Statistical and machine learning

Ordinations

Jannes Muenchow

DAAD summer school

Contents of the tutorial

1. PCA - the mother of all ordinations

Contents of the tutorial

1. PCA - the mother of all ordinations
2. A short glance at DCA and NMDS

Contents of the tutorial

1. PCA - the mother of all ordinations
2. A short glance at DCA and NMDS
3. Spatially predicting ordination scores

Principal Component Analysis (PCA)

Ordination principles

> Ordination is a procedure for adapting a multidimensional swarm of data points in such a way that when it is projected onto a two-space (such as a sheet of paper) any intrinsic pattern the swarm may possess becomes apparent“

Pielou (1984: 133)

Ordination principles

> Ordination is a procedure for adapting a multidimensional swarm of data points in such a way that when it is projected onto a two-space (such as a sheet of paper) any intrinsic pattern the swarm may possess becomes apparent“

Pielou (1984: 133)

```
library("vegan")
data(varechem)
varechem[1:5, 1:11]
```

```
##        N    P     K    Ca    Mg    S    Al   Fe    Mn   Zn  Mo
## 18 19.8 42.1 139.9 519.4  90.0 32.3  39.0 40.9  58.1  4.5 0.3
## 15 13.4 39.1 167.3 356.7  70.7 35.2  88.1 39.0  52.4  5.4 0.3
## 24 20.2 67.7 207.1 973.3 209.1 58.1 138.0 35.4  32.1 16.8 0.8
## 27 20.6 60.8 233.7 834.0 127.2 40.7  15.4  4.4 132.0 10.7 0.2
## 23 23.8 54.5 180.6 777.0 125.8 39.5  24.2  3.0  50.1  6.6 0.3
```

Aims

- Reduce the number of columns (e.g., species, edaphic variables) to two to three visually interpretable columns.

Aims

- Reduce the number of columns (e.g., species, edaphic variables) to two to three visually interpretable columns.
- Just keep the main signal (gradient) and get rid off the noise.

Aims

- Reduce the number of columns (e.g., species, edaphic variables) to two to three visually interpretable columns.
- Just keep the main signal (gradient) and get rid off the noise.
- Principal components should be ordered by most explained variance, i.e., the first axis should explain most of the observed variance.

Aims

- Reduce the number of columns (e.g., species, edaphic variables) to two to three visually interpretable columns.
- Just keep the main signal (gradient) and get rid off the noise.
- Principal components should be ordered by most explained variance, i.e., the first axis should explain most of the observed variance.
- Axes should not be correlated (orthogonality).

Keep in mind

- unsupervised statistical learning method
- linear relationship between variables

PCA by example


```
library("vegan")
data(varechem)
dim(varechem)
```

```
## [1] 24 14
```

```
names(varechem)
```

```
##  [1] "N"        "P"        "K"        "Ca"       "Mg"       "S"
##  [7] "Al"       "Fe"       "Mn"       "Zn"       "Mo"       "Baresoil"
## [13] "Humdepth" "pH"
```

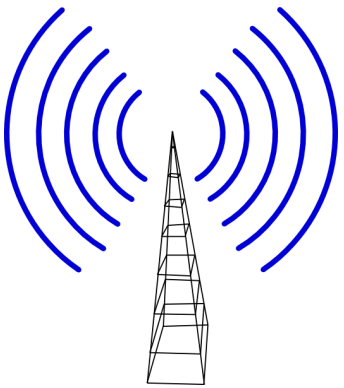
```
varechem[1:3, 1:8]
```

```
##        N    P     K    Ca     Mg    S    Al   Fe
## 18 19.8 42.1 139.9 519.4  90.0 32.3  39.0 40.9
## 15 13.4 39.1 167.3 356.7  70.7 35.2  88.1 39.0
## 24 20.2 67.7 207.1 973.3 209.1 58.1 138.0 35.4
```

In pictures

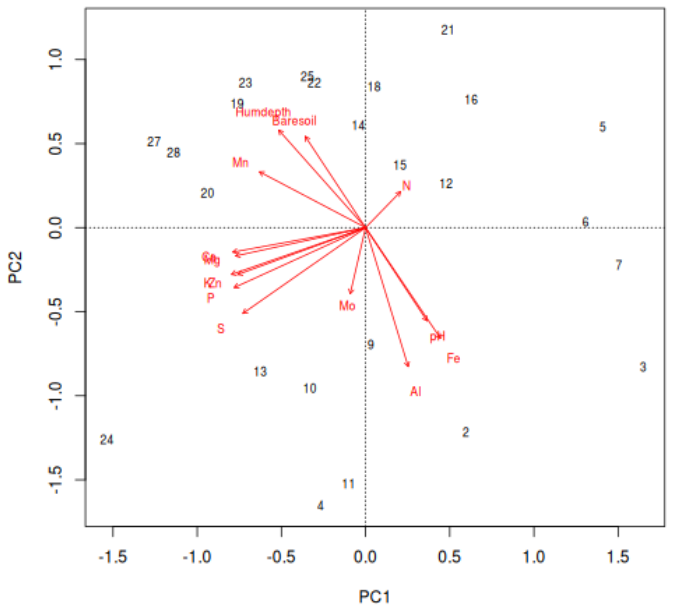
In pictures

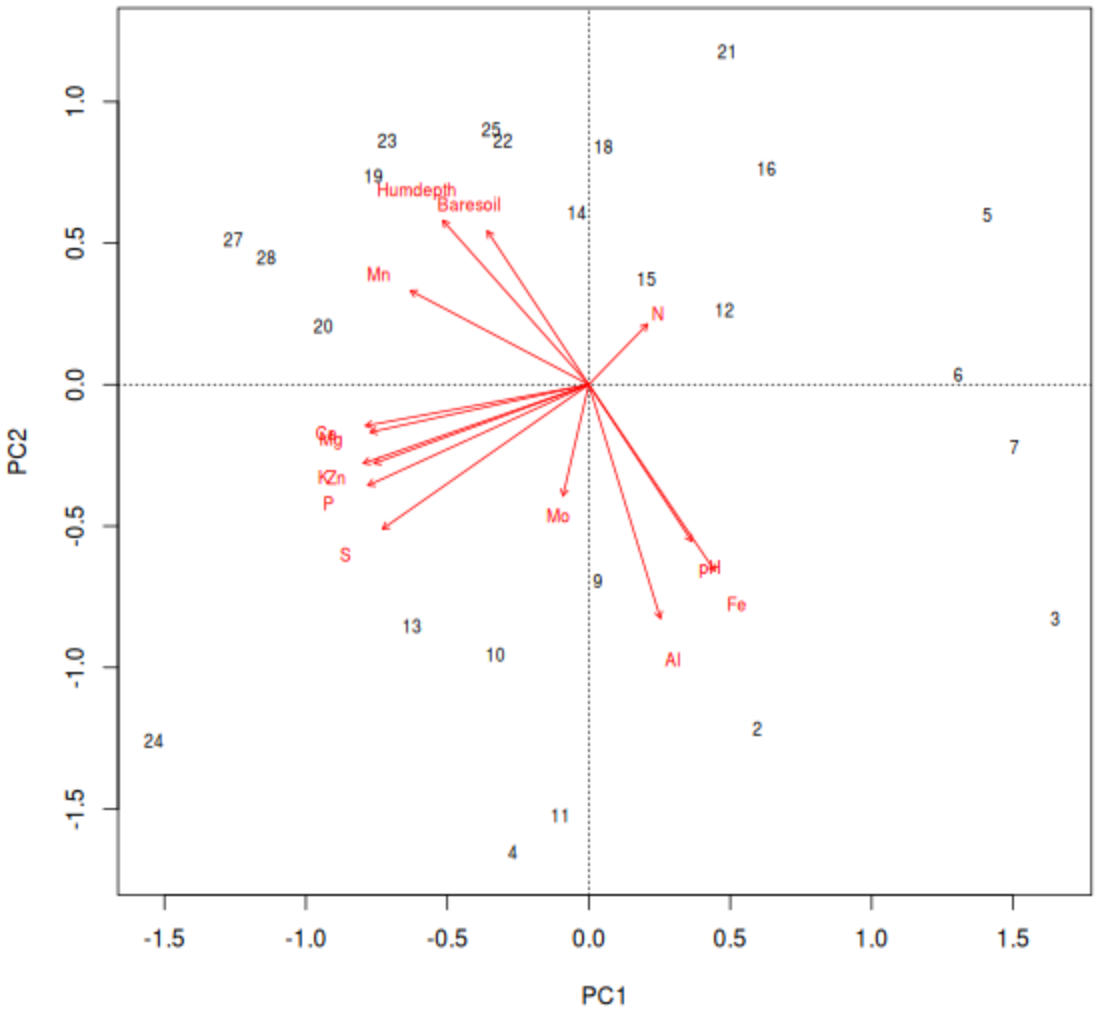
In pictures



PCA by example

```
pca_1 = rda(varechem,
            scale = TRUE)
biplot(pca_1)
```




PC2
PC1
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
21
25
23
22
18
16
19
Humdepth
Baresoil
14
5
27
28
Mn
15
N
12
20
6
Ca
Mg
7
KZn
P
Mo
S
pH
9
Fe
3
13
10
Al
2
24
11
4

Explained variance


```
cumsum(eigenvals(pca_1) / sum(eigenvals(pca_1)))
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 0.3708251 0.5988853 0.7192787 0.7956354 0.8539207 0.9043357 0.9355076
##       PC8       PC9      PC10      PC11      PC12      PC13      PC14
## 0.9618534 0.9740489 0.9847295 0.9908195 0.9958091 0.9983132 1.0000000
```

```
# compare with:
# summary(pca_1)
```

Explained variance


```
cumsum(eigenvals(pca_1) / sum(eigenvals(pca_1)))
```

```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 0.3708251 0.5988853 0.7192787 0.7956354 0.8539207 0.9043357 0.9355076
##       PC8       PC9      PC10      PC11      PC12      PC13      PC14
## 0.9618534 0.9740489 0.9847295 0.9908195 0.9958091 0.9983132 1.0000000
```

```
# compare with:
# summary(pca_1)
```

A very brief word on eigenvalues and eigenvectors

An eigenvalue represents the percentage of the total variance explained.

Eigenvectors (also known as loadings or rotations) tell you how much a principal component is influenced by a (environmental) variable, e.g., in our biplot Al has a big influence on the second PC axis.

Detrended Correspondence Analysis (DCA)

Ecological dataset

Attach the data and have a look:

```
library("reshape2")
library("dplyr")
library("vegan")
library("BiodiversityR")
data("ifri", package = "BiodiversityR")
# find out more about ifri dataset
# ?ifri
head(ifri)  # long table format
```

```
##   forest  plotID  species count  basal
## 1    LOT LOTP001 Lirituli     4 5140.0
## 2    LOT LOTP001 Prunsero     1 1385.4
## 3    LOT LOTP001 Sassalbi     1 1012.2
## 4    LOT LOTP001 Platocci     1  730.6
## 5    LOT LOTP001 Acerrubr     1  317.3
## 6    LOT LOTP001 Cornflor     1  201.1
```

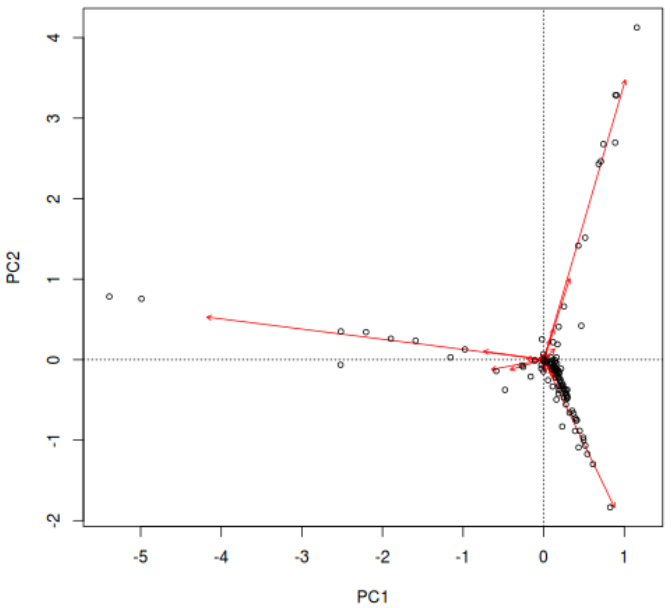
But ordinations (usually) require the wide table format...

```
mat = dcast(ifri, plotID ~ species, value.var = "count", fill = 0)
rownames(mat) = mat$plotID
mat = select(mat, -plotID)
# let's have a look at the first five rows and columns
mat[1:5, 1:5]
```

```
##         Acernegu Acerrubr Acersacc Acersp. Aescglab
## LOTP001        1        1        0       0        0
## LOTP002        0        0        0       0        0
## LOTP003        0        0        0       0        0
## LOTP004        0        1        0       0        0
## LOTP005        0        0        0       0       15
```

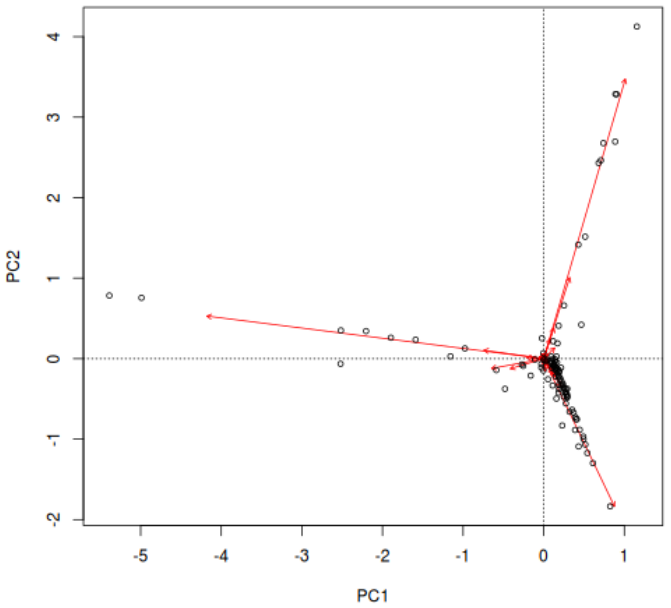
First use a PCA

```
pca_2 = rda(mat)
biplot(pca_2)
```



First use a PCA

```
pca_2 = rda(mat)
biplot(pca_2)
```



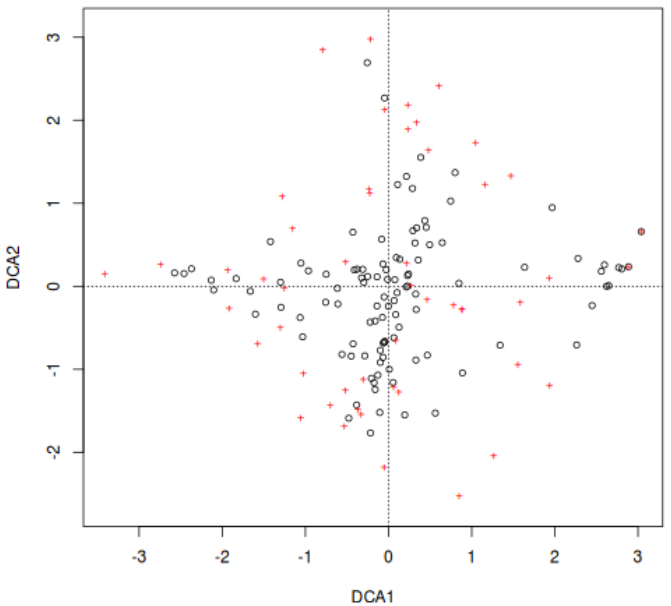

```
cumsum(
  eigenvals(pca_2) /
    sum(eigenvals(pca_2))
  )[1:3]
```

```
##       PC1       PC2       PC3
## 0.2245335 0.4116066 0.5277840
```

First three axes explain only **53%** (not really convincing).

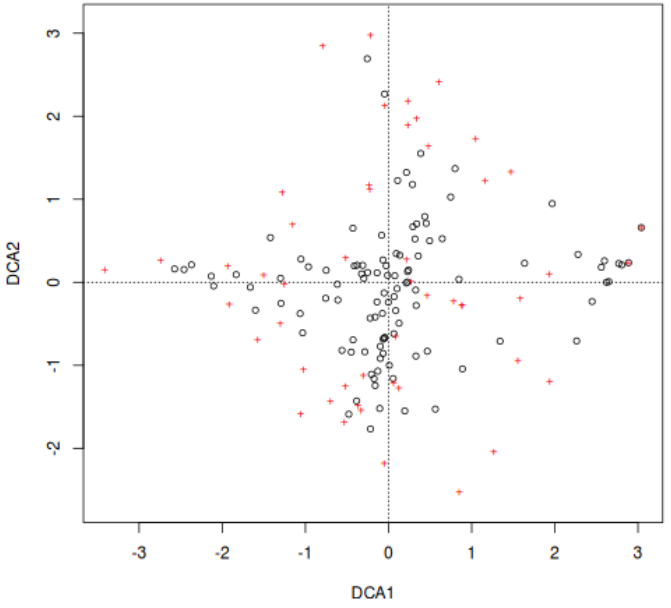
Let's try a DCA

```
dca_1 = decorana(mat,
                 iweigh = 1)
plot(dca_1)
```



Let's try a DCA

```
dca_1 = decorana(mat,
                 iweigh = 1)
plot(dca_1)
```




```
# cumulative proportion
cumsum(
  dca_1$evals /
    sum(dca_1$evals)
  )[1:3]
```

```
##       DCA1       DCA2       DCA3
## 0.3628023 0.6023689 0.8329005
```

First three axes explain **83%**, way better!

A brief word on theory

Reciprocal averaging

A brief word on theory

Reciprocal averaging

- randomly select (unequal) sample scores

A brief word on theory

Reciprocal averaging

- randomly select (unequal) sample scores
- compute new species scores as the weighted average of the sample scores

A brief word on theory

Reciprocal averaging

- randomly select (unequal) sample scores
- compute new species scores as the weighted average of the sample scores
- compute new sample scores as the weighted mean of the previously computed species scores

A brief word on theory

Reciprocal averaging

- randomly select (unequal) sample scores
- compute new species scores as the weighted average of the sample scores
- compute new sample scores as the weighted mean of the previously computed species scores
- rearrange the complete matrix by the species scores

A brief word on theory

Reciprocal averaging

- randomly select (unequal) sample scores
- compute new species scores as the weighted average of the sample scores
- compute new sample scores as the weighted mean of the previously computed species scores
- rearrange the complete matrix by the species scores
- Do this (reciprocal averaging) over and over again until values have stabilized

A brief word on theory

Reciprocal averaging

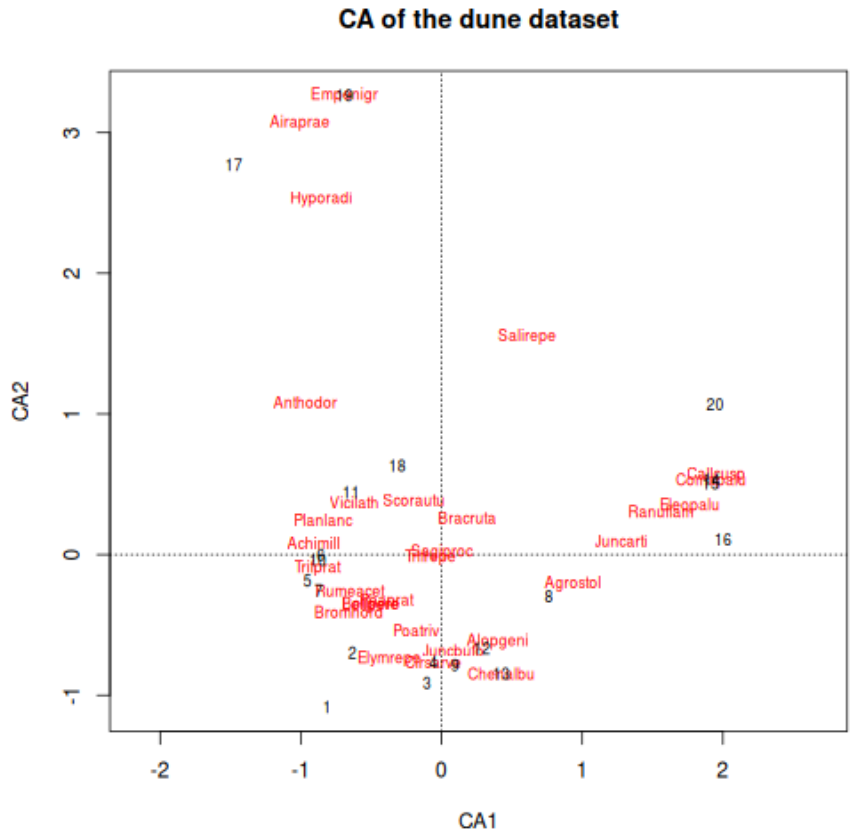
- randomly select (unequal) sample scores
- compute new species scores as the weighted average of the sample scores
- compute new sample scores as the weighted mean of the previously computed species scores
- rearrange the complete matrix by the species scores
- Do this (reciprocal averaging) over and over again until values have stabilized

In fact, this sounds harder than it is. If you are really interested in how the procedure works, let's have a look at Karsten Wesche's slides (# 56-63).

This was just the computation of the first axis. You have to the same for the second, third, etc. axis by making sure that the sample scores are uncorrelated (orthogonal) to those of the previous axis.

Arch effect

However, there is a problem with the CA approch, which is called the Arch effect.



DCA

To make the arch effect dissappear, DCA was invented. In fact, it is just a brute force approach to remove the arch effect through detrending by segments.

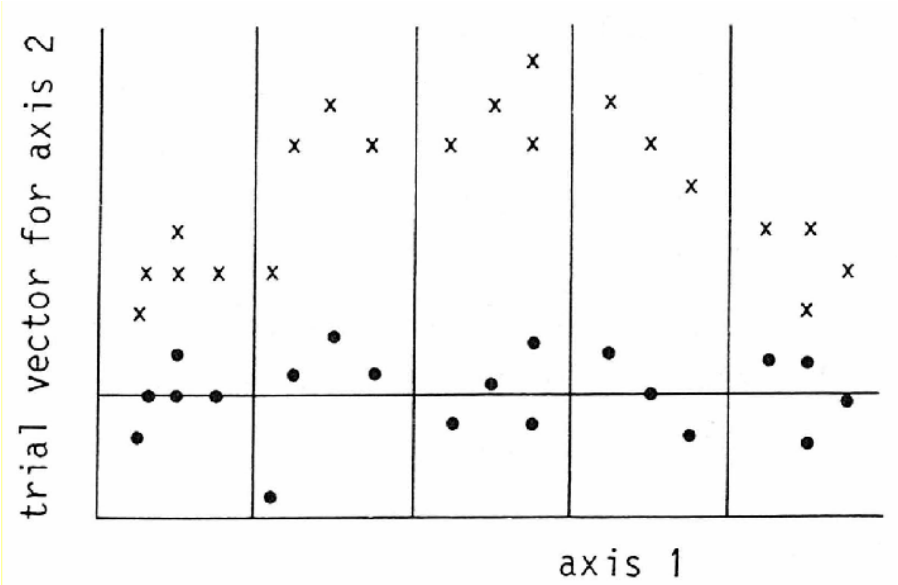
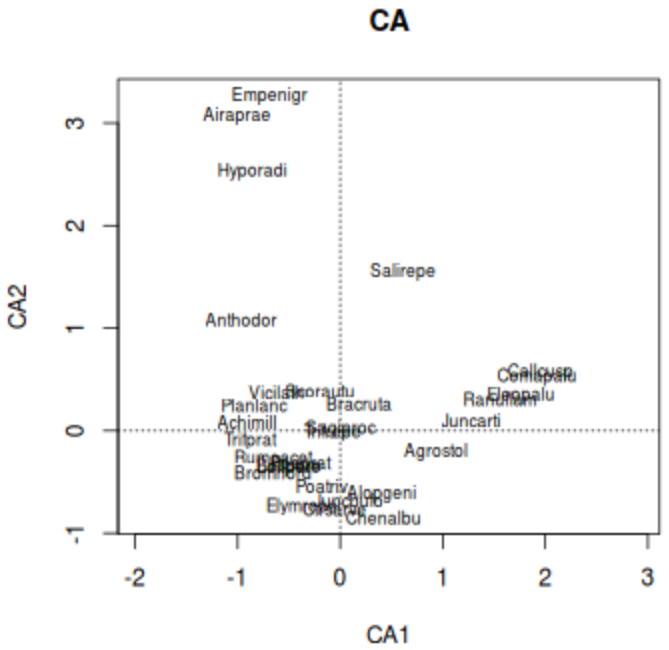
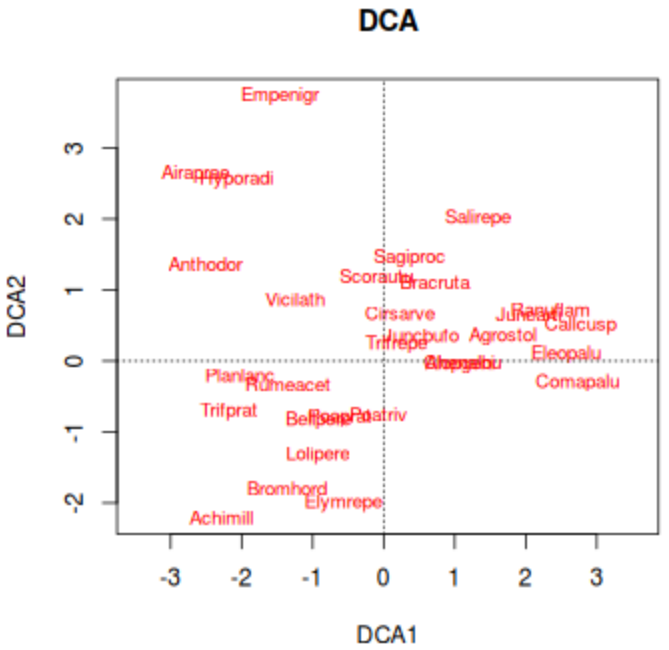


Figure taken from Jongman, Braak, and Van Tongeren (1995).

Comparing CA and DCA





Non-Metric Multidimensional Scaling (NMDS)

Principle

- calculation of ecological dissimilarity (e.g Bray-Curtis)

Principle

- calculation of ecological dissimilarity (e.g Bray-Curtis)
- Choose no. of axes (dimensions) to be tested

Principle

- calculation of ecological dissimilarity (e.g Bray-Curtis)
- Choose no. of axes (dimensions) to be tested
- Distribution (usually random) of sites in ordination space

Principle

- calculation of ecological dissimilarity (e.g Bray-Curtis)
- Choose no. of axes (dimensions) to be tested
- Distribution (usually random) of sites in ordination space
- calculation of distance in ordination space (Euclidean distance)

Principle

- calculation of ecological dissimilarity (e.g Bray-Curtis)
- Choose no. of axes (dimensions) to be tested
- Distribution (usually random) of sites in ordination space
- calculation of distance in ordination space (Euclidean distance)
- Move samples until ecological distance and ordinations distance are strongly correlated

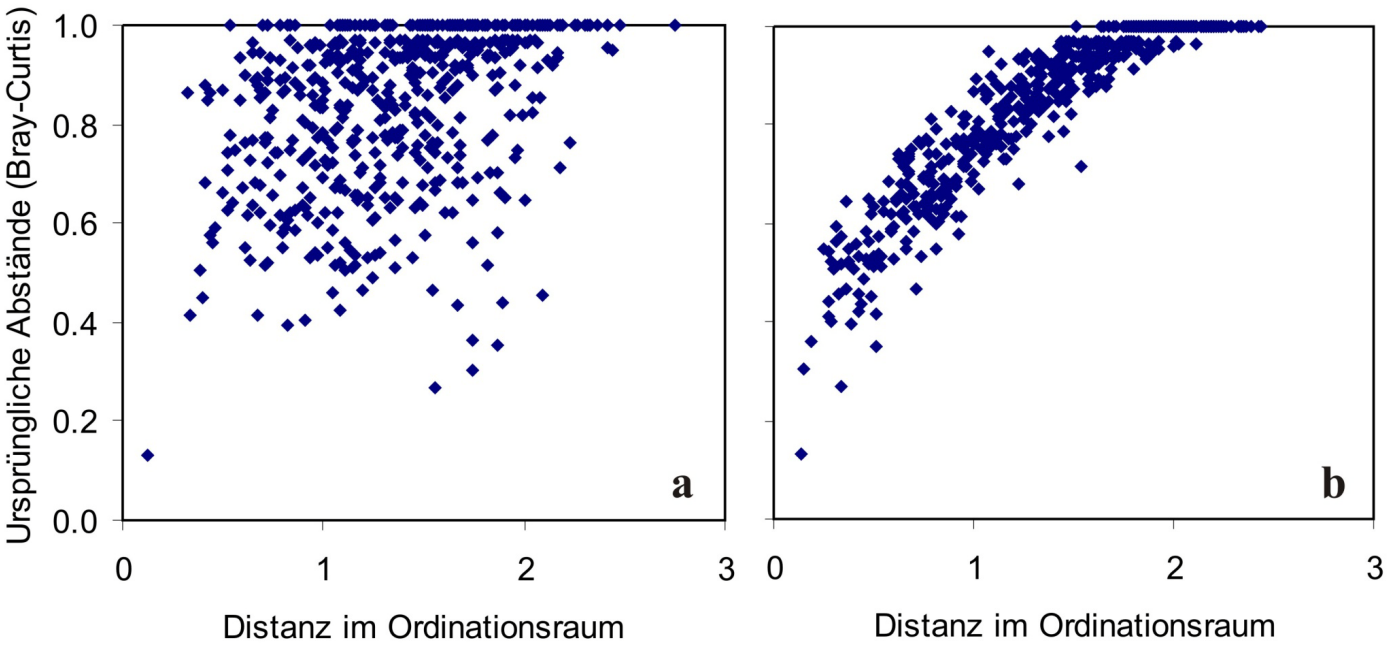


Figure taken from wescheetal_mulva_intro.pdf.

Stress instead of Eigenvalues

- The stress value indicates how good the NMDS represent the multidimensional dataset in just a few dimensions (axes).

Stress instead of Eigenvalues

- The stress value indicates how good the NMDS represent the multidimensional dataset in just a few dimensions (axes).
- The lower the stress value the better. As a rule of thumb, stress values <10 are considered a great representation of the multidimensional data. A stress value of 15 indicates a satisfactory result. And values >25 are considered more or less random noise.

The NMDS start configuration (1st iteration) of the Dune dataset has a stress value of 45 (see figure below).

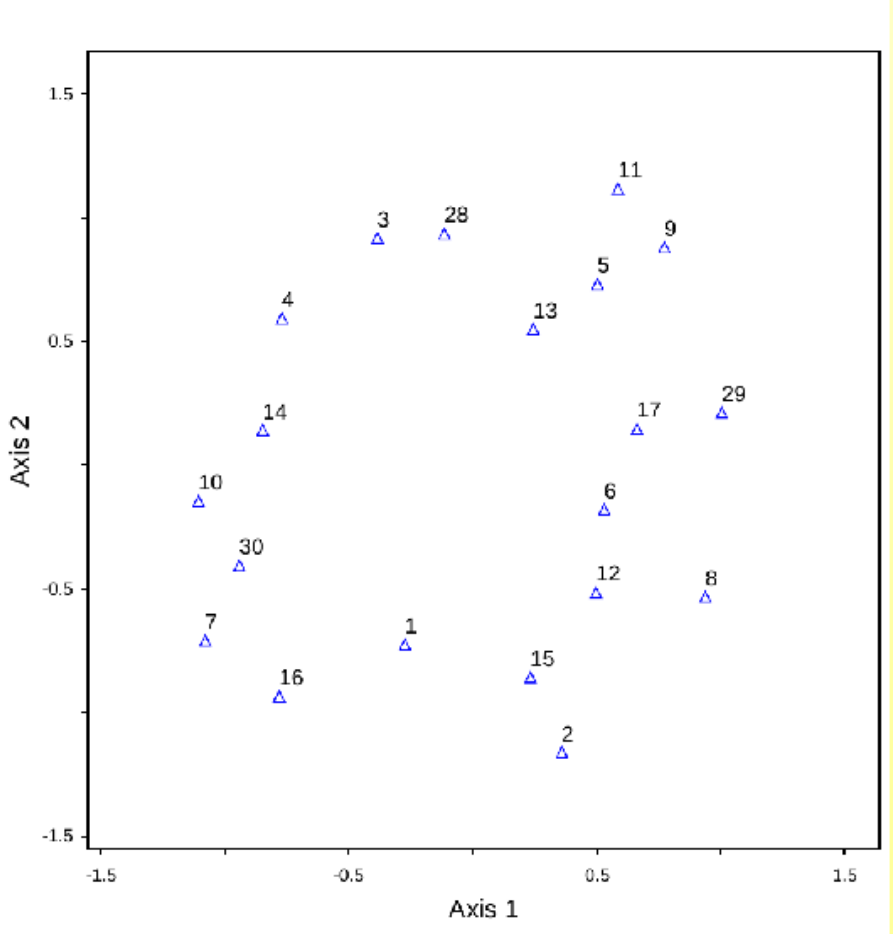


Figure taken from wescheetal_mulva_intro.pdf.

NMDS in practice

```
library("vegan")
# let's use the ifri dataset again (mat is its wide format)
nmds = metaMDS(comm = mat, k = 2)
```

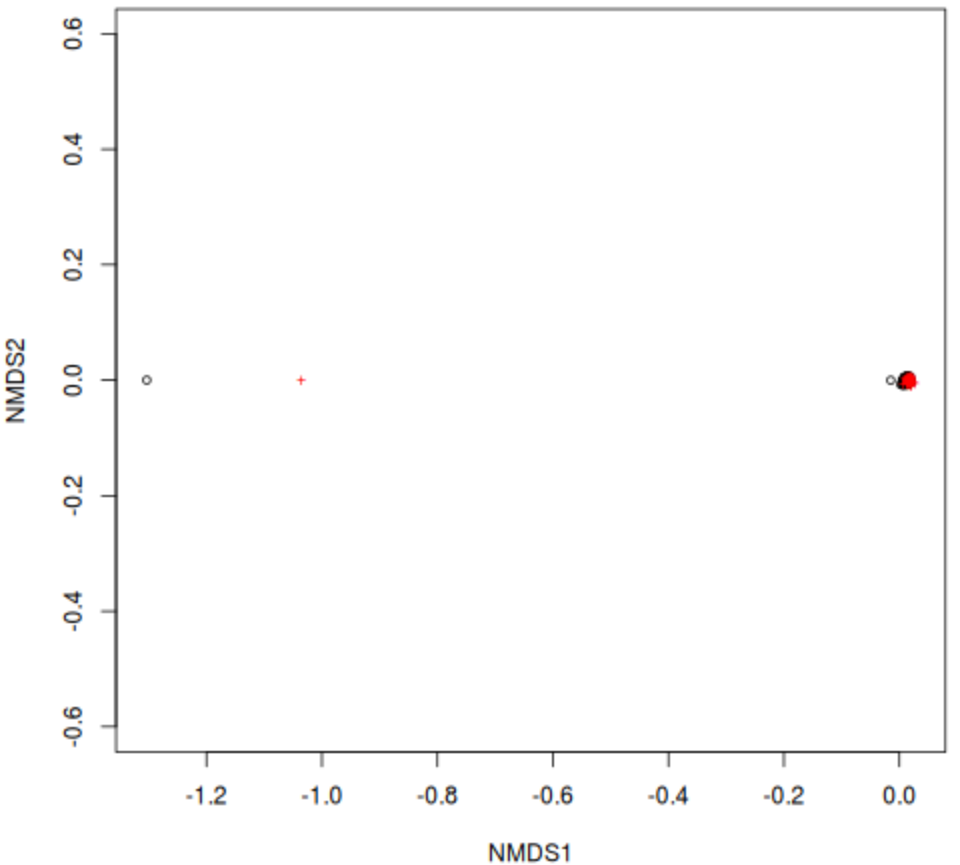
```
## Wisconsin double standardization
## Run 0 stress 0.0959359
## Run 1 stress 0.09594265
## ... Procrustes: rmse 0.001603814  max resid 0.005760848
## ... Similar to previous best
## Run 2 stress 0.09595472
## ... Procrustes: rmse 0.001757011  max resid 0.01121542
## Run 3 stress 0.09593894
## ... Procrustes: rmse 0.001672436  max resid 0.007733106
## ... Similar to previous best
## Run 4 stress 0.09594718
## ... Procrustes: rmse 0.001804019  max resid 0.006821999
## ... Similar to previous best
## Run 5 stress 0.09592982
## ... New best solution
## ... Procrustes: rmse 0.001874431  max resid 0.007823088
## ... Similar to previous best
## Run 6 stress 0.09594619
```

Stress value


```
nmds$stress
```

```
## [1] 0.09592982
```

Plotting the result



Modeling ordination scores

Why doing it?

- Ordination scores represent the floristic gradient of our species-plot matrix.
- We can determine which environmental variables might explain how much of the main gradients (axes).
- The predictive mapping of ordination scores visualizes the floristic gradient in space.
- We need a community matrix, site coordinates and environmental variables.

And this is exactly what you will do in the ecological modeling task.

Your turn

1. Load the `varespec` dataset from the **vegan** package (`data("varespec", package = "vegan")`).
2. Make yourself familiar with the dataset (`?varespec`).
3. Run a PCA, a DCA and a NMDS on the dataset, and justify which ordination technique does represent the data best in ordination space.
4. Use the `varechem` dataset (`data("varechem", package = "vegan")`) to model the scores of the first axis of the best ordination approach.

References

Fielding, Alan H. (2006). *Cluster and classification techniques for the biosciences*. Cambridge University Press.

James, Gareth, Daniela Witten, Trevor Hastie, et al. (2013). *An Introduction to Statistical Learning: With Applications in R*. Springer Science & Business Media. ISBN: 978-1-4614-7138-7.

Jongman, R. H, C. J. F. ter Braak, and O. F. R. Van Tongeren, ed. (1995). *Data analysis in community and landscape ecology*. New ed, with corr. Cambridge ; New York: Cambridge University Press. ISBN: 978-0-521-47574-7.

Pielou, E. C. (1984). *The Interpretation of Ecological Data: A Primer on Classification and Ordination*. En. John Wiley & Sons. ISBN: 978-0-471-88950-2.

Zuur, Alain F, Elena N. Ieno, and Graham M. Smith (2007). *Analysing Ecological Data*. Eng. Statistics for Biology and Health. OCLC: 255677794. New York, NY: Springer. ISBN: 978-0-387-45972-1 978-0-387-45967-7.