Assignment: Local (alpha) Diversity

Venus Kuo; Z620: Quantitative Biodiversity, Indiana University

OVERVIEW

In this exercise, we will explore aspects of local or site-specific diversity, also known as alpha (α) diversity. First we will quantify two of the fundamental components of (α) diversity: **richness** and **evenness**. From there, we will then discuss ways to integrate richness and evenness, which will include univariate metrics of diversity along with an investigation of the **species abundance distribution (SAD)**.

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. For homework, 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, please submit the completed exercise by creating a **pull request** via GitHub. Your pull request should include this file `alpha_assignment.Rmd` and the PDF output of `Knitr` (`alpha_assignment.pdf`).

1) R SETUP

In the R code chunk below, please provide the code to: 1) Clear your R environment, 2) Print your current working directory, 3) Set your working directory to your `/Week2-Alpha` folder, and 4) Load the `vegan` R package (be sure to install if needed).

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

```
## [1] "C:/Users/Venus/Github/QB2017_Kuo/Week2-Alpha"
```

```
setwd("C:/Users/Venus/Github/QB2017_Kuo/Week2-Alpha/data/") #QuantitativeBiodiversity/
#install.packages("vegan")
require("vegan")
```

```
## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.4-1
```

2) LOADING DATA

In the R code chunk below, do the following: 1) Load your dataset, and 2) Display the structure of the dataset (if the structure is long, use `max.level=0` to show just basic information).

```
data("BCI")
str(BCI)
```

```
## 'data.frame':    50 obs. of  225 variables:
##  $ Abarema.macradenia              : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Vachellia.melanoceras           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.diversifolia           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Acalypha.macrostachya           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Adelia.triloba                  : int  0 0 0 3 1 0 0 0 5 0 ...
##  $ Aegiphila.panamensis            : int  0 0 0 0 1 0 1 0 0 1 ...
##  $ Alchornea.costaricensis         : int  2 1 2 18 3 2 0 2 2 2 ...
##  $ Alchornea.latifolia             : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Alibertia.edulis                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Allophylus.psilospermus         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Alseis.blackiana                : int  25 26 18 23 16 14 18 14 16 14 ...
##  $ Amaioua.corymbosa               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Anacardium.excelsum             : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Andira.inermis                  : int  0 0 0 0 1 1 0 0 1 0 ...
##  $ Annona.spraguei                 : int  1 0 1 0 0 0 0 1 1 0 ...
##  $ Apeiba.glabra                   : int  13 12 6 3 4 10 5 4 5 5 ...
##  $ Apeiba.tibourbou                : int  2 0 1 1 0 0 0 1 0 0 ...
##  $ Aspidosperma.desmanthum         : int  0 0 0 1 1 1 0 0 0 1 ...
##  $ Astrocaryum.standleyanum        : int  0 2 1 5 6 2 2 0 2 1 ...
##  $ Astronium.graveolens            : int  6 0 1 3 0 1 2 2 0 0 ...
##  $ Attalea.butyracea               : int  0 1 0 0 0 1 1 0 0 0 ...
##  $ Banara.guianensis               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Beilschmiedia.pendula           : int  4 5 7 5 8 6 5 9 11 14 ...
##  $ Brosimum.alicastrum             : int  5 2 4 3 2 2 6 4 3 6 ...
##  $ Brosimum.guianense              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Calophyllum.longifolium         : int  0 2 0 2 1 2 2 2 2 0 ...
##  $ Casearia.aculeata               : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Casearia.arborea                : int  1 1 3 2 4 1 2 3 9 7 ...
##  $ Casearia.commersoniana          : int  0 0 1 0 1 0 0 0 1 0 ...
##  $ Casearia.guianensis             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Casearia.sylvestris             : int  2 1 0 0 0 3 1 0 1 1 ...
##  $ Cassipourea.guianensis          : int  2 0 1 1 3 4 4 0 2 1 ...
##  $ Cavanillesia.platanifolia       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cecropia.insignis               : int  12 5 7 17 21 4 0 7 2 16 ...
##  $ Cecropia.obtusifolia            : int  0 0 0 0 1 0 0 2 0 2 ...
##  $ Cedrela.odorata                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ceiba.pentandra                 : int  0 1 1 0 1 0 0 1 0 1 ...
##  $ Celtis.schippii                 : int  0 0 0 2 2 0 1 0 0 0 ...
##  $ Cespedesia.spathulata           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chamguava.schippii              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chimarrhis.parviflora           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Maclura.tinctoria               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysochlamys.eclipes           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Chrysophyllum.argenteum         : int  4 1 2 2 6 2 3 2 4 2 ...
##  $ Chrysophyllum.cainito           : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Coccoloba.coronata              : int  0 0 0 1 2 0 0 1 2 1 ...
```

```
##  $ Coccoloba.manzinellensis       : int  0 0 0 0 0 0 0 2 0 0 ...
##  $ Colubrina.glandulosa           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cordia.alliodora               : int  2 3 3 7 1 1 2 0 0 2 ...
##  $ Cordia.bicolor                 : int  12 14 35 23 13 7 5 10 7 13 ...
##  $ Cordia.lasiocalyx              : int  8 6 6 11 7 6 6 3 0 4 ...
##  $ Coussarea.curvigemma           : int  0 0 0 1 0 2 1 0 1 1 ...
##  $ Croton.billbergianus           : int  2 2 0 11 6 0 0 4 2 0 ...
##  $ Cupania.cinerea                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.latifolia              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Cupania.rufescens              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cupania.seemannii              : int  2 2 1 0 3 0 1 2 2 0 ...
##  $ Dendropanax.arboreus           : int  0 3 6 0 5 2 1 6 1 3 ...
##  $ Desmopsis.panamensis           : int  0 0 4 0 0 0 0 0 0 1 ...
##  $ Diospyros.artanthifolia        : int  1 1 1 1 0 0 0 0 0 1 ...
##  $ Dipteryx.oleifera              : int  1 1 3 0 0 0 0 2 1 2 ...
##  $ Drypetes.standleyi             : int  2 1 2 0 0 0 0 0 0 0 ...
##  $ Elaeis.oleifera                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Enterolobium.schomburgkii      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Erythrina.costaricensis        : int  0 0 0 0 0 3 0 0 1 0 ...
##  $ Erythroxylum.macrophyllum      : int  0 1 0 0 0 0 0 1 1 1 ...
##  $ Eugenia.florida                : int  0 1 0 7 2 0 0 1 1 3 ...
##  $ Eugenia.galalonensis           : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Eugenia.nesiotica              : int  0 0 1 0 0 0 5 4 3 0 ...
##  $ Eugenia.oerstediana            : int  3 2 5 1 5 2 2 3 3 3 ...
##  $ Faramea.occidentalis           : int  14 36 39 39 22 16 38 41 33 42 ...
##  $ Ficus.colubrinae               : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Ficus.costaricana              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.insipida                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.maxima                   : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.obtusifolia              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.popenoei                 : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Ficus.tonduzii                 : int  0 0 1 2 1 0 0 0 0 0 ...
##  $ Ficus.trigonata                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ficus.yoponensis               : int  1 0 0 0 0 1 1 0 0 0 ...
##  $ Garcinia.intermedia            : int  0 1 1 3 2 1 2 2 1 0 ...
##  $ Garcinia.madruno               : int  4 0 0 0 1 0 0 0 0 1 ...
##  $ Genipa.americana               : int  0 0 1 0 0 0 1 0 1 1 ...
##  $ Guapira.myrtiflora             : int  3 1 0 1 1 7 3 1 1 1 ...
##  $ Guarea.fuzzy                   : int  1 1 0 1 3 0 0 2 0 3 ...
##  $ Guarea.grandifolia             : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Guarea.guidonia                : int  2 6 2 5 3 4 4 0 1 5 ...
##  $ Guatteria.dumetorum            : int  6 16 6 3 9 7 8 6 2 2 ...
##  $ Guazuma.ulmifolia              : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Guettarda.foliacea             : int  1 5 1 2 1 0 0 4 1 3 ...
##  $ Gustavia.superba               : int  10 5 0 1 3 1 8 4 4 4 ...
##  $ Hampea.appendiculata           : int  0 0 1 0 0 0 0 0 2 1 ...
##  $ Hasseltia.floribunda           : int  5 9 4 11 9 2 7 6 3 4 ...
##  $ Heisteria.acuminata            : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ Heisteria.concinna             : int  4 5 4 6 4 8 2 5 1 5 ...
##  $ Hirtella.americana             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hirtella.triandra              : int  21 14 5 4 6 6 7 14 8 7 ...
##  $ Hura.crepitans                 : int  0 0 0 0 0 2 1 1 0 0 ...
##  $ Hieronyma.alchorneoides        : int  0 2 0 0 0 0 0 0 1 0 ...
##   [list output truncated]
```

```
##  - attr(*, "original.names")= chr  "Abarema.macradenium" "Acacia.melanoceras" "Acalypha.diversifolia
```

3) SPECIES RICHNESS

Species richness (S) is simply the number of species in a system or the number of species observed in a sample.

Observed Richness

In the R code chunk below, do the following:

1. Write a function called `S.obs` to calculate observed richness
2. Use your function to determine the number of species in `site1`, and
3. Compare the output of your function to the output of the `specnumber()` function in vegan.

```
S.obs <- function(x=""){
  rowSums(x>0) *1
   }

site.1 <- BCI[1,]

S.obs(site.1)
```

```
##  1
## 93
```

```
specnumber(site.1)
```

```
##  1
## 93
```

```
sites <- BCI[1:4,]
rich.sites <- S.obs(sites)
```

Question 1: Does `specnumber()` from `vegan` return the same value for observed richness in `site1` as our function `S.obs`? What is the species richness of the first 4 sites (i.e., rows) of the BCI matrix?

> ***Answer 1***: # Yes it does. The first 4 sites (1, 2, 3, and 4) have the richness of 93, 84, 90, and 94 respectively.

Coverage. How Well Did You Sample Your Site?

In the R code chunk below, do the following:

1. Write a function to calculate Good's Coverage, and
2. Use that function to calculate coverage for all sites in the BCI matrix.

```
C <- function(x = ""){
1 - (sum(x == 1) / rowSums(x)) }

all.sites <- BCI[1:50,]
coverage <- C(all.sites)
range(coverage)
```

```
## [1] -4.108824 -1.890183
```

```
site1.coverage <- C(site.1)
```

Question 2: Answer the following questions about coverage:

a. What is the range of values that can be generated by Good's Coverage?
b. What would we conclude from Good's Coverage if n_i equaled N?
c. What portion of taxa in `site1` were represented by singletons?
d. Make some observations about coverage at the BCI plots.

Answer 2a:
The range is from -4.108824 to -1.890183. ***Answer 2b***:
The Good's coverage would be 0. ***Answer 2c***:
0.93 proportion is represented by singletons. ***Answer 2d***:
The coverage all seem to be relatively the same across sites. So the coverage across sites are comparable.

Estimated Richness

In the R code chunk below, do the following:

1. Load the microbial dataset (located in the `/Week2-Alpha/data` folder),
2. Transform and transpose the data as needed (see handout),
3. Create a vector (`soilbac1`) with the bacterial OTU abundances at any site in the dataset,
4. Calculate the observed richness at that particular site, and
5. Calculate the coverage at that particular site

```
soilbac <- read.table("data/soilbac.txt", sep = "\t", header = TRUE, row.names = 1)
soilbac.t <- as.data.frame(t(soilbac))
soilbac1 <- soilbac.t[1,]
soilbac1.richness <- S.obs(soilbac1)
soilbac1.coverage <- C(soilbac1)
soilbac1.N <- sum(soilbac1)
soilbac1.richness
```

```
## T1_1
## 1074
```

```
soilbac1.coverage
```

```
##      T1_1
## 0.6479471
```

```
site1.coverage
```

```
##         1
## 0.9308036
```

```
soilbac1.N
```

```
## [1] 2119
```

Question 3: Answer the following questions about the soil bacterial dataset.

a. How many sequences did we recover from the sample `soilbac1`, i.e. N?
b. What is the observed richness of `soilbac1`?
c. How does coverage compare between the BCI sample (`site1`) and the KBS sample (`soilbac1`)?

Answer 3a:
2119 sequences were recovered from soilbac1 ***Answer 3b***: # 1074 is the observed richness ***Answer 3c***:
the coverage of soilbac1 is 0.65 while the coverage of site1 is 0.93. The coverage of site1 is much higher than in soilbac1.

Richness Estimators

In the R code chunk below, do the following:

1. Write a function to calculate **Chao1**,
2. Write a function to calculate **Chao2**,
3. Write a function to calculate **ACE**, and
4. Use these functions to estimate richness at both `site1` and `soilbac1`.

```
# Chao1
S.chao1 <- function(x = ""){
S.obs(x) + (sum(x == 1)^2) / (2 * sum(x == 2)) }

# Chao2
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)
}

# ACE
S.ace <- function(x = "", thresh=10){
  x <- x[x>0]
  S.abund <- length(which(x>thresh))
  S.rare <- length(which(x <= thresh))
  singlt <- length(which(x == 1))
  N.rare <- sum(x[which(x <= thresh)])
  C.ace <- 1-(singlt/N.rare)
  i <- c(1:thresh)
  count <- function(i,y){
    length(y[y == i])
  }
  a.1 <- sapply(i, count, x)
  f.1 <- (i*(i-1))*a.1
  G.ace <- (S.rare/C.ace)*(sum(f.1)/(N.rare*(N.rare-1)))
  S.ace <- S.abund + (S.rare/C.ace) + (singlt/C.ace) * max(G.ace, 0)
  return(S.ace)
}

S.chao1(site.1)
```

```
##        1
## 119.6944
```

```
S.chao2(1, BCI)
```

```
##        1
## 104.6053
```

```
S.ace(site.1)
```

```
## [1] 159.3404
```

```
S.chao1(soilbac1)
```

```
##     T1_1
## 2628.514
```

```
S.chao2(1, soilbac.t)
```

```
##     T1_1
## 21055.39
```

```
S.ace(soilbac1)
```

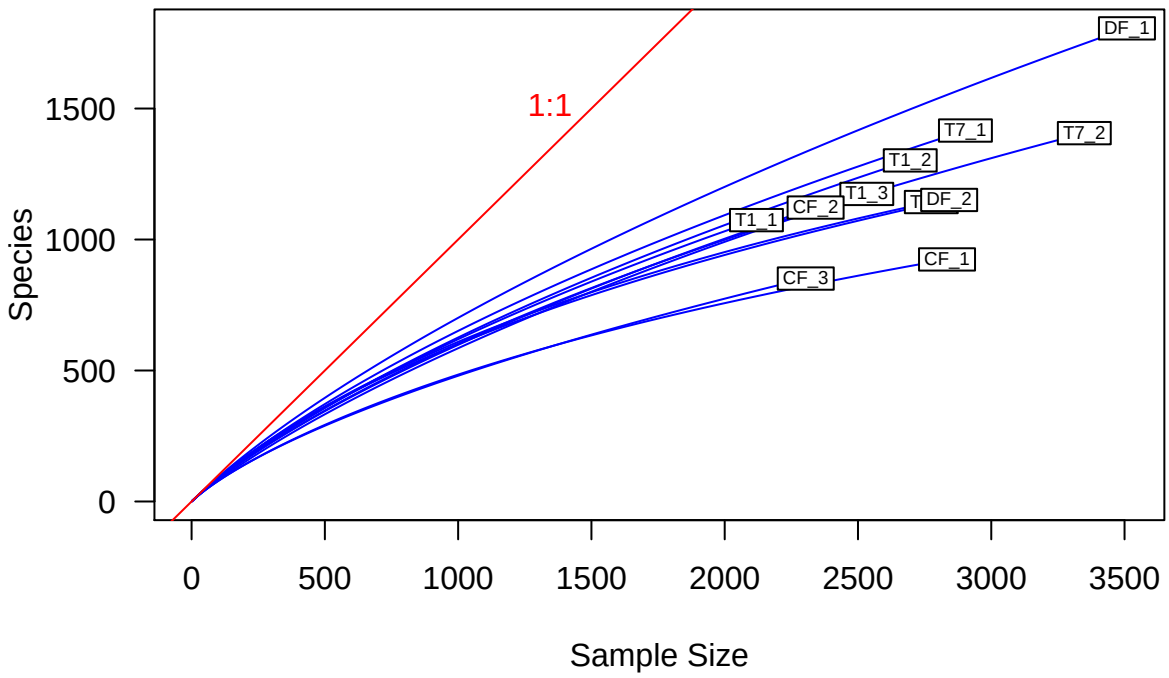
```
## [1] 4465.983
```

Rarefaction

In the R code chunk below, please do the following:

1. Calculate observed richness for all samples in `soilbac`,
2. Determine the size of the smallest sample,
3. Use the `rarefy()` function to rarefy each sample to this level,
4. Plot the rarefaction results, and
5. Add the 1:1 line and label.

```
# Calculating observed richness #
soilbac.S <- S.obs(soilbac.t)
# Smallest value #
min.N <- min(rowSums(soilbac.t))
# Rarefy each sample #
S.rarefy <- rarefy(x = soilbac.t, sample = min.N, se = TRUE)
# Plot rarefaction results
rarecurve(x = soilbac.t, step = 20, col = "blue", cex = 0.6, las=1)
abline(0, 1, col = 'red')
# Add 1:1 line #
text(1500, 1500, "1:1", pos = 2, col = 'red')
```



Question 4: What is the difference between ACE and the Chao estimators?

> ***Answer 4***: # ACE uses a threshold to observe the abundance of other rare species, i.e. species that have 10 or fewer individuals. While the Chao estimators observes the abundance of rare species that are doubletons and singltons. Moreover, ACE, like Chao1 estimator, estimates richess from one sample, while Chao2 estimates richness across sites .

4) SPECIES EVENNESS

Here, we consider how abundance varies among species, that is, **species evenness**.

Visualizing Evenness: The Rank Abundance Curve (RAC)

One of the most common ways to visualize evenness is in a **rank-abundance curve** (sometime referred to as a rank-abundance distribution or Whittaker plot). An RAC can be constructed by ranking species from the most abundant to the least abundant without respect to species labels (and hence no worries about 'ties' in abundance).

In the R code chunk below, do the following:

1. Write a function to construct a RAC,
2. Be sure your function removes species that have zero abundances,
3. Order the vector (RAC) from greatest (most abundant) to least (least abundant), and
4. Return the ranked vector

```
# Cosntruct RAC that removes zero abundances and ordered and returned #
RAC <- function(x = ""){
x = as.vector(x)
x.ab = x[x > 0]
x.ab.ranked = x.ab[order(x.ab, decreasing = TRUE)]
return(x.ab.ranked)
}
```

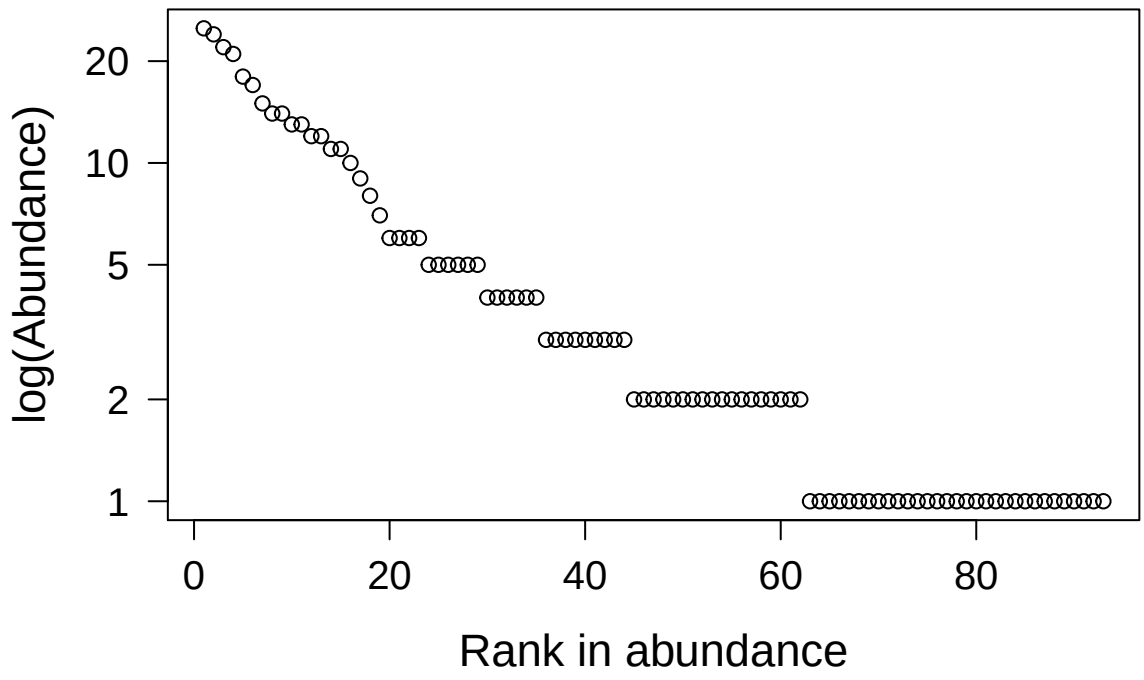
Now, let's examine the RAC for `site1` of the BCI data set.

In the R code chunk below, do the following:

1. Create a sequence of ranks and plot the RAC with natural-log-transformed abundances,
2. Label the x-axis "Rank in abundance" and the y-axis "log(abundance)"

```
# plot the RAC with natural log transformed abundances #
plot.new()
site1 <- BCI[1, ]
rac <- RAC(x = site1)
ranks <- as.vector(seq(1, length(rac)))
opar <- par(no.readonly = TRUE) # Saves default plot parameters

par(mar = c(5.1, 5.1, 4.1, 2.1))
plot(ranks, log(rac), type = 'p', axes = F,
     xlab = "Rank in abundance", ylab = "log(Abundance)",
     las = 1, cex.lab = 1.4, cex.axis = 1.25)
box()
axis(side = 1, labels = T, cex.axis = 1.25)
axis(side = 2, las = 1, cex.axis = 1.25,
      labels = c(1, 2, 5, 10, 20), at = log(c(1, 2, 5, 10, 20)))
```



Question 5: What effect does visualizing species abundance data on a log-scaled axis have on how we interpret evenness in the RAC?

> ***Answer 5***: # The RAC clearly shows that species abundance is quite uneven, as indicated by the higher number of species singletons and doubletons.

Now that we have visualized unevennes, it is time to quantify it using Simpson's evenness ($E_{1/D}$) and Smith and Wilson's evenness index (E_{var}).

Simpson's evenness ($E_{1/D}$)

In the R code chunk below, do the following:

1. Write the function to calculate $E_{1/D}$, and
2. Calculate $E_{1/D}$ for `site1`.

```
# Resets plotting parameters #
par <- opar

# Simpson's eveness index #
SimpE <- function(x = ""){
  S <- S.obs(x)
  x = as.data.frame(x)
  D <- diversity(x, "inv")
  E <- (D)/S
  return(E)
}
```

```
SimpE(site1)
```

```
##         1
## 0.4238232
```

Smith and Wilson's evenness index (E_{var})

In the R code chunk below, please do the following:

1. Write the function to calculate E_{var},
2. Calculate E_{var} for `site1`, and
3. Compare $E_{1/D}$ and E_{var}.

```
# Smith and Wilson's evenness index #
Evar <- function(x){
x <- as.vector(x[x > 0])
1 - (2/pi)*atan(var(log(x)))
}

Evar(site1)
```

```
## [1] 0.5067211
```

```
SimpE(site1)
```

```
##         1
## 0.4238232
```

Question 6: Compare estimates of evenness for `site1` of BCI using $E_{1/D}$ and E_{var}. Do they agree? If so, why? If not, why? What can you infer from the results.

> ***Answer 6***: # Smith and Wilson's evenness index gave a value of 0.5 and the Simpson's evenness index gave a value of 0.42 for site1. I would think that they agree becuase both even index values were roughly 50%. I can infer that site1 species distribution was quite even.

5) INTEGRATING RICHNESS AND EVENNESS: DIVERSITY METRICS

So far, we have introduced two primary aspects of diversity, i.e., richness and evenness. Here, we will use popular indices to estimate diversity, which explicitly incorporate richness and evenness We will write our own diversity functions and compare them against the functions in `vegan`.

Shannon's diversity (a.k.a., Shannon's entropy)

In the R code chunk below, please do the following:

1. Provide the code for calculating H' (Shannon's diversity),
2. Compare this estimate with the output of `vegan`'s diversity function using method = "shannon".

```
# Shannon's diversity #
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)
}

ShanH(site1)
```

```
## [1] 4.018412
```

```
# Vegan package of shannon #
diversity(site1, index="shannon")
```

```
## [1] 4.018412
```

Simpson's diversity (or dominance)

In the R code chunk below, please do the following:

1. Provide the code for calculating D (Simpson's diversity),
2. Calculate both the inverse (1/D) and 1 - D,
3. Compare this estimate with the output of `vegan's` diversity function using method = "simp".

```
# 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)
}

# Calculating the inverse and 1-D #
D.inv <- 1/SimpD(site1)
D.sub <- 1-SimpD(site1)

# Comparing the estimates with vegan diversity function #
diversity(site1, "inv")
```

```
## [1] 39.41555
```

```
diversity(site1, "simp")
```

```
## [1] 0.9746293
```

Question 7: Compare estimates of evenness for `site1` of BCI using $E_{H'}$ and E_{var}. Do they agree? If so, why? If not, why? What can you infer from the results.

> ***Answer 7***: # The evenness value for site 1 using the Evar is 0.5 while the value of shannon diversity index E(H') is 39.415. The shannon's diversity value and Evar value outputs are not really comparable since one considers evenness and the other is a metric for diversity.

Fisher's $\boldsymbol{\alpha}$

In the R code chunk below, please do the following:

1. Provide the code for calculating Fisher's $\boldsymbol{\alpha}$,
2. Calculate Fisher's $\boldsymbol{\alpha}$ for `site1` of BCI.

```
# Calculating Fisher's #
rac <- as.vector(site1[site1 > 0])
invD <- diversity(rac, "inv")
invD
```

```
## [1] 39.41555
```

```
# Fisher's for site1 #
Fisher <- fisher.alpha(rac)
Fisher
```

```
## [1] 35.67297
```

```
Evar(site1)
```

```
## [1] 0.5067211
```

Question 8: How is Fisher's $\boldsymbol{\alpha}$ different from $E_{H'}$ and E_{var}? What does Fisher's $\boldsymbol{\alpha}$ take into account that $E_{H'}$ and E_{var} do not?

> ***Answer 8***: # Fisher's alpha is different from shannon's diversity in that Fishers is asympotitically similar to Simpson's diversity. Simpson's diversity is a dominance index and gives more weight to the common or dominant species. # Fishers is different than the Simpson's diversity and evenness index because Fisher's alpha estimates diversity instead of simply just calculating diversity.

6) MOVING BEYOND UNIVARIATE METRICS OF α DIVERSITY

The diversity metrics that we just learned about attempt to integrate richness and evenness into a single, univariate metric. Although useful, information is invariably lost in this process. If we go back to the rank-abundance curve, we can retrieve additional information – and in some cases – make inferences about the processes influencing the structure of an ecological system.

Species abundance models

The RAC is a simple data structure that is both a vector of abundances. It is also a row in the site-by-species matrix (minus the zeros, i.e., absences).

Predicting the form of the RAC is the first test that any biodiversity theory must pass and there are no less than 20 models that have attempted to explain the uneven form of the RAC across ecological systems.

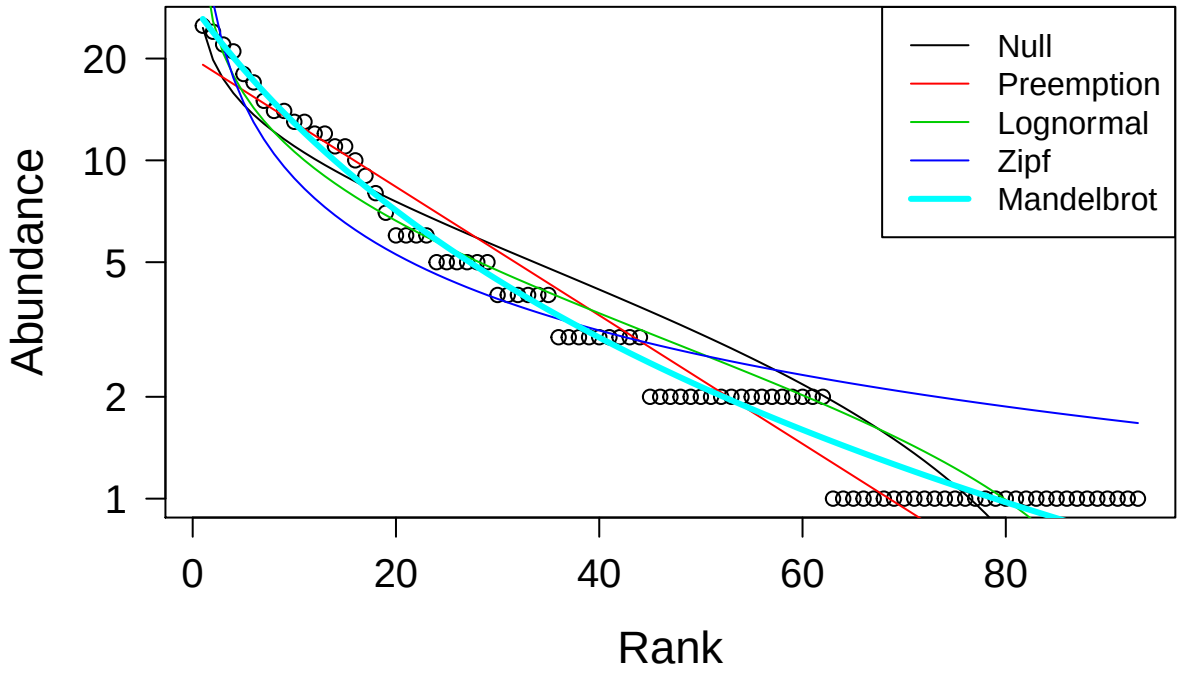
In the R code chunk below, please do the following:

1. Use the `radfit()` function in the `vegan` package to fit the predictions of various species abundance models to the RAC of `site1` in BCI,
2. Display the results of the `radfit()` function, and
3. Plot the results of the `radfit()` function using the code provided in the handout.

```
# Use radfit() #
RACresults <- radfit(site1)
RACresults
```

```
##
## RAD models, family poisson
## No. of species 93, total abundance 448
##
##            par1      par2     par3    Deviance AIC      BIC
## Null                                    39.5261 315.4362 315.4362
## Preemption  0.042797                     21.8939 299.8041 302.3367
## Lognormal   1.0687    1.0186            25.1528 305.0629 310.1281
## Zipf        0.11033  -0.74705           61.0465 340.9567 346.0219
## Mandelbrot  100.52   -2.312    24.084    4.2271 286.1372 293.7350
```

```
# Plot results of the radfit #
plot.new()
plot(RACresults, las = 1, cex.lab = 1.4, cex.axis = 1.25)
```



Question 9: Answer the following questions about the rank abundance curves: a) Based on the output of `radfit()` and plotting above, discuss which model best fits our rank-abundance curve for `site1`? b) Can we make any inferences about the forces, processes, and/or mechanisms influencing the structure of our system, e.g., an ecological community?

> ***Answer 9a***: The mandelbrot seems to fit the best for our rank-abundance curve for site1.
> ***Answer 9b***: BCI is a tropical rainforest that is teeming with rare species, controlling for evenness can improve the fit of the Zipf model significantly. Some mechanisms that may be invoked would be strong negative-density dependent feedbacks.

Question 10: Answer the following questions about the preemption model: a. What does the preemption model assume about the relationship between total abundance (N) and total resources that can be preempted? b. Why does the niche preemption model look like a straight line in the RAD plot?

Answer 10a: N decreases as total resources is used up/ preempted. ***Answer 10b***: The niche preemption model looks like a straight line in the RAD plot because it assumes that the abundance of each new added species would be a fraction of the previous species.

Question 11: Why is it important to account for the number of parameters a model uses when judging how well it explains a given set of data?

Answer 11: It is important to account for the number of parameters becuase it may explain some biological base to explain the patterns of abudnance.

SYNTHESIS

1. As stated by Magurran (2004) the $D = \sum p_i^2$ derivation of Simpson's Diversity only applies to communities of infinite size. For anything but an infinitely large community, Simpson's Diversity index is calculated as $D = \sum \frac{n_i(n_i-1)}{N(N-1)}$. Assuming a finite community, calculate Simpson's D, 1 - D, and Simpson's inverse (i.e. 1/D) for `site 1` of the BCI site-by-species matrix.

Simp_Div <- 0 l <- length(site1) N <- sum(site1) for(i in 1:l){ Simp_Div <- Simp_Div + (site1[,i]*(site1[,i]-1)/N/(N-1)) }

Simp_Div(site1)

0.0231

1/Simp_Div

43.12145

2. Along with the rank-abundance curve (RAC), another way to visualize the distribution of abundance among species is with a histogram (a.k.a., frequency distribution) that shows the frequency of different abundance classes. For example, in a given sample, there may be 10 species represented by a single individual, 8 species with two individuals, 4 species with three individuals, and so on. In fact, the rank-abundance curve and the frequency distribution are the two most common ways to visualize the species-abundance distribution (SAD) and to test species abundance models and biodiversity theories. To address this homework question, use the R function **hist()** to plot the frequency distribution for `site 1` of the BCI site-by-species matrix, and describe the general pattern you see.

hist(rac)

I see that 5 species make up the majority of the indivudals in site1 while ~20 other species occur in low (<10) frequency.

3. We asked you to find a biodiversity dataset with your partner. This data could be one of your own or it could be something that you obtained from the literature. Load that dataset.

rm(list = ls()) setwd("C:/Users/Venus/Github/QB2017BenavidezKuo/data")

Load packages

require("vegan") #Already contains BCI data require("tidyr") require(dplyr)

Load dataset

BCI2010 <- read.delim("BCI2010.txt", header=T) BCI1982 <- read.delim("BCI1982.txt", header=T)

Transform 2010 census data into site by species matrix

BCI.2010.SbyS <- group_by(BCI2010, Quadrat) %>% count(Latin) %>% spread(key=Latin, value=n , fill=0) dim(BCI.2010.SbyS)

Transform 1982 census data

BCI.1982.SbyS <- group_by(BCI1982, Quadrat) %>% count(Latin) %>% spread(key=Latin, value=n , fill=0)

How many sites are there? # 1250

How many species are there in the entire site-by-species matrix? # 298

Any other interesting observations based on what you learned this week? # N/A

SUBMITTING YOUR ASSIGNMENT

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

Unless otherwise noted, this assignment is due on **Wednesday, January 25th, 2015 at 12:00 PM (noon)**.