#### NMDS for Verde Year 1 ####

####Always Run the Next Section of code to load libraries first and verify you have installed all necessary packages####

### Install/Load necessary Packages and set Working Drive###

## I like to set a working drive and have all of my matrices as well as it is a place to deposit the final plots as .jpg or .tiff ##

# if setting a Working Drive can just copy and paste the pathway from your computer and you will need to change backslashes to forward slashes #

# For this meeting likely either set it as your desktop or just watch me go through it and follow along as well as you can until you can set up your specific pathways #

setwd("C:/Users/15hoc/Desktop")

# Not all of these packages are necessary for ordinations, the primary package to conduct ordinations is Vegan. If you don't have the package installed already, just run the code --> install.packages("") #

#install.packages("ade4")

library(ade4)

#install.packages("vegan")

library(vegan)

#install.packages("ggplot2")

library(ggplot2)

#install.packages("raster")

library(raster)

#install.packages("sp")

library(sp)

#install.packages("RColorBrewer")

library(RColorBrewer)

#install.packages("gclus")

library(gclus)

#install.packages("ape")

library(ape)

###################################################################

########################RUN NMS IN VEGAN ##########################

### Now run NMS in R under the package vegan ###

##### NMS for ALL 54 SAMPLES#####################################

##Bring in the species matrix of all 54 samples. Since I set up the working drive it will know where to find the documents I'm bringing in##

verdsp1\_spring<-read.csv("verde\_yr1\_spring.csv")

#head() shows you the columns and first six rows of the imported

head(verdsp)

#observe the dimensions of the imported matrix#

dim(verdsp1\_spring) #54 rows by 118 columns#

##subset the dataframe on which to base the ordination. i.e. subset only the spp data. site names will be assigned again later from second matrix. This step is necessary for ordinations. Data must only be data without ID/groupings.##

#this code subsets columns 2-139

spdat1\_spring<-verdsp1\_spring[2:118]

head(spdat) # double checks that the correct data was subset

#LOG(X+1) transform matrix. Will run both to try to obtain lowest stress final solution. For this data it fulfills the requirements to be able to do a log(x+1) transformation which is (lowest number must be 0 or 1 (e.g. cannot be 0.3). This will provide a lower stress solution in the long run#

sptrns\_spring<-log(spdat1\_spring+1)

## Bring in the Habitat Matrix which will provide site names. columns that contains the descriptive/environmental data ##

#Note: the order of the data in the habitat matrix must match the order of the data from the abundance data in order to properly correlate site names and habitat variables#

verdhab\_spring<-read.csv("verde\_hab\_yr1\_spring.csv")

head(verdhab)

colnames(verdhab)

dim(verdhab\_spring) #54 rows by 19 columns

#ordination by NMDS with RAW Data. Run this if you only want to view the solutions related to the untransformed data. This data has a higher stress solutionn when compared to the transformed data#

NMDS1\_spring <- metaMDS(spdat1\_spring, distance = "bray", k = 2)

#head(NMDS1)

NMDS1\_spring$stress

#ordination by NMDS with log(x+1) transformed Data

sptrns\_spring<-log(spdat\_spring+1)

#Function from Vegan --> metaMDS

#metaMDS(matrix to run NMS, distance metric (here we use Bray-Curtis/Sorensen), k --> seed to start

NMDS2\_spring <- metaMDS(sptrns\_spring, distance = "bray", k = 2)

# R will give you stats on the runs

head(NMDS2)

NMDS2\_spring$stress #provides the lowest stress solution

stressplot(NMDS2) #creates a plot of distance as a function of dissimilarity and shows the fit of the line

##USE NMDS2. Transformed data has lower column and row variance as well as results in a lower final stress (12.67 versus around 20 with untransformed##

###Environmental Correlations Overlay##########################

head(verdhab)

envhab\_spring<-verdhab\_spring[,c(8:21)] #extract only the variables we want to measure

joint\_spring<-envfit(NMDS1\_spring,envhab\_spring,permutations=999,strata=NULL,choices=c(1,2),scaling="sites")

#joint<-envfit(NMDS2,verdhab,permutations=999,strata=NULL,choices=c(1,2),scaling="species")

joint\_spring

colnames(joint\_spring)

class(joint\_spring)

scores(joint\_spring,"vectors")

envpts\_spring<-scores(joint\_spring,"vectors")

class(joint\_spring)

ordiArrowMul(joint\_spring) #0.5843131

envscores\_spring<-as.data.frame(scores(joint\_spring,display="vectors"))

envscores\_spring #this provides the x and y points related to the arrows that will represent correlations of the species/samples

envhab2\_spring<-verdhab\_spring[,c(8,11,17,19)]

joint2\_spring<-envfit(NMDS1\_spring,envhab2\_spring,permutations=999,strata=NULL,choices=c(1,2),scaling="sites")

#########################

##Data visualisation for NMDS with all sites (n=54)

##Using NMDS1##

plot(NMDS1)

aspect\_spring<- factor(verdhab\_spring$order)

color\_spring<- factor(verdhab\_spring$Hab)

co\_spring<-c("black", "magenta", "blue")

shape\_spring<-c(15,16,17)

plot(NMDS1\_spring$points, col=co\_spring[color\_spring],asp=1,pch = shape\_spring[aspect\_spring], cex=1.2, xlab = "NMDS1", ylab = "NMDS2")

plot(NMDS2\_spring$points, col=co\_spring[color],asp=1,pch = shape\_spring[aspect], cex=1.2, xlab = "NMDS1", ylab = "NMDS2")

#Plot ordination so that points are coloured and shaped according to the groups of interest

#aspect <- factor(verdhab$hs)

#color <- factor(verdhab$hs)

#co<-c("darkorchid1", "darkmagenta", "deepskyblue", "deepskyblue4",

"black", "black", "blue", "blue",

"magenta", "magenta", "chocolate4", "chocolate3")

#shape<-c(0,15,2,17,1,16)

#shape<-c(18,18,18,18,0,15,2,17,1,16,18,18)

dev.set()

#jpeg("tribplot1.jpeg", width = 6, height = 6, units = 'in', res = 300)

jpeg(filename="NMDS\_Spring\_2018.jpeg",res=600,height=8,width=12,units="in")

plot(NMDS2$points, col=co[color],asp=1,pch = shape[aspect], cex=1.2, xlab = "NMDS1", ylab = "NMDS2")

#Connect the points that belong to the same treatment with ordispider and create convex hulls around sites habitat (riffle, pool, run)

#ordispider(NMDS2, groups = verdhab$Hab, label = TRUE,cex=.8,col=c(17,76,54))

ordihull(NMDS1\_spring, verdhab\_spring$Micro, display= "sites", draw= c("polygon"),

col=NULL, border=c("magenta", "magenta", "magenta", "black", "black", "black", "blue", "blue", "blue" ) ,

lty= c(1), lwd=2.5)

ordihull(NMDS1\_spring, verdhab\_spring$order, display= "sites", draw= c("polygon"),

col=NULL, border=c("magenta", "black", "blue" ) ,

lty= c(1), lwd=2.5)

#ordihull(NMDS2,groups=verdhab$order,label=FALSE,cex=.6,col=c(17,76,54, 2, 30, 110, 92, 128, 150),draw="polygon",alpha=.05,lty=3)

#Add joint plot overlay of env correlations. We created this above#

#plot(joint,cex=.5,col="black",asp=1,p=0.05,family="serif")

plot(joint\_spring, choices = c(1,2), at = c(0,0),axis = FALSE, p.max = 0.05, col ="gray40", add = TRUE,cex=.5)

plot(joint2\_spring, choices = c(1,2), at = c(0,0),axis = FALSE, p.max = 0.05, col ="gray40", add = TRUE,cex=.7)

vectorfit(joint\_spring)

#Rotate ordination so that axis lines up with env variable

#MDSrotate(...)

#Add legend and additional text

txt <- c("Riffle Fall", "Riffle Spring", "Pool Fall", "Pool Spring", "Run Fall", "Run Spring")

legend('bottomright', txt\_spring , pch=c(15,16,17),col=c("magenta","black","blue"),

cex=1, bty = "y")

text(-1.2,-.7,pos=1,"Stress=15.86",cex=.9)

txt\_spring<- c("Riffle", "Pool", "Run")

legend('bottomright', txt , pch=c(0,15,1,16,2,17),col=c("magenta","magenta","black","black","blue","blue"),

cex=1, bty = "y")

dev.off()

#########################################################################

#########################

#####################

#Bootstrapping and testing for differences between the groups

fit <- adonis(sptrns ~ Hab, data=verdhab, permutations=999, method="bray")

fit

fit1 <- adonis(spdat ~ Hab, data=verdhab, permutations=999, method="bray")

fit1

#####################

#Check assumption of homogeneity of multivariate dispersion

distances\_data <- vegdist(sptrns)

anova(betadisper(distances\_data, verdhab$Hab))

distances\_data1 <- vegdist(spdat)

anova(betadisper(distances\_data1, verdhab$Hab))