7. Worksheet: Diversity Synthesis

Anna Werkowski; Z620: Quantitative Biodiversity, Indiana University

15 February, 2023

OVERVIEW

In this worksheet, you will conduct exercises that reinforce fundamental concepts of biodiversity. Specifically, you will construct a a site-by-species matrix by sampling confectionery taxa. With this primary data structure, you will then answer questions and generate figures using tools from previous weeks, along with wrangling techniques that we learned about in class.

Directions:

1. In the Markdown version of this document in your cloned repo, change "Student Name" on line 3 (above) to your name.
2. Complete as much of the worksheet as possible during class.
3. Refer to previous handouts to help with developing of questions and writing of code.
4. Answer questions in the worksheet. 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 ">". 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, **push** this file to your GitHub repo.
6. For the assignment portion of the worksheet, follow the directions at the bottom of this file.
7. When you are done, **Knit** the text and code into a PDF file.
8. After Knitting, submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `7.DiversitySynthesis_Worskheet.Rmd` and the PDF output of `Knitr` (`DiversitySynthesis_Worskheet.pdf`).

CONFECTIONARY EXERCISE GOALS

We will construct a site-by-species matrix using confectionery taxa (i.e, gummies). The instructors have created distinct **sources communities** that vary in the composition of gummy taxa with even and uneven communities. It might be fun to consider them as distinct geographical regions experiencing different environmental regimes, or different experimental units under different treatments. Each student will sample a source community and then use a taxonomic key to identify gummies and their abundances.

In the end, students will use the site-by-species matrix to:

1) explore their sampling efforts and their effects on species richness using **coverage** and **rarefaction** concept,

2) measure **alpha diversity** for each sub-sample collated from data with their peers from the same source community,

3) examine **beta diversity** between each source community using the data generated across each source community, and

4) use **data wrangling** tools they have learned during the class to accomplish the above goals.

SAMPLING PROTOCOL TO CONSTRUCT A SITE-BY-SPECIES MATRIX

1. Instructors will assign you to sample confectionery taxa from one of the two designated source community bucket (A and B).

2. After randomly sampling one unit (imagine as an equal biomass) from the source community, each student will count the total number of individuals (N), identify the taxa using the species key and quantify the abundance of each taxon.

3. Work with other students in your group to assemble data into a site-by-species matrix on the white board. One person needs to create a .csv or .txt file and share your group's site-by-species matrix with the class using GitHub. Make sure that you include a sample identifier (student name) and what community you sampled from.

```
getwd()
```

```
## [1] "/Users/annawerkowski/Documents/Github/QB2023_Werkowski/2.Worksheets/7.DiversitySynthesis"
```

```
setwd("/Users/annawerkowski/Documents/Github/QB2023_Werkowski/2.Worksheets/7.DiversitySynthesis")
dat <- read.table('~/Documents/Gummies.csv', header = TRUE, sep = ",")

package.list <- c('vegan', 'tidyverse', 'ggplot2', 'dplyr', 'broom')
for (package in package.list) {
  if (!require(package, character.only = TRUE, quietly = TRUE)) {
    install.packages(package)
  }
}
```

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
dat <- dat[,-1]
dat[1:30] = lapply(dat[1:30], FUN = function(y){as.numeric(y)})
str(dat)
```

```
## 'data.frame':    8 obs. of  30 variables:
##  $ SP1 : num  1 0 2 0 11 9 7 11
##  $ SP2 : num  2 4 0 4 3 10 16 8
##  $ SP3 : num  1 3 1 0 2 5 2 6
##  $ SP4 : num  2 0 0 4 1 10 1 3
##  $ SP5 : num  1 3 0 5 4 3 4 4
##  $ SP6 : num  2 2 6 1 2 1 2 1
##  $ SP7 : num  2 1 0 4 5 0 0 2
##  $ SP8 : num  2 2 0 0 0 1 5 4
##  $ SP9 : num  1 1 2 1 4 3 4 4
##  $ SP10: num  1 1 1 0 5 0 2 5
##  $ SP11: num  1 5 1 3 4 1 1 1
##  $ SP12: num  1 4 2 3 4 1 4 3
##  $ SP13: num  2 1 1 4 4 0 1 3
##  $ SP14: num  5 3 3 2 4 4 1 3
##  $ SP15: num  1 1 1 0 2 4 0 2
##  $ SP16: num  5 1 1 4 2 1 2 3
##  $ SP17: num  1 1 2 0 4 4 3 1
##  $ SP18: num  0 0 0 2 1 4 3 2
##  $ SP19: num  2 1 1 2 0 0 0 2
##  $ SP20: num  2 0 2 2 1 1 6 0
##  $ SP21: num  1 3 1 1 0 3 1 3
##  $ SP22: num  1 1 2 0 0 2 1 2
##  $ SP23: num  4 1 0 2 0 1 0 1
##  $ SP24: num  2 1 2 4 1 0 1 1
##  $ SP25: num  0 1 2 3 0 1 1 0
##  $ SP26: num  3 2 5 2 0 1 3 0
##  $ SP27: num  2 2 3 2 0 0 2 0
##  $ SP28: num  4 2 2 2 0 1 0 1
##  $ SP29: num  5 0 1 0 0 1 0 1
##  $ SP30: num  2 1 2 4 1 2 3 0
```

```r
#input Canan's fix for the spec scores problem
`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)
  }
```

GROUP BRAINSTORM

In smaller groups, take 15 minutes to brainstorm questions, code, statistical tests, and “fantasy figures” using the site-by-species matrix the class generated.

1. Using this data, explore how well your sampling effort was. You can use rarefaction and coverage tools you have learned earlier.

2. Investigate alpha diversity based on the methods you have learned in the rest of the handout and accompanying worksheet. For example, you can measure richness, Shannon diversity and Simpson index. You can also convert them to effective number of species using the Hill numbers concept.

3. Measure beta diversity using ordination and multivariate statistical methods. For example, you can create a PCoA plot, based on Bray-Curtis dissimilarity, of sites and communities using different shape and color codes. Use Permanova to test if there are differences between communities.

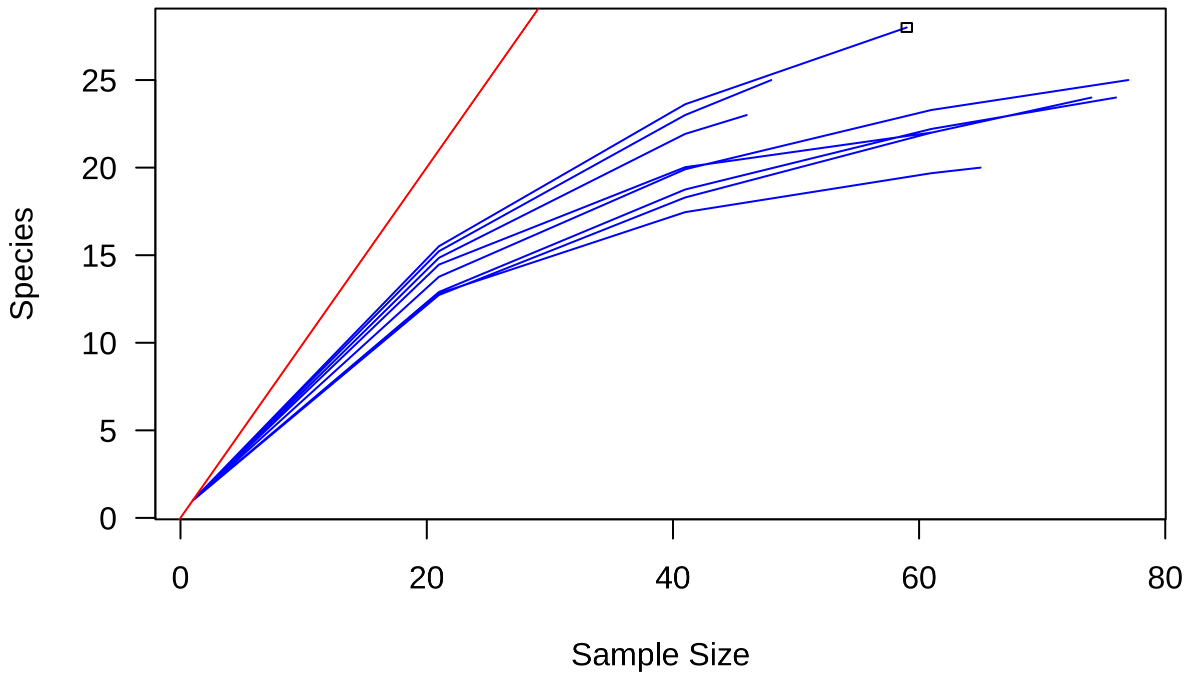
DATA ANALYSIS

1) Sampling coverage and rarefaction curves

Question 1: Using this data, explore how well your sampling effort was. Compare your sampling efforts with other groups. Do you think that your samples cover the actual diversity found in each source community? You can use rarefaction and coverage tools you have learned earlier.

Answer 1: Use the space below to generate a rarefaction curve/sample coverage based on the data we collected in class for each community. Make sure to annotate your code using # symbols so others (including instructors) understand what you have done and why you have done it.

```
#create the function for observed richness
S.obs <- function(x = ""){
  rowSums(x > 0) * 1
}
#generate the Chao2 function
S.chao2 <- function(site = "", SbyS = ""){
  SbyS = as.data.frame(SbyS)
  x = SbyS[site, ]
  Sbys.pa <- (SbyS > 0) * 1
  Q1 = sum(colSums(Sbys.pa) == 1)
  Q2 = sum(colSums(Sbys.pa) == 2)
  S.chao2 = S.obs(x) + (Q1^2)/(2*Q2)
  return(S.chao2)
}
#calculate rarefaction
rare.dat <- S.obs(dat)
min.N <- min(rowSums(dat))
S.rarefy <- rarefy(x = dat, sample = min.N, se = TRUE)
rarecurve(x = dat, step = 20, col = "blue", cex = 0.6, las = 1)
abline(0,1, col = 'red')
text(1500, 1500, "1:1", pos = 2, col = 'red')
```



```r
#calculate Good's coverage
coverage <- function(x = ""){
  1 - (rowSums(x == 1)/rowSums(x))
}
#using the Good's coverage function on the gummy data set
coverage(dat)
```

```
## [1] 0.8135593 0.7291667 0.8043478 0.9508197 0.9230769 0.8513514 0.8947368
## [8] 0.9090909
```

2) Alpha diversity

Question 2: Compare alpha diversity measures within sites and among communities. You can calculate and plot richness, Shannon diversity, and Simpson index. You can also convert these indices to effective number of species using the Hill numbers concept by generating a diversity profile, which will make comparisons easier across sites.

What is the variation among the samples in your group and between the communities of other groups for the alpha diversity indices? Generate a hypothesis around the diversity metrics you chose and test your hypothesis. Interpret your findings.

Answer 2a - Analysis: Use the space below for code that is being used to analyze your data and test your hypotheses on your chosen alpha diversity tool. Make sure to annotate your code using # symbols so others (including instructors) understand what you have done and why you have done it.

```r
#create a Shannon Diversity function
ShanH <- function(x = ""){
  H = 0
  for (n_i in x) {
    if (n_i > 0) {
      p = n_i/sum(x)
      H = H - p*log(p)
    }
  }
  return(H)
}
#calculate the Shannon Diversity of the Gummy data set
diversity(dat, index = "shannon")
```

```
## [1] 3.166063 3.053443 2.975329 2.998721 2.787795 2.834947 2.836906 2.978193
```

```r
#calculate Simpson's Diversity
SimpD <- function(x = ""){
  D = 0
  N = sum(x)
  for (n_i in x) {
    D = D + (n_i^2)/(N^2)
  }
  return(D)
}
#express Simpson's diversity as 1/D (d.inv) and 1-D (d.sub)
d.inv <- 1/SimpD(dat)
d.sub <- 1 - SimpD(dat)
```

```
#use vegan to estimate Simpson's index
diversity(dat, "inv")
```

```
## [1] 20.35673 18.28571 16.53125 18.69849 13.49840 13.16346 12.08368 15.81067
```

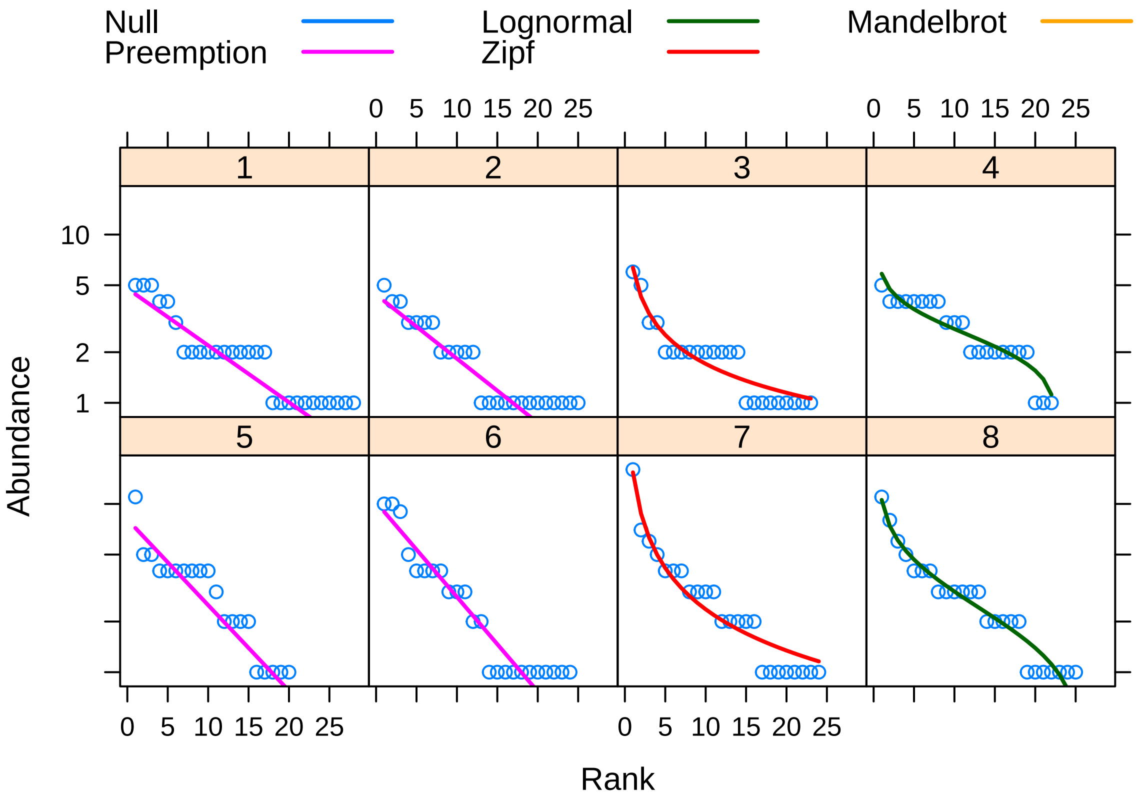
```
diversity(dat, "simp")
```

```
## [1] 0.9508762 0.9453125 0.9395085 0.9465198 0.9259172 0.9240321 0.9172438
## [8] 0.9367516
```

Answer 2b - Plot: With your analysis, create one (and only one, although it can have multiple panels) *publication-quality* figure.

```
RACresults <- radfit(dat)
plot.new()
```

```
plot(RACresults, las = 1, cex.lab = 1.4, cex.axis = 1.25)
```



Answer 2c - Interpret results: Write an informative yet succinct (~5 sentences) caption that creates a "stand-alone" figure. Take a peek at figures and figure captions in a paper published in your favorite journal for inspiration. > Samples from Community A (represented in this figure as Sites 1-4) were incredibly similar to samples from Community B (represented in this figure as Sites 5-8). Some small differences that can be identified between the samples of the two communities are as follows: samples from Community A had an average maximum abundance level of 5 while the samples from Community B had an average maximum abundance level of 10 or above. Due to their average higher maximum abundance level and their steep gradient, it can be inferred that Community B had a lower evenness than Community A. Community B has higher abundances of higher-ranking species compared to the samples from Community A.

3) Beta diversity

Question 3: Measure beta diversity using ordination and multivariate statistics methods. You can create a PCoA plot, based on Bray-Curtis dissimilarity, of sites and communities using different shape and color codes. Then, you can use a Permanova to test if there are differences between communities. Generate a hypothesis around your chosen analysis and test your hypothesis. Interpret your findings.

Can you detect compositional differences between each source community sampled?

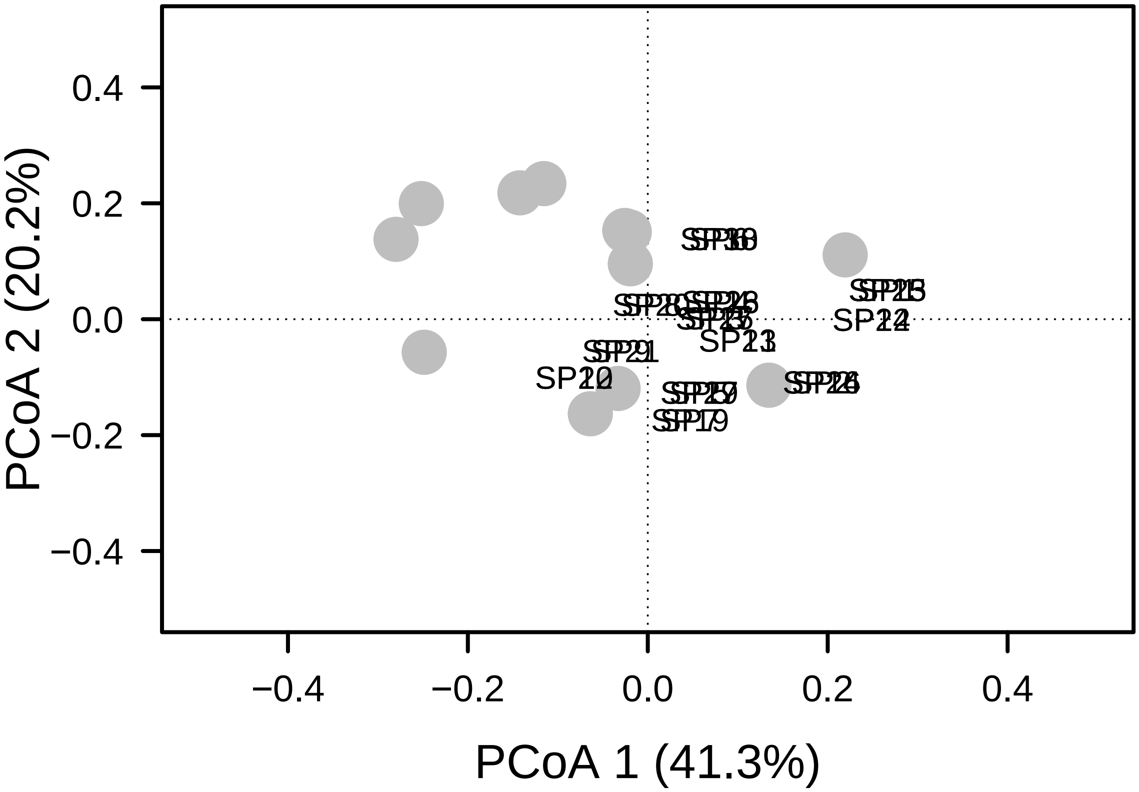
Answer 3a - Analysis: Use the space below for code that is being used to analyze your data and test your hypotheses on your chosen beta diversity tool. Make sure to annotate your code using # symbols so others (including instructors) understand what you have done and why you have done it.

```
#calculate beta diversity
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)
```

```r
  return(b.w)
}
beta.w <- 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(b.w)
  }
}
#calculate Jaccard
gummy.dj <- vegdist(dat, method = "jaccard", binary = TRUE)
#calculate bray-curtis
gummy.db <- vegdist(dat, method = "bray")
#calculate sorenson
gummy.ds <- vegdist(dat, method = "bray", binary = TRUE)
#create a square resemblance matrix
gummy.db <- vegdist(dat, method = "bray", upper = TRUE, diag = TRUE)
print(gummy.db)
```

```
##           1         2         3         4         5         6         7
## 1 0.0000000 0.4018692 0.3904762 0.4000000 0.5645161 0.5789474 0.5407407
## 2 0.4018692 0.0000000 0.4255319 0.4128440 0.4690265 0.5081967 0.4677419
## 3 0.3904762 0.4255319 0.0000000 0.5327103 0.6036036 0.5833333 0.5245902
## 4 0.4000000 0.4128440 0.5327103 0.0000000 0.4920635 0.5851852 0.5182482
## 5 0.5645161 0.4690265 0.6036036 0.4920635 0.0000000 0.4532374 0.4184397
## 6 0.5789474 0.5081967 0.5833333 0.5851852 0.4532374 0.0000000 0.4000000
## 7 0.5407407 0.4677419 0.5245902 0.5182482 0.4184397 0.4000000 0.0000000
## 8 0.5000000 0.4400000 0.6097561 0.4927536 0.2957746 0.3112583 0.3856209
##           8
## 1 0.5000000
## 2 0.4400000
## 3 0.6097561
## 4 0.4927536
## 5 0.2957746
## 6 0.3112583
## 7 0.3856209
## 8 0.0000000
```

```
#calculate ordination and PCoA
gummy.pcoa <- cmdscale(gummy.db, eig = TRUE, k = 3)
explainvar1 <- round(gummy.pcoa$eig[1] / sum(gummy.pcoa$eig), 3) * 100
explainvar2 <- round(gummy.pcoa$eig[2] / sum(gummy.pcoa$eig), 3) * 100
explainvar3 <- round(gummy.pcoa$eig[3] / sum(gummy.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

#define plot parameters & initiate plot
par(mar = c(5,5,1,2) + 0.1)
plot(gummy.pcoa$points[1:4, ], gummy.pcoa$points[5:8, ],
     ylim = c(-0.5, 0.5), xlim = c(-0.5, 0.5),
     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(gummy.pcoa$points[1:4, ], gummy.pcoa$points[5:8, ],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(gummy.pcoa$points[1:4, ], gummy.pcoa$points[5:8, ],
     labels = row.names(gummy.pcoa$points))

#calculate the relative abundances of each species at each site
gummyREL <- dat
  for (i in 1:nrow(dat)){
    gummyREL[i, ] = dat[i, ] / sum(dat[i, ])
  }
#calculate and add species scores
gummy.pcoa <- add.spec.scores.class(gummy.pcoa, gummyREL, method = "pcoa.scores")
text(gummy.pcoa$cproj[1:4, ], gummy.pcoa$cproj[5:8, ],
     labels = row.names(gummy.pcoa$cproj), col = "black")
```



```r
spe.corr <- add.spec.scores.class(gummy.pcoa, gummyREL, method = "corr.scores")$cproj
corrcut <- 0.5
imp.spp <- spe.corr[abs(spe.corr[1,]) >= corrcut | abs(spe.corr[2,]) >= corrcut, ]

#permutation test for species abundance across axes
fit <-  envfit(gummy.pcoa, gummyREL, perm = 999)
```

I really struggled with the PCoA plot and tried to tweak it multiple times. I am unsure as to whether it is correct or not. I tried to follow along with the original handout but I didn't know how to change the variables to reflect this new, different data set.

```r
#calculate PERMANOVA
#create 'factors' vector
gummyquality <- c(rep("A", 4), rep("B", 4))
#run PERMANOVA with adonis function
adonis2(dat ~ gummyquality, method = "bray", permutations = 999)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dat ~ gummyquality, permutations = 999, method = "bray")
##              Df SumOfSqs      R2      F Pr(>F)
## gummyquality  1  0.31895 0.39151 3.8605  0.023 *
## Residual      6  0.49571 0.60849
## Total         7  0.81466 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 3a Answer Generate a hypothesis around your chosen analysis and test your hypothesis. Interpret your findings.Can you detect compositional differences between each source community sampled? > My hypothesis is that there are compositional differences between the two communities sampled. To test this hypothesis, I generated a PCoA plot and conducted a PERMANOVA analysis. The PERMANOVA analysis produced a P-value of 0.026* indicating a relatively significant difference between samples from Community A and Community B. This was further supported by the PCoA plot, where PCoA Axis 1 explained 41.3% of the variance while PCoA Axis 2 only explained 20.2% of the variance. I would say that you could infer compositional differences between the communities sampled but I would need more plot refining to be more confident.

Answer 3b - Plot: With your analysis, create one (and only one, although it can have multiple panels) *publication-quality* figure.

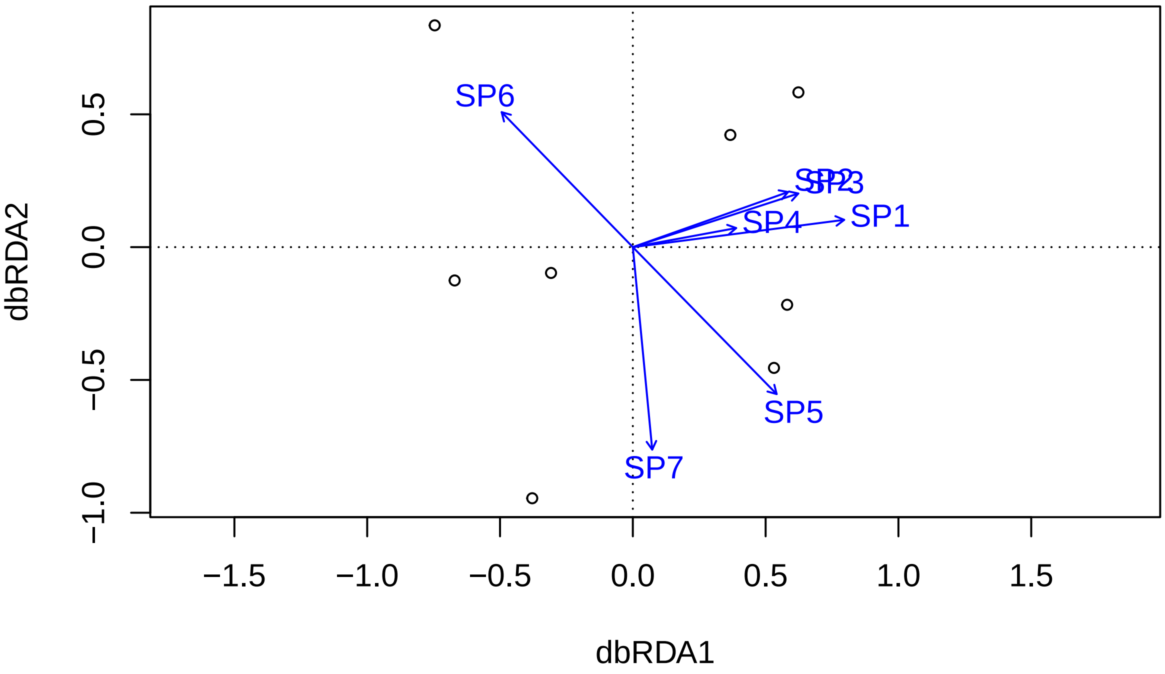
```
#perform dbrda
#dbrda was revised multiple times because the mod0 and mod1 portions that included the OrdiR2Step funct

gummy.dbrda <- dbrda(gummy.db ~ ., as.data.frame(dat))
ordiplot(gummy.dbrda)
gummy.dbrda$call
```

```
## dbrda(formula = gummy.db ~ SP1 + SP2 + SP3 + SP4 + SP5 + SP6 +
##     SP7 + SP8 + SP9 + SP10 + SP11 + SP12 + SP13 + SP14 + SP15 +
##     SP16 + SP17 + SP18 + SP19 + SP20 + SP21 + SP22 + SP23 + SP24 +
##     SP25 + SP26 + SP27 + SP28 + SP29 + SP30, data = as.data.frame(dat))
```

```
gummy.dbrda$anova
```

```
## NULL
```

```
ordiplot(gummy.dbrda)
```



```
permutest(gummy.dbrda, permutations = 999)
```

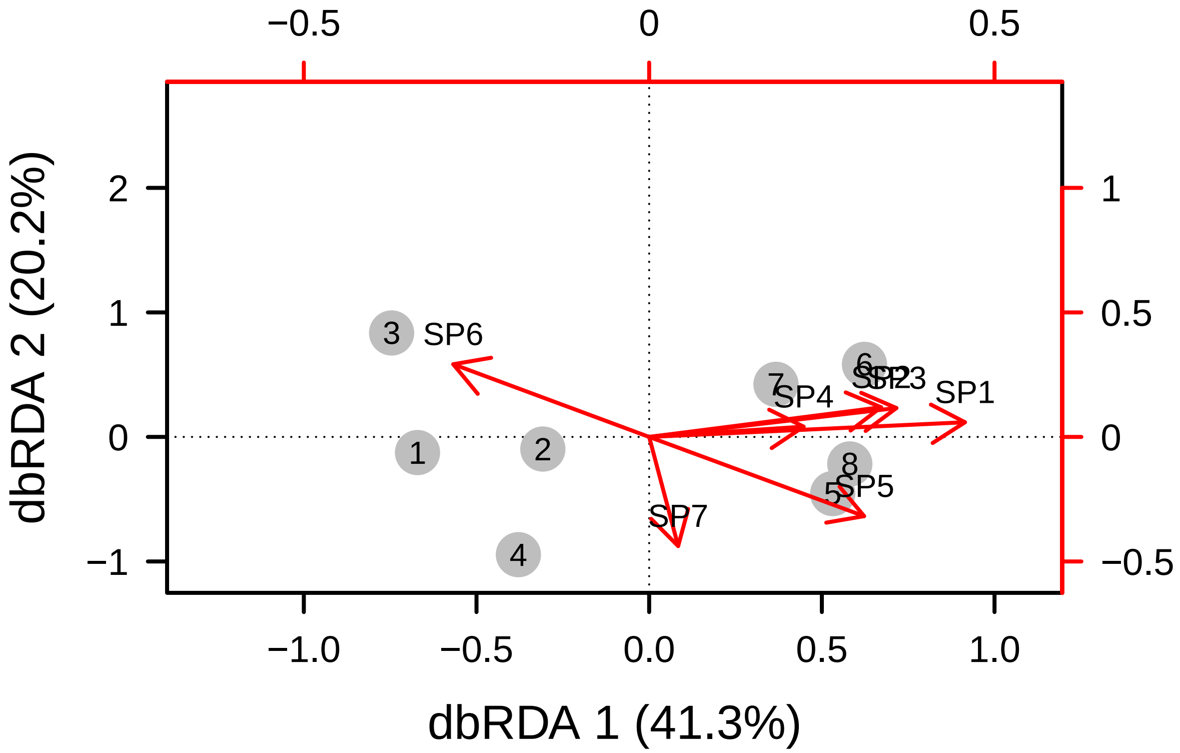
```
##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = gummy.db ~ SP1 + SP2 + SP3 + SP4 + SP5 + SP6 +
## SP7 + SP8 + SP9 + SP10 + SP11 + SP12 + SP13 + SP14 + SP15 + SP16 + SP17
## + SP18 + SP19 + SP20 + SP21 + SP22 + SP23 + SP24 + SP25 + SP26 + SP27 +
## SP28 + SP29 + SP30, data = as.data.frame(dat))
## Permutation test for all constrained eigenvalues
##          Df Inertia   F Pr(>F)
## Model     7 0.81466 NaN
## Residual  0 0.00000
```

```
envfit(gummy.dbrda, dat[,c(4,6,7)], perm = 999)
```

```
##
## ***VECTORS
##
##       dbRDA1   dbRDA2     r2 Pr(>r)
## SP4  0.98342  0.18136 0.2068  0.604
## SP6 -0.69691  0.71716 0.6628  0.032 *
## SP7  0.09539 -0.99544 0.7749  0.018 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## Permutation: free
## Number of permutations: 999
```

```
#calculate explained variation
dbrda.explainvar1 <- round(gummy.dbrda$CCA$eig[1] / sum(c(gummy.dbrda$CCA$eig, gummy.dbrda$CA$eig)), 3)
dbrda.explainvar2 <- round(gummy.dbrda$CCA$eig[2] / sum(c(gummy.dbrda$CCA$eig, gummy.dbrda$CA$eig)), 3)

#plot ordination for selected model
#define plot parameters
par(mar = c(5,5,4,4) + 0.1)
#initiate plot
plot(scores(gummy.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 and labels
points(scores(gummy.dbrda, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
```

```
text(scores(gummy.dbrda, display = "wa"),
     labels = row.names(scores(gummy.dbrda, display = "wa")))
#add vectors
vectors <- scores(gummy.dbrda, display = "bp")
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])))
```



Answer 3c - Interpret results: Write an informative yet succinct (~5 sentences) caption that creates a "stand-alone" figure. Take a peek at figures and figure captions in a paper published in your favorite journal for inspiration. > CAVEAT: I am still very unsure about how to interpret these figures. If we could go over how to interpret this in class it would be highly appreciated! > Most of the variance from the analysis was explained by dbRDA 1 (41.3%) while only 20.2% of the variance was explained by dbRDA 2. Many of the points fell to the right side of the vertical axis at zero. There seemed to be a distinct separation between the left and right sides of the figure, as shown by the red arrows.

SUBMITTING YOUR ASSIGNMENT

Use Knitr to create a PDF of your completed 7.DiversitySynthesis_Worksheet.Rmd document, push it to GitHub, and create a pull request. Please make sure your updated repo includes both the pdf and RMarkdown files.

Unless otherwise noted, this assignment is due on **Wednesday, February 15th, 2023 at 12:00 PM (noon)**.