# eDNA Data Analysis Workshop 2022

## Learning Outcomes:

1. Students are able to appropriately analyse data from an eDNA field sampling campaign;
2. Students are able to interpret and understand statistical outputs from the R package vegan.

library(vegan)

library(flextable)

library(stringr)

set\_flextable\_defaults(theme\_fun = "theme\_zebra", font.size = 9)

## Import the data

**Notes**:

1. This code imports the data from a file where the **samples are rows** and the **species are columns**.
2. We convert the column Site to a factor (categorical information in R)

eDNA2022 <- read.csv(file = "eDNA2022.csv")

eDNA2022$Site <- as.factor(eDNA2022$Site)

If you have species as rows and samples as columns, we will need to do a bit of wrangling!

speciesAsRows <- read.csv(file = "speciesAsRows.csv", row.names = 1)

speciesAsRows <- rbind(str\_remove(colnames(speciesAsRows), "X"),speciesAsRows)

row.names(speciesAsRows)[1] <- "SiteID"

speciesAsRows <- as.matrix(speciesAsRows) # convert to matrix for next step

speciesAsRows <- t(speciesAsRows) # t() means transpose

eDNA2022 <- as.data.frame(speciesAsRows)

eDNA2022$Site <- as.factor(eDNA2022$Site)

## List of all the species and their common names

altnames <-

c("MosqFish", "Goby", "Mullet", "Bream", "BnjoFrog", "GToadlet", "MlrdDuck",

"SpotDove", "SwampHen", "Crake", "WFHeron", "Lorikeet", "Cattle", "Sheep",

"Mouse", "Rat", "HydroidB", "HydroidO", "BwnJelly", "MoonJely", "Dog")

names\_table <-

data.frame(Taxonomic = gsub("."," ",colnames(eDNA2022[,4:24]), fixed = T),

Common=c("Mosquito Fish","Bridled Goby","Gray Mullet","Black Bream",

"Western Banjo Frog", "Gunther's Toadlet", "Mallard Duck",

"Spotted Turtle Dove", "Purple Swamphen", "Spotted Crake",

"White Faced Heron", "Rainbow Lorikeet", "Cattle", "Sheep",

"Mouse", "Rat", "HydroidB", "HydroidO", "Brown Jellyfish",

"Moon Jellyfish", "Domestic Dog"),

Abbreviated=altnames)

flextable(names\_table,cwidth = c(3,2,2))

|  |  |  |
| --- | --- | --- |
| **Taxonomic** | **Common** | **Abbreviated** |
| *Gambusia holbrooki* | Mosquito Fish | MosqFish |
| *Arenigobius bifrenatus* | Bridled Goby | Goby |
| *Mugil cephalus* | Gray Mullet | Mullet |
| *Acanthopagrus butcheri* | Black Bream | Bream |
| *Limnodynastes dorsalis* | Western Banjo Frog | BnjoFrog |
| *Pseudophryne guentheri* | Gunther's Toadlet | GToadlet |
| *Anas platyrhynchos* | Mallard Duck | MlrdDuck |
| *Spilopelia chinensis* | Spotted Turtle Dove | SpotDove |
| *Porphyrio porphyrio* | Purple Swamphen | SwampHen |
| *Porzana tabuensis* | Spotted Crake | Crake |
| *Egretta novaehollandiae* | White Faced Heron | WFHeron |
| *Trichoglossus rubritorquis* | Rainbow Lorikeet | Lorikeet |
| *Bos taurus* | Cattle | Cattle |
| *Ovis sp* | Sheep | Sheep |
| *Mus musculus* | Mouse | Mouse |
| *Rattus sp* | Rat | Rat |
| *Blackfordia polytentaculata* | HydroidB | HydroidB |
| *Obelia bidentata* | HydroidO | HydroidO |
| *Phyllorhiza punctata* | Brown Jellyfish | BwnJelly |
| *Aurelia sp* | Moon Jellyfish | MoonJely |
| *Canis lupus familiaris* | Domestic Dog | Dog |

allSpecies <- eDNA2022[,4:24] # not columns 1:3 which are sample IDs etc.

colnames(allSpecies) <- names\_table[,3]

## Difference and Similarity measures

We start by assessing how similar or different our samples are – to do this we need to set up a resemblance matrix. Ideally our 4 replicated samples are more similar to one another than any other samples – but with field sampling this is not always the case. To determine similarity we need to use a similarity measure.

### Similarity Measures:

A similarity measure is conventionally defined to take values between 0 and 100% (or 0 and 1) with the ends of the range representing the extreme possibilities:

* Similarity = 100% if two samples are totally similar
* Similarity = 0 if two samples are totally dissimilar

What constitutes total similarity, and particularly total dissimilarity, of two samples depends on the specific criteria that are adopted. For example, most biologists would feel that similarity should equal zero when two samples have no species in common, and similarity should equal 100% if two samples have identical entries. Similarities are calculated between every pair of samples creating a similarity matrix – a number of similarity measures can be used to create this matrix.

**Bray Curtis** and **Jaccard** are two different similarity measures we can use -– both are commonly used in ecology and have desirable properties -– similarity is measure based on taxa that samples have in common. Similarity takes a value of 0 if two samples have no species in common and joint absences have no effect (i.e. a taxon jointly absent in two samples does not contribute to its similarity). The difference between them is that Bray Curtis will use abundance information (where it is present) in assessing similarity, whereas Jaccard converts abundances to binary (i.e. 0s and 1s) to assess similarity. In our case it does not matter much which we use, since we have already converted to binary – so let’s proceed with Jaccard.

**Ecological Question – If two samples have no taxa present each does this make them more similar to one another?**

To assess similarity in the vegan R package carry out the following steps. The output shows the first few rows and columns of the dissimilarity matrix; each number represent the Jaccard distance between samples based on the presence or absence of species.

AF\_diss\_aquat <- vegdist(allSpecies,

distance = "jaccard") # default is distance = "bray"

# just look at the top corner of the dissimilarity matrix:

txt0 <- capture.output(print(AF\_diss\_aquat,digits=3));txt0[1:10];rm(txt0)

## [1] " 1 2 3 4 5 6 7 8 9 10 11"

## [2] "2 0.0769 "

## [3] "3 0.2727 0.1667 "

## [4] "4 0.0769 0.0000 0.1667 "

## [5] "5 0.3333 0.3846 0.6364 0.3846 "

## [6] "6 0.3333 0.3846 0.6364 0.3846 0.0000 "

## [7] "7 0.3333 0.3846 0.6364 0.3846 0.0000 0.0000 "

## [8] "8 0.2727 0.3333 0.6000 0.3333 0.0909 0.0909 0.0909 "

## [9] "9 0.8000 0.6364 0.5556 0.6364 0.6000 0.6000 0.6000 0.7778 "

## [10] "10 0.6000 0.6364 0.5556 0.6364 0.6000 0.6000 0.6000 0.7778 0.2500 "

# Ordinations

Next we will construct our ordination – a mechanism to visualise the similarity matrix. There are two different types of ordination that we will use – **Multi-Dimensional-Scaling** (nmMDS) and **Principal Coordinates Analysis** (PCO). If you have previously used PCA – Principal Component Analysis - you may be familiar with ordination methods. PCA is useful for environmental datasets – these are represented by a similarity matrix derived via use of Euclidean Distance as a measure of similarity. PCA is more suited to environmental variables because of the type of data – there are no large blocks of zeros (usually) and it is no longer necessary to select a similarity measure that ignores joint absences. When looking at ordination plots in general you should note that sample points that are spatially located together share common characteristics and sample points that are distantly located from one another share less characteristics. This holds whether we are working with environmental variables (pH, EC, moisture content, heavy metals etc.) or species abundances.

The main difference between MDS and PCO relates to the underlying use of the resemblance matrix.

**MDS** takes the similarity measures calculated by Jaccard or Bray Curtis and ranks them in order. Thus pairs are samples are ranked according to highest to lowest similarity and are thus plotted in multidimensional space.

**PCO** takes the value of underlying measure of similarity and plots pairs of samples in multi-dimensional space starting with the pairs of samples with highest level of similarity and working through the pairs of samples.

For both types of ordinations the software attempts to preserve the similarity of each pair of samples –- while this is feasible in multidimensional space, in order for us to view the ordination we see this in 2 or 3 dimensions -– thus the software attempts to preserve as much as is feasible of the sample pair similarity as calculated –- but in reality we need to consider how much the 2- dimensional view is a true representation of the data set. In nmMDS we use the stress value to evaluate this, in PCO we can plot multiple axes, i.e. axis 1 vs. axis 2, axis 2 vs. axis 3 and so on. In reality much of the variability is often explained in the first two axes (i.e. 2 dimensions) and we will limit our analysis to these axes for this dataset.

## Non-metric multidimensional scaling (nmMDS)

In vegan this is done using the metaMDS() function. We need to specify an input matrix of just the species data (e.g. allSpecies which we made earlier). The default is to find 2 dimensions (k) using a Bray-Curtis dissimilarity matrix, but we can set these options to different values using k = and distance = options.

AF\_nmds\_all <- metaMDS(allSpecies, distance = "jaccard")

AF\_nmds\_all3 <- metaMDS(allSpecies, k=3, distance = "jaccard")

AF\_nmds\_all

##

## Call:

## metaMDS(comm = allSpecies, distance = "jaccard")

##

## global Multidimensional Scaling using monoMDS

##

## Data: allSpecies

## Distance: jaccard

##

## Dimensions: 2

## Stress: 0.1554552

## Stress type 1, weak ties

## Two convergent solutions found after 20 tries

## Scaling: centring, PC rotation, halfchange scaling

## Species: expanded scores based on 'allSpecies'

The 2D stress value is 0.16. This can be interpreted as follows:

The stress value is relatively high as we are dealing with a large number of samples with lots of 1s and 0s in a 2 dimensional space – resulting in a challenging similarity environment.

👉 Check the stress value of the 3-D ordination – is it lower? Remember that we made this object in the code block above:

AF\_nmds\_all3

The following plot mimics the nmmds plot in Deirdre’s handout (note that the sign of dimension MDS2 is reversed)

palette(c("black","grey60","grey42",

rainbow(4,v=0.75,end=0.25),rainbow(4,v=0.75,start=0.45,end=0.8)))

plot(AF\_nmds\_all$points, pch=c(rep(21:25, 2),21)[eDNA2022$Site],

xlim = c(-1.6,2.2), ylim = c(1.2,-1.2),

bg=seq(1,11)[eDNA2022$Site],

cex = 1.4, main = "All species", col.main = "steelblue")

text(AF\_nmds\_all$points, labels=eDNA2022$SiteID, cex=0.9, pos = 4,

col=c("black","grey40","grey70",

rainbow(4,v=0.75,end=0.25),

rainbow(4,v=0.75,start=0.45,end=0.8))[eDNA2022$Site])

text(AF\_nmds\_all, display = "species", col = "#80008080", font=3, cex = 1.1)

legend("bottomright", inset = 0.01, box.col = 3, ncol = 4, title = "Site",

legend = seq(1,11), pch=c(rep(21:25, 2),21),

pt.bg = seq(1,11), pt.cex = 1.4, cex = 1.2)

You can play around with the code to make additional changes – you won’t break it by trying so play around as much as you like. You will notice that there are only 2 points showing for Site 2 – this is because the points are overlaid on top of one another (as we are picking up only a few taxa, some of the samples are very similar to one another). However, we can generally see that there are sample groupings and that likely location has an effect (*i.e*. samples that are from the same location (Site) are closer together). We will statistically test this a little later.

## Principal Coordinates Analysis

In the ‘vegan’ package we use Weighted classical multidimensional scaling, also known as Weighted Principal Coordinates Analysis. This is implemented using the function wcmdscale().

AF\_pco\_all <- wcmdscale(vegdist(allSpecies), eig=TRUE)

AF\_pco\_all

## Call: wcmdscale(d = vegdist(allSpecies), eig = TRUE)

##

## Inertia Rank

## Total 8.169

## Real 10.272 15

## Imaginary -2.103 21

##

## Results have 44 points, 15 axes

##

## Eigenvalues:

## [1] 3.377 2.101 1.537 1.297 0.613 0.434 0.342 0.214 0.158 0.100

## [11] 0.039 0.032 0.014 0.010 0.004 -0.006 -0.007 -0.009 -0.016 -0.021

## [21] -0.027 -0.035 -0.046 -0.052 -0.059 -0.074 -0.089 -0.094 -0.115 -0.134

## [31] -0.143 -0.163 -0.190 -0.212 -0.242 -0.371

##

## Weights: Constant

plot(AF\_pco\_all)

This is a basic PCO plot based on all species, showing the locations of individual samples by row number.

We can make a more informative plot by using the information stored in the output object (AF\_pco\_all).

palette(c("black","grey60","grey42",

rainbow(4,v=0.75,end=0.25),rainbow(4,v=0.75,start=0.45,end=0.8)))

plot(AF\_pco\_all$points[,1:2], xlim = c(0.4,-0.6), ylim = c(0.4,-0.6),

pch=3, cex = 0.5, col = "grey")

text(AF\_pco\_all$points[,1:2],labels = eDNA2022$SiteID,

col=seq(1,11)[eDNA2022$Site])

AF\_pco\_scores <- scores(AF\_pco\_all)

# PERMANOVA

In order to test where location has a significant effect on the taxa at our field site we need to run an ANOVA – however as we have multiple species to test together we cannot run a basic ANOVA – we must run a multivariate permutational ANOVA or PERMANOVA.

PERMANOVA shares some resemblance to ANOVA where they both measure the sum-of-squares within and between group and make use of an F test to compare within-group to between-group variance. However, while ANOVA bases the significance of the result on assumption of normality, PERMANOVA draws tests for significance by comparing the actual F test result to that gained from random permutations of the objects between the groups. Moreover, whilst PERMANOVA tests for similarity based on a chosen distance measure, ANOVA tests for similarity of the group averages.

We could calculate a univariate measure of diversity (Simpson or Shannon for example) and run an ANOVA on that univariate value. However in doing so we are reducing the complex dataset to a simplified version and losing discrete information in the process. Remember that ANOVA also has an underlying assumption of normality – generally speaking biological species data is not normally distributed and thus does not satisfy this assumption. This makes applying a permutational ANOVA a much better option. Before the advent of computational capacity a permutational ANOVA would have taken quite some time to run – luckily now it only takes a matter of seconds to minutes depending on the complexity of the dataset.

In the vegan package, the adonis2() function implements Permutational Multivariate Analysis of Variance Using Distance Matrices (i.e. PERMANOVA).

We use a formula in the adonis2() function which expresses the community matrix (i.e. the columns of species presence or absence) as a function of a factor (in this case, Site).

AF\_permanova\_all <- adonis2(eDNA2022[,4:24] ~ Site, data = eDNA2022,

permutations = 9999)

AF\_permanova\_all

## Permutation test for adonis under reduced model

## Terms added sequentially (first to last)

## Permutation: free

## Number of permutations: 9999

##

## adonis2(formula = eDNA2022[, 4:24] ~ Site, data = eDNA2022, permutations = 9999)

## Df SumOfSqs R2 F Pr(>F)

## Site 10 7.7475 0.94839 60.644 1e-04 \*\*\*

## Residual 33 0.4216 0.05161

## Total 43 8.1691 1.00000

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

P value by permutation = 0.0001 indicating a significant effect of location on the community composition of our samples. Note that 9999 permutations were undertaken to arrive at this value.

source("FUN\_pairwise\_adonis2.R")

source("FUN\_plain\_pairwise.R")

AF\_PWpermanova\_all <- pairwise.adonis2(eDNA2022[,4:24] ~ Site, data = eDNA2022)

AF\_plainPW\_all <- plainPW(AF\_PWpermanova\_all)

AF\_plainPW\_all

## Comparison P\_value

## 1 1\_vs\_2 0.024

## 2 1\_vs\_3 0.040

## 3 1\_vs\_4 0.038

## 4 1\_vs\_5 0.031

## 5 1\_vs\_6 0.027

## 6 1\_vs\_7 0.024

## 7 1\_vs\_8 0.025

## 8 1\_vs\_9 0.029

## 9 1\_vs\_10 0.037

## 10 1\_vs\_11 0.037

## 11 2\_vs\_3 0.032

## 12 2\_vs\_4 0.023

## 13 2\_vs\_5 0.032

## 14 2\_vs\_6 0.030

## 15 2\_vs\_7 0.025

## 16 2\_vs\_8 0.027

## 17 2\_vs\_9 0.028

## 18 2\_vs\_10 0.028

## 19 2\_vs\_11 0.034

## 20 3\_vs\_4 0.026

## 21 3\_vs\_5 0.032

## 22 3\_vs\_6 0.025

## 23 3\_vs\_7 0.034

## 24 3\_vs\_8 0.045

## 25 3\_vs\_9 0.019

## 26 3\_vs\_10 0.038

## 27 3\_vs\_11 0.028

## 28 4\_vs\_5 0.034

## 29 4\_vs\_6 0.021

## 30 4\_vs\_7 0.023

## 31 4\_vs\_8 0.025

## 32 4\_vs\_9 0.036

## 33 4\_vs\_10 0.029

## 34 4\_vs\_11 0.026

## 35 5\_vs\_6 0.027

## 36 5\_vs\_7 0.023

## 37 5\_vs\_8 0.037

## 38 5\_vs\_9 0.036

## 39 5\_vs\_10 0.030

## 40 5\_vs\_11 0.027

## 41 6\_vs\_7 0.035

## 42 6\_vs\_8 0.035

## 43 6\_vs\_9 0.023

## 44 6\_vs\_10 0.090

## 45 6\_vs\_11 0.020

## 46 7\_vs\_8 0.030

## 47 7\_vs\_9 0.021

## 48 7\_vs\_10 0.030

## 49 7\_vs\_11 0.025

## 50 8\_vs\_9 0.031

## 51 8\_vs\_10 0.030

## 52 8\_vs\_11 0.034

## 53 9\_vs\_10 0.024

## 54 9\_vs\_11 0.023

## 55 10\_vs\_11 0.034

**Notes**:

* the pairwise Adonis2 source code is by Pedro Martinez Arbizu at <https://github.com/pmartinezarbizu/pairwiseAdonis>.
* The code for the plainPW() function is as follows:

# function inputs output object from Pedro Martinez Arbizu's pairwise.adonis2()

# code at https://github.com/pmartinezarbizu/pairwiseAdonis

plainPW <- function(PWobj) {

UL\_PW\_perm <- unlist(PWobj) # convert list to long named vector

names\_PW\_perm <- names(UL\_PW\_perm) # extract the names...

rows\_pvals <- grep("F)1",names\_PW\_perm) # find the rows we need

justThePW <- # use the row indices to find the rown in the long vector

data.frame(Comparison = str\_remove(names(UL\_PW\_perm[rows\_pvals]),

fixed(".Pr(>F)1")),

P\_value = as.numeric(UL\_PW\_perm[rows\_pvals]))

return(justThePW) # and the resulting data frame is the output!

}

# Extras

### Stress plot for nmMDS based on Jaccard dissimilarities, all species

stressplot(AF\_nmds\_all, main="Stress plot")

## Analysis with just the aquatic species

These are: Gambusia holbrooki, Arenigobius bifrenatus, Mugil cephalus, Acanthopagrus butcheri, Limnodynastes dorsalis, Pseudophryne guentheri, Blackfordia polytentaculata, Obelia bidentata, Phyllorhiza punctata, Aurelia sp.

fish <- allSpecies[,c(1:6,17:20)]

head(fish)

## MosqFish Goby Mullet Bream BnjoFrog GToadlet HydroidB HydroidO BwnJelly

## 1 1 0 0 0 1 0 0 0 0

## 2 1 0 0 0 1 1 0 0 0

## 3 1 0 0 0 0 1 0 0 0

## 4 1 0 0 0 1 1 0 0 0

## 5 1 0 0 0 1 0 0 0 0

## 6 1 0 0 0 1 0 0 0 0

## MoonJely

## 1 0

## 2 0

## 3 0

## 4 0

## 5 0

## 6 0

### for interest look at the dissimilarity matrix

AF\_diss\_aquat <- vegdist(fish,

distance = "bray")

options(max.print=2000)

print(AF\_diss\_aquat, digits = 2)

## 1 2 3 4 5 6 7 8 9 10 11 12

## 2 0.200

## 3 0.500 0.200

## 4 0.200 0.000 0.200

## 5 0.000 0.200 0.500 0.200

## 6 0.000 0.200 0.500 0.200 0.000

## 7 0.000 0.200 0.500 0.200 0.000 0.000

## 8 0.000 0.200 0.500 0.200 0.000 0.000 0.000

## 9 0.500 0.200 0.000 0.200 0.500 0.500 0.500 0.500

## 10 0.333 0.500 0.333 0.500 0.333 0.333 0.333 0.333 0.333

## 11 1.000 0.500 0.333 0.500 1.000 1.000 1.000 1.000 0.333 1.000

## 12 0.500 0.200 0.000 0.200 0.500 0.500 0.500 0.500 0.000 0.333 0.333

AF\_nmds\_aquat <- metaMDS(fish, trymax = 500,

distance = "bray")

AF\_nmds\_aquat

##

## Call:

## metaMDS(comm = fish, distance = "bray", trymax = 500)

##

## global Multidimensional Scaling using monoMDS

##

## Data: fish

## Distance: bray

##

## Dimensions: 2

## Stress: 0.08664754

## Stress type 1, weak ties

## Two convergent solutions found after 20 tries

## Scaling: centring, PC rotation, halfchange scaling

## Species: expanded scores based on 'fish'

plot(AF\_nmds\_aquat, type="p",display = "sites", cex = 1.2)

text(AF\_nmds\_aquat, display = "species", col = "dodgerblue", cex = 0.8)

### alternative to vegan plots

palette(c("black","grey80","grey92",

rainbow(4,s=0.5,end=0.25),rainbow(4,s=0.5,start=0.45,end=0.8)))

plot(AF\_nmds\_aquat$points, pch=c(rep(21:25, 2),21)[eDNA2022$Site],

xlim = c(-1.2,1.8),

bg = seq(1,11)[eDNA2022$Site],

cex = 1.4, col.main = "steelblue", main = "Just aquatic species")

text(jitter(AF\_nmds\_aquat$points[,1], factor = 500),

jitter(AF\_nmds\_aquat$points[,2], factor=50),

labels=eDNA2022$SiteID, cex=1,

col=c("black","grey40","grey70",

rainbow(4,v=0.75,end=0.25),

rainbow(4,v=0.75,start=0.45,end=0.8))[eDNA2022$Site])

text(AF\_nmds\_aquat, display = "species", col = "grey33", font=3, cex = 0.9)

### Detrended correspondence analysis - all species

AF\_dca\_all <- decorana(allSpecies)

AF\_dca\_all

##

## Call:

## decorana(veg = allSpecies)

##

## Detrended correspondence analysis with 26 segments.

## Rescaling of axes with 4 iterations.

##

## DCA1 DCA2 DCA3 DCA4

## Eigenvalues 0.5537 0.3602 0.2545 0.17011

## Decorana values 0.5632 0.3154 0.1915 0.07493

## Axis lengths 3.2842 2.6926 2.1577 1.75539

plot(AF\_dca\_all)

### For comparison if wanted: nmMDS again with Bray-Curtis distances

AF\_nmds\_all\_B <- metaMDS(allSpecies) # uses Bray by default

AF\_nmds\_all\_B

##

## Call:

## metaMDS(comm = allSpecies)

##

## global Multidimensional Scaling using monoMDS

##

## Data: allSpecies

## Distance: bray

##

## Dimensions: 2

## Stress: 0.1554552

## Stress type 1, weak ties

## Two convergent solutions found after 20 tries

## Scaling: centring, PC rotation, halfchange scaling

## Species: expanded scores based on 'allSpecies'

The results should be identical to nmMDS with Jaccard, since the community matrix is already converted to presence-absence.