8. Non-metric Multidimensional Scaling

FISH 560: Applied Multivariate Statistics for Ecologists

Topics

- Non-metric Multidimensional scaling (NMDS)

R Packages: vegan, cluster
R Source: biostats

BACKGROUND

Principal component analysis (and its variants) utilizes matrix algebra to derive unique, successive linear axes such that distances among objects in multivariate space are well represented in a low-dimensional ordination space. Non-metric multidimensional scaling (NMDS) obtains a similar representation using only the rank order of distances, rather than the distance values themselves (Digby and Kempton 1987). The theoretical advantage of this rank-order approach to ordination is that underlying assumptions of linearity (as in PCA and variants) are not required nor specified. The lack of underlying assumptions, however, necessitates the use of a computationally intensive iterative algorithm to derive an optimized ordination configuration. At each NMDS iteration the rank order relationship between ordination and variable space distances is improved through successive approximation. Iteration continues until the stress function, which measures the correspondence between ranked ordination and multivariate space distances, is minimized. In other words, the *stress* value indicates the faithfulness of the ordination configuration to the original dissimilarity matrix. The final solution is an optimized rank-order mapping of the sampling units in an ordination space of specified dimensionality. The solution obtained from a single run may not be globally optimal, however, since NMDS is based on an iterative algorithm. It is therefore imperative that multiple NMDS solutions be obtained in order to ensure that a stable and optimal ordination configuration is found. Because only rank order relationships are used, NMDS solutions are unstable or even degenerate when applied to small data sets or to poorly structured data.

Previously, the numerically intensive calculations required to achieve a stable NMDS solution precluded its use in ecology. But with increased computing power this approach has grown in popularity and is now arguably one of the most commonly used ordination techniques in community ecology. Part of this popularity is owed to NMDS's ability to handle any (dis)similarity coefficient. For species abundance data sets, simulation testing has indicated that NMDS may outperform other commonly used ordination techniques (e.g., PCA, PCoA, CA, DCA; Gauch 1982, Reynolds et al. 1988, McCune 2002). That is, the object configuration depicted in the NMDS ordination better captured the simulated species abundance patterns.

SET-UP

In this exercise you will be working with the MAHA species abundance dataset. But first remember to set-up your R work session by defining the current work directory to your folder of choice and loading the vegan library. Also, make sure to source the BIOSTATS file from the *File* pull-down menu. You can also do this using the functions **`setwd`**, **`library`** and **`source`**. Import the dataset by typing:

```
speabu <- read.csv('MAHA_speciesabu.csv',header=TRUE, row.names=1)
```

Let's transform the data before diving into the analysis. We will do this because the species abundance dataset is highly skewed and contains some rather large values which are valid but highly influential. Remember, the log of zero is undefined so we'll add 1 to each value in our data set.

```
speabu.log<-log(speabu+1)
```

NMDS

To perform NMDS we'll use the function metaMDS() which is part of the vegan library. Its usage is:

metaMDS(comm, distance = "bray", k = 2, trymax = 20, autotransform =TRUE)

Where:

1. comm is the community data set (object-by-descriptor matrix)
2. distance is the dissimilarity coefficient used (calculated using vegdist). Default is Bray-Curtis.
3. k, the number of ordination axis to generate
4. autotransformation, logical. If TRUE, metaMDS autotransfoms the data (usually using square root transformation) using simple heurisitics. Default is TRUE, but set this to FALSE.
5. trymax, the maximum number of random starts in search for stable solution. Default is 20.

metaMDS combines several functions from isoMDS (in the MASS package) that automate the NMDS iterative process. To perform NMDS piecemeal, you can type ?metaMDS for additional documentation of the underlying functions. Let's perform metaMDS using our log-transformed dataset.

```
spe.nmds<-metaMDS(speabu.log, distance='bray', k=2, autotransform=FALSE,
trymax=100)
```

Type **spe.nmds**. Your results should be somewhat similar as below.

```
Call:
metaMDS(comm = speabu.log, k = 2, trymax = 100, autotransform = FALSE)

Nonmetric Multidimensional Scaling using isoMDS (MASS package)

Data:     speabu.log
Distance: bray

Dimensions: 2
Stress:     0.2320339
Two convergent solutions found after 36 tries
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on 'speabu.log'
```

You can type **names(spe.nmds)** to obtain a list of objects resulting from the analysis.

```
[1] "points"    "dims"      "stress"    "data"      "distance"  "converged"
[7] "tries"     "species"   "call"
```

For example, spe.nmds$points contains the co-ordinates from the first two axes (in case you wanted to plot the results in another program). From the above summary we can see that the stress level is relatively high and therefore indicates a poor fit between the original distance matrix and the final ordination configuration. To improve the fit, we could try a different transformation (presence/absence) or perform NMDS with three axis (k=3). We'll repeat the procedure here.

```
spe.nmds2<-metaMDS(speabu.log, distance='bray', k=3, autotransform=FALSE,
trymax=100)
```

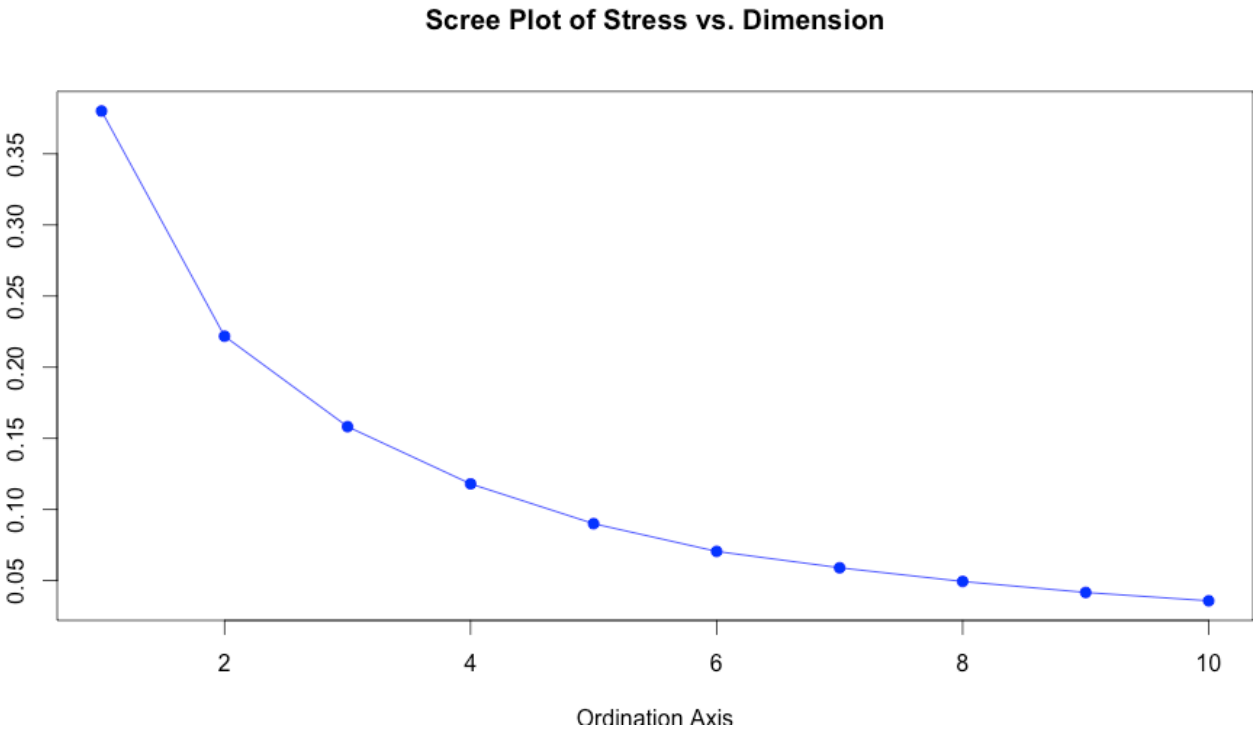
On this run, the stress improved to 16.4 indicating that the major gradients in the data set can be sufficiently captured by three dimensions. Of course, increasing the number of ordination axis will

reduce the stress value (at k = n species, stress must = 0 because all species would be represented by a unique ordination axis), but the utility of NMDS, as with any ordination technique, is to summarize as much variation as possible using the fewest number of axis. Ultimately, it is the user who must decide whether the addition of dimensions is justified by the reduction in stress.

Examining a scree plot of stress versus the number of dimensions can help you make this decision. To perform this type:

```
nmds.scree(speabu.log, distance='bray', k=10, autotransform=FALSE, trymax=20)
```

You will be provided a scree plot that looks similar to this:



This function basically calls the metaMDS function as before, but this time it calls it once for each number of dimensions and then plots the final stress value against the number of dimensions.

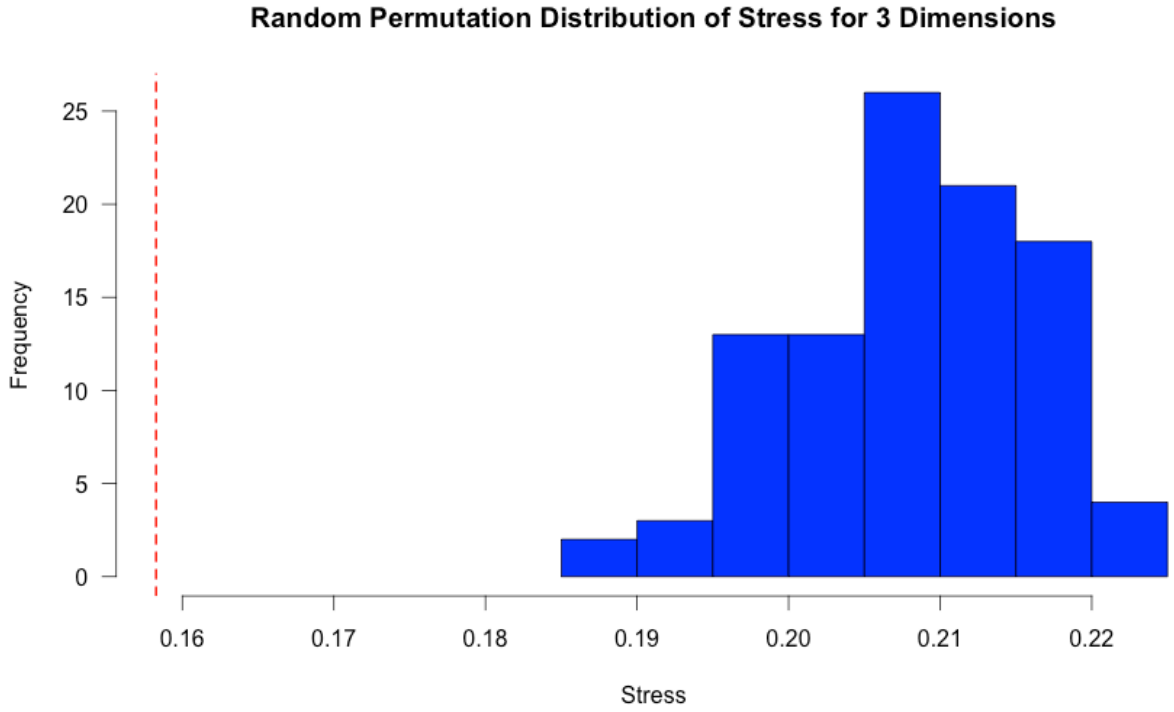
Once the final number of dimensions has been decided upon, a Monte Carlo randomization test of the final stress value can be conducted as follows. Note that this will take a couple minutes to complete.

```
nmds.monte(speabu.log, distance='bray', k=3, autotransform=FALSE, trymax=20)
```

This will return the permuted stress values (and histogram) and calculated p-value.

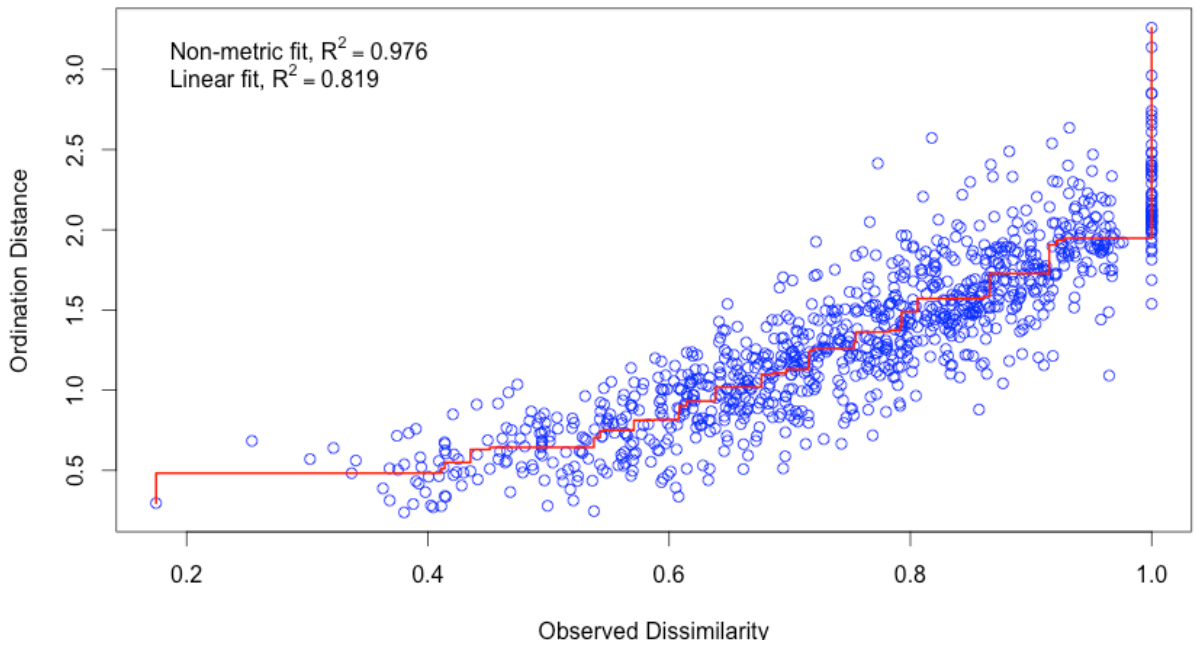
```
Randomization Test of Stress:
Permutation stress values:
  [1] 20.81755 21.27511 20.59101 21.77180 21.60949 20.86109 21.47368 19.93579
  [9] 21.10165 21.90203 21.65971 20.55388 21.27061 20.55968 21.74566 20.94443
 [17] 21.62887 21.04472 21.32917 21.90709 20.66803 20.82322 19.69421 20.85727
 [25] 21.79995 20.46723 20.99515 21.60599 20.84756 21.62835 20.23694 20.29475
…
```

```
      [,1]
Observed stress 16.36179378
P-value          0.00990099
```



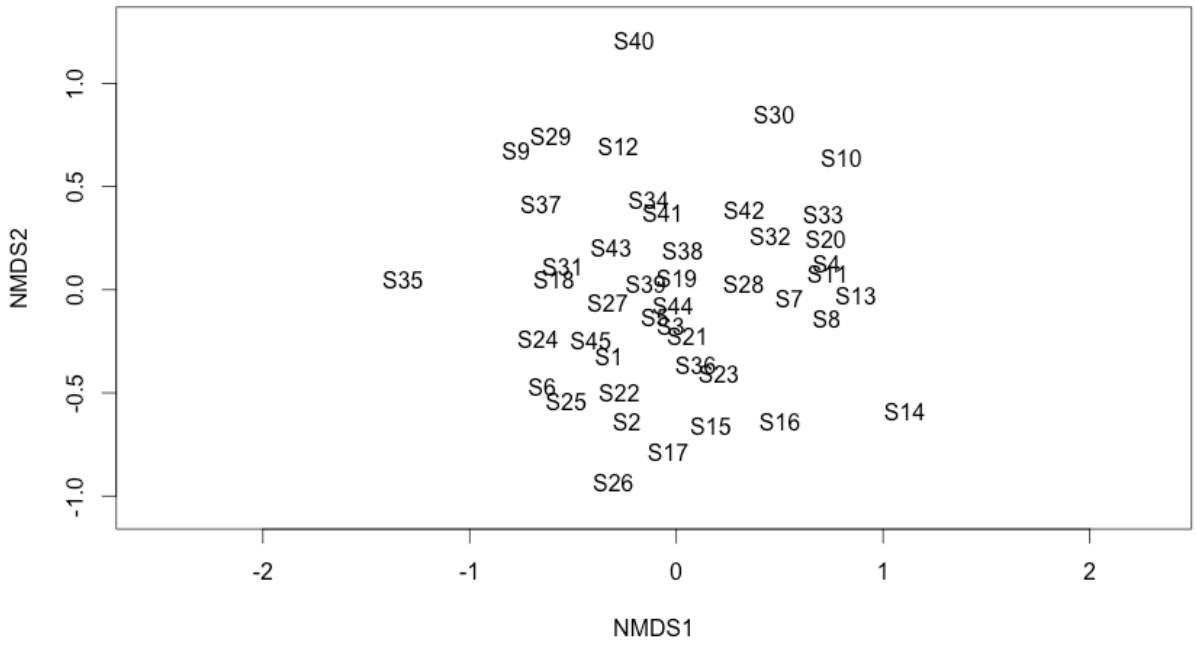
How good a job does NMDS do? Well, another way to determine this is to look at the correlation between the calculated dissimilarities and the plotted values (after all, that's what it's trying to maximize). Specifically, we can plot the relationship between original dissimilarities and Euclidean distances in the ordination using the stressplot() function. Try typing:

```
stressplot(spe.nmds2)
```



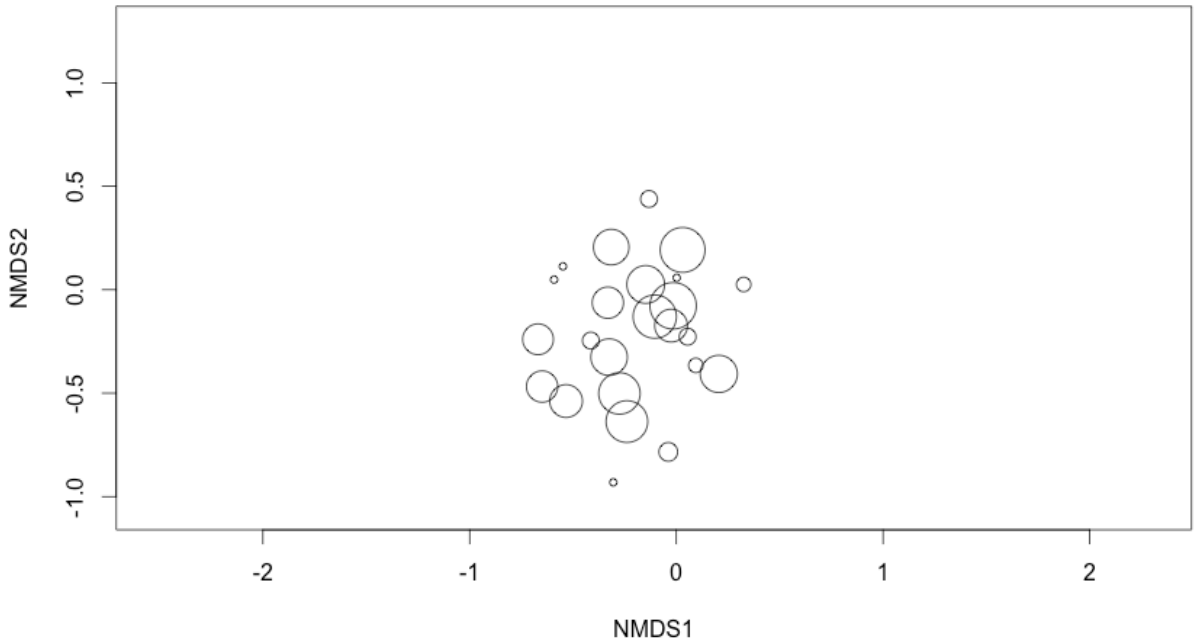
Finally, let's examine the 2-dimensional NMDS configuration for presentation purposes. First we will plot the objects (sites) in ordinate space to visualize the default settings.

```
plot(spe.nmds,type='n')
text(spe.nmds,labels=row.names(speabu))
```



Say we'd like to see how a particular descriptor (in this case, rosyside dace abundance) changes with location. We can make the symbol size proportional to log abundance. Try typing,

```
plot(spe.nmds,type='n')
points(spe.nmds,cex=speabu.log$ROSYDACE)
```



Notice that sites without rosyside dace (i.e., abundance = 0) are not depicted on the plot because their symbol size is zero. The range of options for depicting ordination figures is enormous and beyond the scope of this document. Other ordination graphing functions are available in the vegan package (e.g., **ordiplot**) and include many useful examples that may help you present your own data.

NMDS produces sample scores which are the coordinates of the samples in the k-dimensional ordination space, and these are stored in the result object list in the component named 'points'. To see the sample scores, type:

```
spe.nmds$points
```

Calculate the loadings (i.e., variable weights) on each NMDS axis

To calculate and depict species loadings (i.e., variable weights) on each derived axis from the NMDS we'll use the function envfit() along with the NMDS scores (recall we used the same function for PCoA). The function envfit() simply performs a linear correlation analysis based on standardized data (in other words, a simple linear regression) between each of the original descriptors (i.e., species) and the scores from each NMDS axis. A permutation test is used to assess statistical significance, rather than using the F distribution.

```
vec.sp<-envfit(spe.nmds$points, speabu.log, perm=1000)
```

This should return the values (note yours will be slightly different because it is based on random permutations) listed below for the first 8 species:

```
vec.sp
```

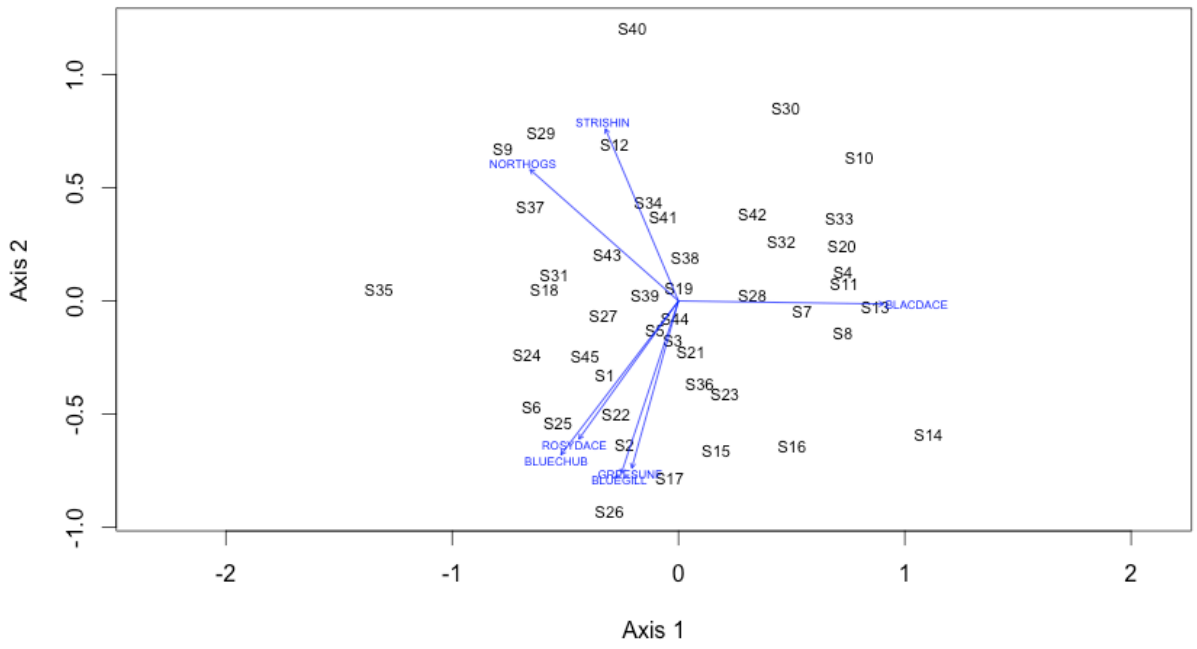
```
***VECTORS

              Dim1      Dim2     r2 Pr(>r)
BANDDART -0.521765  0.853089 0.2247  0.003 **
BANDSCUL -0.278855  0.960333 0.0803  0.175
BLACDACE  0.996128 -0.087917 0.4085 <0.001 ***
BLUECHUB -0.651379 -0.758753 0.3852 <0.001 ***
BLUEGILL -0.380731 -0.924686 0.3195 <0.001 ***
BLUNMINN -0.999536  0.030458 0.0455  0.384
BROOTROU  0.942122 -0.335271 0.0646  0.230
CCHUBSUC -0.123188 -0.992383 0.1760  0.018 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
P values based on 1000 permutations.
```

Based on these results you will notice that banded darter (BANDDART), black dace (BLACDACE), blue chub (BLUECHUB), bluegill (BLUEGILL) and creek chub sucker (CCHUBSUC) and common shiner (COMMSHIN) show statistically significant loadings on the first two principal components. These species could be used to interpret the position of the stream sites (objects) in ordination space.

Now, plot these loadings on the ordination plot.

```
ordiplot(spe.nmds, choices = c(1, 2), type="text", display="sites",
xlab="Axis 1", ylab="Axis 2")
plot(vec.sp, p.max=.01, col="blue")
```

where p.max is the significance level that the species occurrence data must have with either axis in order to be depicted (these p-values were presented in vec.sp).



Combining clustering results and ordination

It is time to start integrating across multivariate approaches. Let's do this by symboling the objects (sites) in the NMDS ordination plot according to clusters idenitied using hierarchical clustering.

First, let's conduct Ward clustering of the species abundance matrix according to Bray-Curtis dissimilarity by typing:

```
speabu.d<-vegdist(speabu.log, "bray")
sitecl.ward<-hclust(speabu.d,method='ward.D2')
sitecl.class<-cutree(sitecl.ward,k=4)
groups<-levels(factor(sitecl.class))
```

Now let's combine with the NMDS results by typing:

```
site.sc <- scores(spe.nmds)
p <- ordiplot(site.sc, type="n", main="NMDS combined with clustering")
for (i in 1:length(groups))
{
  points(site.sc[sitecl.class==i,], pch=(14+i), cex=2, col=i+1)
}
text(site.sc, row.names(speabu), pos=4, cex=0.7)
```

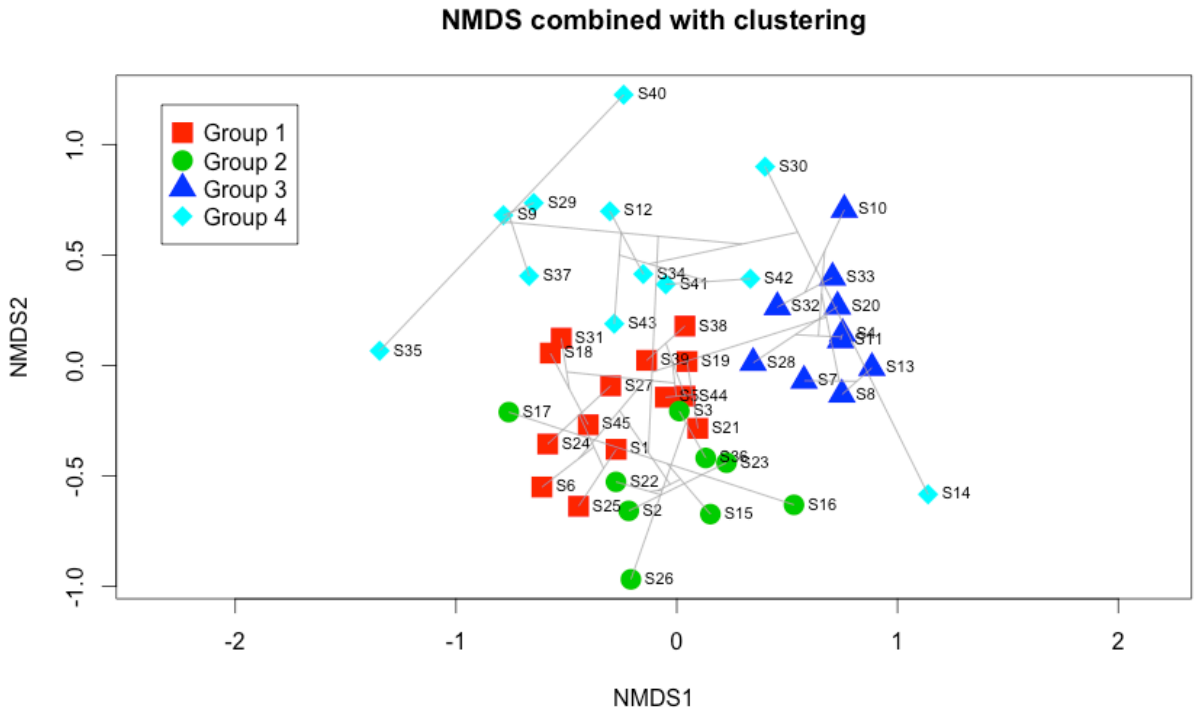
We can add the dendrogram results (optional) by typing:

```
ordicluster(p, sitecl.ward, col="dark grey")
```

Finally, we can a legend interactively by typing:

```
legend(locator(1), paste("Group",c(1:length(groups))),
pch=14+c(1:length(groups)), col=1+c(1:length(groups)), pt.cex=2)
```

Use your mouse to click on the preferred location of the legend.



OPTIONAL READINGS (* recommended)

Kruskal, J.B. 1964a. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29: 1–27.

Kruskal, J.B. 1964b. Nonmetric multidimensional scaling: a numerical method. Psychometrika 29: 115–129.

Kruskal, J. B., and Wish. M. 1977. Multidimensional Scaling. Sage Publications. Beverly Hills. CA.

James, F.C. and McCulloch. 1990. Multivariate analysis in ecology and systematics: Panacea or Pandora's box. Annual Review in Ecology and Systematics 21:129-166.

EXERCISE

Purpose

Upon completion of this chapter, you should be able to do the following: (1) Carry out a NMDS; (2) assess the most powerful solution with respect to the number of dimensions; (3) interpret NMDS axis scores; and (4) consider how NMDS scores may be used in further analyses.

Tasks

- Perform a NMDS using an appropriate resemblance matrix.
    - Is a 2-dimensional solution suitable? What about 3-dimensions?
    - How does the maximum number of random starts affect your ability to find a stable solution?
    - Interpret the bi-plot (i.e., ordination with object scores and variable weights)