8. Worksheet: Among Site (Beta) Diversity – Part 1

Joy O'Brien; Z620: Quantitative Biodiversity, Indiana University

31 January, 2023

OVERVIEW

In this worksheet, 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. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) with your name.
2. Complete as much of the worksheet as possible during class.
3. Use the handout as a guide; it contains a more complete description of data sets along with examples of proper scripting needed to carry out the exercises.
4. Answer questions in the worksheet. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio (color may vary if you changed the editor theme).
5. Before you leave the classroom today, it is *imperative* that you **push** this file to your GitHub repo, at whatever stage you are. Ths will enable you to pull your work onto your own computer.
6. When you have completed the worksheet, **Knit** the text and code into a single PDF file by pressing the `Knit` button in the RStudio scripting panel. This will save the PDF output in your '6.BetaDiversity' folder.
7. After Knitting, please submit the worksheet by making a **push** to your GitHub repo and then create a **pull request** via GitHub. Your pull request should include this file (**6.BetaDiversity_1_Worksheet.Rmd**) with all code blocks filled out and questions answered) and the PDF output of `Knitr` (**6.BetaDiversity_1_Worksheet.pdf**).

The completed exercise is due on **Wednesday, February 1st, 2023 before 12:00 PM (noon)**.

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 "*/6.BetaDiversity*" folder, and
4. load the `vegan` R package (be sure to install if needed).

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

```
## [1] "/Users/joyobrien/GitHub/QB2023_OBrien/2.Worksheets/6.BetaDiversity"
```

```
setwd("~/GitHub/QB2023_OBrien/2.Worksheets/6.BetaDiversity")
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

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

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.

```
# note, please do not print the dataset when submitting
package.list <- c('vegan', 'ade4', 'viridis', 'gplots', 'BiodiversityR', 'indicspecies')
for (package in package.list) {
  if (!require(package, character.only = TRUE, quietly = TRUE)) {
    install.packages(package)
    library(package, character.only = TRUE)
  }
}
```

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

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

```
## BiodiversityR 2.15-1: Use command BiodiversityRGUI() to launch the Graphical User Interface;
## to see changes use BiodiversityRGUI(changeLog=TRUE, backward.compatibility.messages=TRUE)
```

```
#library(vegan)
#library(ade4)
#library(viridis)
#library(gplots)
#library(BiodiversityR)
#library(indicspecies)
# Loading doubs data
#install.packages("sem")
#install.packages("leaps")
#install.packages("rgl")
#install.packages("aplpack")
data(doubs)

#Structure of dataset
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
```

```
head(doubs$species)
```

```
##                 Scientific        French           English code
## 1             Cottus gobio        chabot european bullhead Cogo
## 2       Salmo trutta fario  truite fario       brown trout Satr
## 3        Phoxinus phoxinus        vairon            minnow Phph
## 4   Nemacheilus barbatulus loche franche       stone loach Neba
## 5      Thymallus thymallus         ombre          grayling Thth
## 6 Telestes soufia agassizi       blageon           blageon Teso
```

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: 4 objects: env, fish, xy, species ***Answer 1b***: 27 species ***Answer 1c***: 30 sites

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 varies along the Doubs river in that richness decreases moving towards upstream. ***Answer 2b***: Brown trout is more abundant upstream than downstream. ***Answer 2c***: Fish richness does not allow us to make predictions regarding abundance since we see less fish richness in the upstream area but more abundance of the Brown trout.

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



# Modify function to calculate pairwise beta diversity for turnover as follows:
beta.w <- function(site.by.species = "", sitenum1 = "", sitenum2 = "",
                   pairwise = FALSE){
  # only if we specify TRUE do this:
  if(pairwise == TRUE){
    # As a check, let's print an error if we do not provide the arguments
    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)
  }
  #Otherwise pairwise defaults to FALSE. so do this like before:
  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(b.w)
  }
}

# Using function to calculate diversity in the Doubs river
beta.w(doubs$fish) # 2.16
```

```
## [1] 2.16
```

```
# Comparing site 1 and 2
beta.w(doubs$fish, 1, 2, pairwise = TRUE) # 0.5
```

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

```
# Comparing site 1 and 10
beta.w(doubs$fish, 1, 10, pairwise = TRUE) # 0.714
```

```
## [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 beta diversity function written above incorporates richness across all sites (gamma diversity) and the richness of each site (alpha diversity). Local richness and turnover contribute to gamma diversity because the relationship between local and regional diversity is multiplicative. In this case, the beta diversity of the Doubs is 2.16, indicating that there is a low level of similar species in the river. ***Answer 3b***: The beta diversity of site 1 compared to site 2 is 0.5 while the beta diversity of site 1 compared to site 10 is 0.714. Therefore, the assemblage of fish species in site 1 of the Doubs River is more similar to site 2. ***Answer 3c***: If we defined beta additively, we would be calculating how many more species exist in the regional pool when compared to local sites instead of measuring how many more times diverse the region is when compared to local sites.

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***: Incidence and abudance based metrics differ because they weight rare species differently. Jaccard incidence based metric places more emphasis on taxa not shared between sites, meaning rare taxa. Sorenson which is an incidence-based metric is characterized as emphasizing similarity of samples that have shared species. Abundance-based metrics are influenced by abundant species as they measure compositional overlap.

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 Sørensen's Similarity ("fish.ds"), and
4. construct a resemblance matrix based on Bray-Curtis Distance ("fish.db").

```
fish <- doubs$fish
fish <- fish[-8, ] # Remove site 8 from data because it has no observations

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

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

# Calculate Sorenson
fish.ds <- vegdist(fish, method = "bray", binary = TRUE)
# Generating a square resemblance matrix
fish.db_square <- vegdist(fish, method = "bray", upper = TRUE , diag = TRUE)

fish.db
```

```
##            1         2         3         4         5         6         7
## 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 0.66666667
## 10 0.88235294 0.38461538 0.40000000 0.37142857 0.54166667 0.25714286 0.26666667
## 11 0.57142857 0.30434783 0.40740741 0.43750000 0.68888889 0.43750000 0.33333333
## 12 0.71428571 0.20000000 0.23529412 0.33333333 0.69230769 0.38461538 0.17647059
## 13 0.72727273 0.29032258 0.31428571 0.45000000 0.73584906 0.55000000 0.37142857
## 14 0.80645161 0.40000000 0.31818182 0.34693878 0.67741935 0.42857143 0.36363636
## 15 0.83333333 0.51111111 0.46938776 0.40740741 0.55223881 0.37037037 0.38775510
## 16 0.86046512 0.65384615 0.57142857 0.47540984 0.45945946 0.37704918 0.53571429
## 17 0.91489362 0.67857143 0.63333333 0.50769231 0.51282051 0.44615385 0.60000000
## 18 0.95555556 0.74074074 0.72413793 0.58730159 0.50000000 0.52380952 0.68965517
```

```
## 19 1.00000000 0.79310345 0.70967742 0.61194030 0.50000000 0.52238806 0.67741935
## 20 1.00000000 0.91176471 0.88888889 0.74025974 0.48888889 0.68831169 0.86111111
## 21 1.00000000 0.94594595 0.92307692 0.78313253 0.50000000 0.73493976 0.89743590
## 22 1.00000000 0.97619048 0.95454545 0.82795699 0.52830189 0.78494624 0.93181818
## 23 1.00000000 1.00000000 1.00000000 0.92000000 0.89473684 0.84000000 0.90000000
## 24 1.00000000 1.00000000 1.00000000 0.88888889 0.79591837 0.77777778 0.93548387
## 25 1.00000000 1.00000000 0.92592593 0.81250000 0.68888889 0.68750000 0.85185185
## 26 1.00000000 0.96363636 0.93220339 0.78125000 0.55844156 0.68750000 0.89830508
## 27 1.00000000 0.97333333 0.94936709 0.83333333 0.56701031 0.76190476 0.92405063
## 28 1.00000000 0.97560976 0.95348837 0.82417582 0.57692308 0.78021978 0.93023256
## 29 0.97777778 0.93939394 0.92233010 0.81481481 0.53719008 0.77777778 0.90291262
## 30 1.00000000 1.00000000 0.98095238 0.87272727 0.59349593 0.83636364 0.96190476
##             9         10         11         12         13         14         15
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10 0.57142857
## 11 0.76000000 0.44000000
## 12 0.68750000 0.37500000 0.24137931
## 13 0.81818182 0.57575758 0.33333333 0.18918919
## 14 0.76190476 0.47619048 0.43589744 0.21739130 0.19148936
## 15 0.65957447 0.40425532 0.50000000 0.33333333 0.38461538 0.24590164
## 16 0.70370370 0.51851852 0.64705882 0.55172414 0.59322034 0.44117647 0.26027397
## 17 0.68965517 0.51724138 0.63636364 0.58064516 0.61904762 0.50000000 0.40259740
## 18 0.64285714 0.57142857 0.69811321 0.66666667 0.70491803 0.60000000 0.46666667
## 19 0.66666667 0.63333333 0.82456140 0.75000000 0.81538462 0.67567568 0.56962025
## 20 0.68571429 0.77142857 0.91044776 0.89189189 0.92000000 0.83333333 0.70786517
## 21 0.76315789 0.81578947 0.91780822 0.92500000 0.95061728 0.86666667 0.76842105
## 22 0.76744186 0.86046512 0.95180723 0.95555556 0.97802198 0.90000000 0.77142857
## 23 0.77777778 0.88888889 0.86666667 0.90909091 1.00000000 0.93750000 0.94594595
## 24 0.72413793 0.79310345 0.92307692 0.93939394 1.00000000 0.90697674 0.87500000
## 25 0.84000000 0.76000000 0.90909091 0.93103448 1.00000000 0.84615385 0.81818182
## 26 0.71929825 0.82456140 0.92592593 0.93442623 0.96774194 0.85915493 0.76315789
## 27 0.76623377 0.84415584 0.94594595 0.95061728 0.97560976 0.89010989 0.77083333
## 28 0.76190476 0.85714286 0.95061728 0.95454545 0.97752809 0.89795918 0.78640777
## 29 0.78217822 0.84158416 0.89795918 0.90476190 0.90566038 0.84347826 0.73333333
## 30 0.84466019 0.90291262 0.98000000 0.98130841 1.00000000 0.93162393 0.81967213
##            16         17         18         19         20         21         22
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
```

```
## 15
## 16
## 17 0.26190476
## 18 0.34146341 0.13953488
## 19 0.39534884 0.31111111 0.25000000
## 20 0.58333333 0.42000000 0.32653061 0.23529412
## 21 0.62745098 0.49056604 0.40384615 0.29629630 0.10169492
## 22 0.66071429 0.55172414 0.47368421 0.38983051 0.18750000 0.10447761
## 23 0.90909091 0.83333333 0.82608696 0.84000000 0.86666667 0.87878788 0.89473684
## 24 0.81818182 0.69491525 0.64912281 0.63934426 0.57746479 0.61038961 0.65517241
## 25 0.76470588 0.74545455 0.66037736 0.61403509 0.67164179 0.69863014 0.73493976
## 26 0.63855422 0.54022989 0.45882353 0.32584270 0.21212121 0.20000000 0.25217391
## 27 0.66990291 0.57009346 0.48571429 0.37614679 0.19327731 0.13600000 0.12592593
## 28 0.69090909 0.57894737 0.50000000 0.41379310 0.22222222 0.16666667 0.12676056
## 29 0.65354331 0.51145038 0.44186047 0.41353383 0.24475524 0.18120805 0.11949686
## 30 0.72093023 0.57894737 0.52671756 0.48148148 0.29655172 0.23178808 0.18012422
##            23         24         25         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 0.57894737
## 25 0.46666667 0.46153846
## 26 0.82978723 0.48275862 0.59259259
## 27 0.88059701 0.61538462 0.70270270 0.18867925
## 28 0.89189189 0.64705882 0.72839506 0.23893805 0.09774436
## 29 0.91208791 0.70588235 0.77551020 0.33846154 0.18666667 0.14649682
## 30 0.91397849 0.71153846 0.78000000 0.36363636 0.19736842 0.15723270 0.14772727
```

```
fish.ds
```

```
##            1          2          3          4          5          6          7
## 2  0.50000000
## 3  0.60000000 0.14285714
## 4  0.77777778 0.45454545 0.33333333
## 5  0.83333333 0.57142857 0.46666667 0.15789474
## 6  0.81818182 0.53846154 0.42857143 0.11111111 0.04761905
```

```
## 7  0.66666667 0.25000000 0.33333333 0.38461538 0.37500000 0.33333333
## 9  1.00000000 0.50000000 0.55555556 0.38461538 0.37500000 0.33333333 0.40000000
## 10 0.71428571 0.33333333 0.40000000 0.28571429 0.29411765 0.25000000 0.09090909
## 11 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000 0.27272727
## 12 0.71428571 0.33333333 0.40000000 0.42857143 0.52941176 0.50000000 0.27272727
## 13 0.71428571 0.33333333 0.40000000 0.57142857 0.64705882 0.62500000 0.45454545
## 14 0.81818182 0.53846154 0.42857143 0.33333333 0.42857143 0.40000000 0.46666667
## 15 0.83333333 0.57142857 0.60000000 0.36842105 0.36363636 0.33333333 0.37500000
## 16 0.88888889 0.70000000 0.61904762 0.36000000 0.28571429 0.25925926 0.54545455
## 17 0.91304348 0.76000000 0.69230769 0.46666667 0.39393939 0.37500000 0.62962963
## 18 0.91666667 0.76923077 0.70370370 0.48387097 0.41176471 0.39393939 0.64285714
## 19 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545 0.71428571
## 20 1.00000000 0.84000000 0.76923077 0.53333333 0.39393939 0.43750000 0.70370370
## 21 1.00000000 0.84615385 0.77777778 0.54838710 0.41176471 0.45454545 0.71428571
## 22 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 23 1.00000000 1.00000000 1.00000000 0.81818182 0.71428571 0.69230769 0.75000000
## 24 1.00000000 1.00000000 1.00000000 0.75000000 0.68421053 0.66666667 0.84615385
## 25 1.00000000 1.00000000 0.83333333 0.62500000 0.36842105 0.44444444 0.69230769
## 26 1.00000000 0.91666667 0.84000000 0.58620690 0.43750000 0.48387097 0.76923077
## 27 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 28 1.00000000 0.92000000 0.84615385 0.60000000 0.45454545 0.50000000 0.77777778
## 29 0.92592593 0.79310345 0.73333333 0.52941176 0.40540541 0.44444444 0.67741935
## 30 1.00000000 1.00000000 0.92000000 0.65517241 0.50000000 0.54838710 0.84615385
##             9         10         11         12         13         14         15
## 2
## 3
## 4
## 5
## 6
## 7
## 9
## 10 0.45454545
## 11 0.45454545 0.33333333
## 12 0.45454545 0.33333333 0.00000000
## 13 0.63636364 0.50000000 0.16666667 0.16666667
## 14 0.60000000 0.37500000 0.25000000 0.25000000 0.25000000
## 15 0.50000000 0.29411765 0.29411765 0.29411765 0.29411765 0.14285714
## 16 0.54545455 0.47826087 0.56521739 0.56521739 0.56521739 0.33333333 0.28571429
## 17 0.62962963 0.57142857 0.57142857 0.57142857 0.57142857 0.37500000 0.33333333
## 18 0.64285714 0.58620690 0.58620690 0.58620690 0.58620690 0.39393939 0.35294118
## 19 0.64285714 0.65517241 0.79310345 0.79310345 0.79310345 0.57575758 0.52941176
## 20 0.62962963 0.64285714 0.78571429 0.78571429 0.85714286 0.62500000 0.57575758
## 21 0.64285714 0.65517241 0.79310345 0.79310345 0.86206897 0.63636364 0.58823529
## 22 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
## 23 0.50000000 0.77777778 0.77777778 0.77777778 1.00000000 0.84615385 0.85714286
## 24 0.69230769 0.71428571 0.85714286 0.85714286 1.00000000 0.77777778 0.78947368
## 25 0.69230769 0.57142857 0.85714286 0.85714286 1.00000000 0.66666667 0.68421053
## 26 0.69230769 0.70370370 0.85185185 0.85185185 0.92592593 0.67741935 0.62500000
## 27 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
## 28 0.70370370 0.71428571 0.85714286 0.85714286 0.92857143 0.68750000 0.63636364
## 29 0.67741935 0.62500000 0.68750000 0.68750000 0.68750000 0.50000000 0.45945946
## 30 0.76923077 0.77777778 0.92592593 0.92592593 1.00000000 0.74193548 0.68750000
##            16         17         18         19         20         21         22
## 2
```

```
## 3
## 4
## 5
## 6
## 7
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17 0.12820513
## 18 0.15000000 0.02222222
## 19 0.25000000 0.15555556 0.13043478
## 20 0.28205128 0.18181818 0.15555556 0.02222222
## 21 0.30000000 0.20000000 0.17391304 0.04347826 0.02222222
## 22 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222
## 23 0.80000000 0.76000000 0.76923077 0.76923077 0.76000000 0.76923077 0.76000000
## 24 0.68000000 0.60000000 0.54838710 0.48387097 0.46666667 0.48387097 0.46666667
## 25 0.60000000 0.60000000 0.54838710 0.48387097 0.46666667 0.48387097 0.46666667
## 26 0.36842105 0.25581395 0.22727273 0.09090909 0.06976744 0.04545455 0.02325581
## 27 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222 0.00000000
## 28 0.33333333 0.22727273 0.20000000 0.06666667 0.04545455 0.02222222 0.00000000
## 29 0.25581395 0.12500000 0.10204082 0.06122449 0.08333333 0.06122449 0.08333333
## 30 0.36842105 0.25581395 0.22727273 0.09090909 0.06976744 0.04545455 0.02325581
##            23         24         25         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 0.45454545
## 25 0.45454545 0.37500000
## 26 0.75000000 0.44827586 0.44827586
## 27 0.76000000 0.46666667 0.46666667 0.02325581
## 28 0.76000000 0.46666667 0.46666667 0.02325581 0.00000000
```

```
## 29 0.79310345 0.52941176 0.52941176 0.10638298 0.08333333 0.08333333
## 30 0.75000000 0.44827586 0.44827586 0.04761905 0.02325581 0.02325581 0.10638298
```

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 Sørensen or Bray-Curtis distance influence your interpretation of site (dis)similarity?

Answer 5a: The 'fish.db' matrix represents dissimilarity not only because we used the Bray-Curtis metric to calculate the matrix (which is a measure of dissimilarity) but also because the matrix is triangular, the diagonal of comparing each site to itself is missing as a default characteristic of dissimilarity. ***Answer 5b***: When looking at the Sorensen distance matrix and the Bray-Cutris distance matrix they look simlar in terms of the values they return but it is clear that Sorenson is weighting shared taxa within samples more than the Bray-Curtis matrix which may lead us to suggest that there is more similarity between the composition of species than there actually is. With Sorensen, the sites look more similar. With Bray-Curtis, the sites appear to be more dissimilar.

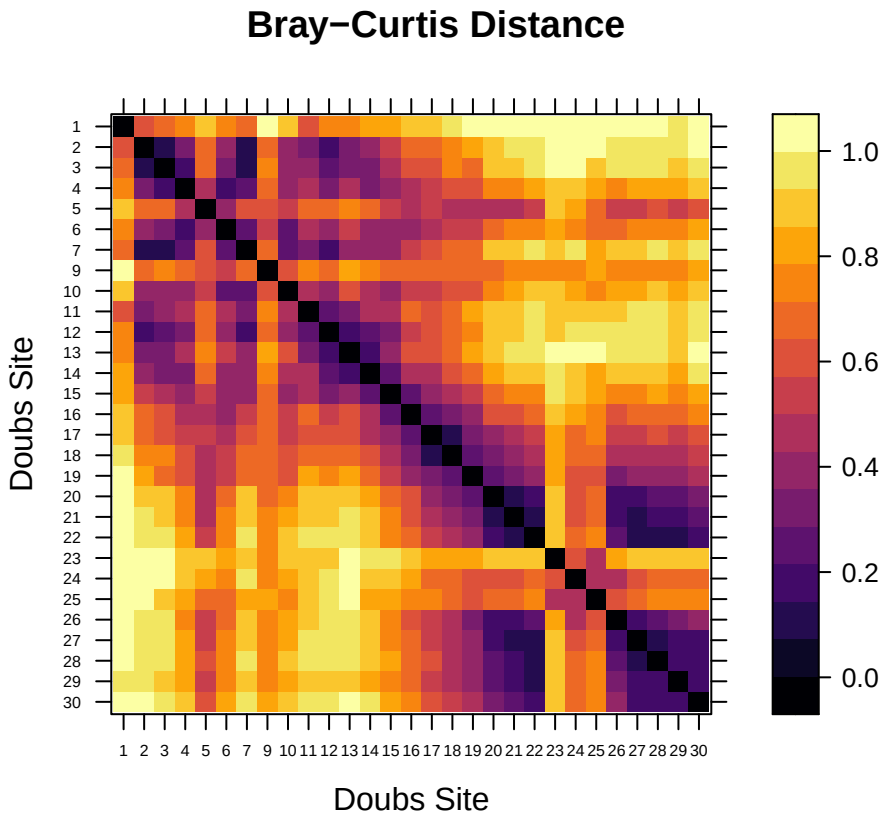
4) VISUALIZING BETA-DIVERSITY

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 color palette
#install.packages('viridis')
library(viridis)
library(lattice)
# 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")
```

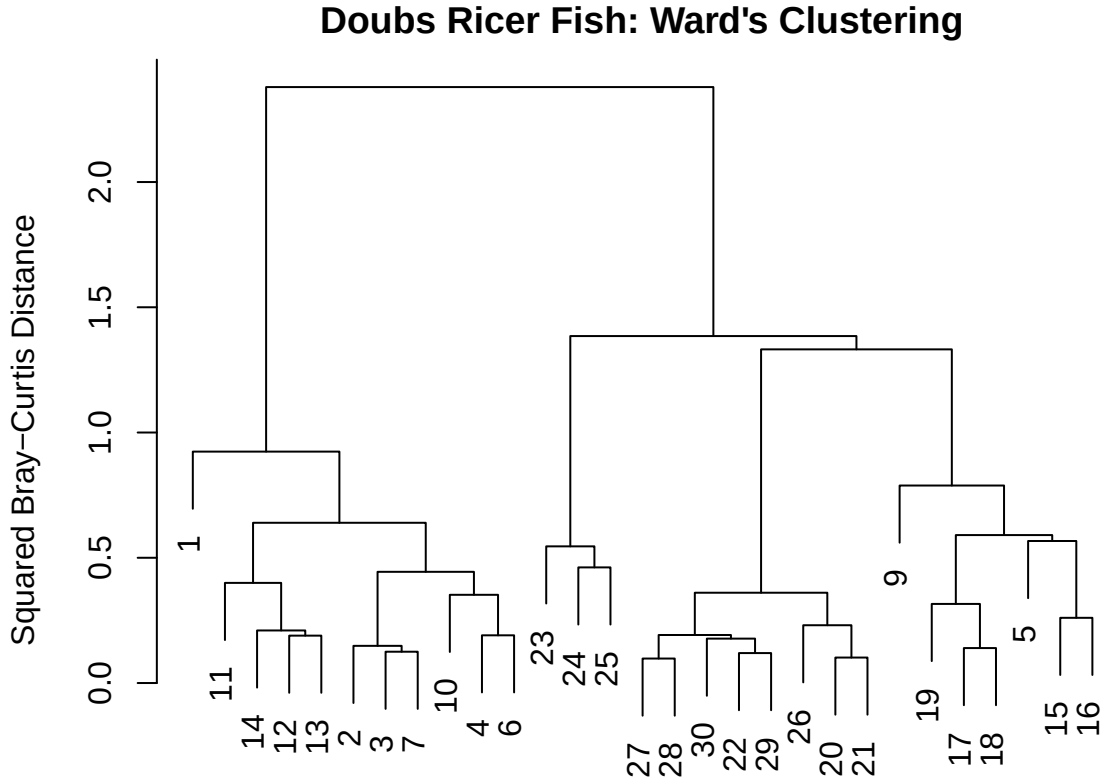


B. Cluster Analysis

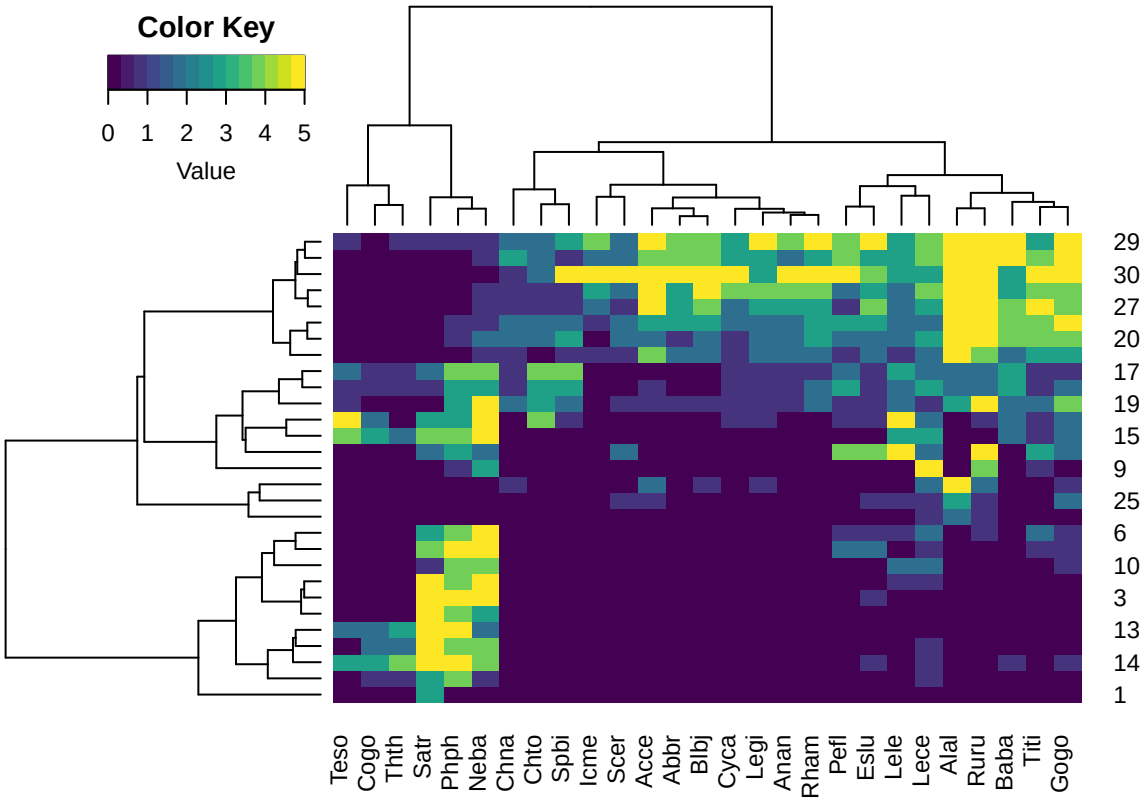
In the R code chunk below, do the following:

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

```
# library(vegan)
# 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 Ricer Fish: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```




```
# Plotting cluster using heatmap.2
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***: Sample sites within the Doubs river that are close to eachother will exhibit similar fish diversity than sample sites that are not located near eachother (such as upstream vs. downstream).

C. Ordination

Principal Coordinates Analysis (PCoA)

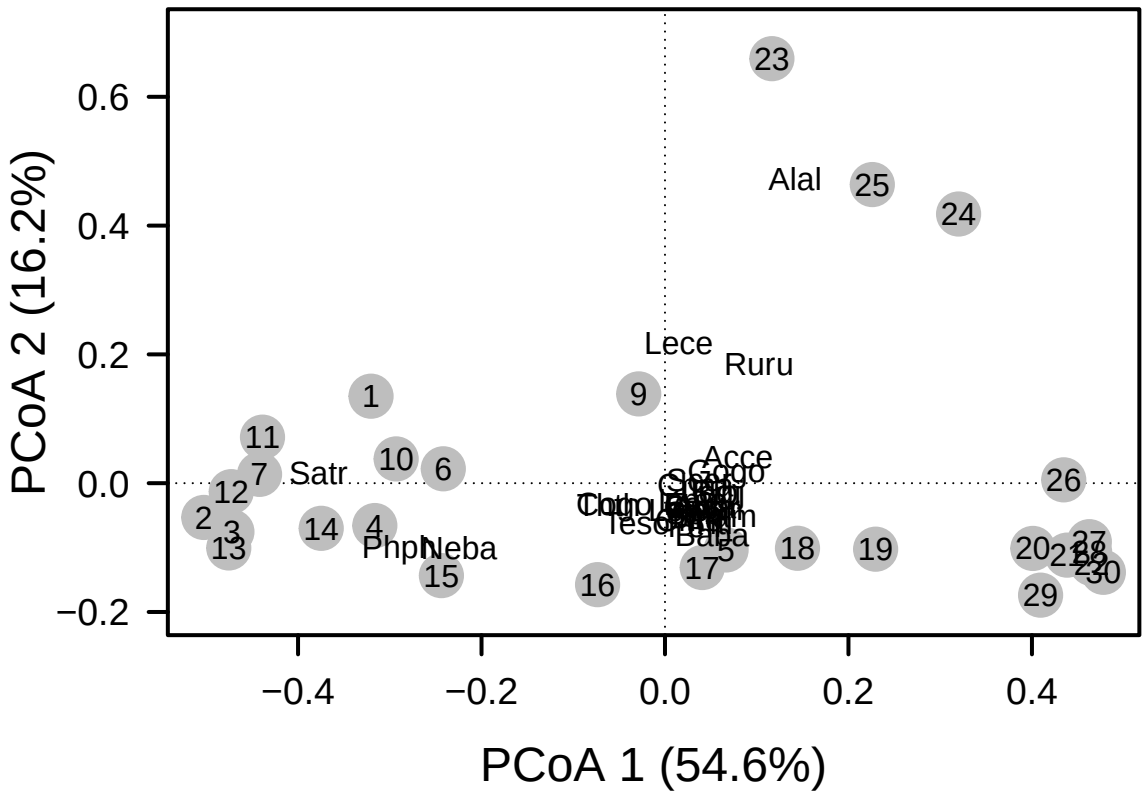
In the R code chunk below, do the following:

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_square, eig = TRUE, k = 3)

# Quantify percent variation explained by the first three axes
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
```

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

# 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 and 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))

# Identify and visualize influential species
fishREL <- fish
for(i in 1:nrow(fish)){
  fishREL[i, ] = fish[i, ] / sum(fish[i, ])
}



# Added from Canan's slack message since the add.spec.scores isn't working
`add.spec.scores.class` <-
  function(ordi,comm,method="cor.scores",multi=1,Rscale=F,scaling="1") {
    ordiscores <- scores(ordi,display="sites")
    n <- ncol(comm)
    p <- ncol(ordiscores)
    specscores <- array(NA,dim=c(n,p))
    rownames(specscores) <- colnames(comm)
    colnames(specscores) <- colnames(ordiscores)
    if (method == "cor.scores") {
      for (i in 1:n) {
        for (j in 1:p) {specscores[i,j] <- cor(comm[,i],ordiscores[,j],method="pearson")}
      }
    }
    if (method == "wa.scores") {specscores <- wascores(ordiscores,comm)}
    if (method == "pcoa.scores") {
      rownames(ordiscores) <- rownames(comm)
      eigenv <- ordi$eig
      accounted <- sum(eigenv)
      tot <- 2*(accounted/ordi$GOF[2])-(accounted/ordi$GOF[1])
      eigen.var <- eigenv/(nrow(comm)-1)
```

```
      neg <- length(eigenv[eigenv<0])
      pos <- length(eigenv[eigenv>0])
      tot <- tot/(nrow(comm)-1)
      eigen.percen <- 100*eigen.var/tot
      eigen.cumpercen <- cumsum(eigen.percen)
      constant <- ((nrow(comm)-1)*tot)^0.25
      ordiscores <- ordiscores * (nrow(comm)-1)^-0.5 * tot^-0.5 * constant
      p1 <- min(p, pos)
      for (i in 1:n) {
        for (j in 1:p1) {
          specscores[i,j] <- cor(comm[,i],ordiscores[,j])*sd(comm[,i])/sd(ordiscores[,j])
          if(is.na(specscores[i,j])) {specscores[i,j]<-0}
        }
      }
      if (Rscale==T && scaling=="2") {
        percen <- eigen.var/tot
        percen <- percen^0.5
        ordiscores <- sweep(ordiscores,2,percen,"/")
        specscores <- sweep(specscores,2,percen,"*")
      }
      if (Rscale==F) {
        specscores <- specscores / constant
        ordiscores <- ordi$points
      }
      ordi$points <- ordiscores
      ordi$eig <- eigen.var
      ordi$eig.percen <- eigen.percen
      ordi$eig.cumpercen <- eigen.cumpercen
      ordi$eigen.total <- tot
      ordi$R.constant <- constant
      ordi$Rscale <- Rscale
      ordi$scaling <- scaling
    }
    specscores <- specscores * multi
    ordi$cproj <- specscores
    return(ordi)
  }

# Use this info to calculate and add species scores (I am getting an error when I knit the pdf due to t
fish.pcoa <- add.spec.scores.class(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.class(fish.pcoa, fishREL, method = "cor.scores")$cproj
corrcut <- 0.7
imp.spp <- spe.corr[abs(spe.corr[, 1]) >= corrcut | abs(spe.corr[, 2]) >= corrcut, ]

# Permutation test
fit <- envfit(fish.pcoa, fishREL, perm = 999)
fit
```

```
##
## ***VECTORS
##
##          Dim1     Dim2     r2 Pr(>r)
## Cogo -0.83884 -0.54438 0.2982  0.014 *
## Satr -0.99904  0.04371 0.4326  0.001 ***
## Phph -0.94110 -0.33813 0.7814  0.001 ***
## Neba -0.91413 -0.40543 0.6234  0.001 ***
## Thth -0.87692 -0.48063 0.2634  0.027 *
## Teso -0.44704 -0.89452 0.1700  0.111
## Chna  0.99707 -0.07644 0.4612  0.001 ***
## Chto  0.42032 -0.90738 0.2579  0.022 *
## Lele  0.33041 -0.94384 0.0495  0.513
```

```
## Lece  0.06856  0.99765 0.3399  0.016 *
## Baba  0.54118 -0.84091 0.6752  0.001 ***
## Spbi  0.57341 -0.81927 0.4138  0.002 **
## Gogo  0.97507  0.22188 0.3753  0.006 **
## Eslu  0.72044 -0.69352 0.1673  0.092 .
## Pefl  0.43762 -0.89916 0.3048  0.011 *
## Rham  0.72476 -0.68901 0.8301  0.001 ***
## Legi  0.93461 -0.35568 0.7016  0.001 ***
## Scer  0.98569  0.16858 0.3533  0.011 *
## Cyca  0.68181 -0.73153 0.7743  0.001 ***
## Titi  0.64378 -0.76521 0.4586  0.002 **
## Abbr  0.77254 -0.63497 0.7128  0.001 ***
## Icme  0.75626 -0.65427 0.5270  0.002 **
## Acce  0.88799  0.45986 0.6294  0.001 ***
## Ruru  0.48379  0.87518 0.5177  0.003 **
## Blbj  0.95802 -0.28671 0.6766  0.001 ***
## Alal  0.28755  0.95777 0.8592  0.001 ***
## Anan  0.74277 -0.66954 0.7894  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

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***: Sites that are similar in their beta diversity measurements of fish composition (as we have seen above) are also sites that we see grouping together on the PCoA plot. For example, there is a cluster of sites on the bottom left of the plot and that is comprised of the "Satr" fish species and a small cluster of sites on the top right of the plot that are associated with the "Alal" species. The cluster of sites in the center are comprised of many fish species (hard to tell which ones from the plot) and seem to have a higher amount of richness than other sites. ***Answer 7b***: Based on the PCoA plot above, we may be able to say that fish species such as "Acce" and "Teso" are potential indicators of river quality because they are clusted with other species in the downstream area of the river, meaning that the better river quality, the more potential there is for the river to support different fish species, which is what we see in the downstream sites and the clustering in the center of the PCoA.

SYNTHESIS

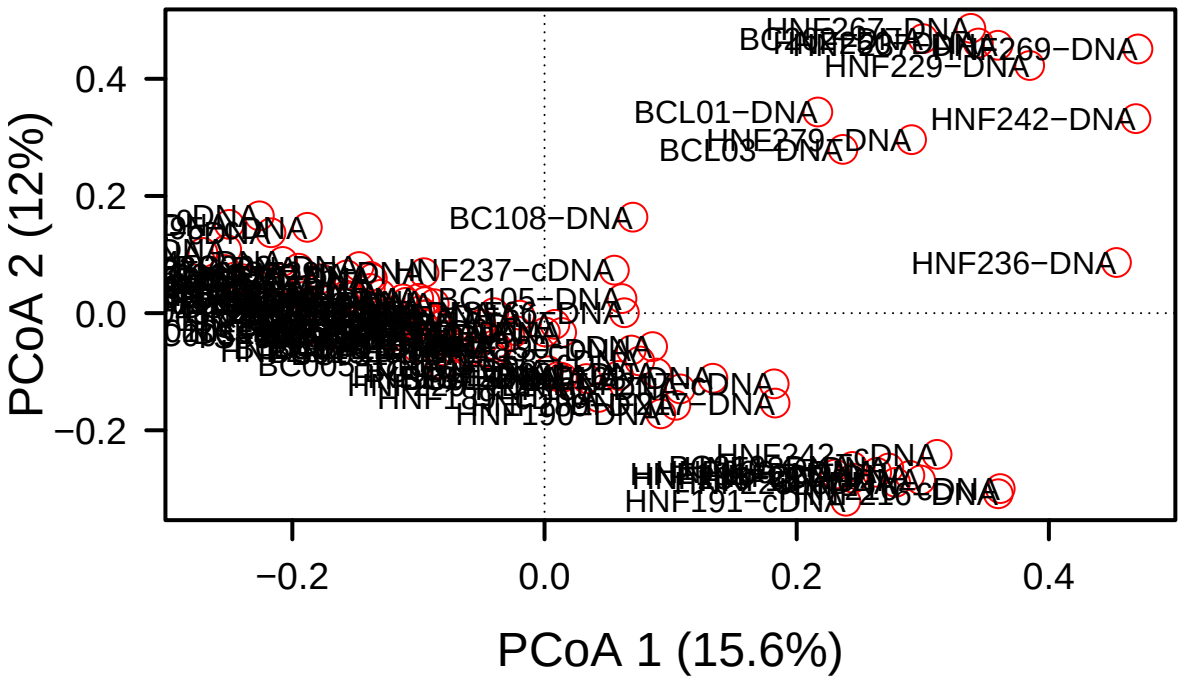
Load the dataset from that you and your partner are using for the team project. Use one of the tools introduced in the beta diversity module to visualize your data. Describe any interesting patterns and identify a hypothesis is relevant to the principles of biodiversity.

> Answer: Based on the PCoA plot shown below with the sample names overlain on the points, we can see that there is a lot of clustering on the left side of the plot, comprised of both active (cDNA) and total (DNA) samples. This could lead us to think that there is little to no difference in the composition of active and total taxa within the ponds, which may lead us to conclude that dormant microbes may not be as common in the ponds as we originally thought.

```r
# Inputting Data
# relative abundance matrix
Ponds.rel <- load("/Users/joyobrien/Desktop/INPond_Initial.RData")
as.matrix(Ponds.rel)
```

```
##       [,1]
##  [1,] ".Random.seed"
##  [2,] "coverage"
##  [3,] "design"
##  [4,] "i"
##  [5,] "Pond97"
##  [6,] "PondsPA"
##  [7,] "PondsREL"
##  [8,] "read.otu"
##  [9,] "read.tax"
## [10,] "rich"
## [11,] "se"
## [12,] "shared"
```

```r
#sites 54-58 are missing some environmental data

# Bray Curtis resemblance matrix
Ponds.db <- vegdist(Pond97, method="bray")

# Principal Component Analysis - TOTAL
Ponds.pcoa <- cmdscale(Ponds.db, eig=TRUE, k=3)

exvar1 <- round(Ponds.pcoa$eig[1] / sum(Ponds.pcoa$eig), 3) * 100
exvar2 <- round(Ponds.pcoa$eig[2] / sum(Ponds.pcoa$eig), 3) * 100
exvar3 <- round(Ponds.pcoa$eig[3] / sum(Ponds.pcoa$eig), 3) * 100
total.sum.eig <- sum(exvar1, exvar2, exvar3)

# PCoA Plot PC1 x PC2
plot(Ponds.pcoa$points[ ,1], Ponds.pcoa$points[ ,2],
     xlab= paste("PCoA 1 (", exvar1, "%)", sep = ""),
     ylab= paste("PCoA 2 (", exvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5,
     cex.axis=1.2, axes=FALSE);
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1);
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1);
abline(h = 0, v = 0, lty = 3);
box(lwd = 2);
points(Ponds.pcoa$points[ ,1], Ponds.pcoa$points[ ,2],
       pch = 1, cex = 2, bg = "red", col = "red");
text(Ponds.pcoa$points[ ,1], Ponds.pcoa$points[ ,2],
     labels = row.names(Ponds.pcoa$points), adj=1)
```


HNF267−DNA
HNF269−DNA
HNF229−DNA
BCL01−DNA
HNF242−DNA
HNF279−DNA
BCL03−DNA
BC108−DNA
HNF236−DNA
HNF237−cDNA
BC105−DNA
HNF242−cDNA
HNF191−cDNA
0.4
0.2
0.0
−0.2
−0.2
0.0
0.2
0.4
PCoA 2 (12%)
PCoA 1 (15.6%)