Basic Ordination

Gavin L. Simpson

U Adelaide 2017 • Feb 13–17 2017

Introduction

Introduction

- Ordination comes from the German word *ordnung*, meaning to put things in order
- This is exactly what we we do in ordination — we arrange our samples along gradients by fitting lines and planes through the data that describe the main patterns in those data
- Linear and unimodal methods
- Principle Components Analysis (PCA) is a linear method — most useful for environmental data or sometimes with species data and short gradients
- Correspondence Analysis (CA) is a unimodal method — most useful for species data, especially where non-linear responses are observed

Principal Components Analysis

Ordination

- Regression gives us a basis from which to work
- Instead of doing many regressions, do one with all the species data once
- Only now we don't have any explanatory variables, we wish to uncover these underlying gradients
- PCA fits a line through our cloud of data in such a way that it maximises the variance in the data captured by that line (i.e.~minimises the distance between the fitted line and the observations)
- Then we fit a second line to form a plane, and so on, until we have one PCA line or axis for each of our species
- Each of these subsequent axes is uncorrelated with previous axes — they are orthogonal — so the variance each axis explains is uncorrelated

Principal Components Analysis

Figure 1: © HappyWaldo CC BY-SA

Principal Components Analysis

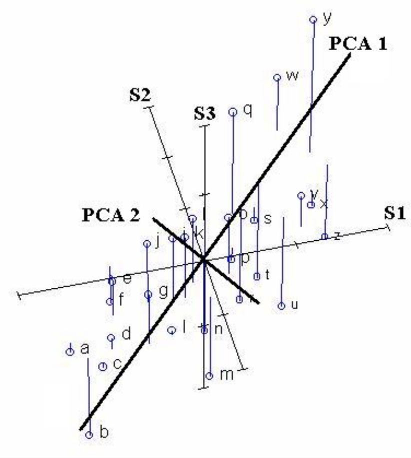
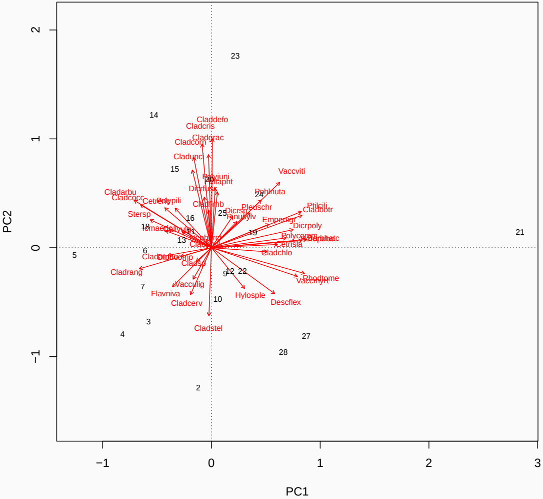


Figure 2:

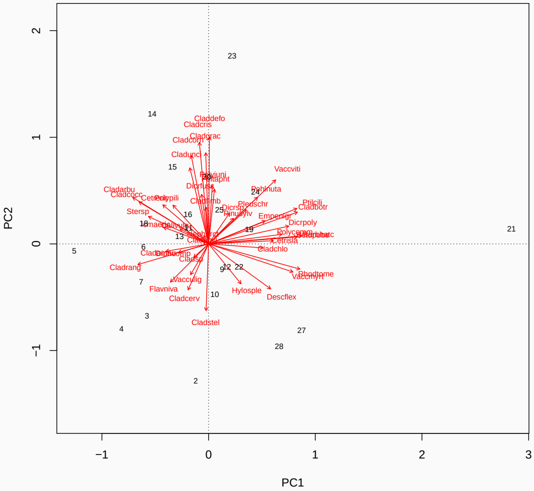
Vegetation in lichen pastures — PCA

Data are cover values of 44 understorey species recorded at 24 locations in lichen pastures within dry *Pinus sylvestris* forests



Vegetation in lichen pastures — PCA biplots

- Have two sets of scores
    - Species scores
    - Site scores
- Sample (species) points plotted close together have similar species compositions (occur together)
- In PCA, species scores often drawn as arrows — point in direction of increasing abundance
- Species arrows with small angles to an axis are highly correlated with that axis



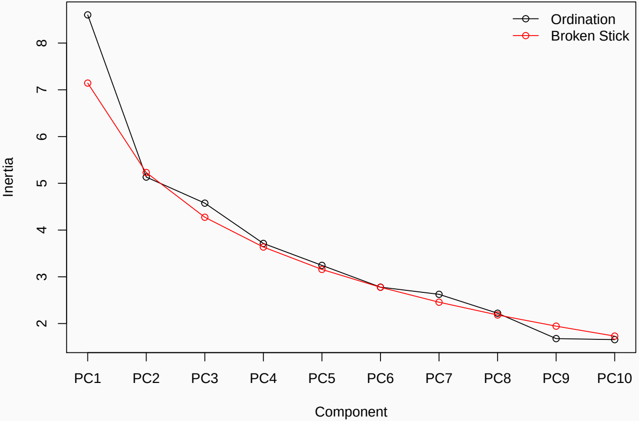
Eigenvalues

Eigenvalues λ are the amount of variance (inertia) explained by each axis

```
head(eigenvals(pca), 6L)
```

```
     PC1      PC2      PC3      PC4      PC5      PC6 
8.602826 5.133623 4.575623 3.713926 3.244925 2.779195 
```

```
screeplot(pca, bstick = TRUE, type = "l", main = NULL)
```



Correspondence Analysis

Correspondence Analysis

Correspondence analysis (CA) is very similar to PCA — in fct it is a just a weighted form of PCA

The row and column sums are used as weights and this has the effect of turning the analysis into one of relative composition

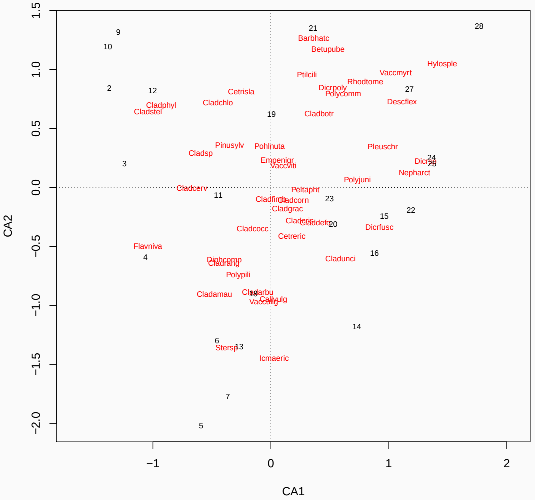
The weighting is a trick to get linear-based software to fit non-linear responses

These nonlinear response are assumed to unimodal Gaussian curves, all with equal height and tolerance widths, and equally spaced optima

So, not very realistic, but it is suprisingl robust at times to violation of this assumption

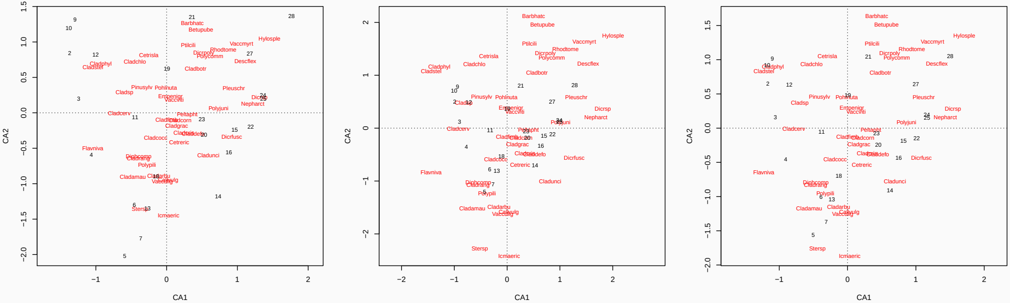
Vegetation in lichen pastures — CA biplots

- Have two sets of scores
    1. Species scores
    2. Site scores
- Sample (species) points plotted close together have similar species compositions (occur together)
- In CA, species scores drawn as points — this is the fitted optima along the gradients
- Abundance of species declines in concentric circles away from the optima



Vegetation in lichen pastures — CA biplots

- Species scores plotted as weighted averages of site scores, or
- Site scores plotted as weighted averages of species scores, or
- A symmetric plot



Vegan usage

Vegan basics

- The majority of vegan functions work with a single vector, or more commonly an entire data frame
- This data frame may contain the species abundances
- Where subsidiary data is used/required, these two are supplied as data frames
- For example; the environmental constraints in a CCA
- It is not a problem if you have all your data in a single file/object; just subset it into two data frames after reading it into R

```
spp <- allMyData[, 1:20] ## columns 1-20 contain the species data
env <- allMyData[, 21:26] ## columns 21-26 contain the environmental data
```

Simple vegan usage

First we start with a simple correspondence analysis (CA) to illustrate the basic features

Here I am using one of the in-built data sets on lichen pastures

For various reasons to fit a CA we use the `cca()` function

Store the fitted CA in `ca1` and print it to view the results

```r
ca1 <- cca(varespec)
ca1
```

```
Call: cca(X = varespec)

              Inertia Rank
Total           2.083     
Unconstrained   2.083   23
Inertia is mean squared contingency coefficient

Eigenvalues for unconstrained axes:
   CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8 
0.5249 0.3568 0.2344 0.1955 0.1776 0.1216 0.1155 0.0889 
(Showed only 8 of all 23 unconstrained eigenvalues)
```

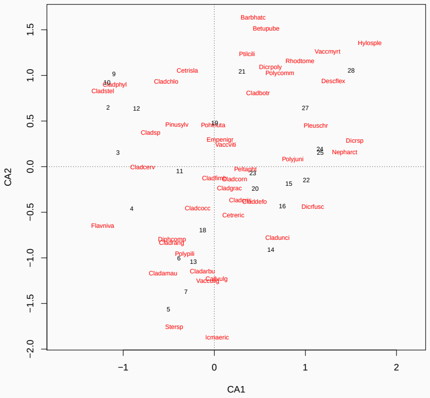
`scores()` & `scaling` in `cca()`, `rda()`

- When we draw the results of many ordinations we display 2 or more sets of data
- Can’t display all of these and maintain relationships between the scores
- Solution; scale one set of scores relative to the other
- Controlled via the `scaling` argument
    - `scaling = 1` — Focus on species, scale site scores by λ_i
    - `scaling = 2` — Focus on sites, scale species scores by λ_i
    - `scaling = 3` — Symmetric scaling, scale both scores by $\sqrt{\lambda_i}$
    - `scaling = -1` — As above, but
    - `scaling = -2` — For `cca()` multiply results by $\sqrt{(1/(1-\lambda_i))}$
    - `scaling = -3` — this is Hill’s scaling
    - `scaling < 0` — For `rda()` divide species scores by species’ σ
    - `scaling = 0` — raw scores

Basic ordination plots

- Basic plotting can be done using the `plot()` method
- `choices = 1:2` — select which axes to plot
- `scaling = 3` — scaling to use
- `display = c("sites","species")` — which scores (default is both)
- `type = "text"` — display scores using labels or points (`"points"`)
- Other graphics arguments can be supplied but the apply for all scores

```r
plot(ca1, scaling = "symmetric")
```



Non-Metric Multidimensional Scaling

Non-Metric Multidimensional Scaling

- Aim is to find a low-dimensional mapping of dissimilarities
- Similar idea to PCoA, but does not use the actual dissimilarities
- NMDS attempts to find a low-dimensional mapping that preserves as best as possible the rank order of the original dissimilarities (d_{ij})
- Solution with minimal `stress` is sought; a measure of how well the NMDS mapping fits the d_{ij}
- Stress is sum of squared residuals of monotonic regression between distances in NMDS space (d^*_{ij}) & d_{ij}
- Non-linear regression can cope with non-linear responses in species data
- Iterative solution; convergence is not guaranteed
- Must solve separately different dimensionality solutions

Non-Metric Multidimensional Scaling

- Use an appropriate dissimilarity metric that gives good gradient separation `rankindex()`
    - Bray-Curtis
    - Jaccard
    - Kulczynski
- Wisconsin transformation useful; Standardize species to equal maxima, then sites to equal totals `wisconsin()`
- Iterative solution; use many random starts and look at the fits with lowest stress
- Only conclude solution reached if lowest stress solutions are similar (Procrsutes rotation)
- Fit NMDS for 1, 2, 3, ...dimensions; stop after a sudden drop in stress observed in a screeplot
- NMDS solutions can be rotated at will; common to rotate to principal components
- Also scale axes in half-change units; samples separated by a distance of 1 correspond, on average, to a 50% turnover in composition

NMDS in vegan

Vegan implements all these ideas via the `metaMDS()` wrapper

```r
data(dune)
set.seed(10)
(sol <- metaMDS(dune, trace = FALSE))
```

```

Call:
metaMDS(comm = dune, trace = FALSE)

global Multidimensional Scaling using monoMDS

Data:     dune
Distance: bray

Dimensions: 2
Stress:     0.1183186
Stress type 1, weak ties
Two convergent solutions found after 20 tries
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on 'dune'
```

NMDS in vegan

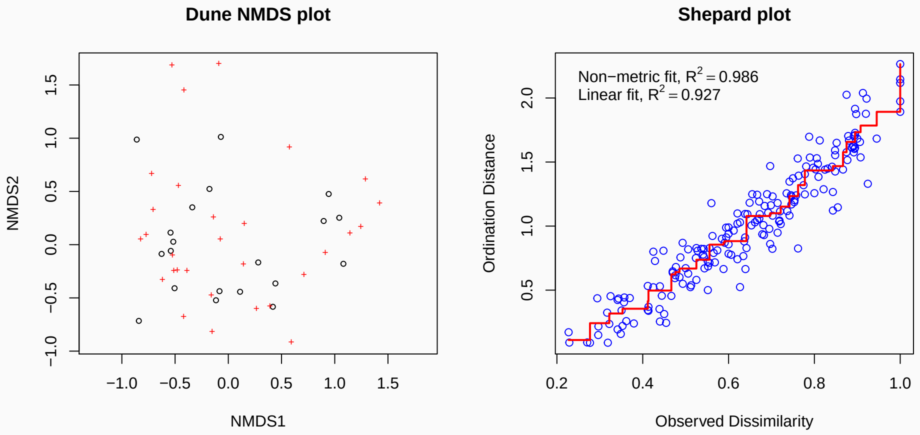
If no convergent solutions, continue iterations from previous best solution

```r
(sol <- metaMDS(dune, previous.best = sol, trace = FALSE))
```

```

Call:
metaMDS(comm = dune, trace = FALSE, previous.best = sol)

global Multidimensional Scaling using monoMDS

Data:     dune
Distance: bray

Dimensions: 2
Stress:     0.1183186
Stress type 1, weak ties
Two convergent solutions found after 40 tries
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on 'dune'
```

NMDS in vegan

```r
layout(matrix(1:2, ncol = 2))
plot(sol, main = "Dune NMDS plot")
stressplot(sol, main = "Shepard plot")
layout(1)
```



Links

I have several **vegan**-related posts on my blog. For a list of posts see http://www.fromthebottomoftheheap.net/blog/

Re-use