

eDNA Data Analysis Workshop 2023

Learning Outcomes:

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

```
library(vegan)      # for multivariate analyses
library(stringr)    # for manipulation of character strings
library(viridis)    # for making colourblind-friendly colour palettes
library(flextable)   # for making formatted tables
  set_flextable_defaults(theme_fun = "theme_zebra", font.size = 10)
library(sf)          # handling spatial data
library(maptiles)   # making tile-raster-based maps
library(prettymapr) # essential map annotations
```

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).

The following code imports the data from a CSV file and converts the numeric `Site` column into a factor (*i.e.* categorical column in R). The data file has species (and sample identifying data) as *columns* and samples as *rows*.

```
eDNA2023 <- read.csv(file = "eDNA2023.csv")
eDNA2023$Site <- as.factor(eDNA2023$Site)
```

If you have *species* as rows and *samples* as columns, we will need to do a bit of wrangling! (This could also be done in Excel but, of course, we like R)

```
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
eDNA2023 <- as.data.frame(speciesAsRows)
eDNA2023$Site <- as.factor(eDNA2023$Site)
```

```

git <- "https://raw.githubusercontent.com/Ratey-AtUWA/"
afr_map <- read.csv(file=paste0(git,"spatial/main/afr_map_v3.csv"),
                     stringsAsFactors = TRUE)
UTM50S <- st_crs(32750)
extent <- st_as_sf(x = data.frame(x = c(399860,400520), y = c(6467920,6468350)),
                     coords = c("x","y"), crs=UTM50S)
aftiles <- get_tiles(extent, provider="OpenStreetMap", crop=TRUE, zoom=17)
par(oma=c(3,3,1,1), mgp=c(1.5,0.2,0), tcl = -0.2,
    lend = "square", ljoin = "mitre", lheight=0.85)
plot_tiles(aftiles)
axis(1);axis(2);box(which="plot")
mtext("Easting (m)", 1, 1.6, font=2, cex=1.2)
mtext("Northing (m)", 2, 1.6, font=2, cex=1.2)
with(afr_map, polygon(wetland_E, wetland_N, col="lightblue", lty=1,
                      border="steelblue", lend="round", lwd=1))
with(afr_map, lines(drain_E, drain_N, col = "#8080FFB0", lwd=3))
with(afr_map, polygon(veg_E, veg_N, border = "#60806040", lwd = 5))
addnorthand("topleft", border=1, text.col=1)
addscalebar(pos="bottomright", plotepsg=32750, linecol=1, label.col=1,
            htin=0.15, label.cex=1.2, widthhint = 0.15)

# ----#---- A D D   D A T A  ----#----
with(eDNA2023[seq(1,NROW(eDNA2023),4),], points(Easting, Northing, col=10,
                                                     bg="orchid1", pch=21, cex=1.5, lwd=2))
with(eDNA2023[seq(1,NROW(eDNA2023),4),], text(Easting, Northing, labels=Site,
                                                col="purple", pos=4, cex = 1, font=4, offset = 0.3))

```

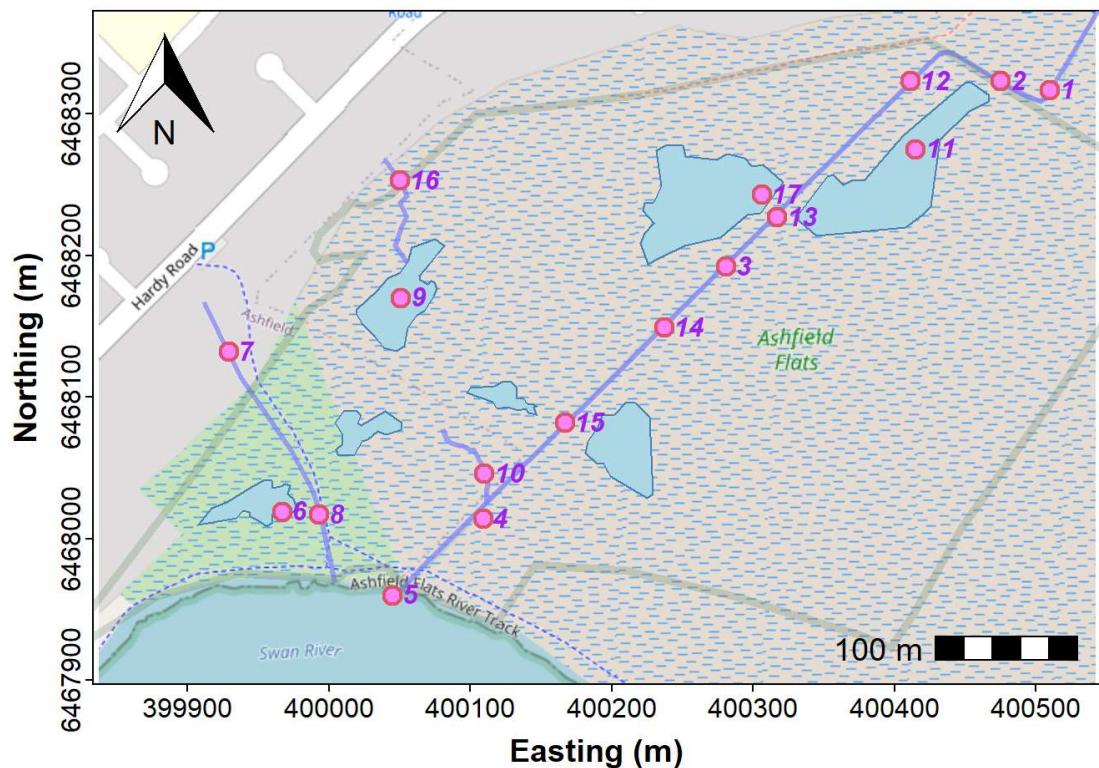


Figure 1: Map of locations where eDNA samples were taken in March 2023 at Ashfield Flats Reserve.

```

altnames <-
c("MosqFish", "Goby", "Mullet", "Bream", "BnjoFrog", "GToadlet", "MlrdDuck",
  "SpotDove", "SwampHen", "SpoCrake", "WFHeron", "Loriikeet", "Cattle", "Sheep",
  "Mouse", "Rat", "HydroidB", "HydroidO", "BwnJelly", "MoonJely", "Dog")
names_table <-
data.frame(Taxonomic = gsub(".", " ", colnames(eDNA2023[,6:26])), 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)) |> italic(j=1,part="body") |>
  set_caption(caption="List of all the species detected using eDNA, their common names, and abbreviated names used in plots in this document.")

```

Table 1: List of all the species detected using eDNA, their common names, and abbreviated names used in plots in this document.

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

We need to make a subset of our imported data which contains just the species presence/absence information (not sample IDs, sites, reps). We also overwrite the original column names with abbreviated names for better visualisation later.

```

allSpecies <- eDNA2023[,6:26] # not columns 1:5 which are sample IDs etc.
colnames(allSpecies) <- names_table[,3]

```

"I think that the formation of [DNA's] structure by Watson and Crick may turn out to be the greatest developments in the field of molecular genetics in recent years."

— Linus Pauling (https://en.wikipedia.org/wiki/Linus_Pauling), April 1953

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% (or 1) 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. There are a number of similarity measures (ways to calculate similarity) which 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 calculated based on the 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, with 1 representing presence and 0 meaning absence) 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 represents the Jaccard distance between samples based on the presence or absence of species.

```
AF_diss_all <- vegdist(allSpecies,
                         distance = "jaccard") # default is distance = "bray"

# just look at the top corner of the dissimilarity matrix:
txt0 <- capture.output(print(AF_diss_all, digits=3)); txt0[1:10]; rm(txt0)
```

```
##      1    2    3    4    5    6    7    8    9    10   11
## 2  0.0769
## 3  0.2727 0.1667
## 4  0.0769 0.0000 0.1667
## 5  0.3333 0.3846 0.6364 0.3846
## 6  0.3333 0.3846 0.6364 0.3846 0.0000
## 7  0.3333 0.3846 0.6364 0.3846 0.0000 0.0000
## 8  0.2727 0.3333 0.6000 0.3333 0.0909 0.0909 0.0909
## 9  0.8000 0.6364 0.5556 0.6364 0.6000 0.6000 0.6000 0.7778
## 10 0.6000 0.6364 0.5556 0.6364 0.6000 0.6000 0.6000 0.7778 0.2500
## 11 0.8000 0.6364 0.5556 0.6364 0.8000 0.8000 0.8000 1.0000 0.2500 0.2500
```

Ordinations

Next we will construct our ordination – a procedure to visualise the similarity matrix by simplifying (scaling) it into fewer dimensions than our original data. This reduction of dimensions is useful as the original data will have as many dimensions as there are species, which is impossible to visualise! 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. Since it uses the rank and not the actual values, it's commonly called “non-[para]metric” multi-dimensional scaling, or **nmMDS**.

PCO takes the actual values of the underlying measure of similarity and plots pairs of samples in multi-dimensional space. It will then look for the direction (vector) of greatest variance in the multi-dimensional “cloud” of points, which becomes the first PCO dimension. Successive dimensions explain the most possible remaining variance, until all variance is accounted for. So, many PCO dimensions are possible, but usually only the first 2 or 3 are useful for visualizing our data.

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. The amount of variance explained in each PCO dimension is called the *eigenvalue*, and the output will include these values; the greater the eigenvalue, the more useful a PCO dimension is. In practice, 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. The dissimilarity matrix is calculated within the `metaMDS()` function, so for consistency we set `distance = "jaccard"`.

```
# default with 2 dimensions
AF_nmmds_all <- metaMDS(allSpecies, distance = "jaccard")
# 3 dimensions for comparison
AF_nmmds_all3 <- metaMDS(allSpecies, k=3, distance = "jaccard")
```

```
# show output for 2 nmMDS dimensions
AF_nmmds_all
```

```

## 
## Call:
## metaMDS(comm = allSpecies, distance = "jaccard")
##
## global Multidimensional Scaling using monoMDS
##
## Data:      allSpecies
## Distance: jaccard
##
## Dimensions: 2
## Stress:     0.1876453
## 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:



Figure 2: Interpretation of the stress parameter in non-metric multidimensional scaling ordination.

The stress value in this example is relatively high as we are dealing with a large number of samples with lots of 1 and 0 values 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_nmmds_all3
```

The following plot (Figure 3) mimics the nmmds plot in Deirdre's handout (note that the sign of dimension MDS2 is reversed).

```

palette(c("black",magma(15,alpha = 0.8),"white"))
par(mar=c(3,3,1,1), mgp=c(1.6,0.3,0), tcl=0.25, font.lab=2)
plot(AF_nmmds_all$points, pch=c(rep(21:25, 3),21,22)[eDNA2023$Site],
      xlim = c(-1.6,2.2), ylim = c(1.2,-1.2),
      bg=c(1:17)[eDNA2023$Site],
      cex = 1.4, main = NA, col.main = "steelblue")
# text(AF_nmmds_all$points, Labels=eDNA2023$Site, cex=0.9, pos = 4,
#       col="grey40")
text(AF_nmmds_all, display = "species", col = "#20208080", font=3, cex = 0.9)
legend("bottomright", inset = 0.01, box.col = 3, ncol = 4,
       x.intersp=0.75, title = "Site",
       legend = seq(1,17), pch=c(rep(21:25, 3),21,22),
       pt.bg = c(1:17), pt.cex = 1.4, cex = 1.1)

```

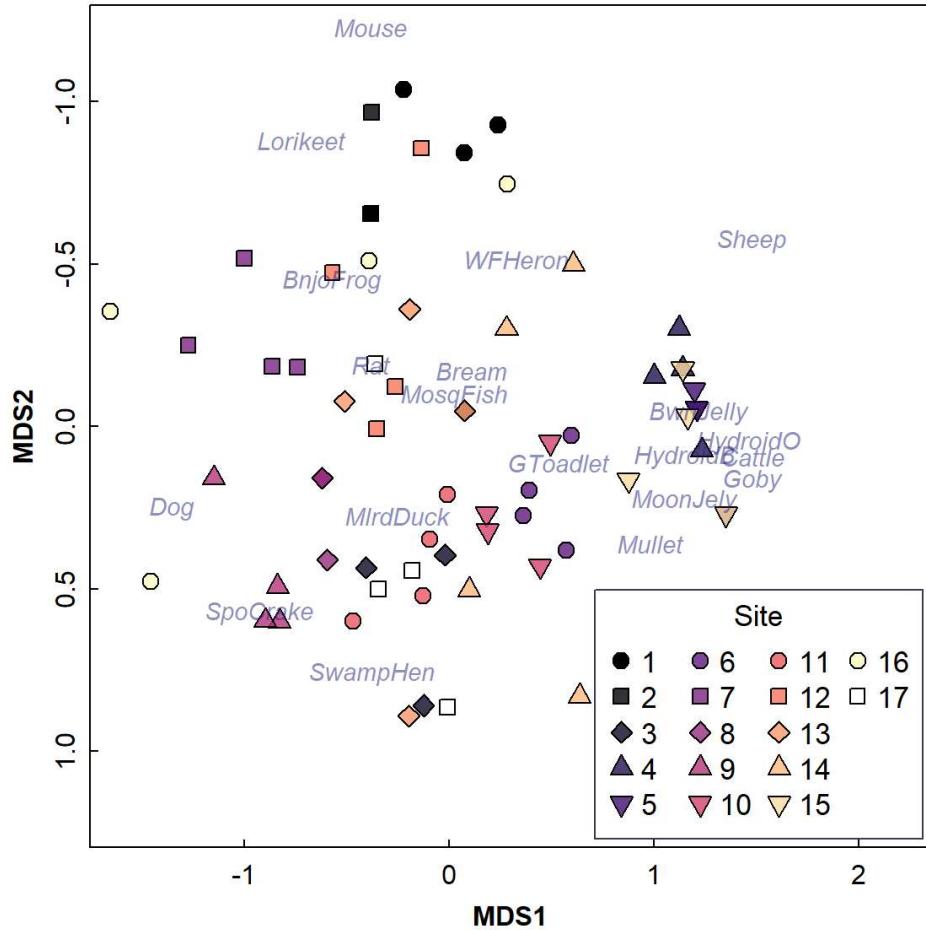


Figure 3: Two-dimensional nmMDS plot, showing ordination of species and samples, for the Ashfield Flats eDNA data including DNA from all species except humans.

We can plot the nmMDS ordination in different ways (e.g. Figure 4) – choose which one you think is easiest to interpret, or code your own!

```

palette(c("#00000080",magma(15,alpha = 0.5),"white"))
par(mar=c(3,3,1,1), mgp=c(1.6,0.3,0), tcl=0.25, font.lab=2)
plot(AF_nmmds_all$points, pch=21,
      xlim = c(-1.6,2.2), ylim = c(1.2,-1.2),
      bg=c(1:17)[eDNA2023$Site],
      cex = 2.6, main=NA, col.main = "steelblue")
palette(c("black",magma(15),"white"))
text(AF_nmmds_all$points, labels=eDNA2023$Site, cex=0.9,
      col=c(seq(17,13,-1),1,1,17,1,1,seq(5,1,-1))[eDNA2023$Site])
TeachingDemos::shadowtext(AF_nmmds_all$species,
                           labels = row.names(AF_nmmds_all$species),
                           col = "#20208080", bg="#FFFFFF80", font=3, cex = 0.8)

```

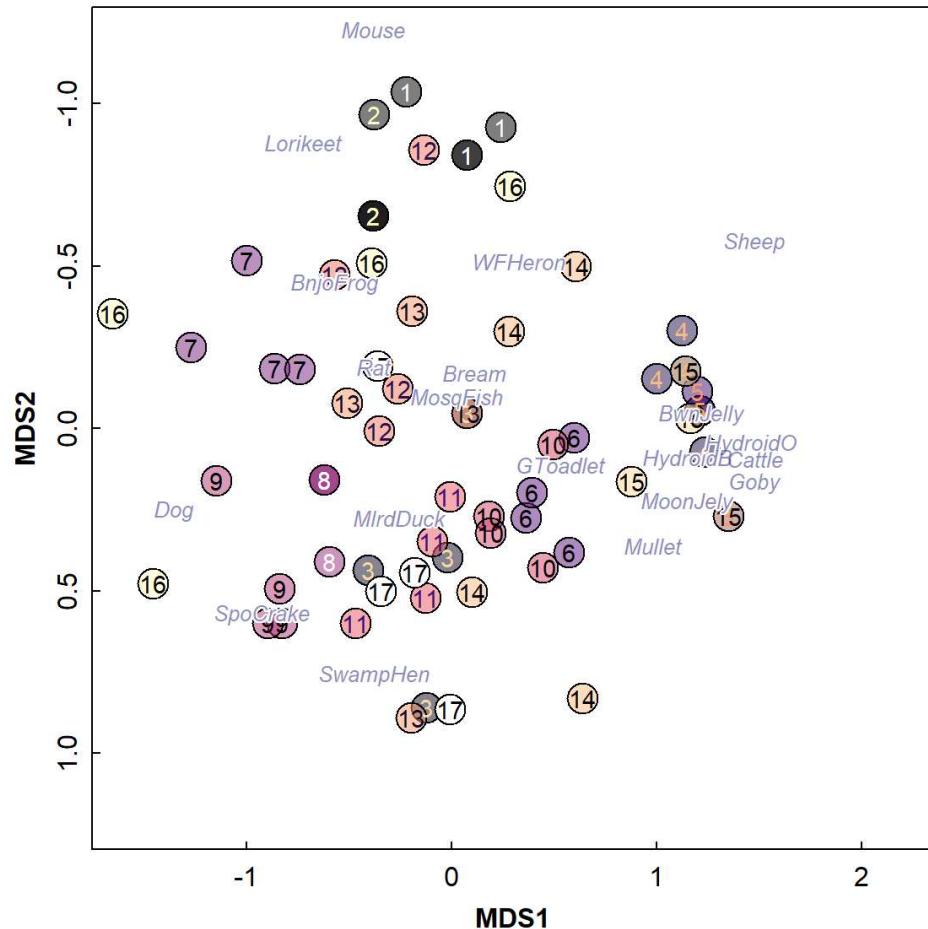


Figure 4: NMDS ordination plot (alternative plot style) for all species detected using eDNA analysis of water from Ashfield Flats.

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 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 are closer together). We will test this statistically a little later.

Principal Coordinates Analysis

In the `vegan` R 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      12.99
## Real       18.02   17
## Imaginary  -5.03   41
##
## Results have 68 points, 17 axes
##
## Eigenvalues:
## [1] 5.032 2.784 2.289 2.050 1.255 1.082 0.948 0.774 0.471 0.434
## [11] 0.323 0.183 0.174 0.085 0.073 0.044 0.017 -0.004 -0.005 -0.006
## [21] -0.008 -0.009 -0.010 -0.011 -0.015 -0.017 -0.020 -0.025 -0.029 -0.032
## [31] -0.036 -0.037 -0.046 -0.050 -0.056 -0.061 -0.063 -0.065 -0.081 -0.089
## [41] -0.096 -0.101 -0.109 -0.129 -0.135 -0.146 -0.151 -0.166 -0.179 -0.201
## [51] -0.240 -0.276 -0.308 -0.336 -0.361 -0.387 -0.418 -0.513
##
## Weights: Constant

```

```
plot(AF_pco_all)
```

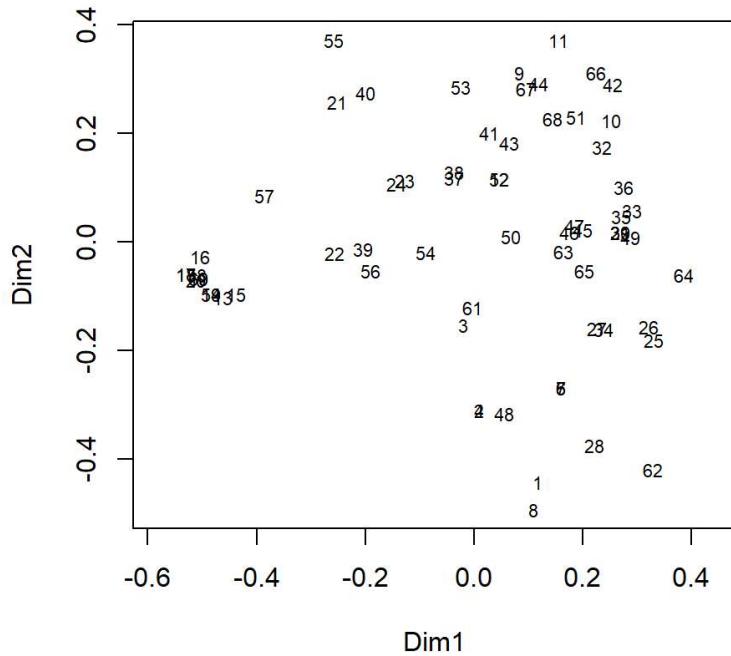


Figure 5: Principal coordinates analysis plot of an R object made using the `wcmdscale()` function from the `vegan` package.

Figure 5 is a basic PCO plot based on all species, showing the locations of individual samples by row number.

We can make more informative plots (Figure 6) by using the information stored in the output object (`AF_pco_all`).

```

palette(c("black","grey60","grey42",viridis(8),"white"))
par(mfrow = c(1,2), mar = c(4,4,1,1), mgp = c(1.7,0.3,0), tcl = 0.25,
font.lab = 2)
plot(AF_pco_all$points[,c(1,2)], xlim = c(0.4,-0.6), ylim = c(0.4,-0.6),
pch=3, cex = 0.75, col = 5)
text(AF_pco_all$points[,c(1,2)],labels = eDNA2023$SiteID,
col=seq(1,11)[eDNA2023$Site])
mtext("(a)",line=-1.3, adj=0.03, cex=1.2)
plot(AF_pco_all$points[,c(1,3)], xlim = c(0.4,-0.6), ylim = c(0.4,-0.6),
pch=4, cex = 0.75, col = 7)
text(AF_pco_all$points[,c(1,3)],labels = eDNA2023$SiteID,
col=seq(1,11)[eDNA2023$Site])
mtext("(b)",line=-1.3, adj=0.03, cex=1.2)

```

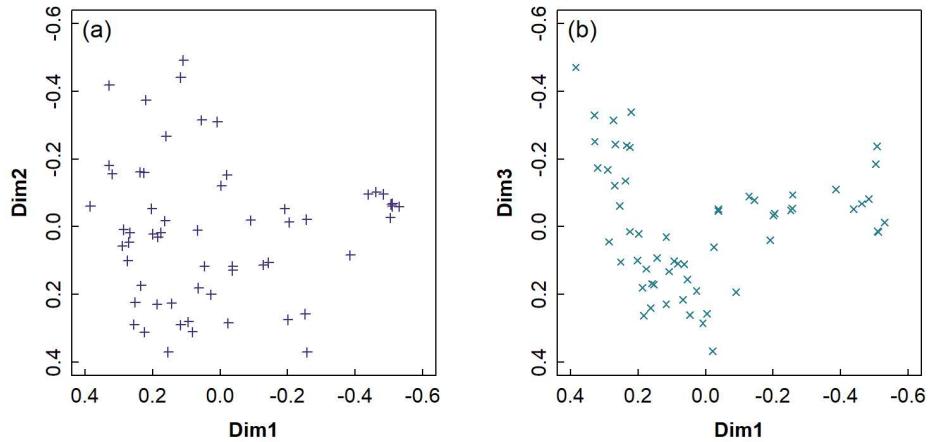


Figure 6: Alternative principal coordinates analysis plots of an R object made using the `wcmdscale()` function from the `vegan` package: (a) first and second dimensions; (b) first and third dimensions.

```
AF_pco_scores <- scores(AF_pco_all) # not sure if we need this!?
```



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 groups, 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 (a Simpson or Shannon index, for example) and run an ANOVA on that univariate value. In doing so, however, 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 are not normally distributed and thus do 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(eDNA2023[,6:26] ~ Site, data = eDNA2023,
                               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 = eDNA2023[, 6:26] ~ Site, data = eDNA2023, permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Site      16    9.766 0.75204 9.6674 1e-04 ***
## Residual 51    3.220 0.24796
## Total     67   12.986 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P value by permutation = `1e-04` indicating a significant effect of location on the community composition of our samples. Note that 9999 permutations were undertaken to arrive at this value.

The output from the `pairwise.adonis2()` function is quite lengthy (to see it, run `print(AF_PWpermanova_all)`). We can tidy it up for clarity with a custom function (`plainPW()` – see Table 2 below).

```
source("https://github.com/Ratey-AtUWA/eDNA/raw/master/FUN_pairwise_adonis2.R")
source("https://github.com/Ratey-AtUWA/eDNA/raw/master/FUN_plain_pairwise.R")
AF_PWpermanova_all <- pairwise.adonis2(eDNA2023[,6:26] ~ Site, data = eDNA2023)
AF_plainPW_all <- plainPW(AF_PWpermanova_all) # see custom function code below
colnames(AF_plainPW_all)[1] <- "Comparison.between.Sites"
flextable(AF_plainPW_all, cwidth = c(2.5,2)) |>
  align(align="center", part = "all") |>
  font(i = ~ P_value <=0.05, j = ~P_value, font = "Arial Black") |>
  set_caption(caption="Pairwise comparison p-values (p<=0.05 in bold font) from PERMANOVA analysis of eDNA data from water samples at Ashfield Flats.")
```

Table 2: Pairwise comparison p-values (p≤0.05 in bold font) from PERMANOVA analysis of eDNA data from water samples at Ashfield Flats.

Comparison.between.Sites	P_value
1_vs_2	0.031
1_vs_3	0.030
1_vs_4	0.032
1_vs_5	0.029
1_vs_6	0.023

Comparison.between.Sites	P_value
1_vs_7	0.027
1_vs_8	0.019
1_vs_9	0.035
1_vs_10	0.031
1_vs_11	0.029
1_vs_12	0.025
1_vs_13	0.033
1_vs_14	0.030
1_vs_15	0.024
1_vs_16	0.293
1_vs_17	0.033
2_vs_3	0.029
2_vs_4	0.024
2_vs_5	0.034
2_vs_6	0.027
2_vs_7	0.032
2_vs_8	0.030
2_vs_9	0.031
2_vs_10	0.032
2_vs_11	0.036
2_vs_12	0.970
2_vs_13	0.032
2_vs_14	0.037
2_vs_15	0.030
2_vs_16	0.341
2_vs_17	0.024
3_vs_4	0.026
3_vs_5	0.028
3_vs_6	0.027
3_vs_7	0.045
3_vs_8	0.026
3_vs_9	0.027
3_vs_10	0.032
3_vs_11	0.025
3_vs_12	0.054
3_vs_13	0.787
3_vs_14	0.516
3_vs_15	0.034
3_vs_16	0.400
3_vs_17	0.153
4_vs_5	0.029

Comparison.between.Sites	P_value
4_vs_6	0.035
4_vs_7	0.025
4_vs_8	0.034
4_vs_9	0.025
4_vs_10	0.020
4_vs_11	0.027
4_vs_12	0.035
4_vs_13	0.032
4_vs_14	0.054
4_vs_15	0.594
4_vs_16	0.027
4_vs_17	0.026
5_vs_6	0.035
5_vs_7	0.027
5_vs_8	0.037
5_vs_9	0.029
5_vs_10	0.026
5_vs_11	0.033
5_vs_12	0.023
5_vs_13	0.029
5_vs_14	0.028
5_vs_15	0.777
5_vs_16	0.026
5_vs_17	0.019
6_vs_7	0.032
6_vs_8	0.033
6_vs_9	0.023
6_vs_10	0.066
6_vs_11	0.037
6_vs_12	0.035
6_vs_13	0.031
6_vs_14	0.186
6_vs_15	0.028
6_vs_16	0.036
6_vs_17	0.024
7_vs_8	0.033
7_vs_9	0.030
7_vs_10	0.020
7_vs_11	0.033
7_vs_12	0.031
7_vs_13	0.020

Comparison.between.Sites	P_value
7_vs_14	0.028
7_vs_15	0.024
7_vs_16	0.495
7_vs_17	0.033
8_vs_9	0.031
8_vs_10	0.025
8_vs_11	0.028
8_vs_12	0.031
8_vs_13	0.030
8_vs_14	0.040
8_vs_15	0.033
8_vs_16	0.406
8_vs_17	0.034
9_vs_10	0.035
9_vs_11	0.029
9_vs_12	0.030
9_vs_13	0.032
9_vs_14	0.029
9_vs_15	0.028
9_vs_16	0.553
9_vs_17	0.026
10_vs_11	0.038
10_vs_12	0.031
10_vs_13	0.027
10_vs_14	0.169
10_vs_15	0.033
10_vs_16	0.126
10_vs_17	0.033
11_vs_12	0.030
11_vs_13	0.414
11_vs_14	0.163
11_vs_15	0.025
11_vs_16	0.052
11_vs_17	1.000
12_vs_13	0.923
12_vs_14	0.051
12_vs_15	0.027
12_vs_16	0.609
12_vs_17	0.307
13_vs_14	0.678
13_vs_15	0.029

Comparison.between.Sites	P_value
13_vs_16	0.553
13_vs_17	0.731
14_vs_15	0.064
14_vs_16	0.306
14_vs_17	0.206
15_vs_16	0.034
15_vs_17	0.027
16_vs_17	0.366

Notes:

- the pairwise Adonis2 source code is by Pedro Martinez Arbizu at github.com/pmartinezarbizu/pairwiseAdonis (<https://raw.githubusercontent.com/pmartinezarbizu/pairwiseAdonis/master/pairwiseAdonis/R/pairwise.adonis2.R>).
- 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("F1",names_PW_perm) # find the rows we need
  justThePW <- # use the row indices to find the rows in the Long vector
    data.frame(Comparison = str_remove(names(UL_PW_perm[rows_pvals]),
                                       fixed(".Pr(>F1"))),
               P_value = as.numeric(UL_PW_perm[rows_pvals]))
  return(justThePW) # and the resulting data frame is the output!
}
```

You can also get the `plainPW()` function code at github.com/Ratey-AtUWA/.../FUN_plain_pairwise.R (https://github.com/Ratey-AtUWA/eDNA/raw/master/FUN_plain_pairwise.R).

Optional Extras

```
stressplot(AF_nmds_all)
```

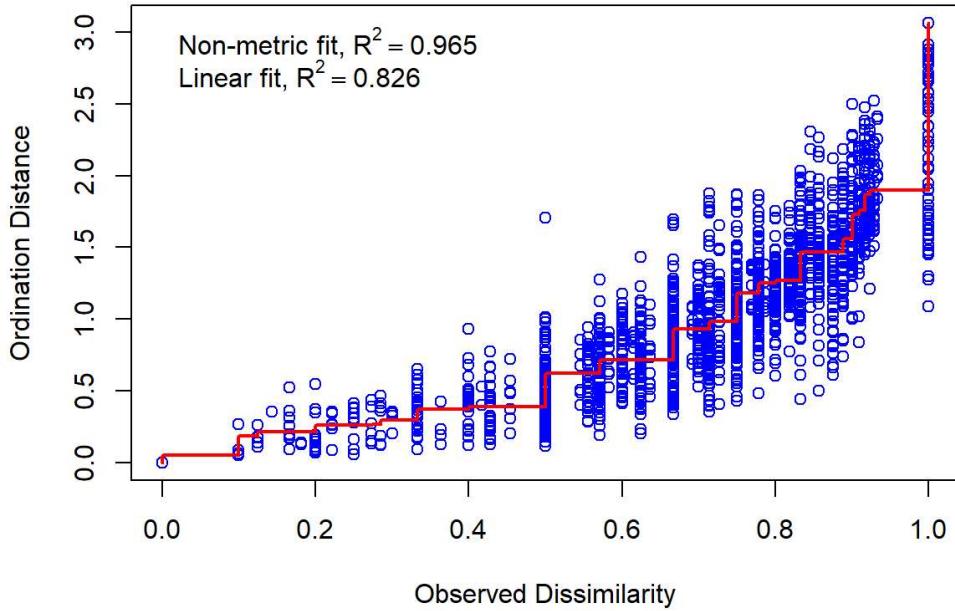


Figure 7: Stress plot for nmMDS based on Jaccard dissimilarities, based on presence/absence of all species from eDNA analysis of Ashfield Flats water samples.

This stress plot (Figure 7) shows the decrease in ordination stress with an increase in the number of ordination dimensions, indicating the fit between ordination distances and observed dissimilarities. Large scatter around the line suggests that original dissimilarities are not well preserved in the reduced number of dimensions. You would proceed to steps of plotting your NMDS if you identify a minimized stress solution.

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")
txt0 <- capture.output(print(AF_diss_aquat,digits=3));txt0[1:10];rm(txt0)
```

```

##      1     2     3     4     5     6     7     8     9     10    11
## 2  0.2000
## 3  0.5000  0.2000
## 4  0.2000  0.0000  0.2000
## 5  0.0000  0.2000  0.5000  0.2000
## 6  0.0000  0.2000  0.5000  0.2000  0.0000
## 7  0.0000  0.2000  0.5000  0.2000  0.0000  0.0000
## 8  0.0000  0.2000  0.5000  0.2000  0.0000  0.0000  0.0000
## 9  0.5000  0.2000  0.0000  0.2000  0.5000  0.5000  0.5000  0.5000
## 10 0.3333  0.5000  0.3333  0.5000  0.3333  0.3333  0.3333  0.3333  0.3333
## 11 1.0000  0.5000  0.3333  0.5000  1.0000  1.0000  1.0000  0.3333  1.0000

```

```

AF_nmds_aquat <- metaMDS(fish, trymax = 500,
                           distance = "jaccard")

```

```
AF_nmds_aquat
```

```

##
## Call:
## metaMDS(comm = fish, distance = "jaccard", trymax = 500)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      fish
## Distance: jaccard
##
## Dimensions: 2
## Stress:     0.0969898
## 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)

```

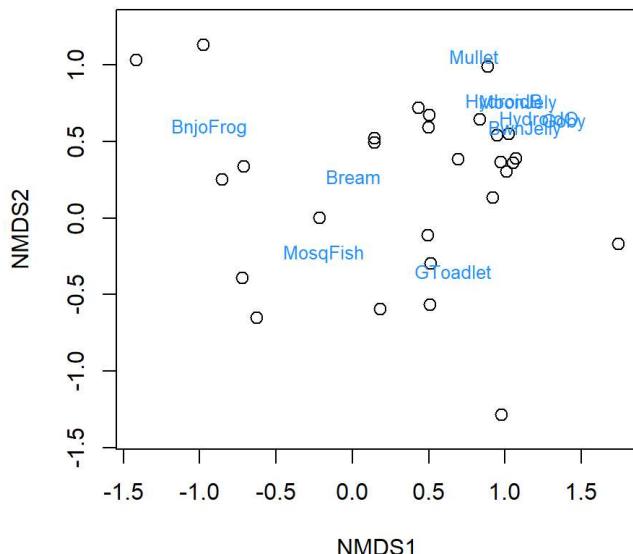


Figure 8: Ashfield Flats eDNA nmMDS plot based on just the aquatic species and Bray dissimilarities.

alternative to vegan plots

```
palette(viridis::plasma(17))
plot(AF_nmds_aquat$points, pch=c(rep(21:25, 3),21,22)[eDNA2023$Site],
      xlim = c(-1.2,1.8),
      bg = seq(1,17)[eDNA2023$Site],
      cex = 1.4, col.main = "steelblue", main = "Just aquatic species")
# text(AF_nmds_aquat$points, pos = rep(c(1,2,4),6)[eDNA2023$Site],
#       Labels=eDNA2023$Site, cex=1,
#       col=c(1:17)[eDNA2023$Site])
text(AF_nmds_aquat, display = "species", col = "#20208080", font=3, cex = 0.9)
legend("bottomleft", inset = 0.01, box.col = 3, ncol = 6,
      x.intersp=0.75, title = "Site",
      legend = seq(1,17), pch=c(rep(21:25, 3),21,22),
      pt.bg = seq(1,17), pt.cex = 1, cex = 0.9)
```

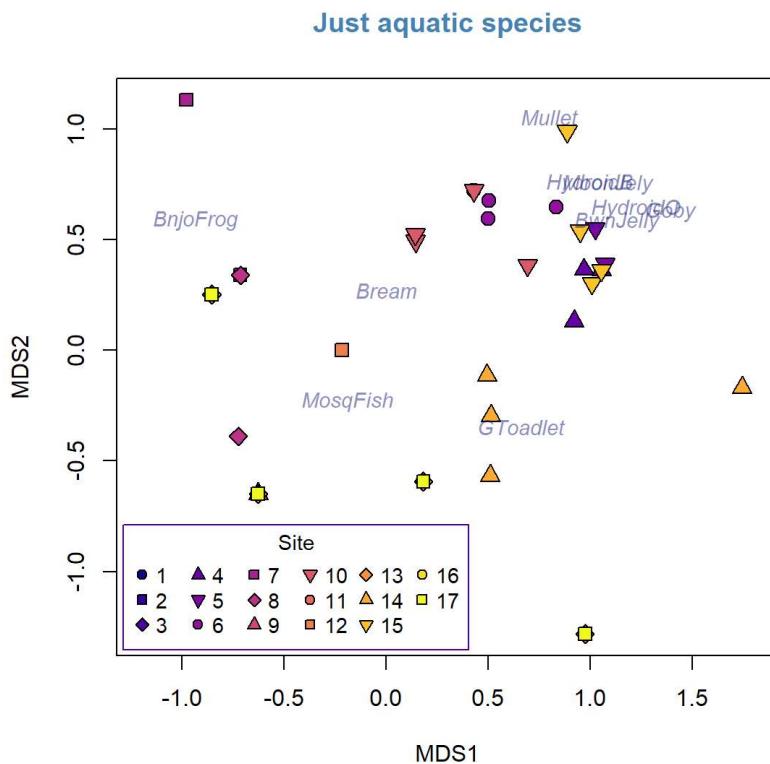


Figure 9: Ashfield Flats eDNA nmMDS plot based on just the aquatic species and Bray dissimilarities, plotted using base R functions.

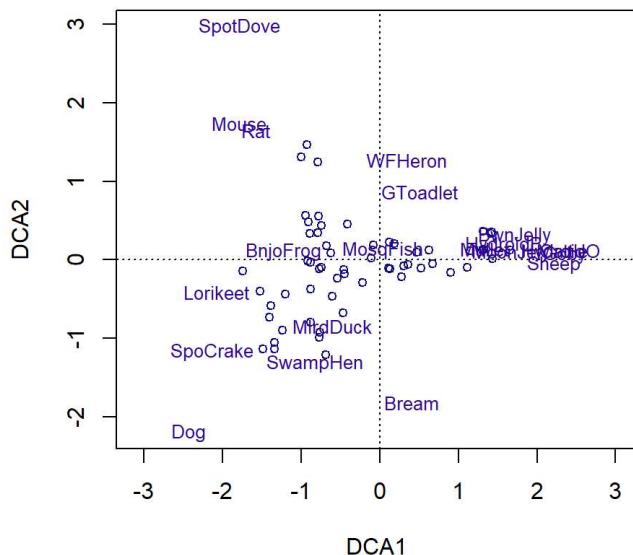
Detrended correspondence analysis – all species

```
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  0.5468 0.2969 0.2391 0.1746
## Decorana values 0.5609 0.2572 0.1638 0.0935
## Axis lengths  3.4015 2.6765 2.2793 2.1290
```

```
plot(AF_dca_all)
```



(#fig:decorana plot) Ashfield Flats eDNA detrended correspondence analysis plot based on all species.

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

```
AF_nmmds_all_B <- metaMDS(allSpecies) # uses Bray by default
```

```
AF_nmmds_all_B
```

```
##  
## Call:  
## metaMDS(comm = allSpecies)  
##  
## global Multidimensional Scaling using monoMDS  
##  
## Data:      allSpecies  
## Distance: bray  
##  
## Dimensions: 2  
## Stress:     0.1876453  
## 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 (binary ones and zeros).

References and R Packages

Garnier S, Ross N, Rudis R, Camargo AP, Sciai M, Scherer C (2021). *Rvision - Colorblind-Friendly Color Maps for R (viridis)*. R package version 0.6.2. <https://sjmgarnier.github.io/viridis/> (<https://sjmgarnier.github.io/viridis/>)

Oksanen J, Simpson G, Blanchet F, Kindt R, Legendre P, Minchin P, O'Hara R, Solymos P, Stevens M, Szoecs E, Wagner H, Barbour M, Bedward M, Bolker B, Borcard D, Carvalho G, Chirico M, De Caceres M, Durand S, Evangelista H, FitzJohn R, Friendly M, Furneaux B, Hannigan G, Hill M, Lahti L, McGlinn D, Ouellette M, Ribeiro Cunha E, Smith T, Stier A, Ter Braak C, Weedon J (2022). *vegan: Community Ecology Package*. R package version 2.6-2, <https://CRAN.R-project.org/package=vegan> (<https://CRAN.R-project.org/package=vegan>).

Gohel D, Skintzos P (2022). *flextable: Functions for Tabular Reporting*. R package version 0.8.1, <https://CRAN.R-project.org/package=flextable> (<https://CRAN.R-project.org/package=flextable>).

Wickham H (2022). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.4.1, <https://CRAN.R-project.org/package=stringr> (<https://CRAN.R-project.org/package=stringr>).

Created with `rmarkdown` in RStudio.