First R Exercise (Bioinformatics 2019)

– an introduction to matrices and basic community analysis.

*Many elements of this exercise were taken from a packet used in the Microbial Metagenomics Course (Summer 2011, Michigan State University)

Objectives:

- Load and manipulate data
- Examine standard community matrices
- Start developing a tool-kit for analyzing your Mothur output

NOTE: see Chris' list of important R stuff as a companion to this and as a quick reference for basic things R.

1) Brief introduction to matrices and R

a) Matrices

-A matrix is an ordered set of numbers listed in rectangular form (think Excel sheet!).

$$A = \begin{matrix} 2 & 3 & 4 & 4 \\ 6 & 2 & 1 & 9 \\ 9 & 5 & 2 & 8 \end{matrix}$$

- The matrix above (matrix A) has 3 rows and 4 columns, it is a 3 x 4 (read as "three by four") matrix.
- A matrix with a single row or a single column is often called a "vector".
- The elements of a matrix are referred to by their row number and column number (always in that order). From the matrix above, $A_{3,2} = 5$.
- Ecological data, especially community composition data, often takes the form of a matrix or vector.

b) R Statistics Package

- R is an open source, command prompt-based software that allows loading and manipulation of data in matrix form.
- Individuals are able to contribute **packages** to the Comprehensive R Archive Network (CRAN). Packages provide useful statistical, mathematical, and graphical processes **(functions)** in an easy to access form
- R and its packages are easy to download and install from the R Project for Statistical Computing website (http://www.r-project.org)
- A matrix is one type of **object** in R; others include: vector, list, and data frame
- To manipulate matrices in R, we have to use the previously described matrix notation.
- Each object (and therefore matrix) in R has a name (e.g. "A", "data", or "stuart")
- The same row by column format is used to access portions of a matrix

    - To access a whole matrix, use: A
    - To access a single element, use: A[l,3]
    - To access a whole row, use: A[2,]
    - To access a whole column, use: A[,4]
  - The basic package of R and any additional packages you install include a series of functions. Rstudio is an interface that uses R, but makes organization easier. I will demo using the free R studio. (note, Rstudio requires that you download it along with R, Rstudio can be downloaded from: https://www.rstudio.com/products/rstudio/download/)
  - Functions are called by typing their name followed by an open and close parentheses, e.g. plot(). However, the function can't do what it is designed to do without giving it object(s) and rules to work with. These pieces of information go within the parentheses and are called **arguments.**
  - Typing help(plot) or ?plot will give you the help file for the function plot; within the help file is a list of arguments that must be passed to the function.

Final Note: There is a fairly steep learning curve with R, but it is an extremely powerful tool. There are numerous books and tutorials available on the web. I believe the best way to become proficient is to commit to using R for a project even if it seems like it will take a really long time to do something you know how to do in some other program (Excel, etc.). Don't get frustrated and use the web and others as a resource! A simple google search for “R ???” will often give you relevant results. I have always been fond of the “Cookbook for R” series of resources.

2) Loading data, data transformations, and visualization

- To start Rstudio (and consequently, R), click the Rstudio icon in the dock at the bottom of your screen or from your applications directory.
- A four-panel window should open – the panels are: a text editor (where you can keep only the code that works), an R window (where the commands will be executed), a list of your objects in the R environment, and a panel for graphs/packages and other useful stuff.
- R only "sees" the files in the working directory. To access information from a file, you must put that file in your current working directory or change your current working directory to the directory (folder) containing the file.
- To check your current working directory (i.e., your path) type “getwd()”.
- To set your working directory type “setwd()” – if you want this to be your desktop put the path to your Desktop in the parentheses, and in quotes – setwd(“~/Desktop”)

****Installation of the vegan package: We will rely heavily on a multivariate statistics package called "vegan". To install this use the install.packages command – install.packages(“vegan”, dependencies = TRUE)

***Note: Any line in this document that begins with the R line prompt, ">", is intended for you to type in at the line prompt on your machine. Expected output from R will be printed in smaller, italicized text in this document.

a) Loading a data table

The function read.table() is used to read in delimited text files. Often R will assign the data type "data frame" to loaded text files, but we want it to be of the type "matrix" so we use the function as.matrix() along with read.table(). Don't forget to assign the matrix a name so you can continue to use it (e.g. "data").

```
     >data=as.matrix(read.table("Table1.txt",header=TRUE, sep="\t",row.names=1))
     > data
              Sample1 Sample2 Sample3 Sample4
           OTU1     25     80     60     90
           OTU2     25      0      5      0
           OTU3     25      0      5      0
           OTU4     25      0      0      0
           OTU5      0     20     30     10
    > dim(data)
     {l} 5 4
     >  data[1,]
        Samplel Sample2 Sample3 Sample4
     OTUI     25     80     60     90
     > data[,3]
     {1} 60  5  5  0 30
```

Questions

1) What does the output of the function dim() report?

2) For which OTU did we recover the most sequences in our study?

3) An ordination is a visual depiction of complex data often found in ecology (or other types of multivariate data). Usually these are two-dimensional scatter plots where the pairwise distances between points represent the multivariate, compositional distance between samples. Draw an ordination of the four samples in this dataset. When making your drawing remember that ordination is grouping points by similarity (points that are closer in the 2D space of your figure are more similar in their species composition).

b) Data transformations

Often, community composition data is transformed from raw abundance to presence-absence or relative abundance. We can also easily calculate sample richness using the presence-absence matrix.

```
>dataPA=(data>0)*1
>dataPA
        Samplel  Sample2 Sample3 Sample4
OTUl      1      1     1     1
OTU2      1      0     1     0
OTU3      1      0     1     0
OTU4      1      0     0     0
OTUS      0      1     1     1
> rich =colSums(dataPA)
>rich
Samplel Sample2 Samp/e3 Sample4
   4      2      4      2
```

There is a slower and faster way to calculate relative abundance from raw data. The faster approach requires something called a "for loop". In a "for loop", a process is repeated for each member of a list. In our case each column *i* of the data matrix is divided by the sum of that column *i.*

Slower method

```
> dataREL=matrix(0,5,4)
> dataREL[,l]=data[,l]/sum(data[,l])
> dataREL[,2]=data[,2]/sum(data [,2])
> dataREL[,3]=data [,3]/sum(data [,3])
> dataREL [,4]=data[,4]/sum(data[,4])
> dataREL
         Samplel  Samp/e2 Sample3 Sample4
OTUl    0.25    0.8   0.60    0.9
OTU2    0.25    0.0   0.05    0.0
OTU3    0.25    0.0   0.05    0.0
OTU4    0.25    0.0   0.00    0.0
OTU5    0.00    0.2   0.30    0.1
>  colSums(dataREL)
SamplelSample2Sample3Sample4
    1      1      1      1
```

Faster method

```
> dataREL2=data
>for(i in 1:4){dataREL2[,i]=data[,i]/sum(data[,i])}
> dataREL2
        SampleISample2Samp/e3Sample4
OTUI   0.25   0.8  0.60   0.9
OTU2   0.25   0.0  0.05   0.0
OTU3   0.25   0.0  0.05   0.0
OTU4   0.25   0.0  0.00   0.0
OTU5   0.00   0.2  0.30   0.1
>  colSums(dataREL2)
Sample1Sample2  Samp/e3 Sample4
    1     1     1     1
```

Another tranformation that is useful at times is to transpose a matrix (switch the rows and columns). This is important because the orientation of the matrix determines whether you evaluate community composition or species occurrence patterns

```
> t(dataREL2)
         OTUI OTU2 OTU3 OTU4 OTU5
SampleI 0.25 0.25 0.25 0.25  0.0
Samp/e2 0.80 0.00 0.00 0.00  0.2
Sample3 0.60 0.05 0.05 0.00  0.3
Sample4 0.90 0.00 0.00 0.00  0.1
```

The basis of many multivariate statistical approaches is a pairwise distance matrix. This is a symmetrical (reflected across the diagonal) square matrix. Sometimes only the lower triangle of this matrix is shown for brevity. Each element of the matrix is an index of distance or dissimilarity between two samples. If we calculate a distance matrix, DIST, DIST[l,1] is the dissimilarity between sample 1and sample 1and therefore equals 0. The diagonal is made up of all "self comparisons and is filled with 0s. DIST[2, 1] is the distance between sample one and sample two. Common dissimilarity indices include: Euclidean, Bray-Curtis, Sorenson's, and Jaccard (see Appendix 1for calculations). A really useful function for calculating distance matrices in R is vegdistQ. This is in a package contributed by Jari Oksanen called vegan. Packages can be installed using a graphical interface under the Packages & Data pulldown menu. To load a package, the function library() is used.

```
> library(vegan)


>samplePA.dist=vegdist(t( dataPA),method="jaccard")
> samplePA.dist
        SampleI  Sample2 Sample3
Sample2     0.8
Sample3     0.4     0.5
Sample4     0.8     0.0     0.5
>otuPA.dist=vegdist( dataPA,method="jaccard")
>otuPA.dist
```

```
   OTUl OTU2
OTU3 OTU4
OTU2 0.50
OTU3 0.50 0.00
OTU4 0.75 0.50 0.50
OTU5 0.25 0.750.75 1.00
> sampleREL.dist=vegdist(t(dataREL2),method="bray")
> sampleREL.dist
         Samp/el  Samp/e2 Samp/e3
Sample2    0.75
Samp/e3    0.65    0.20
Sample4    0.75    0.10    0.30
```

Questions

1) Which of our samples had the highest a-diversity (richness)?

2) Why would it be a good idea to standardize each sample to a sum of one?

3) Were the presence-absence and relative distance matrices the same? Why or why not?

c) Visualization: Ordination, Clustering, and Heatmaps

A common ordination technique used in ecology is principle coordinates analysis (PCoA). This is a useful way to visualize complex community composition data in two dimensions. The function for PCoA in R is cmdscale(). Other common ordination techniques include Correspondence Analysis (CA; R function is cca()) and non-metric multidimensional scaling (NMMDS; R function is metaMDS()).

```
> samplePA.pcoa=cmdscale( samplePA.dist)
> samplePA.pcoa
                 [,1]      [,2]
Samplel   0.4813025  0.08424920
Sample2 -0.3168243  0.02953543
Sample3  0.1523462 -0.14332005
Sample4 -0.3168243  0.02953543
>  sampleREL.pcoa=cmdscale( sampleREL.dist)
> sampleREL.pcoa
              [,1]        [,2]
Samplel  0.52990579  0.024747578
Sample2 -0.22049875  0.008313729
Sample3 -0.09518054 -0.154254092
Sample4 -0.21422650  0.121192785
```

Another common approach for visualizing multivariate data is clustering. Here we will use Unweighted Pair Group Method with Arithmetic Mean (UPGMA) clustering with the R function hclust().

```
>samplePA.clust=hclust(samplePA.dist)
>  sampleREL.clust=hclust(sampleREL.dist)
```

The most common function for plotting data is plot().

```
> par(mfrow=c(2,2)) #this set of functions sets up a 2 x 2 grid for your plots.
#you can change the 2,2 to be different combinations of row and column combos.
> plot(samplePA.pcoa[, l],samplePA.pcoa[,2],cex=0) #the first arguments are just
#your x and y variables, and cex is a scaling the plot symbol size.
> text(samplePA.pcoa [,l],samplePA.pcoa[,2],seq (l,4),cex= 1.5)
#this text simply says put a 1 through 4 at each of the coordinates and the cex
#scales them.
> plot(sampleREL.pcoa[ ,1],sampleREL.pcoa [,2],cex=0)
>text(sampleREL.pcoa[, 1],sampleREL.pcoa [,2],seq(1,4),cex=1.5)
>plot(samplePA.clust)
>samplePA.dist
          Sample1Sample2  Samp/e3
Sample2   0.8
Sample3   0.4   0.5
Samp/e4   0.8   0.0   0.5
> plot(sampleREL.clust)
> sampleREL.dist
         Samplel Sample2 Sample3
Sample2   0.75
Sample3   0.65  0.20
Sample4   0.75   0.10   0.30
```

Another popular method for depicting microbial community data is a heatmap. This figure type combines cluster analysis with shaded cells indicating the relative abundance of each "species" in each sample.

```
>heatmap(dataREL2, scale="none",labCol=c('Sl','S2' ,'S3','S4'))
```

Questions

1) Were the presence-absence and relative PCOA ordinations the same? Why or why not?

2) Do sample differences depicted in your cluster diagram agree with your heatmap?

Appendix 1

Community Distance Indices for samples j and k calculated across all species i

Euclidean: $dj.k = \}\overline{L(xij - X_{;,k})2}$

Bray – Curtis: $dj.k = \frac{|X_{;j} — X_{;,k}|}{L..J(X_{;j} + X_{;,k})}$

Sorenson's: Bray – Curtis, but with presence – absence data

Jaccard : $_1$ *BB, where B == Bray – Curtis distance*

Appendix 2

INDIRECT GRADIENT ANALYSES (only species data is used to create ordinations; unconstrained ordinations)

Distance-based approaches

-Principal Coordinates Analysis (PCoA; Metric Dimensional Scaling)

Projects multidimensional distances into two or three distances. Maximizes correlation between multidimensional distances and reduced dimension distances. Eigenanalysis on distance matrix. When the distance metric used is Euclidean PCoA=PCA Assumes a linear response to gradients.

-Nonmetric Multidimensional Scaling (NMDS/MDS)

Maintains rank order of distances rather than absolute distances. This avoids assumptions used in PCoA, but can at times find local solutions rather than global. This occurs because an iterative approach is used to minimize disagreement between the distance matrix and the ordination using the "stress" value. Also, information about species identity or relationships is lost. The likelihood of finding local solutions can be minimized by estimating the NMDS ordination from multiple starting configurations.

Eigenanalysis-based

-Linear

+Principal Components Analysis (PCA)

The simplest and oldest eigenanalysis-based method. Rotats the original data matrix onto a set of new axes, such that the maximum variance is represented by the first axis and the second most variance is represented by an orthogonal axis. Assumes linear responses of species. Suffers from "the horseshoe effect" where the second axis is curved and/or twisted relative to the first and does not represent a true, independent secondary gradient.

-Unimodal

+Correspondence Analysis (CA; Reciprocal Averaging)

Also known as reciprocal averaging because one method for finding the solution involves repeated averaging of sample scores and species scores. CA maximizes the correspondence between species scores and sample scores. CA assumes unimodal responses of species along the axes. This allows for the potentially useful interpretation of species scores as optima along unmeasured environmental gradients. The second and higher axes can suffer from "the arch effect", which is similar to the horseshoe effect of PCA, but less severe.

+Detrended Correspondence Analysis (DCA)

Developed in response to the arch effect in CA The arch effect is eliminated by detrending using polynomials and segments. Using polynomials is better: a regression is performed in which the second axis is a polynomial function of the first axis and the second axis is replace by the residuals of that regression. This is repeated for any subsequent axes. Sometimes this approach doesn't completely remove the arch effect and detrending by segment is used. In this approach, points along the first axis are split into groups and are centered to have a zero mean on the second axis. This post hoc detrending is desirable as it removes the arch effect, but it isn't the most elegant approach.

Useful sources of information:

-The ordination webpage at Oklahoma State University (http://ordination.okstate.edu); A good "digested" source

-Numerical Ecology

Legendre & Legendre 1998; A huge detailed source that is difficult to digest at times