM4           0.4651218
## - PCNM1           0.4076020
## - PCNM5           0.3970300
```

```
## - PCNM3           0.3691841
## - PCNM2           0.1269210
##
##          Df    AIC      F Pr(>F)
## + PCNM13  1 39.346 3.4612  0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5212427
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13
##
##                 R2.adjusted
## <All variables>   0.6260113
## + PCNM16          0.5767968
## + PCNM15          0.5715331
## + PCNM14          0.5698343
## + PCNM6           0.5475140
## + PCNM7           0.5392074
## + PCNM8           0.5379134
## + PCNM11          0.5281106
## + PCNM9           0.5267003
## + PCNM10          0.5265029
## + PCNM12          0.5255581
## <none>            0.5212427
## + PCNM17          0.5171800
## + PCNM4           0.5152311
## - PCNM13          0.4721469
## - PCNM1           0.4511582
## - PCNM5           0.4350790
## - PCNM3           0.4111185
## - PCNM2           0.2307026
##
##          Df   AIC      F Pr(>F)
## + PCNM16  1 36.48 4.0192   0.01 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.5767968
## Call: fish.db ~ PCNM2 + PCNM3 + PCNM5 + PCNM1 + PCNM13 + PCNM16
##
##                 R2.adjusted
## <All variables>   0.6260113
## + PCNM6           0.6043089
## + PCNM8           0.5970286
## + PCNM12          0.5946888
## + PCNM7           0.5946475
## + PCNM9           0.5883735
## + PCNM10          0.5851333
## + PCNM15          0.5846468
## <none>            0.5767968
## + PCNM17          0.5748533
## + PCNM4           0.5733749
## + PCNM11          0.5711176
## + PCNM14          0.5652509
```

```
## - PCNM16          0.5212427
## - PCNM13          0.5208668
## - PCNM1           0.5136241
## - PCNM5           0.4764463
## - PCNM3           0.4676690
## - PCNM2           0.2646853
##
##         Df    AIC      F Pr(>F)
## + PCNM6  1 35.182 2.5296  0.054 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 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
```

```
##                 R2.adj Df    AIC      F Pr(>F)
## + PCNM2         0.23537  1 49.574 9.6190  0.002 **
## + PCNM3         0.34293  1 46.083 5.4196  0.004 **
## + PCNM5         0.40760  1 43.941 3.8385  0.016 *
## + PCNM1         0.47215  1 41.411 4.0570  0.016 *
## + PCNM13        0.52124  1 39.346 3.4612  0.016 *
## + PCNM16        0.57680  1 36.480 4.0192  0.010 **
```

```
## <All variables> 0.62601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

# 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))

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

```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ env.mod + Condition(space.mod))
## Permutation test for all constrained eigenvalues
## Pseudo-F:     4.423025 (with 3, 18 Degrees of Freedom)
## Significance:     0.001
```

```
permutest(doubs.space.cond.env,permutations =999)
```

```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ space.mod + Condition(env.mod))
## Permutation test for all constrained eigenvalues
## Pseudo-F:     4.174109 (with 7, 18 Degrees of Freedom)
## Significance:     0.001
```

```
permutest(doubs.total.env,permutations =999)
```

```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ env.mod)
## Permutation test for all constrained eigenvalues
## Pseudo-F:     10.2619 (with 3, 25 Degrees of Freedom)
## Significance:     0.001
```

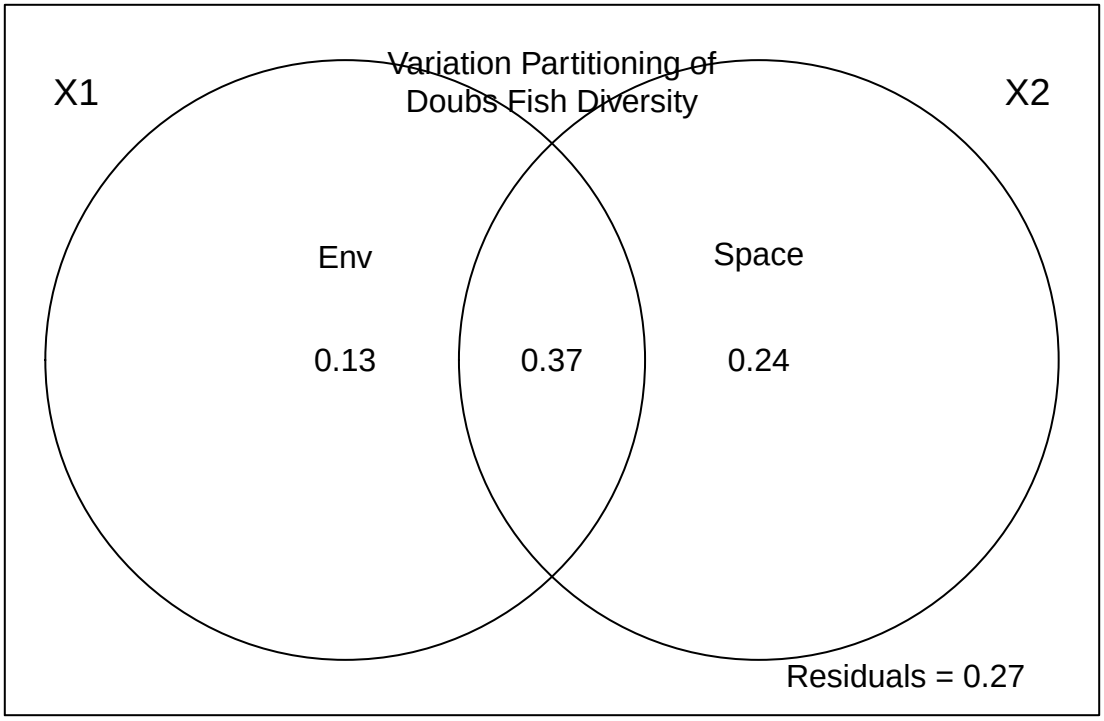
```r
permutest(doubs.total.space,permutations =999)
```

```
##
## Permutation test for dbrda
##
## Permutation: free
## Number of permutations: 999
##
## Call: dbrda(formula = fish.db ~ space.mod)
## Permutation test for all constrained eigenvalues
## Pseudo-F:     7.108896 (with 7, 21 Degrees of Freedom)
## Significance:     0.001
```

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

```
##
## Partition of squared Bray distance in dbRDA
##
## Call: varpart(Y = fish.db, X = env.mod, space.mod)
##
## Explanatory tables:
## X1:  env.mod
## X2:  space.mod
##
## No. of explanatory tables: 2
## Total variation (SS): 6.7621
## No. of observations: 29
##
## Partition table:
##                       Df R.squared Adj.R.squared Testable
## [a+b] = X1             3   0.55186       0.49808     TRUE
## [b+c] = X2             7   0.70323       0.60431     TRUE
## [a+b+c] = X1+X2       10   0.82917       0.73426     TRUE
## Individual fractions
## [a] = X1|X2            3                 0.12995     TRUE
## [b]                    0                 0.36813    FALSE
## [c] = X2|X1            7                 0.23618     TRUE
## [d] = Residuals                          0.26574    FALSE
## ---
## Use function 'capscale' to test significance of fractions of interest
```

```r
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)
```



Question 11: Interpret the variation partitioning results.

> ***Answer 11***: 13% of the variation in diversity by site can be attributed to the environment, or unique environmental variation. 37% of the variation in fish diversity can be attributed to space and the environment, or spacially structured environmental variation. Further, 24% of the variation in diverstiy can be attributed to space, or unique spatial variation.

SYNTHESIS

Load the dataset you are using for your project. Perform an ordination to visualize your dataset. Using this ordination, develop some hypotheses relevant to β-diversity and identify the appropriate tools you would use to test them.

```
fungi.data <- read.csv("finalprojectdata1csv.csv", sep = ",", header = TRUE)


fungi.data <- fungi.data[,-1]


#Bray-Curtis
fungi.db <- vegdist(fungi.data, method = "bray")

#Sorensen
fungi.ds <- vegdist(fungi.data, method = "bray", binary = TRUE)

#Bray-Curtis
```

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

#Sorensen
fungi.pcoa.ds <- cmdscale(fungi.ds, eig = TRUE, k = 3)

explainvar1.db <- round(fungi.pcoa.db$eig[1] / sum(fungi.pcoa.db$eig), 3) * 100
explainvar2.db <- round(fungi.pcoa.db$eig[2] / sum(fungi.pcoa.db$eig), 3) * 100
explainvar3.db <- round(fungi.pcoa.db$eig[3] / sum(fungi.pcoa.db$eig), 3) * 100
33
```

```
## [1] 33
```

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

#Ordination Plot (Bray-Curtis)
par(mar = c(5,5,1,2) + 0.1)

#Initiate Plot
plot(fungi.pcoa.db$points[ ,1], fungi.pcoa.db$points[ ,2],
xlab = paste("PCoA 1 (", explainvar1.db, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar2.db, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE, ylim = c(-.4,.4), xlim = c

#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 and Labels
points(fungi.pcoa.db$points[,1], fungi.pcoa.db$points[,2],
pch = 19, cex = 1.5, bg = "gray", col = "blue")

text(fungi.pcoa.db$points[,1], fungi.pcoa.db$points[,2],
labels = row.names(fungi.pcoa.db$points))
fungi.pcoa.db <- add.spec.scores(fungi.pcoa.db,fungi.data,method = "pcoa.scores")
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

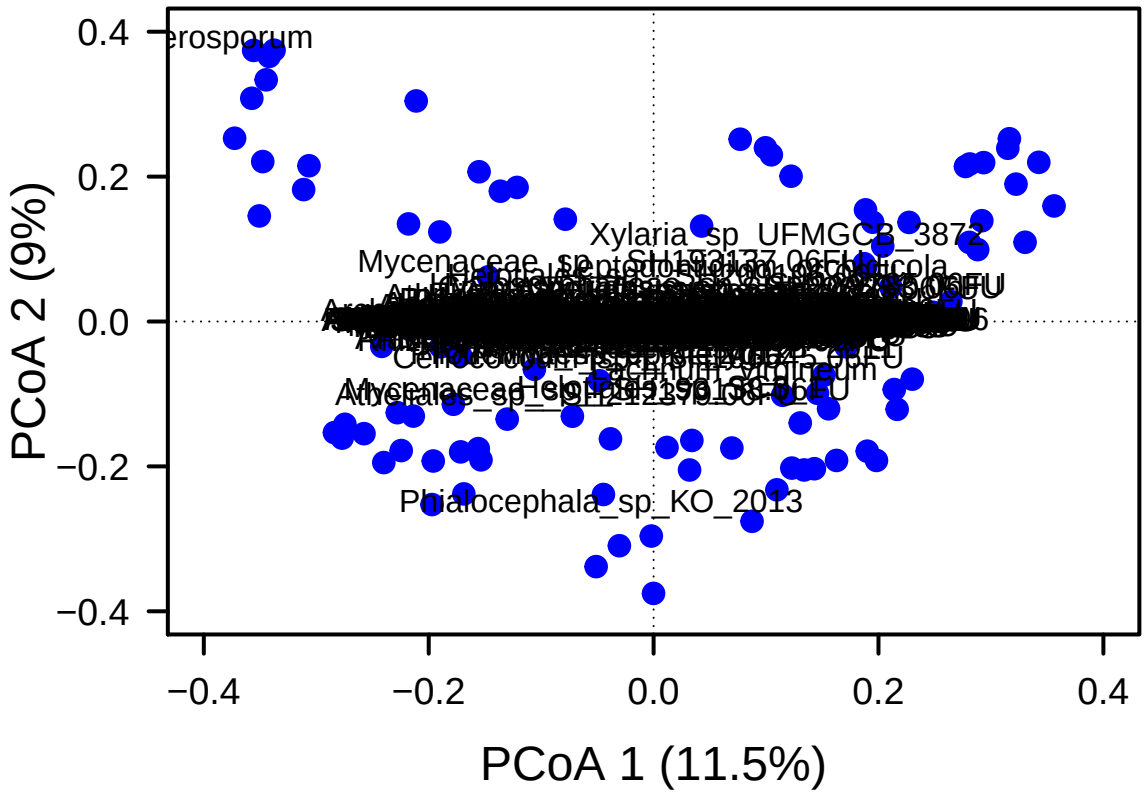
```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
#Calculate and add species scores- did not calculate relative abundances because they were already calc

text(fungi.pcoa.db$cproj[ ,1], fungi.pcoa.db$cproj[, 2],
labels = row.names(fungi.pcoa.db$cproj), col = "black")
```



>The Bray-Curtis index would be an appropriate test for our dataset because it has relative abundances of endophyte OTUs at each sampling site. This ordination plot (Bray-Curtis index version) shows most of the points clustered in the middle of the plot. There do not appear to be other groupings of data points on this plot aside from several points towards the bottom of the plot, and several in the top left. The x and y axes of this plot only account for 11.5% and 9% of the variation in the data. Overall, it is difficult to discern whether there are any patterns based on this plot because the points are all clustered together in the middle of the plot.

```
#Ordination Plot (Sorensen)

fungi.pcoa.ds <- cmdscale(fungi.ds, eig = TRUE, k = 3)
```

```r
#Quantify Percent Variation
explainvar1.ds <- round(fungi.pcoa.ds$eig[1] / sum(fungi.pcoa.ds$eig), 3) * 100
explainvar2.ds <- round(fungi.pcoa.ds$eig[2] / sum(fungi.pcoa.ds$eig), 3) * 100
explainvar3.ds <- round(fungi.pcoa.ds$eig[3] / sum(fungi.pcoa.ds$eig), 3) * 100
sum.eig.fungi.ds <- sum(explainvar1.ds, explainvar2.ds, explainvar3.ds)


#Ordination Plot (Sorensen)

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

#Inidiate Plot
plot(fungi.pcoa.ds$points[ ,1], fungi.pcoa.ds$points[ ,2],
xlab = paste("PCoA 1 (", explainvar1.ds, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar2.ds, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE, ylim = c(-.4,.4), xlim = 

#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)

#Points and Labels
points(fungi.pcoa.ds$points[,1], fungi.pcoa.ds$points[,2],
pch = 19, cex = 1.5, bg = "gray", col = "blue")

text(fungi.pcoa.ds$points[,1], fungi.pcoa.ds$points[,2],
labels = row.names(fungi.pcoa.ds$points))

fungi.pcoa.ds <- add.spec.scores(fungi.pcoa.ds,fungi.data,method = "pcoa.scores")
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

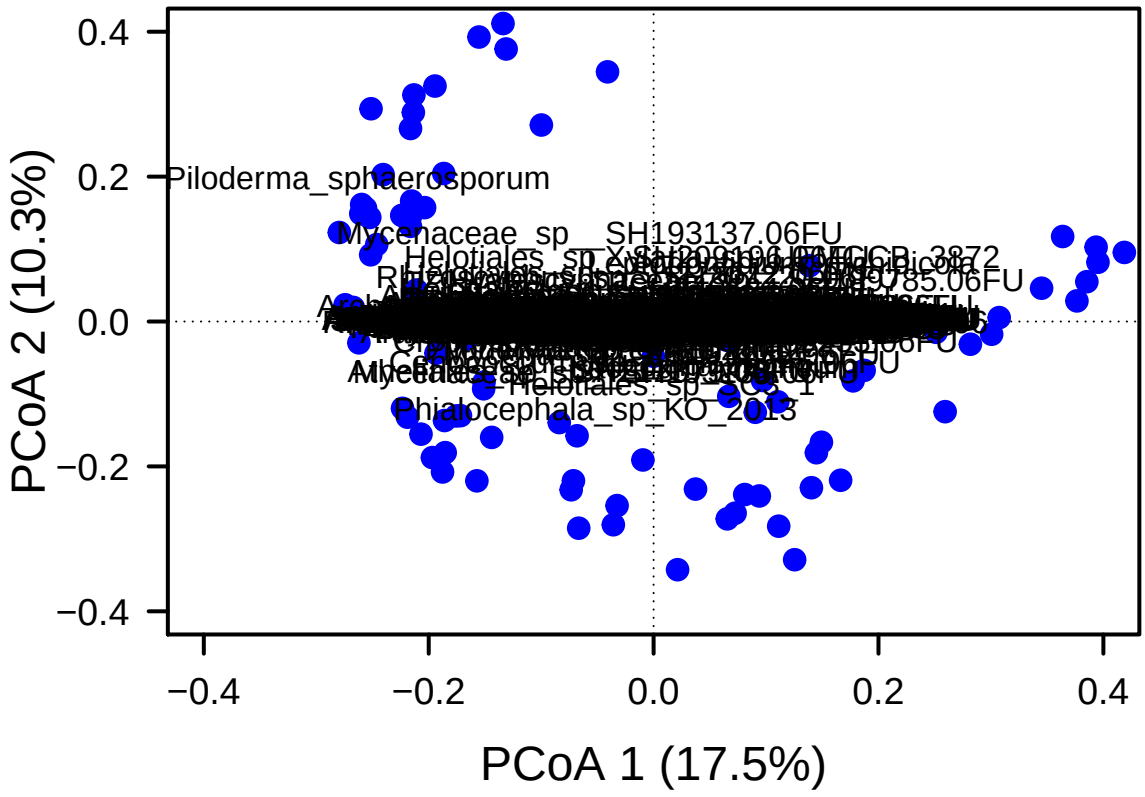
```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
text(fungi.pcoa.ds$cproj[ ,1], fungi.pcoa.ds$cproj[, 2],
labels = row.names(fungi.pcoa.ds$cproj), col = "black")
```



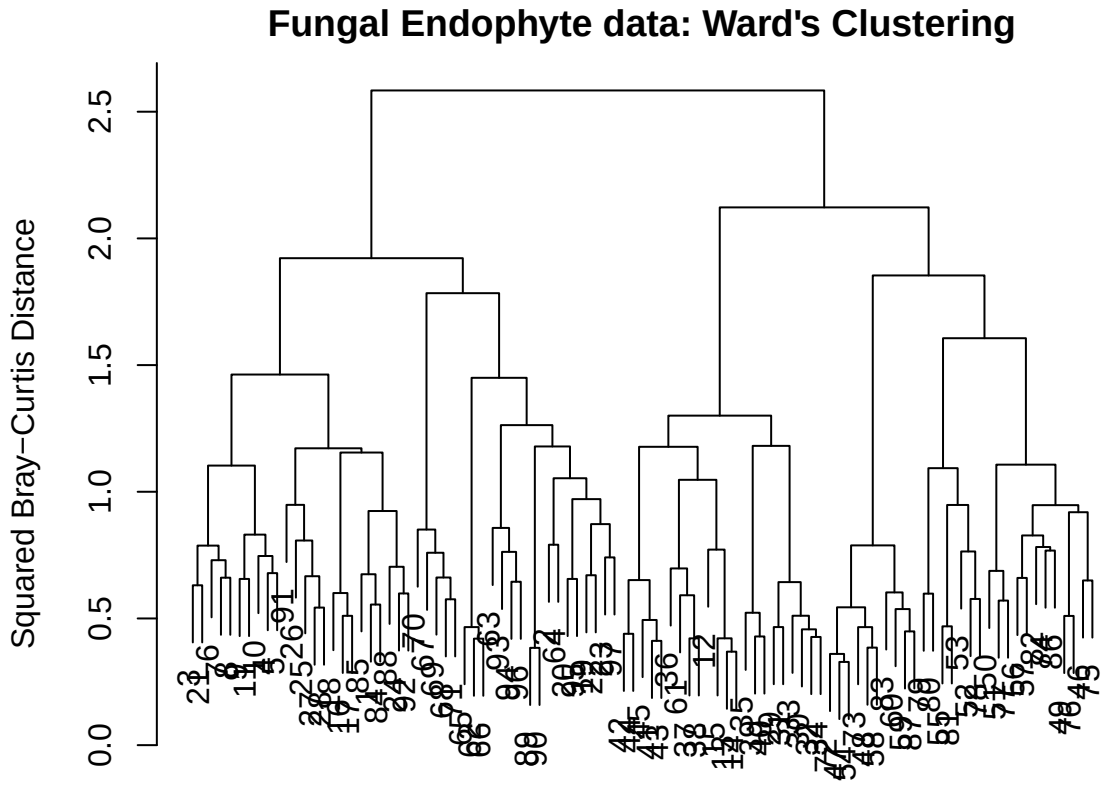
>Although the Bray-Curtis index version of the ordination plot seems to be a better fit for our dataset (because it measures relative abundances), I did the Sorensen version as well to see whether any different patterns would arise. However, similar to the other ordination plot, most of the points are clustered in the center of the plot again. The x and y axes on this plot explain 17.5% and 10.3% of the variation in the data, which is greater than that of the first plot.The first plot cut off part of the species name in the top left corner, but it seems that the same species are represented in the top left corner of each plot (Piloderma sphaerosporum).Perhaps other patterns would be visible if it was easier to read the species in the center. Also, it would be helpful to visualize the sites or site numbers on this plot.

> Based on these ordination plots, it is difficult to develop hypotheses about beta diversity for this data set. Broadly, I hypothesize there will be high beta diversity among communities in different geographic regions. Beta diversity among the different tree species will also be high. In other words, diversity will differ among separate geographic locations and among tree species studied. Analyses and visualizations of beta diversity would test this hypothesis. Data on environmental variables would enable additional tests to be performed, but such data were not included in this

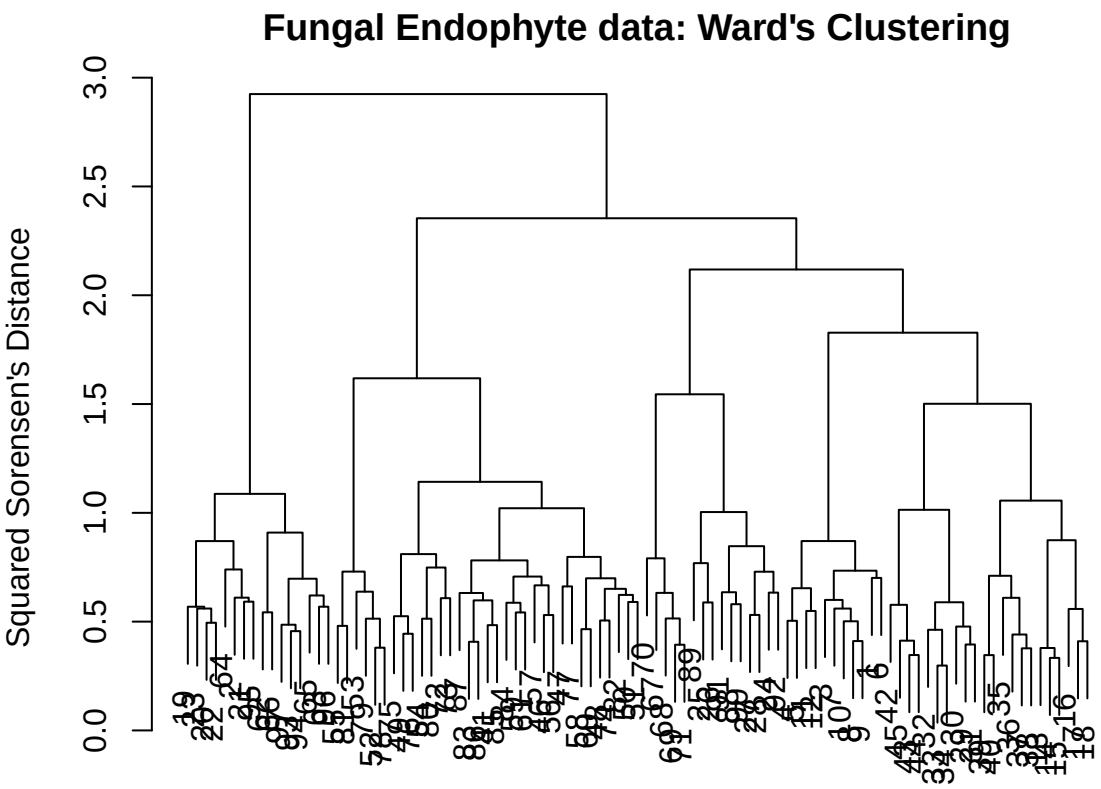
data set.

I did the cluster analysis for both the Bray-Curtis index and Sorensen index analyses to visualize any potential similarities or differences in diversity by site.

```
#Cluster Analysis (Bray-Curtis)

fungi.ward.db <- hclust(fungi.db, method = "ward.D2")

#Plot Cluster
par(mar = c(1,5,2,2) + 0.1)
plot(fungi.ward.db, main = "Fungal Endophyte data: Ward's Clustering", ylab = "Squared Bray-Curtis Dista
```




```
#Cluster Analysis (Sorensen)

fungi.ward <- hclust(fungi.ds, method = "ward.D2")

#Plot Cluster
par(mar = c(1,5,2,2) + 0.1)
plot(fungi.ward, main = "Fungal Endophyte data: Ward's Clustering", ylab = "Squared Sorensen's Distance
```



Based on the figures above, it is difficult to discern whether certain groups appear to be clustered together. Is there a way to make the sites more visible, or reduce the overlap of the sites on this figure? Looking at the cluster analysis from the Bray-Curtis index, the sites clustered together on the far left seem to be from one or two sites because it looks as if the numbers range from 10-20 (which refer to sites from Pinus Contorta in the USA). The sites on the far right seem to include sites 40-50, which include the European Pinus Contorta sites. However, further analyses are needed to determine whether diversity differs among the different geographic locations, and among the three species investigated. Using the Mantel test or a constrained ordination might be helpful in explaining beta diversity, but these data were not available in our dataset.

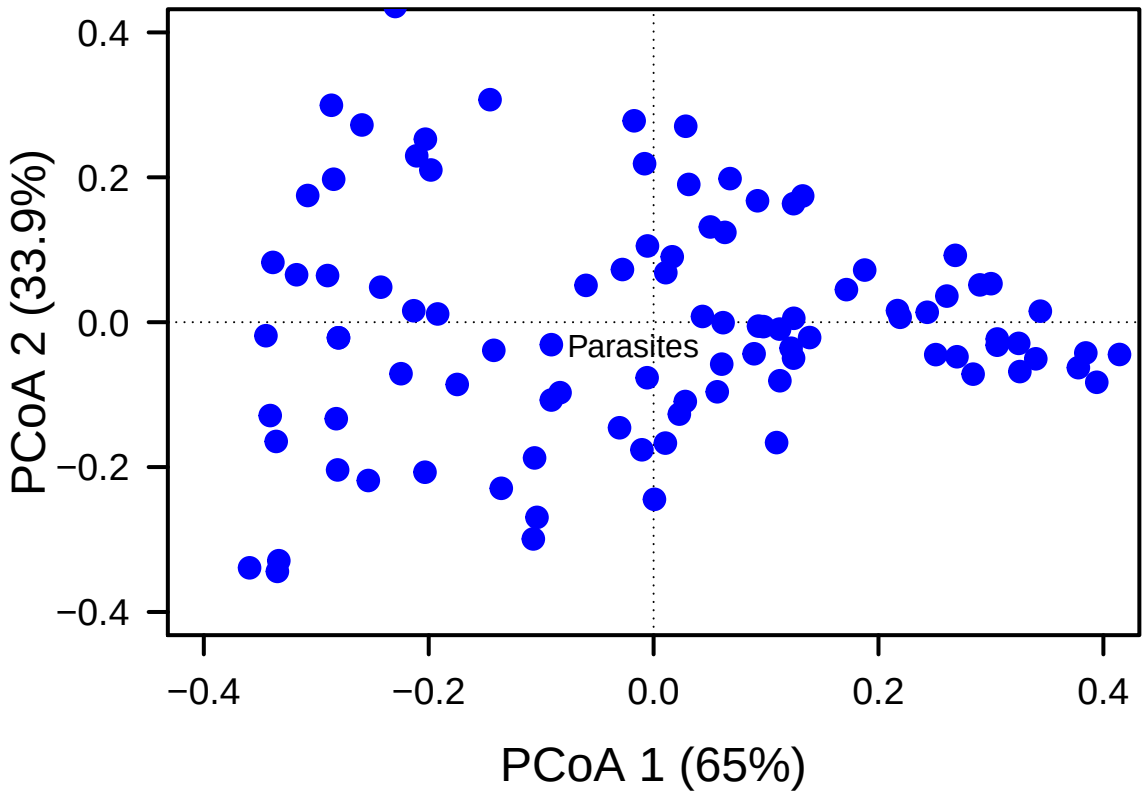
I also thought it might be interesting to examine beta diversity with respect to fungal functional groups (saprotrophs, EMF, parasites, and unknowns). Our dataset has relative abundances for each functional group. Perhaps analyzing beta diversity in terms of functional groups might yield new insights into differences among the different sites and tree species in this study. In the R-chunk below, I redid the ordination plots with the relative abundances of the functional groups at each site.

```
fungi2.data <- read.csv("functionalgroupdata.csv", sep = ",", header = TRUE)

fungi2.data <- fungi2.data[,-1]
fungi2.data <-fungi2.data[,1:4]

#Bray-Curtis
fungi2.db <- vegdist(fungi2.data, method = "bray")

#Sorensen
fungi2.ds <- vegdist(fungi2.data, method = "bray", binary = TRUE)
```

```
#Bray-Curtis
fungi2.pcoa.db <- cmdscale(fungi2.db, eig = TRUE, k = 3)

#Sorensen
fungi2.pcoa.ds <- cmdscale(fungi2.ds, eig = TRUE, k = 3)

explainvar1.db <- round(fungi2.pcoa.db$eig[1] / sum(fungi2.pcoa.db$eig), 3) * 100
explainvar2.db <- round(fungi2.pcoa.db$eig[2] / sum(fungi2.pcoa.db$eig), 3) * 100
explainvar3.db <- round(fungi2.pcoa.db$eig[3] / sum(fungi2.pcoa.db$eig), 3) * 100
33
```

```
## [1] 33
```

```
sum.eig.fungi2.db <- sum(explainvar1.db, explainvar2.db, explainvar3.db)

#Ordination Plot (Bray-Curtis)
par(mar = c(5,5,1,2) + 0.1)

#Initiate Plot
plot(fungi2.pcoa.db$points[ ,1], fungi2.pcoa.db$points[ ,2],
xlab = paste("PCoA 1 (", explainvar1.db, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar2.db, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE, ylim = c(-.4,.4), xlim = c

#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 and Labels
points(fungi2.pcoa.db$points[,1], fungi2.pcoa.db$points[,2],
pch = 19, cex = 1.5, bg = "gray", col = "blue")

text(fungi2.pcoa.db$points[,1], fungi2.pcoa.db$points[,2],
labels = row.names(fungi2.pcoa.db$points))
fungi2.pcoa.db <- add.spec.scores(fungi2.pcoa.db,fungi2.data,method = "pcoa.scores")

#Calculate and add species scores- did not calculate relative abundances because they were already calc

text(fungi2.pcoa.db$cproj[ ,1], fungi2.pcoa.db$cproj[, 2],
labels = row.names(fungi2.pcoa.db$cproj), col = "black")
```



>This ordination plot does not seem to show any clustering of points as they seem to be scattered throughout. I added points and labels to this plot, but it doesn't show labels for each point. COnsequently, it is difficult to determine which functional groups are clustered together.

Further, our data set has site by species matrices for each functional group. In the R-chunk below, I explored the diversity of EMF by site with an ordination plot.

```
fungi3.data <- read.csv("emfdiversitydata.csv", sep = ",", header = TRUE)

fungi3.data <- fungi3.data[,-1]


#Bray-Curtis
fungi3.db <- vegdist(fungi3.data, method = "bray")

#Sorensen
fungi3.ds <- vegdist(fungi3.data, method = "bray", binary = TRUE)

#Bray-Curtis
fungi3.pcoa.db <- cmdscale(fungi3.db, eig = TRUE, k = 3)

#Sorensen
fungi3.pcoa.ds <- cmdscale(fungi3.ds, eig = TRUE, k = 3)

explainvar1.db <- round(fungi3.pcoa.db$eig[1] / sum(fungi3.pcoa.db$eig), 3) * 100
explainvar2.db <- round(fungi3.pcoa.db$eig[2] / sum(fungi3.pcoa.db$eig), 3) * 100
explainvar3.db <- round(fungi3.pcoa.db$eig[3] / sum(fungi3.pcoa.db$eig), 3) * 100
33
```

```
## [1] 33
```

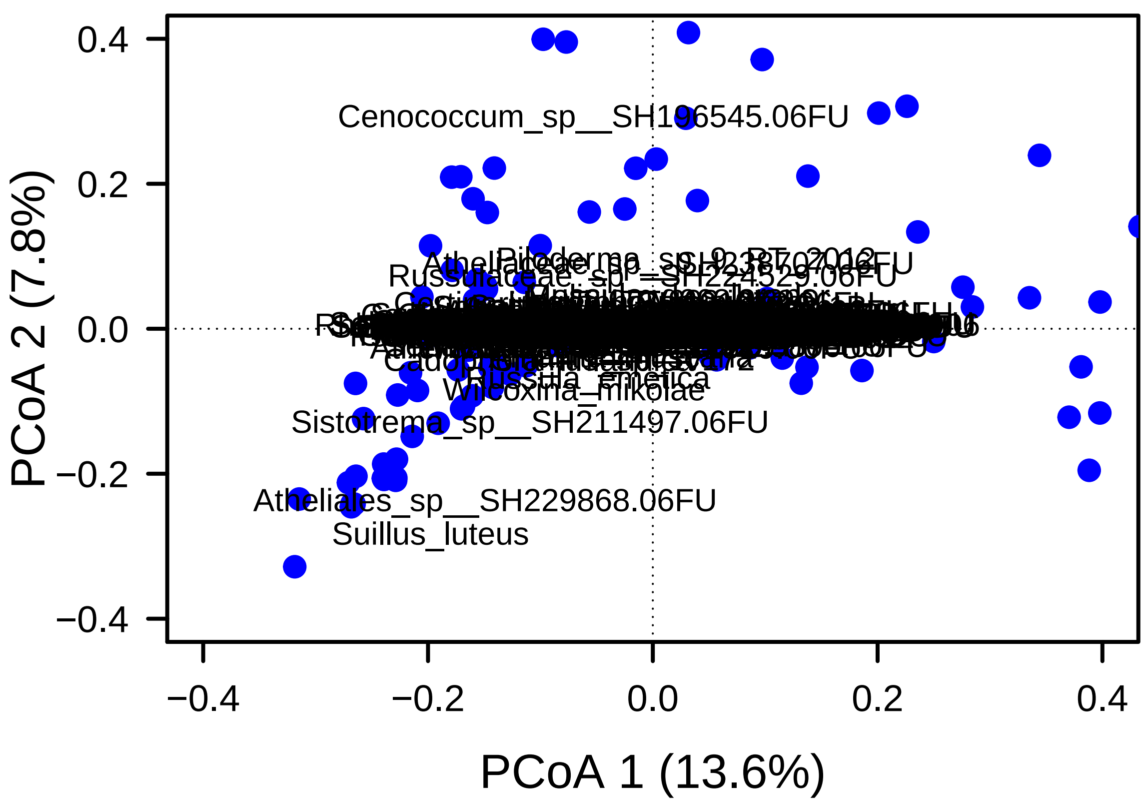
```
sum.eig.fungi3.db <- sum(explainvar1.db, explainvar2.db, explainvar3.db)

#Ordination Plot (Bray-Curtis)
par(mar = c(5,5,1,2) + 0.1)

#Initiate Plot
plot(fungi3.pcoa.db$points[ ,1], fungi3.pcoa.db$points[ ,2],
xlab = paste("PCoA 1 (", explainvar1.db, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar2.db, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE, ylim = c(-.4,.4), xlim = c

#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 and Labels
points(fungi3.pcoa.db$points[,1], fungi3.pcoa.db$points[,2],
pch = 19, cex = 1.5, bg = "gray", col = "blue")

text(fungi3.pcoa.db$points[,1], fungi3.pcoa.db$points[,2],
labels = row.names(fungi3.pcoa.db$points))
fungi3.pcoa.db <- add.spec.scores(fungi3.pcoa.db,fungi3.data,method = "pcoa.scores")
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

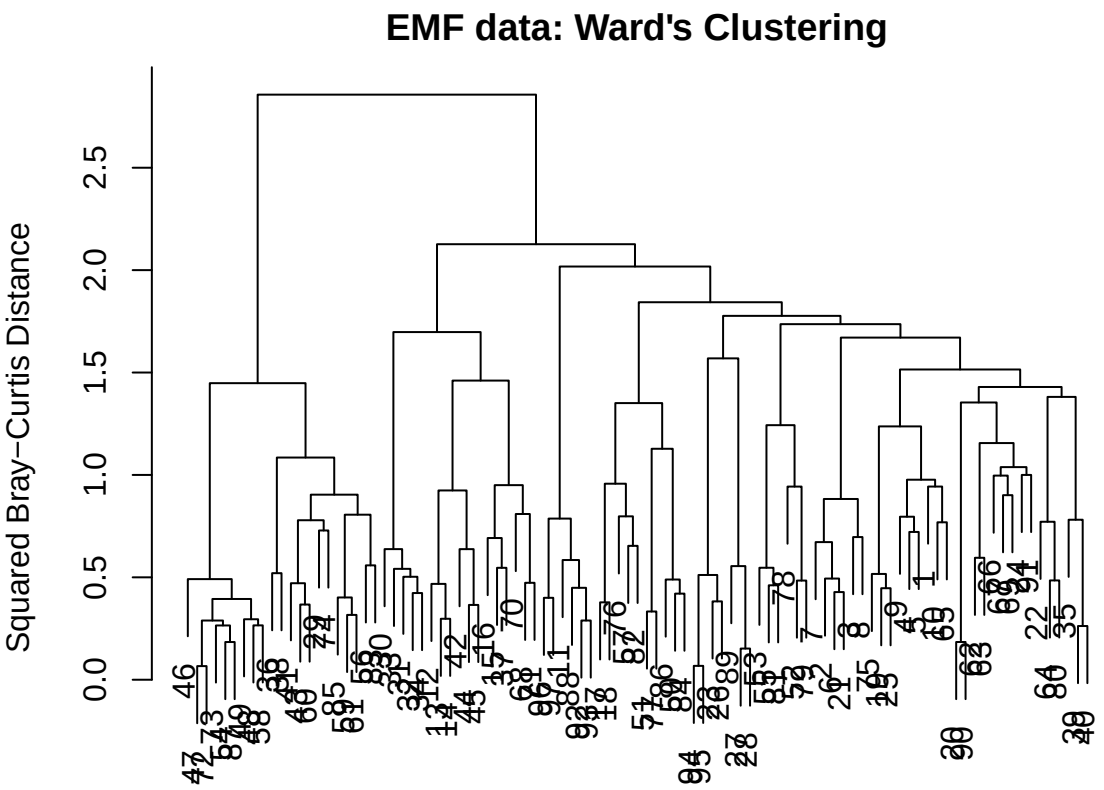
```
#Calculate and add species scores- did not calculate relative abundances because they were already calc

text(fungi3.pcoa.db$cproj[ ,1], fungi3.pcoa.db$cproj[, 2],
labels = row.names(fungi3.pcoa.db$cproj), col = "black")
```



>Similar to the ordination plots with all fungal OTUS, the ordination plot for EMF OTUs simply has a cluster of points in the middle of the plot. It is difficult to discern whether there are clusters of points, and the x and y axes only account for 13.6% and 7.8% of the total variation. Below I performed a cluster analysis

on the emf data to determine whether this visualization would provide other trends/indications about our data set.

```
#Cluster Analysis (Bray-Curtis)

fungi3.ward.db <- hclust(fungi3.db, method = "ward.D2")

#Plot Cluster
par(mar = c(1,5,2,2) + 0.1)
plot(fungi3.ward.db, main = "EMF data: Ward's Clustering", ylab = "Squared Bray-Curtis Distance")
```



The cluster analysis for the EMF OTUs looks relatively similar to that of total OTUs. The far left includes sites 46, 47, 49, and 58, which were all from Pinus Contorta trees in Europe.

I performed another ordination on the saprotroph data below.

```
fungi4.data <- read.csv("saprotrophdiversitydata.csv", sep = ",", header = TRUE)

fungi4.data <- fungi4.data[,-1]


#Bray-Curtis
fungi4.db <- vegdist(fungi4.data, method = "bray")

#Sorensen
fungi4.ds <- vegdist(fungi4.data, method = "bray", binary = TRUE)

#Bray-Curtis
```

```
fungi4.pcoa.db <- cmdscale(fungi4.db, eig = TRUE, k = 3)

#Sorensen
fungi4.pcoa.ds <- cmdscale(fungi4.ds, eig = TRUE, k = 3)

explainvar1.db <- round(fungi4.pcoa.db$eig[1] / sum(fungi4.pcoa.db$eig), 3) * 100
explainvar2.db <- round(fungi4.pcoa.db$eig[2] / sum(fungi4.pcoa.db$eig), 3) * 100
explainvar3.db <- round(fungi4.pcoa.db$eig[3] / sum(fungi4.pcoa.db$eig), 3) * 100
33
```

```
## [1] 33
```

```
sum.eig.fungi4.db <- sum(explainvar1.db, explainvar2.db, explainvar3.db)

#Ordination Plot (Bray-Curtis)
par(mar = c(5,5,1,2) + 0.1)

#Initiate Plot
plot(fungi4.pcoa.db$points[ ,1], fungi4.pcoa.db$points[ ,2],
xlab = paste("PCoA 1 (", explainvar1.db, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar2.db, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE, ylim = c(-.4,.4), xlim = c

#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 and Labels
points(fungi4.pcoa.db$points[,1], fungi4.pcoa.db$points[,2],
pch = 19, cex = 1.5, bg = "gray", col = "blue")

text(fungi4.pcoa.db$points[,1], fungi4.pcoa.db$points[,2],
labels = row.names(fungi3.pcoa.db$points))
fungi4.pcoa.db <- add.spec.scores(fungi4.pcoa.db,fungi4.data,method = "pcoa.scores")
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

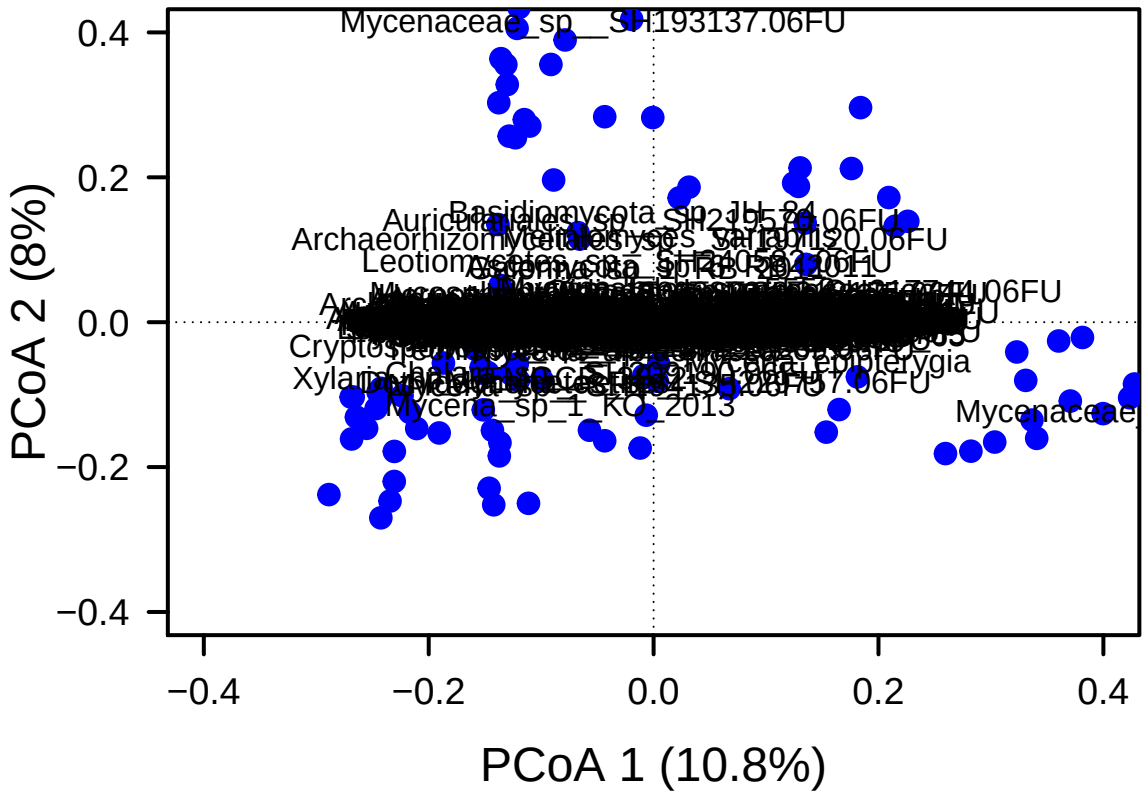
```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

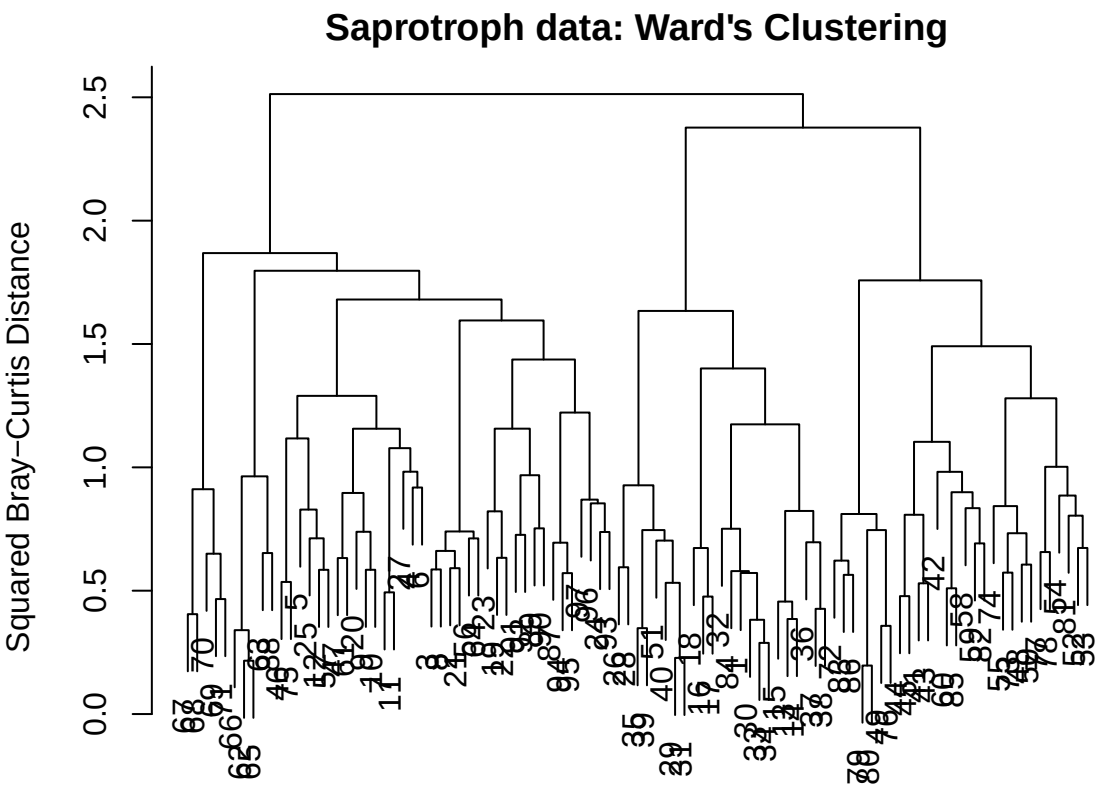
```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
#Calculate and add species scores- did not calculate relative abundances because they were already calc

text(fungi4.pcoa.db$cproj[ ,1], fungi4.pcoa.db$cproj[, 2],
labels = row.names(fungi4.pcoa.db$cproj), col = "black")
```



The ordination plot with saprotrophs has the same cluster of points in the middle of the plot. However, there are two other clusters of points, one at the top of the plot (Mycenaceae), and another on the far right (Mycenaceae). Perhaps this species is highly prevalant at specific sites, or at one or two geographic locations.

```
#Cluster Analysis (Bray-Curtis)

fungi4.ward.db <- hclust(fungi4.db, method = "ward.D2")

#Plot Cluster
par(mar = c(1,5,2,2) + 0.1)
plot(fungi4.ward.db, main = "Saprotroph data: Ward's Clustering", ylab = "Squared Bray-Curtis Distance")
```



The cluster analysis shows sites shows the Nothofagus sites in a cluster on the far right. This might indicate that diversity is similar at these sites. On the far left, European sites with Pinus Contorta are clustered together, indicating they might have similar diversity values.

‘‘