Introduction to non-metric multidimensional scaling

This is a brief tutorial on non-metric multidimensional scaling using the `vegan` package in `R`. I use data from an experiment that tested the effects of urchin grazing on the community structure of subtidal rock walls (Elahi and Sebens 2012).

Load packages

```
library(vegan)
library(tidyverse)
theme_set(theme_bw(base_size = 14) +
            theme(strip.background = element_blank(),
                  panel.grid = element_blank()))

## Define the repository from which we get the data
repo_url <- "https://raw.githubusercontent.com/elahi/phd_elahi/master/"
```

Load data

```
sjc_sessile <- read_csv(
  paste(repo_url,
        "Elahi_sessile_quadrat/output/sjc_master_compiled/sessile_compiled_final_wide.csv",
        sep = ""))
dat <- sjc_sessile %>%
  filter(dataset == "elahi_urchin_addition") %>%
  select(-X1, -X)
names(dat)[1:10]
```

```
##  [1] "quadrat"         "transect"        "site"            "year"
##  [5] "dataset"         "real.date"       "comm.date"       "urchin_addition"
##  [9] "urchin_removal"  "chiton_removal"
```

Explore data

```
dat %>%
  count(site, urchin_addition, transect)
```

```
## # A tibble: 18 x 4
##    site  urchin_addition transect     n
##    <chr> <chr>           <chr>    <int>
##  1 ON    Control         ON0         12
##  2 ON    Control         ON2         12
##  3 ON    Control         ON4         12
##  4 ON    Urchin          ONA         12
##  5 ON    Urchin          ONB         12
##  6 ON    Urchin          ONC         12
##  7 PG    Control         PG1         12
##  8 PG    Control         PG3         12
```

```
##  9 PG    Control         PG6         12
## 10 PG    Urchin          PA          12
## 11 PG    Urchin          PB          12
## 12 PG    Urchin          PC          12
## 13 SC    Control         4           12
## 14 SC    Control         5           12
## 15 SC    Control         7           12
## 16 SC    Urchin          A           12
## 17 SC    Urchin          B           12
## 18 SC    Urchin          C           12
```

```
dat %>% count(dateR)
```

```
## # A tibble: 2 x 2
##   dateR          n
##   <date>     <int>
## 1 2009-06-27   108
## 2 2009-09-24   108
```

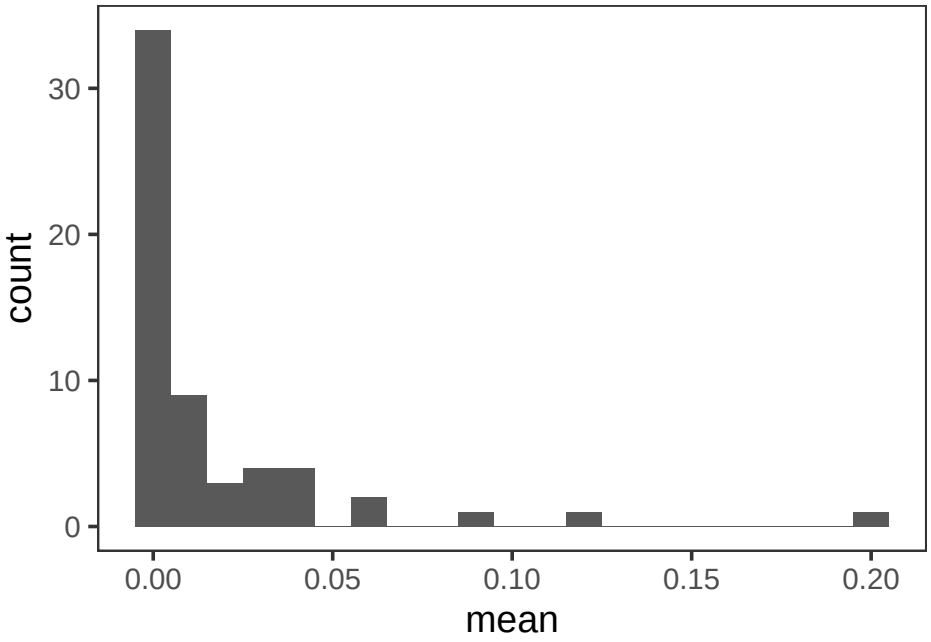
```
dat <- dat %>% filter(dateR == "2009-09-24")
```

Examine percent cover of all sessile taxa and functional groups

```
datL <- dat %>%
  gather(key = species, value = cover, bare_rock:rubble)

cover_means <- datL %>%
  group_by(species) %>%
  summarise(mean = mean(cover)) %>%
  arrange(mean)

cover_means %>%
  ggplot(aes(mean)) +
  geom_histogram(binwidth = 0.01)
```



Remove uncommon sessile taxa

Here, I use a 1% cover threshold. I do this for simplicity - not for biological relevance.

```
species_subset <- cover_means %>% filter(mean >= 0.01)
species_subset
```

```
## # A tibble: 19 x 2
##    species                          mean
##    <chr>                           <dbl>
##  1 shell_debris                   0.0111
##  2 Pycnoclavella_stanleyi         0.0144
##  3 bare_rock                      0.0147
##  4 tube_calcareous                0.0171
##  5 bryozoan_encrusting            0.0177
##  6 Balanus_crenatus               0.0188
##  7 hydroid_other                  0.0300
##  8 sponge_other                   0.0303
##  9 Haliclona_spp1                 0.0317
## 10 Balanophyllia_elegans          0.0330
## 11 Didemnum_carnulentum           0.0373
## 12 Schizoporella_japonica         0.0379
## 13 fuzz                           0.0385
## 14 Eurystomella_bilabiata         0.0431
## 15 hydro_bryo_complex             0.0623
## 16 red_filamentous_algae          0.0639
## 17 Metandrocarpa_taylori          0.0903
## 18 encrusting_coralline_algae     0.117
## 19 encrusting_non_calcified_algae 0.204
```

Prepare data for ordination

```
# create spp matrix
abun <- dat[, names(dat) %in% species_subset$species]

# transform matrix
abun_sq <- sqrt(abun)

# presence/absence
abun_pa <- decostand(abun, method="pa")

# create group matrix
dat_group <- dat[,1:8]
```

To illustrate the approach, I use the default settings in the function `metaMDS` to run the mds on proportional cover data (i.e., these data were not transformed). But later, I will run the mds on square-root transformed data, and presence/absence data.

```
## Rename the matrix
mds_mat <- abun

## Run nmds, using default settings
mds_results <- metaMDS(mds_mat)
```

```
## Run 0 stress 0.2079075
## Run 1 stress 0.2160132
## Run 2 stress 0.2084744
```

```
## Run 3 stress 0.2079075
## ... New best solution
## ... Procrustes: rmse 1.138468e-05  max resid 7.534283e-05
## ... Similar to previous best
## Run 4 stress 0.2319878
## Run 5 stress 0.2337023
## Run 6 stress 0.2079076
## ... Procrustes: rmse 5.637358e-05  max resid 0.0002382564
## ... Similar to previous best
## Run 7 stress 0.2079075
## ... Procrustes: rmse 2.384004e-05  max resid 0.0001443116
## ... Similar to previous best
## Run 8 stress 0.2433538
## Run 9 stress 0.2084744
## Run 10 stress 0.251587
## Run 11 stress 0.4122132
## Run 12 stress 0.2220487
## Run 13 stress 0.2084745
## Run 14 stress 0.2084746
## Run 15 stress 0.2084744
## Run 16 stress 0.2079075
## ... Procrustes: rmse 1.664218e-05  max resid 7.794969e-05
## ... Similar to previous best
## Run 17 stress 0.2084744
## Run 18 stress 0.2084746
## Run 19 stress 0.2436327
## Run 20 stress 0.2226901
## *** Solution reached
```

```
# Using the scores function from vegan to extract the site scores and convert to a data.frame
data_scores <- as.data.frame(scores(mds_results))

# Combining with group dataframe
data_scores <- cbind(data_scores, dat_group)
data_scores %>% head()
```

```
##        NMDS1      NMDS2       quadrat transect site year               dataset
## 1 -0.5220007 -0.1298724 P1010004_crop        A   SC 2009 elahi_urchin_addition
## 2 -0.5589906 -0.1494296 P1010005_crop        A   SC 2009 elahi_urchin_addition
## 3 -0.1143992  0.1254921 P1010006_crop        A   SC 2009 elahi_urchin_addition
## 4 -0.1934488  0.4538695   P1010008_IJ        A   SC 2009 elahi_urchin_addition
## 5 -0.3171015  0.1178059   P1010009_IJ        A   SC 2009 elahi_urchin_addition
## 6 -0.1214966  0.4643560   P1010010_IJ        A   SC 2009 elahi_urchin_addition
##   real.date comm.date urchin_addition
## 1   9/24/09 9/24/2009          Urchin
## 2   9/24/09 9/24/2009          Urchin
## 3   9/24/09 9/24/2009          Urchin
## 4   9/24/09 9/24/2009          Urchin
## 5   9/24/09 9/24/2009          Urchin
## 6   9/24/09 9/24/2009          Urchin
```

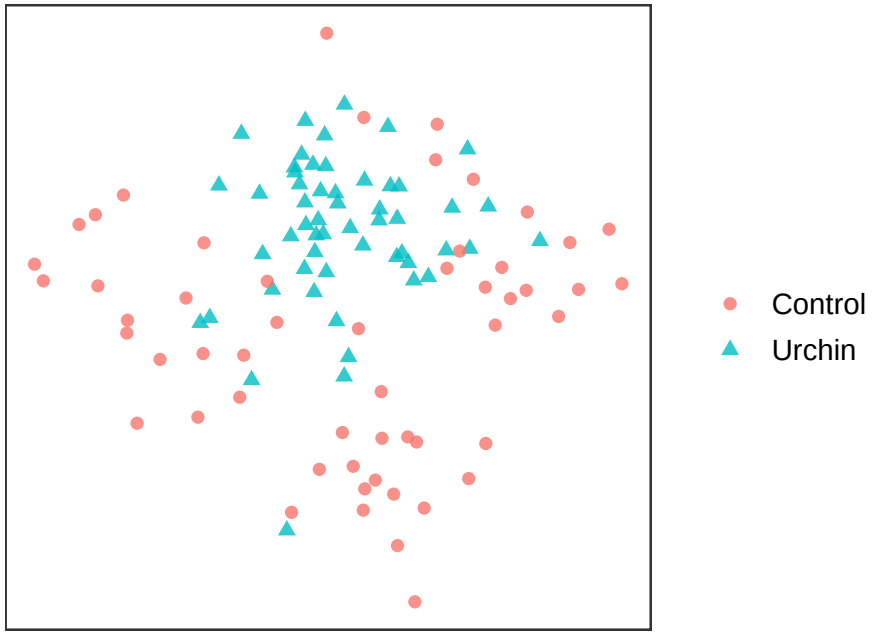
```
# Using the scores function from vegan to extract the species scores and convert to a data.frame
species_scores <- as.data.frame(scores(mds_results, "species"))

# Create a column of species, from the rownames of species.scores
species_scores$species <- rownames(species_scores)
head(species_scores)  #look at the data
```

```
##                                      NMDS1       NMDS2
```

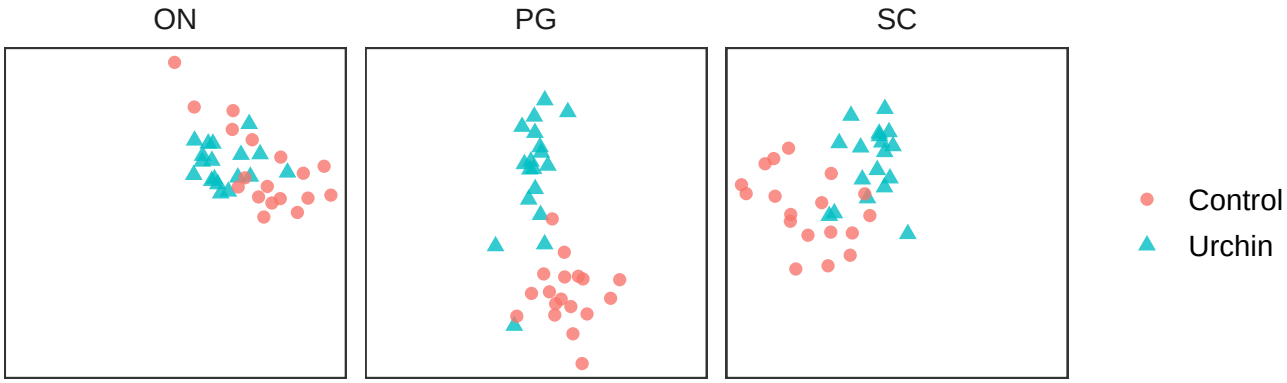
```
## bare_rock                       -0.13220926  0.22631846
## encrusting_coralline_algae       0.02115456 -0.09514028
## encrusting_non_calcified_algae   0.02409301  0.45758276
## red_filamentous_algae            0.10033776 -0.74075941
## hydro_bryo_complex               1.01632882  0.12396096
## Haliclona_spp1                   0.95338033  0.22350400
##                                                      species
## bare_rock                                          bare_rock
## encrusting_coralline_algae         encrusting_coralline_algae
## encrusting_non_calcified_algae encrusting_non_calcified_algae
## red_filamentous_algae                   red_filamentous_algae
## hydro_bryo_complex                         hydro_bryo_complex
## Haliclona_spp1                                 Haliclona_spp1
```

```
data_scores %>%
  ggplot(aes(x = NMDS1, y = NMDS2, shape = urchin_addition,colour = urchin_addition)) +
  geom_point(size = 2, alpha = 0.8) + # add the point markers
  coord_equal() +
  theme(axis.text = element_blank(),  # remove axis text
        axis.ticks = element_blank(),  # remove axis ticks
        axis.title = element_blank(), # remove axis labels
        legend.title = element_blank())
```



Now separate by site.

```
data_scores %>%
  ggplot(aes(x = NMDS1, y = NMDS2, shape = urchin_addition,colour = urchin_addition)) +
  geom_point(size = 2, alpha = 0.8) + # add the point markers
  coord_equal() +
  theme(axis.text = element_blank(),  # remove axis text
        axis.ticks = element_blank(),  # remove axis ticks
        axis.title = element_blank(), # remove axis labels
        legend.title = element_blank()) +
  facet_wrap(~ site)# remove y-axis labels)
```



Comparing NMDS after transformations

```
# Perform the NMDS in 2 dimensions on raw data
set.seed(132)
abun_MDS2 <- metaMDS(abun, distance="bray", k=2, engine = 'monoMDS',
                     autotransform=FALSE, noshare=0.1, trymax=40, zerodist='add')

# Perform the NMDS in 2 dimensions on sqrt data
set.seed(132)
abun_sq_MDS2 <- metaMDS(abun_sq, distance="bray", k=2, engine = 'monoMDS',
                        autotransform=FALSE, noshare=0.1, trymax=40, zerodist='add')

# Perform the NMDS in 2 dimensions on presence / absence data
set.seed(12345678) # was difficult to converge
abun_pa_MDS2 <- metaMDS(abun_pa, distance="bray", k=2, engine = 'monoMDS',
                        autotransform=FALSE, noshare=0.1, trymax=40, zerodist='add')
```

```
extract_data_scores <- function(mds_results, dat_group){
# Using the scores function from vegan to extract the site scores and convert to a data.frame
data_scores <- as.data.frame(scores(mds_results))
# Combining with group dataframe
data_scores <- cbind(data_scores, dat_group)
return(data_scores)
}

extract_species_scores <- function(mds_results){
# Using the scores function from vegan to extract the species scores and convert to a data.frame
species_scores <- as.data.frame(scores(mds_results, "species"))
# Create a column of species, from the rownames of species.scores
species_scores$species <- rownames(species_scores)
return(species_scores)
}
```

```
abun_MDS2$stress
```

```
## [1] 0.2079075
```

```
abun_sq_MDS2$stress
```

```
## [1] 0.2090932
```

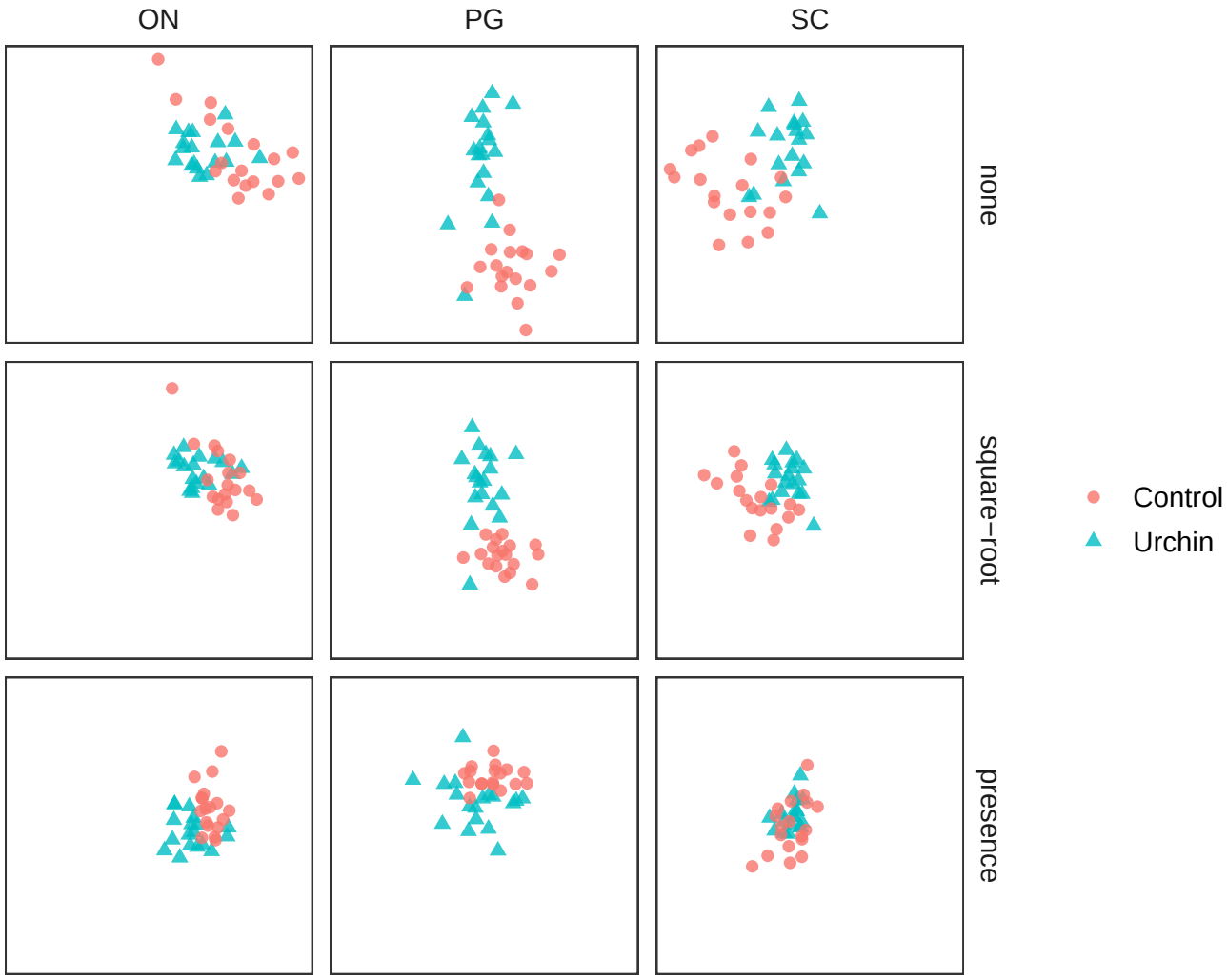
```
abun_pa_MDS2$stress
```

```
## [1] 0.2278666
```

```
abun_ds <- extract_data_scores(mds_results = abun_MDS2, dat_group = dat_group) %>%
  mutate(transformation = "none")
abun_sq_ds <- extract_data_scores(mds_results = abun_sq_MDS2, dat_group = dat_group) %>%
  mutate(transformation = "square-root")
abun_pa_ds <- extract_data_scores(mds_results = abun_pa_MDS2, dat_group = dat_group) %>%
  mutate(transformation = "presence")

data_scores <- rbind(abun_ds, abun_sq_ds, abun_pa_ds) %>%
  mutate(transformation = factor(transformation,
                                 levels = c("none", "square-root", "presence")))
```

```
data_scores %>%
  ggplot(aes(x = NMDS1, y = NMDS2, shape = urchin_addition,colour = urchin_addition)) +
  geom_point(size = 2, alpha = 0.8) + # add the point markers
  coord_equal() +
  theme(axis.text = element_blank(),  # remove axis text
        axis.ticks = element_blank(),  # remove axis ticks
        axis.title = element_blank(), # remove axis labels
        legend.title = element_blank()) +
  facet_grid(transformation ~ site)# remove y-axis labels)
```



–> –>