# Workshop: NMDS IN R

This workshop is based on the following blog post;

https://jonlefcheck.net/2012/10/24/nmds-tutorial-in-r/

Before we start it is good practice to clear your environment in R or R Studio. Start a new script for this workshop and save it from the start. I also recommend annotating your script - I have added my annotations but it’s good to write it in your own words. Use # to prevent the annotation from being read in R as code. Annotation will help you understand the commands as you use them and is handy as a future reference guide.

Often in ecological research, we are interested not only in comparing univariate descriptors of communities, like diversity, but looking at the composition changes from one community to the next.

One tool to do this is non-metric multidimensional scaling (NMDS). The goal of NMDS is to collapse information from multiple variables into just a few, so that they can be visualised and interpreted. Unlike other ordination techniques that rely on (primarily Euclidean) distances, such as Principal Coordinates Analysis, NMDS uses rank orders. This is an extremely flexible technique that can accommodate a variety of different kinds of data.

## Plot multiple species abundances within three communities

Consider one axis representing the abundance of a single species. We will call this Species 1. Along this axis, we can plot the communities in which Species 1 appears, based on its abundance within each.

#create a plot

plot(0:10,0:10,type="n",axes=F,xlab="Abundance of Species 1",ylab="")

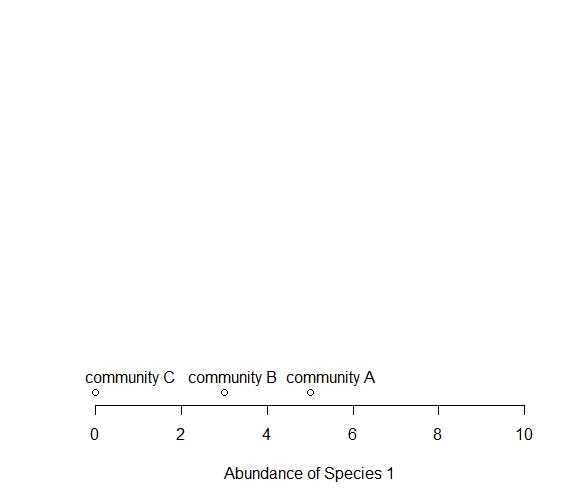
axis(1)

#add the data points for each community

points(5,0); text(5.5,0.5,labels="community A")

points(3,0); text(3.2,0.5,labels="community B")

points(0,0); text(0.8,0.5,labels="community C")



Now consider a *second* axis of abundance, representing another species. We will call this Species 2. We can now plot each of our three communities along two axes (Species 1 and Species 2).

#create a plot with two axes

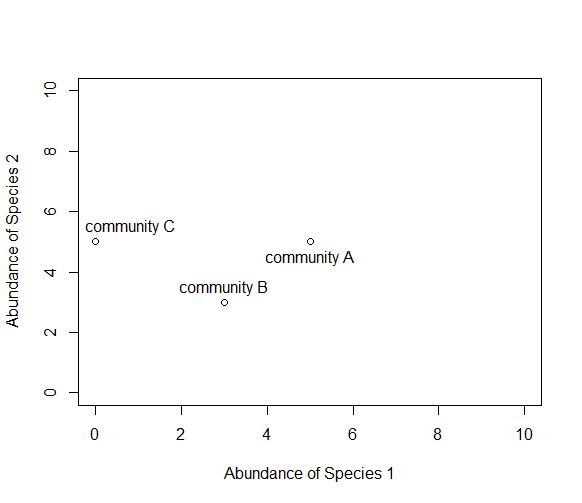
plot(0:10,0:10,type="n",xlab="Abundance of Species 1", ylab="Abundance of Species 2")

#add the data points for each community

points(5,5); text(5,4.5,labels="community A")

points(3,3); text(3,3.5,labels="community B")

points(0,5); text(0.8,5.5,labels="community C")



Now consider a *third*axis of abundance representing yet another species. To plot this you will need to install a new package **scatterplot3d.**

Use the menu Tools>Install packages to find the package **scatterplot3d**

#load it from your library

library(scatterplot3d)

Now use the **scatterplot3d** package to create a graph of three species abundances for your three communities

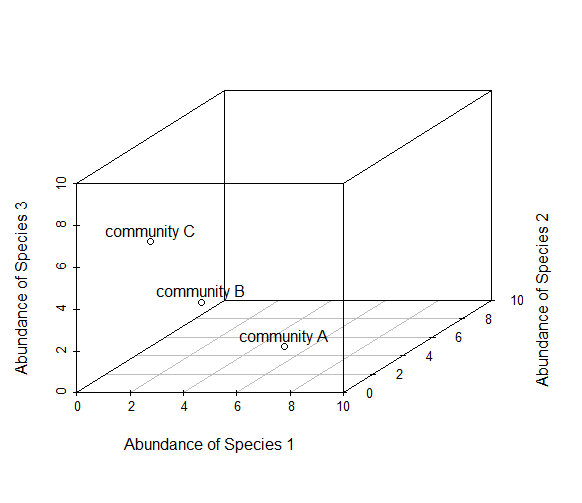
d=scatterplot3d(0:10,0:10,0:10,type="n",xlab="Abundance of Species 1",

ylab="Abundance of Species 2",zlab="Abundance of Species 3"); d

d$points3d(5,5,0); text(d$xyz.convert(5,5,0.5),labels="community A")

d$points3d(3,3,3); text(d$xyz.convert(3,3,3.5),labels="community B")

d$points3d(0,5,5); text(d$xyz.convert(0,5,5.5),labels="community C")



Now imagine how you might want to represent 4 species on a graph, but what about 10 species, or 200 species! Does your head hurt yet? The goal of NMDS is to represent the original position of communities in multidimensional space as accurately as possible using a reduced number of dimensions that can be easily plotted and visualized.

## How does NMDS work?

NMDS does not use the absolute abundances of species in communities, but rather their *rank orders*. The use of ranks omits some of the issues associated with using absolute distance (e.g., sensitivity to transformation), and as a result is much more flexible technique that accepts a variety of types of data. It’s also where the “non-metric” part of the name comes from. Although you don’t need to understand all of the processes involved in NMDA it is useful to have an overview of how it works. Don’t worry too much if the details don’t make sense immediately.

To begin, NMDS requires a distance matrix, or a matrix of dissimilarities. Raw Euclidean distances are not ideal for this purpose: they’re sensitive to total abundances, so may treat sites with a similar number of species as more similar, even though the identities of the species are different. They’re also sensitive to species absences, so may treat sites with the same number of absent species as more similar.

Consequently, ecologists use the Bray-Curtis dissimilarity calculation, which has a number of ideal properties:

* It is invariant to changes in units
* It is unaffected by additions/removals of species that are not present in two communities
* It is unaffected by the addition of a new community
* It can recognize differences in total abundances when relative abundances are the same

The NMDS procedure is iterative and takes place over several steps:

1. Define the original positions of communities in multidimensional space.
2. Specify the number of reduced dimensions (typically 2).
3. Construct an initial configuration of the samples in 2-dimensions.
4. Regress distances in this initial configuration against the observed (measured) distances.
5. Determine the stress, or the disagreement between 2-D configuration and predicted values from the regression. If the 2-D configuration perfectly preserves the original rank orders, then a plot of one against the other must be monotonically increasing. The extent to which the points on the 2-D configuration differ from this monotonically increasing line determines the degree of stress. This relationship is often visualized in what is called a Shepard plot.
6. If stress is high, reposition the points in 2 dimensions in the direction of decreasing stress, and repeat until stress is below some threshold.\*A good rule of thumb: stress < 0.05 provides an excellent representation in reduced dimensions, < 0.1 is great, < 0.2 is good/ok, and stress < 0.3 provides a poor representation.

It is worth noting that the final configuration may differ depending on the initial configuration (which is often random), and the number of iterations, so it is advisable to run the NMDS multiple times and compare the interpretation from the lowest stress solutions. Remember, as is also true in life - high stress is bad, low stress is good!

## NMDS in R

To run the NMDS, we will use the function metaMDS from the **vegan**package. This package is widely used in community ecology and may be familiar from statistical methods sections of papers. The metaMDS function requires only a community-by-species matrix (which we will create randomly).

Use the menu Tools>Install packages to install the **vegan** package, then load it in R

library(vegan)

The set.seed() function in R sets the starting number used to generate a sequence of random numbers. This ensures that you get the same result if you start with that same **seed** each time you run the same process.

#set seed to 2

set.seed(2)

#randomly create a community matrix

community\_matrix=matrix(sample(1:100,300,replace=T),nrow=10,

dimnames=list(paste("community",1:10,sep=""),paste("sp",1:30,sep="")))

#run the NMDS

#K is the number of reduced dimensions, set this as k=2

example\_NMDS=metaMDS(community\_matrix, k=2)

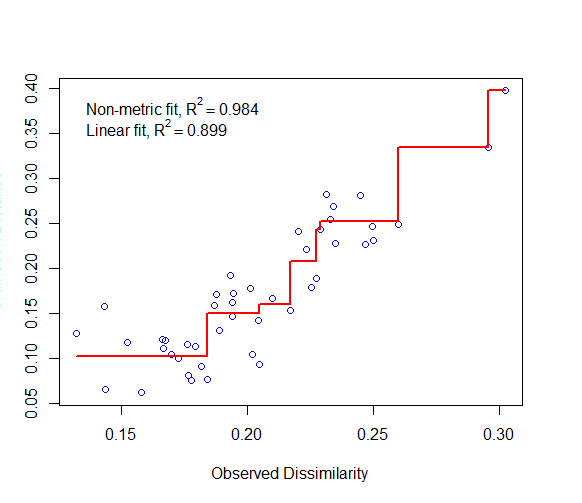
You should see each iteration of the NMDS until a solution is reached (i.e., stress was minimized after some number of reconfigurations of the points in 2 dimensions). It will say ‘solution reached’ in the R console. If a solution is not reached you can increase the number of default iterations or the number of dimensions. To increase the number of default iterations use the argument trymax=. If high stress is a problem, try increasing the number of dimensions to k=3.

example\_NMDS=metaMDS(community\_matrix,k=2,trymax=100)

You’ll see that metaMDS has automatically applied a square root transformation and calculated the Bray-Curtis distances for our community-by-site matrix.

Let’s examine a Shepard plot, which shows scatter around the regression between the interpoint distances in the final configuration (i.e., the distances between each pair of communities) against their original dissimilarities.

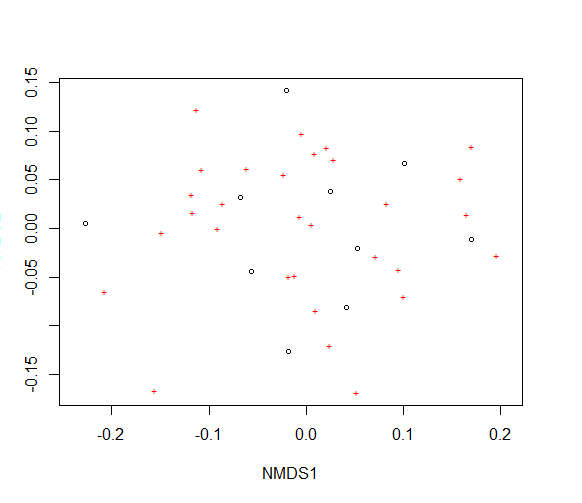
stressplot(example\_NMDS)



Large scatter around the line suggests that original dissimilarities are not well preserved in the reduced number of dimensions. This plot looks pretty ok so we’ll continue.

Now we can plot the NMDS. The plot shows us both the communities (“sites”, open circles) and species (red crosses), but we don’t know which circle corresponds to which site, and which species corresponds to which cross.

plot(example\_NMDS)



We can use the functions ordiplot and orditorp to add text to the plot in place of points to make some sense of this rather non-intuitive mess.

#creates the blank plotting space

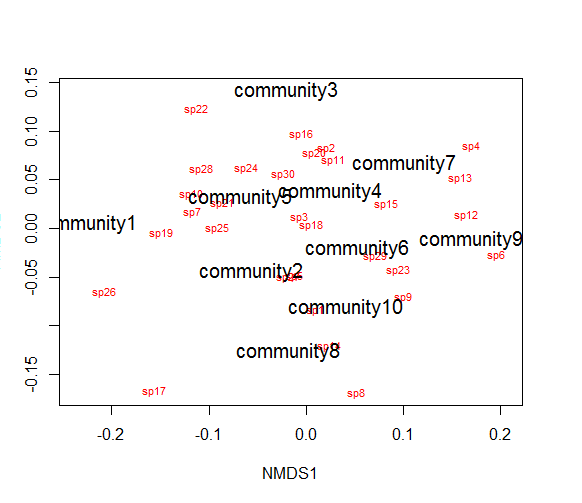
ordiplot(example\_NMDS,type="n")

#adds the species point labels in red

orditorp(example\_NMDS,display="species",col="red",air=0.01)

#adds the community point labels in black

orditorp(example\_NMDS,display="sites",cex=1.25,air=0.01)



This is a basic plot, but it can be modified for better presentation and to add more information. For example, the cex= function defines the text size. It’s currently quite large so have a go at reducing it to make the community labels more readable.

## Using NMDS plots to examine treatment differences in communities

Usually we have some reason to assume that the communities we have sampled might be different in some way. Let’s assume that communities 1-5 had some treatment applied, and communities 6-10 a different treatment. We can draw convex hulls, using the function ordihull, to connect the vertices of the points made by these communities on the plot.

treat=c(rep("Treatment1",5),rep("Treatment2",5))

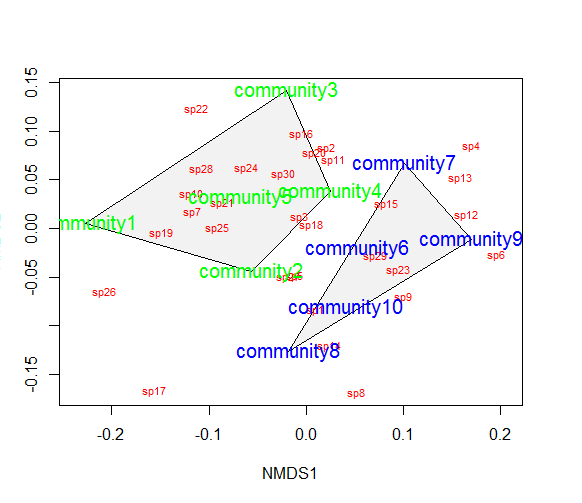
ordiplot(example\_NMDS,type="n")

ordihull(example\_NMDS,groups=treat,draw="polygon",col="grey90",label=F)

orditorp(example\_NMDS,display="species",col="red",air=0.01)

orditorp(example\_NMDS,display="sites",col=c(rep("green",5),rep("blue",5)),

air=0.01,cex=1.25)



There are a number of other functions you can use to create alternative plots. Spider graphs can be plotted using the function orderspider or ellipses using the function ordiellipse. Have a go at creating your own versions of these plots.

Another plot type is a minimum spanning tree (MST), which uses the function ordicluster to overlay a cluster dendrogram onto ordination. This can be useful to see if treatments are effective in controlling community structure. For ordicluster to work it needs the result from a hierarchic clustering such as hclust or agnes. The hclust function in R uses the complete linkage method for hierarchical clustering by default. This clustering method defines the cluster distance between two clusters to be the maximum distance between their individual components. At every stage of the clustering process, the two nearest clusters are merged into a new cluster. The process is repeated until the whole data set is agglomerated into one single cluster.

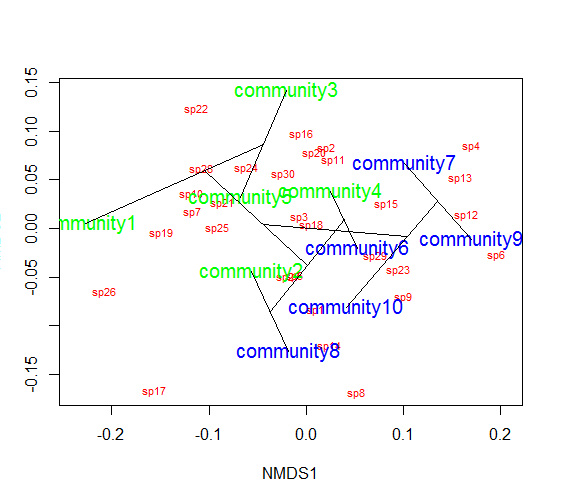
ordiplot(example\_NMDS,type="n")

orditorp(example\_NMDS,display="species",col="red",air=0.01)

orditorp(example\_NMDS,display="sites",col=c(rep("green",5),

rep("blue",5)), air=0.01,cex=1.25)

ordicluster(example\_NMDS,hclust(vegdist(community\_matrix,"bray")))



You could also colour in the convex hulls, ellipses, etc. by treatment. Although you may wish to use a less garish colour scheme!

#create a vector of colour values corresponding to the same length as the vector of treatment values

colors=c(rep("red",5),rep("blue",5))

ordiplot(example\_NMDS,type="n")

#plot convex hulls with colours based on treatment

for(i in unique(treat)) {

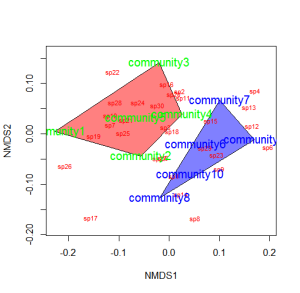
ordihull(example\_NMDS$point[grep(i,treat),],draw="polygon",

groups=treat[treat==i],col=colors[grep(i,treat)],label=F) }

orditorp(example\_NMDS,display="species",col="red",air=0.01)

orditorp(example\_NMDS,display="sites",col=c(rep("green",5),

rep("blue",5)), air=0.01,cex=1.25)

[](https://jslefche.files.wordpress.com/2012/10/convex_hulls_color.png)

If the treatment is continuous, such as an environmental gradient, then it might be useful to plot contour lines rather than convex hulls. We can simply make up some, say, elevation data for our original community matrix and overlay them onto the NMDS plot using ordisurf. You could even do this for other continuous variables, such as temperature.

#define random elevations for previous example

elevation=runif(10,0.5,1.5)

#use the function ordisurf to plot contour lines

ordisurf(example\_NMDS,elevation,main="",col="forestgreen")

#display species on plot

orditorp(example\_NMDS,display="species",col="grey30",air=0.1, cex=1)

