Tutorial 11: Diversity, Indicator Species Analysis, Cluster Analysis

Calculating Diversity Indices
The ‘vegan’ package contains the command `diversity()` for calculating Shannon and Simpson diversity indices.
The input data matrix (‘datamatrix’ here) is a species abundance matrix – do not include extra columns for plot numbers, etc.
The ‘index=’ argument specifies which calculation to perform: Shannon index (“shannon”), Simpson index (“simpson”), or Inverse Simpson index (“invsimpson”).
The ‘base=’ argument specifies the logarithm base to be used; “exp(1)” specifies the natural logarithm, which is the default.

```r
library(vegan)

diversity(datamatrix, index="shannon", base=exp(1))
diversity(datamatrix, index="simpson")
diversity(datamatrix, index="invsimpson")
```

Indicator Species Analysis
Indicator species analysis (ISA) provides a list of species for each study group (perhaps based on habitat, treatment, etc.) that are indicative or representative of the community present in that group of plots/sites/samples. The indicator species are often either dominant (highly frequent or abundant) or rare species in their particular habitat/group but are markedly less dominant or non-existent in the other habitats/groups.

The commands for performing ISA are in the ‘labdsv’ package.

```r
library(labdsv)

indval(abundmatrix, plots$grouping)
```

Data needs to be in two separate data objects: (1) Species abundance matrix (called ‘abundmatrix’ here) and (2) a data frame (called ‘plots’ here) containing the grouping variable.

The basic `indval()` command will produce a large output of the many values calculated, including relative frequency (relfrq), relative abundance (relabu), indicator value (indval), and p-values.
The ‘plots$grouping’ argument chooses a single grouping variable (called ‘grouping’ here) from the ‘plots’ data frame.

```r
indval(abundmatrix, plots$grouping)
```
The output can be summarized, using the “summary” command, to only show the significant indicator species. The output must first be assigned to a name (called ‘output’ here).

```r
output=indval(abundmatrix, plots$grouping)
summary(output)
```

**Similarity and Dissimilarity Matrices**

Community matrices can be used to create distance (also called similarity or dissimilarity) matrices. The ‘vegan’ package is required.

```r
library(vegan)
```

The input data matrix (‘matrix’ here) should be raw data. Relative measures (e.g. relative abundance) will not produce correct ordinations if the output is to be used with NMDS (see Tutorial 12). The output will be assigned to a new name, ‘dist.matrix’.

The ‘method=’ argument refers to the distance measure or index used. Choices are “manhattan”, “euclidean”, “canberra”, “bray”, “kulczynski”, “jaccard”, “gower”, “altGower”, “morisita”, “horn”, “mountford”, “raup”, “binomial”, or “chao”.

The “bray” method is Bray-Curtis dissimilarity index, equal to 1 minus Sorensen similarity ($1 - S_S$). It is commonly used on species community data.

The ‘binary=’ argument determines whether the resulting distance matrix is produced based on presence-absence information only. If FALSE, abundance measures will be used in calculating the dissimilarity matrix. If TRUE, abundance values in the input ‘matrix’ will be converted to 0 and 1 (binary) presence-absence data.

```r
dist.matrix=vegdist(matrix, method="bray", binary=FALSE)
```

**Cluster Analysis**

Cluster analysis is used to group (cluster) samples, plots, or sites based on their species compositions. Sites that are more similar in composition are more likely to be clustered together.

There are many different methods and options for performing a cluster analysis, so we’ll just cover some of the common ones here. This is one of those situations where there’s no “right” way to do the analysis and everyone has their own preferred method.

All of the cluster analysis commands that follow require a distance (dissimilarity) matrix.

The ‘cluster’ package contains the commands necessary for cluster analysis.

```r
library(cluster)
```
Agglomerative hierarchical clustering works by starting with each site as its own cluster and then adding clusters together based on their similarity (the most similar clusters are added together first, and so on).

\[
\text{agnes(dist.matrix, method= "single")}
\]
\[
\text{agnes(dist.matrix, method= "average")}
\]
\[
\text{agnes(dist.matrix, method= "complete")}
\]

Divisive hierarchical clustering works by starting with all sites in one cluster and then dividing it into smaller and smaller clusters.

\[
\text{diana(dist.matrix)}
\]

Non-hierarchical methods of clustering work by assigning sites to a preset number of clusters. The output from pam() is more robust than kmeans() and is also more easily plotted.

\[
\text{kmeans(dist.matrix, centers)}
\]
\[
\text{pam(dist.matrix, k)}
\]

Dendrograms are used to display hierarchical clustering analysis outputs. The pltree() command creates the a dendrogram for the outputs of agnes() and diana().

\[
\text{output=agnes(dist.matrix, method= "single")}
\]
\[
\text{pltree(output)}
\]

The output from pam() can be visualized with either a silhouette plot (using the plot() command) or a cluster plot (using the clusplot() command).

\[
\text{output=pam(dist.matrix, 4)}
\]
\[
\text{plot(output)}
\]
\[
\text{clusplot(output)}
\]

**Cophenetic Correlation**

Cophenetic Correlation tells how well the clustering analysis result portrays the ecological distance between sites. A large correlation coefficient (‘Mantel statistic r’) value indicates good portrayal. It’s probably a good idea to run multiple types of cluster analyses and check the coefficients of all of them to see which one best represents the data. The cophenetic() command generates a matrix of cophenetic distances. A Mantel test is then used to test how well the dendrogram distances correlate with the actual site distances.

\[
\text{output=agnes(dist.matrix, method="single")}
\]
\[
\text{cophenetic.dist=cophenetic(output)}
\]
\[
\text{mantel(dist.matrix, cophenetic.dist, permutations=1000)}
\]
\[
\text{plot(dist.matrix, cophenetic.dist)}
\]
Tutorial Code

```r
setwd("/Users/johndoe/Desktop/")
example=read.csv("R_example_community_data.csv")
matrix=read.csv("example_stem_matrix.csv")
head(matrix)

#Is the first column of the matrix ‘X’?
#If yes, remove it: either open in Excel and delete the column, then re-read into R
#OR run the following line of code
matrix=matrix[-1]

head(matrix)  #Check and see if it’s fixed

#Diversity Indices
library(vegan)
shannon=diversity(matrix, index="shannon")
shannon

simpson=diversity(matrix, index="simpson")
simpson

invsimp=diversity(matrix, index="invsimpson")
invsimp

#Add diversity to the existing ‘plotdata’ data frame
plotdata$Shannon=diversity(matrix, index="shannon")
head(plotdata)

plotdata$Simpson=diversity(matrix, index="simpson")
head(plotdata)

plotdata$Invsimp=diversity(matrix, index="invsimpson")
head(plotdata)

plotdata$Aspect=c(rep("N", 30), rep("E", 20), rep("S", 30))
plotdata$Treatment=c(rep("A", 20), rep("B", 20), rep("C", 20),
rep("D", 20))
head(plotdata)
```
write.csv(plotdata, file="example_plotdata.csv")

#Indicator Species Analysis
library(labdsv)
indval(matrix, plotdata$Treatment)
summary(indval(matrix, plotdata$Treatment))

#Distance (Dissimilarity) Matrix
library(vegan)
dist.matrix=vegdist(matrix, method="bray")
dist.matrix

#Cluster Analyses
library(cluster)

#Hierarchical agglomerative
agnes(dist.matrix, method="single")
agnes(dist.matrix, method="complete")
agnes(dist.matrix, method="average")

#Hierarchical divisive
diana(dist.matrix)

#Non-hierarchical methods
kmeans(dist.matrix, 3)
pam(dist.matrix, 3)

#Plot the hierarchical agglomerative output (dendrogram)
output=agnes(dist.matrix, method="complete")
pltree(output)

#Plot the non-hierarchical output
output2=pam(dist.matrix, 3)
plot(output2) #silhouette plot
clusplot(output2)
#Cophenetic Correlation

cophenetic.dist=cophenetic(output)
mantel(dist.matrix, cophenetic.dist, permutations=1000)
plot(dist.matrix, cophenetic.dist)