Chapter 5
Unconstrained Ordination

5.1 Objectives

While cluster analysis looks for discontinuities in a dataset, ordination extracts the main trends in the form of continuous axes. It is therefore particularly well adapted to analyse data from natural ecological communities, which are generally structured in gradients.

Practically, you will:

- Learn how to choose among various ordination techniques (PCA, CA, PCoA and NMDS), compute them using the correct options, and properly interpret the ordination diagrams
- Apply these techniques to the Doubs river data
- Overlay the result of a cluster analysis on an ordination diagram to improve the interpretation of the clustering results
- Interpret the structures in the species data using the environmental variables from a second dataset
- Write your own PCA function

5.2 Ordination Overview

5.2.1 Multidimensional Space

A multivariate data set can be viewed as a collection of sites positioned in a space where each variable defines one dimension. There are thus as many dimensions as variables. To reveal the structure of the data, it would be interesting to represent the main trends in the form of scatterplots of the sites. Since ecological data generally contain more than two variables, it is tedious and not very informative to draw the objects in a series of planes defined by all possible pairs of descriptors. For instance, if the matrix contains ten descriptors, the number of planes to draw would

D. Borcard et al., *Numerical Ecology with R*, Use R,
DOI 10.1007/978-1-4419-7976-6_5, © Springer Science+Business Media, LLC 2011

be equal to $(10 \times 9)/2 = 45$. Such a series of scatterplots would allow neither to bring out the most important structures of the data, nor to visualize the relationships among descriptors (which, in general, are not linearly independent of one another). The aim of ordination methods is to represent the data along a reduced number of orthogonal axes, constructed in such a way that they represent, in decreasing order, the main trends of the data. These trends can then be interpreted visually or in association with other methods such as clustering or regression. Here, we shall address four basic techniques. All these methods are descriptive: no statistical test is provided to assess the significance of the structures detected. That is the role of constrained ordination, a family of methods that are presented in Chap. 6.

5.2.2 Ordination in Reduced Space

Most ordination methods (except NMDS) are based on the extraction of the eigenvectors of an association matrix. They can be classified according to the distance preserved among sites and to the type of variables that they can handle. Legendre and Legendre (1998, Table 9.1, p. 388) provide a table showing their domains of application.

The basic principle of ordination in reduced space is the following. Imagine an $n \times p$ data set containing n objects and p variables. The n objects can be represented as a cluster of points in the p-dimensional space. Now, this cluster is generally not spheroid: it is elongated in some directions and flattened in others. These directions are not necessarily aligned with a single dimension (= a single variable) of the multidimensional space. The direction where the cluster is most elongated corresponds to the direction of largest variance of the cluster. This is the first axis that an ordination will extract. The next axis to be extracted is the second most important in variance, provided that it is *orthogonal* (linearly independent, uncorrelated) to the first one. The process continues until all axes have been computed.

When there are a few major structures in the data (gradients or groups) and the method has been efficient at extracting them, then the few first axes contain most of the useful information, i.e. they have extracted most of the variance of the data. In that case, the distances among sites in the projection in reduced space (most often two-dimensional) are relatively similar to the distances among objects in the multidimensional space. Note, however, that an ordination can be useful even when the first axes account for small proportions of the variance. This may happen when there are some interesting structures in an otherwise noisy data set. The question arising is then: how many axes should one retain and interpret? In other words, how many axes represent interpretable structures? The answer depends on the method; several helping procedures are explained in due course to answer this question.

The methods that are presented in this chapter are:

- *Principal component analysis (PCA)*: the main eigenvector-based method. Works on raw, quantitative data. Preserves the Euclidean distance among sites.
- *Correspondence analysis (CA)*: works on data that must be frequencies or frequency-like, dimensionally homogeneous, and non-negative. Preserves the χ^2 distance among rows or columns. Mainly used in ecology to analyse species data tables.
- *Principal coordinate analysis (PCoA)*: devoted to the ordination of distance matrices, most often in the Q mode, instead of site-by-variables tables. Hence, great flexibility in the choice of association measures.
- *Nonmetric multidimensional scaling (NMDS)*: unlike the three others, this is not an eigenvector-based method. NMDS tries to represent the set of objects along a predetermined number of axes while preserving the ordering relationships among them.

PCoA and NMDS can produce ordinations from any square distance matrix.

5.3 Principal Component Analysis

5.3.1 Overview

Imagine a data set whose variables are normally distributed. This data set is said to show a multinormal distribution. The first principal axis (or principal-component axis) of a PCA of this data set is the line that goes through the greatest dimension of the concentration ellipsoid describing this multinormal distribution. The following axes, which are orthogonal to one another and successively shorter, go through the following greatest dimensions of the ellipsoid (Legendre and Legendre 1998). One can derive a maximum of p principal axes from a data set containing p variables.

Stated otherwise, PCA carries out a rotation of the original system of axes defined by the variables, such that the successive new axes (called principal components) are orthogonal to one another, and correspond to the successive dimensions of maximum variance of the scatter of points. The principal components give the positions of the objects in the new system of coordinates. PCA works on a *dispersion matrix* $\boldsymbol{S}$, i.e. an association matrix among variables containing the variances and covariances of the variables, or the correlations computed from dimensionally heterogeneous variables. It is exclusively devoted to the analysis of quantitative variables. The distance preserved is the Euclidean distance and the relationships detected are linear. Therefore, it is not generally appropriate to the

analysis of raw species abundance data. These can, however, be subjected to PCA after an appropriate pre-transformation (Sects. 2.2.4 and 5.3.3).

In a PCA ordination diagram, following the tradition of scatter diagrams in Cartesian coordinate systems, *objects* are represented as *points* and *variables* are displayed as *arrows.*

Later in this chapter, we show how to program a PCA in **R** using matrix equations. But for everyday users, PCA is available in several **R** packages. A convenient function for ecologists is **`rda()`** in package **`vegan`**. The name of the function refers to redundancy analysis, a method that is presented in Chap. 6. Other possible functions (not detailed here) are **`dudi.pca()`** (package **`ade4`**) and **`prcomp()`** (package **`stats`**).

5.3.2 PCA on the Environmental Variables of the Doubs Data Set Using `rda()`

Let us work again on the Doubs data. We have 11 quantitative environmental variables at our disposal. How are they correlated? What can we learn from the ordination of the sites?

Since the variables are expressed in different measurement scales, we compute a PCA on the correlation matrix. Correlations are the covariances of standardized variables.

5.3.2.1 Preparation of the Data

```
# Load required  packages
library(ade4)
library(vegan)
library(gclus)
library(ape)

# Import the data from CSV files
spe <- read.csv("DoubsSpe.csv", row.names=1)
env <- read.csv("DoubsEnv.csv", row.names=1)
spa <- read.csv("DoubsSpa.csv", row.names=1)
# Remove empty site 8
spe <- spe[-8,]
env <- env[-8,]
spa <- spa[-8,]
# A reminder of the content of the env dataset
summary(env)  # Descriptive statistics
```

5.3.2.2 PCA on a Correlation Matrix

```
# PCA on the full dataset (correlation matrix: scale=TRUE)
# *******************************************************

env.pca <- rda(env, scale=TRUE)  # Argument scale=TRUE calls for
                                 # a standardization of the
                                 # variables
env.pca
summary(env.pca) # Default scaling 2
summary(env.pca, scaling=1)

  # Note that the scaling (see below) is called at the step of
  # the summary (or, below, for the drawing of biplots) and not
  # for the analysis itself.
```

The "summary" output is presented as follows for scaling 2 (some results have been deleted):

```
Call:
rda(X = env, scale = TRUE)

Partitioning of correlations:
              Inertia Proportion
Total              11          1
Unconstrained      11          1

Eigenvalues, and their contribution to the correlations

Importance of components:
                          PC1   PC2    PC3   PC4   PC5   PC6...
Eigenvalue            6.098 2.167 1.0376 0.704 0.352 0.319...
Proportion Explained  0.554 0.197 0.0943 0.064 0.032 0.029...
Cumulative Proportion 0.554 0.751 0.8457 0.910 0.942 0.971...

Scaling 2 for species and site scores
* Species are scaled proportional to eigenvalues
* Sites are unscaled: weighted dispersion equal on all
* dimensions
* General scaling constant of scores:  4.189264
```

```
Species scores

          PC1      PC2        PC3       PC4       PC5       PC6
das  1.08432  0.5148 -0.257430 -0.16170  0.21140 -0.09500
alt -1.04356 -0.5946  0.179904  0.12274  0.12527  0.14024
(...)

Site scores (weighted sums of species scores)

         PC1       PC2       PC3       PC4      PC5       PC6
1  -1.41239 -1.47577 -1.74581 -2.95537  0.2312  0.49150
2  -1.04170 -0.81766  0.34078  0.54374  0.9252 -1.77040
```

The ordination output uses some vocabulary that requires explanations.

- *Inertia*: in **vegan**'s language, this is the general term for "variation" in the data. This term comes from the world of CA (Sect. 5.4). In PCA, the "inertia" is either the sum of the variances of the variables (PCA on a covariance matrix) or, as in this case (PCA on a correlation matrix), the sum of the diagonal values of the correlation matrix, i.e. the sum of all correlations of the variables with themselves, which corresponds to the number of variables (11 in this example).
- *Constrained and unconstrained*: see Chap. 6 (canonical ordination). In PCA, the analysis is unconstrained, and so are the results.
- *Eigenvalues*: symbolized λ_j, these are measures of the importance (variance) of the axes. They can be expressed as *Proportions Explained*, or proportions of variation accounted for, by dividing them by the total inertia.
- *Scaling*: not to be confused with the argument `scale` calling for standardization of variables. "Scaling" refers to the way ordination results are projected in the reduced space for graphical display. There is no single way to optimally display objects and variables together in a PCA biplot, i.e. a plot showing two types of results, here the sites and the variables. Two main types of scaling are generally used. Each of them has properties that must be kept in mind for proper interpretation of the biplots. Here, we give the essential features of each scaling. Please refer to Legendre and Legendre (1998, pp. 403–404) for a complete account.
  - *Scaling 1* = distance biplot: the eigenvectors are scaled to unit length. (1) **Distances among objects in the biplot are approximations of their Euclidean distances in multidimensional space**. (2) *The angles among descriptor vectors are meaningless.*

  - *Scaling 2* = correlation biplot: each eigenvector is scaled to the square root of its eigenvalue. (1) *Distances among objects in the biplot are not approximations of their Euclidean distances in multidimensional space.* (2) **The angles between descriptors in the biplot reflect their correlations**.
  - In both cases, projecting an object at right angle on a descriptor approximates the position of the object along that descriptor.
  - Bottom line: if the main interest of the analysis is to interpret the relationships among **objects**, choose **scaling 1**. If the main interest focuses on the relationships among **descriptors**, choose **scaling 2**.
- *Species scores*: coordinates of the arrow heads of the variables. For historical reasons, response variables are always called "species" in **`vegan`**, no matter what they represent.
- *Site scores*: coordinates of the sites in the ordination diagram. Objects are always called "Sites" in **`vegan`** output files.

5.3.2.3 Extracting, Interpreting and Plotting Results from a `vegan` Ordination Output Object

`vegan` output objects are complex entities, and extraction of their elements does not follow the basic rules of **R**. Type **`?cca.object`** in the **R** console. This calls for a help file explaining all features of an **`rda()`** or **`cca()`** output object. The examples at the end of that help file show how to access some of the ordination results directly. Here, we access some important results as examples. Further results are examined later when useful.

Eigenvalues

First, let us examine the eigenvalues. Are the first few clearly larger than the following ones? Here, a question arises: how many ordination axes are meaningful to display and interpret?

PCA is not a statistical test, but a heuristic procedure: it aims at representing the major features of the data along a reduced number of axes (hence, the expression "ordination in reduced space"). Usually, the user examines the eigenvalues, and decides how many axes are worth representing and displaying on the basis of the amount of variance explained. The decision can be completely arbitrary (for instance, interpret the number of axes necessary to represent 75% of the variance of the data), or assisted by one of several procedures proposed to set a limit between the axes that represent interesting variation of the data and axes that merely display the remaining, essentially random variance. One of these procedures

(called the Kaiser–Guttman criterion) consists in computing the *mean of all eigenvalues* and interpreting only the axes whose eigenvalues are larger than that mean. Another is to compute a *broken stick model*, which randomly divides a stick of unit length into the same number of pieces as there are PCA axes. The theoretical equation for the broken stick model is known. The pieces are then put in order of decreasing length and compared to the eigenvalues. One interprets only the axes whose eigenvalues are larger than the length of the corresponding piece of the stick, or, alternately, one may compare the sum of eigenvalues, from 1 to k, to the sum of the values from 1 to k predicted by the broken stick model. One can compute these two procedures by hand as follows (Fig. 5.1)[1]:

```
# Examine and plot partial results from PCA output
# ************************************************

?cca.object  # Explains how an ordination object produced by
             # vegan is structured and how to extract its
             # results.

# Eigenvalues
(ev <- env.pca$CA$eig)

# Apply Kaiser-Guttman criterion to select axes
ev[ev > mean(ev)]

# Broken stick model
n <- length(ev)
bsm <- data.frame(j=seq(1:n), p=0)
bsm$p[1] <- 1/n
for (i in 2:n) {
 bsm$p[i] = bsm$p[i-1] + (1/(n + 1 - i))
}
bsm$p <- 100*bsm$p/n
bsm

# Plot eigenvalues and % of variance for each axis
par(mfrow=c(2,1))
barplot(ev, main="Eigenvalues", col="bisque", las=2)
abline(h=mean(ev), col="red")    # average eigenvalue
legend("topright", "Average eigenvalue", lwd=1, col=2, bty="n")
barplot(t(cbind(100*ev/sum(ev),bsm$p[n:1])), beside=TRUE,
 main="% variance", col=c("bisque",2), las=2)
```

[1] Comparison of a PCA result with the broken stick model can also be done by using function **`PCAsignificance()`** of package **`BiodiversityR`**.

```
legend("topright", c("% eigenvalue", "Broken stick model"),
 pch=15, col=c("bisque",2), bty="n")

 # Is the same number of axes retained by the two rules?
```

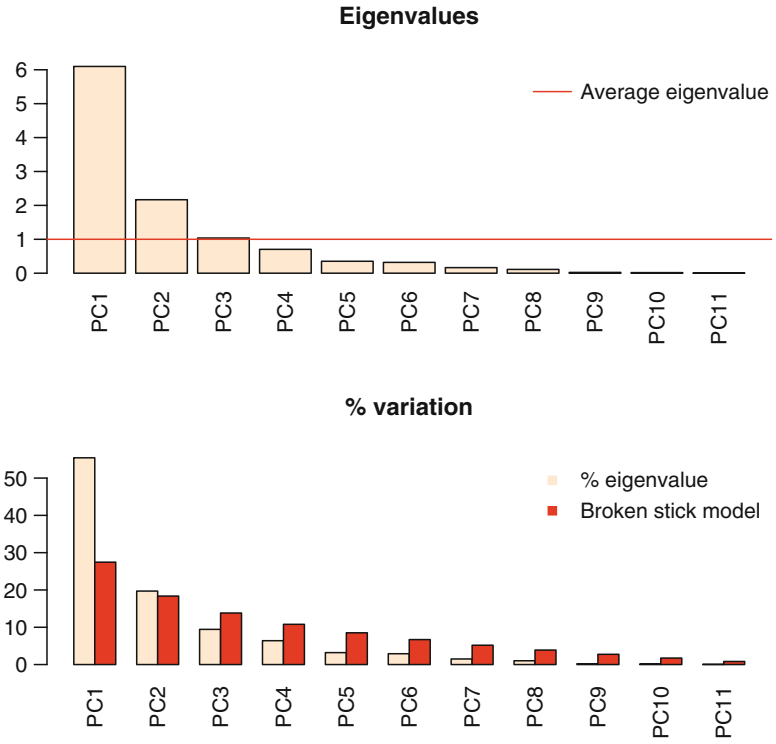

Fig. 5.1 Kaiser–Guttman and broken-stick plots to help assess the number of interpretable axes in PCA. Application to the Doubs environmental data

To make things easier, the code above has been framed in a function called **evplot()**, which is used as follows:

```
# Same plots using a single function:
# Plot eigenvalues and % of variance for each axis
source("evplot.R")
evplot(ev)
```

Biplots of Sites and Variables

To plot PCA results in a proper manner, one has to show objects as points and variables as arrows. Two plots are produced here, the first in scaling 1 (optimal display of objects), the second in scaling 2 (optimal display of variables) (Fig. 5.2). We present two functions: **vegan**'s **biplot.rda()** and a function directly drawing scaling 1 and 2 biplots from **vegan** results: **cleanplot.pca()**.

```
# Two PCA biplots: scaling 1 and scaling 2
# ****************************************

# Plots using biplot.rda
par(mfrow=c(1,2))
biplot.rda(env.pca, scaling=1, main="PCA - scaling 1")
biplot.rda(env.pca, main="PCA - scaling 2")  # Default scaling 2

# Plots using cleanplot.pca
# A rectangular graphic window is needed for the two plots
source("cleanplot.pca.R")
# With points for sites and arrowheads
cleanplot.pca(env.pca, point=TRUE)
# With site labels only (vegan's standard)
cleanplot.pca(env.pca)
cleanplot.pca(env.pca, ahead=0)   # ... and without arrowheads

 # What does the circle in the left-hand plot mean? See below...
```

Hint *Check in the* **cleanplot.pca()** *function how the plots are progressively built. First, one extracts two data tables (scaling 1 and 2, site scores and species scores) from the rda output object by means of the* **scores()** *function. Then an empty plot is produced using a special* **vegan** *plotting function called* **plot.cca()**. *Site points (function* **points()***) and site labels (function* **text()***) are added afterwards. Finally, species arrows and their labels are drawn, and a circle (see below) is added to the scaling 1 biplot.*

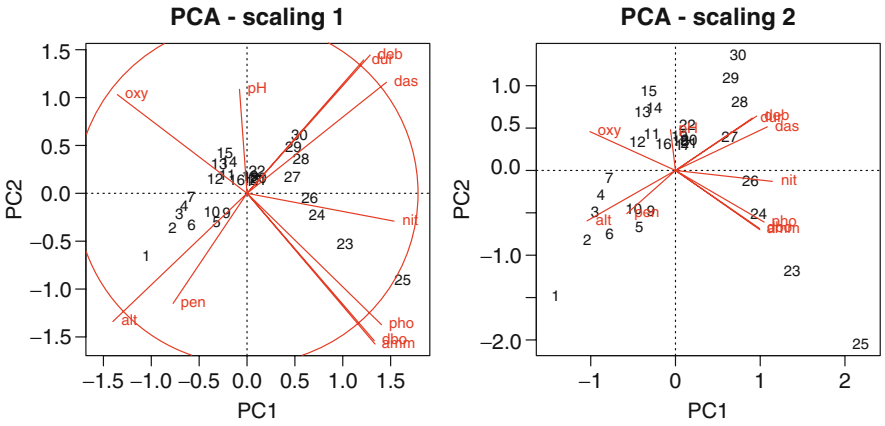

Fig. 5.2 PCA biplots of the Doubs environmental data, drawn with function **`cleanplot.pca()`**

Now, it is time to interpret the two biplots. First, the proportion of variance accounted for by the first two axes is 0.751 or 75.1%. This high value makes us confident that our interpretation of the first pair of axes extracts most relevant information from the data. Here is an example of how such a biplot can be interpreted.

First, the *scaling 1 biplot* displays a feature that must be explained. The circle is called a *circle of equilibrium contribution*. Its radius is equal to $\sqrt{d / p}$, where d is the number of axes represented in the biplot (usually $d = 2$) and p is the number of dimensions of the PCA space (i.e. usually the number of variables of the data matrix).[2] The radius of this circle represents the length of the vector representing a variable that would contribute equally to all the dimensions of the PCA space. Therefore, for any given pair of axes, the variables that have vectors longer than this radius make a higher contribution than average and can be interpreted with confidence.

The *scaling 1 biplot* shows a gradient from left to right, starting with a group formed by sites 1–10 which display the highest values of altitude (`alt`) and slope (`pen`), and the lowest values in river discharge (`deb`) and distance from the source (`das`); hardness (`dur`), which increases in the downstream direction, is also correlated

[2] Note, however, that **`vegan`** uses an internal constant to rescale its results, so that the vectors and the circle represented here are not equal but proportional to their original values. See the code of the **`cleanplot.pca()`** function.

to these variables. The second group of sites (11–16), has the highest values in oxygen content (`oxy`) and the lowest in nitrate concentration (`nit`). A third group of very similar sites (17–22) shows intermediate values in almost all the measured variables; they are not spread out by the variables contributing to axes 1 and 2. Phosphate (`pho`) and ammonium (`amm`) concentrations, as well as biological oxygen demand (`dbo`) show their maximum values around sites 23–25; the values decrease afterwards. Overall, the progression from oligotrophic, oxygen-rich to eutrophic, oxygen-deprived water is clear.

The *scaling 2 biplot* shows that the variables are organized in groups. The lower left part of the biplot shows that altitude and slope are very highly, positively correlated, and that these two variables are very highly, negatively correlated with another group comprising distance from the source, river discharge and calcium concentration. Oxygen content is positively correlated with slope (`pen`) and altitude, but very negatively with phosphate and ammonium concentration and, of course, with biological oxygen demand. The right part of the diagram shows the variables associated with the lower section of the river, i.e. the group discharge (`deb`) and hardness (`dur`), highly correlated with the distance from the source, and the group of variables linked to eutrophication, i.e. phosphate and ammonium concentration and biological oxygen demand. Positively correlated with these two groups is nitrate concentration (`nit`). Nitrate and pH have nearly orthogonal arrows, indicating a correlation close to 0. pH displays a shorter arrow, showing its lesser importance for the ordination of the sites in the ordination plane. A plot of axes 1 and 3 would emphasize its contribution to axis 3.

This example shows how useful a biplot representation can be in summarizing the main features of a data set. Clusters and gradients of sites are obvious, as are the correlations among the variables. The correlation biplot (scaling 2) is far more informative than the visual examination of a correlation matrix among variables; the latter can be obtained by typing **`cor(env)`**.

Technical remark: **`vegan`** provides a simple plotting function for ordination results, called **`plot.cca()`**. However, the basic use of this function provides PCA plots where sites as well as variables are represented by points. This is misleading, since the points representing the variables are actually the apices (tips) of vectors that must be drawn for the plot to be interpreted correctly.

Supplementary sites and species can be added to a PCA plot through the function **`predict.cca()`**. Explanatory variables can be added through the function **`envfit()`**.

5.3.2.4 Combining Clustering and Ordination Results

Comparing a cluster analysis and an ordination can be fruitful to explain or confirm the differences between groups of sites. Here, you will see two ways of combining these results. The first differentiates clusters of sites by colours on the ordination plot, the second overlays a dendrogram on the plot. Both are done on a single PCA plot here (Fig. 5.3), but they can be drawn separately, of course.

```
# Combining clustering and ordination results
# ******************************************

# Clustering the objects using the environmental data: Euclidean
# distance after standardizing the variables, followed by Ward
# clustering

env.w <- hclust(dist(scale(env)), "ward")

# Cut the dendrogram to yield 4 groups
gr <- cutree(env.w, k=4)
grl <- levels(factor(gr))

# Get the site scores, scaling 1
sit.sc1 <- scores(env.pca, display="wa", scaling=1)

# Plot the sites with cluster symbols and colours (scaling 1)
p <- plot(env.pca, display="wa", scaling=1, type="n",
 main="PCA correlation + clusters")
abline(v=0, lty="dotted")
abline(h=0, lty="dotted")
for (i in 1:length(grl)) {
 points(sit.sc1[gr==i,], pch=(14+i), cex=2, col=i+1)
 }
text(sit.sc1, row.names(env), cex=.7, pos=3)

# Add the dendrogram
ordicluster(p, env.w, col="dark grey")
legend(locator(1), paste("Group",c(1:length(grl))),
 pch=14+c(1:length(grl)),
 col=1+c(1:length(grl)), pt.cex=2)
```

Hint *See how the coding of the symbols and colours is conditioned on the number of groups automatically: object* `grl` *has been set to contain numbers from 1 to the number of groups.*

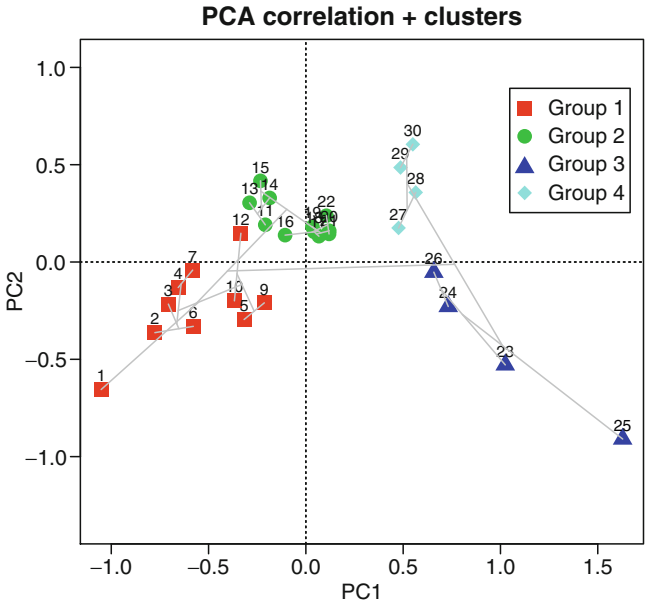

Fig. 5.3 PCA biplot (scaling 1) of the Doubs environmental data with overlaid clustering results

5.3.3 PCA on Transformed Species Data

PCA being a linear method preserving the Euclidean distance among sites, it is not naturally adapted to the analysis of species abundance data. However, transforming these after Legendre and Gallagher (2001) alleviates this problem (Sect. 2.2.4). Here is a quick application with a Hellinger pre-transformation on the fish data (Fig. 5.4).

```
# PCA on the fish abundance data
# ******************************

# Hellinger pre-transformation of the species data
spe.h <- decostand(spe, "hellinger")
spe.h.pca <- rda(spe.h)
spe.h.pca

# Plot eigenvalues and % of variance for each axis
ev <- spe.h.pca$CA$eig
evplot(ev)

# PCA biplots
cleanplot.pca(spe.h.pca, ahead=0)

 # The species do not form clear groups like the environmental
 # variables. However, see how the species replace one another
 # along the site sequence.
 # On the scaling 1 biplot, observe that 8 species contribute
 # strongly to axes 1 and 2. Are these species partly of
 # completely the same as those identified as indicator of the
 # groups in Section 4.10.3?
```

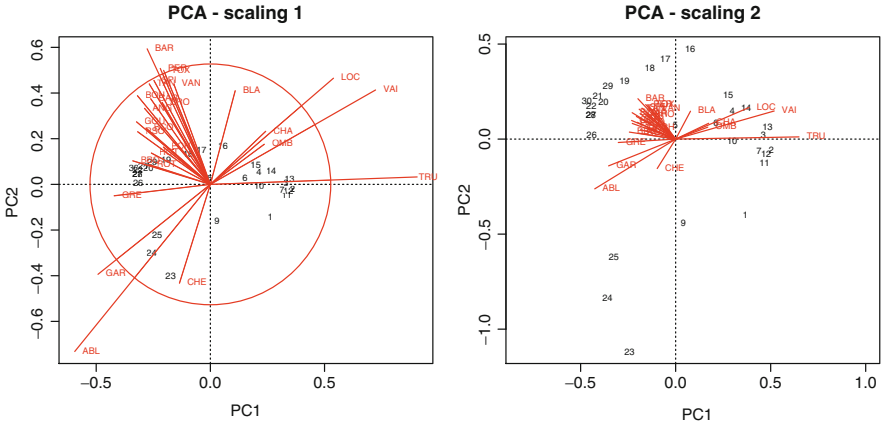

Fig. 5.4 PCA biplots of the Hellinger-transformed fish species data

For comparison, repeat the PCA on the original file `spe` without transformation. Which ordination shows better the gradient of species contributions along the course of the river?

Although PCA has a long history as a method devoted to tables of physical and chemical variables, the recent introduction of species data pre-transformations has opened up this powerful technique to the analysis of community data. Although PCA itself is not modified and remains a linear ordination model, the pre-transformations ensure that the species data are treated according to their specificity, i.e. without undue importance being given to double zeros. A scaling 1 PCA biplot thus reveals the underlying gradients structuring the community; the sites are ordered along the axes according to their positions along these gradients. The circle of equilibrium contribution allows the identification of the species contributing most to the plotted pair of axes. A scaling 2 biplot reveals the relationships among species in a correlation-like fashion; since the data have been transformed, the correlations are not completely equivalent to Pearson's *r* computed on raw data.

The chi-square transformation can also be applied to species data prior to PCA. In that case, the PCA solution is very similar, but not identical to a CA of the species data. Although the two methods preserve the chi-square distance among the sites, the calculation of the eigen-decomposition is not done in exactly the same way and leads to different sets of eigenvalues and eigenvectors.

5.3.4 Domain of Application of PCA

Principal component analysis is a very powerful technique, but it has its limits. The main application of PCA in ecology is the ordination of sites on the basis of quantitative environmental variables or, after an appropriate transformation, of community composition data. PCA has originally been defined for data with multinormal distributions. In its applications in ecology, however, PCA is not very sensitive to departure from multinormality, as long as the distributions are not exaggeratedly skewed. The main computational step of PCA is the eigen-decomposition of a dispersion matrix (linear covariances or correlations). Covariances must in turn be computed on quantitative data – but see below for binary data. Here are, in more detail, the conditions of application of PCA:

- PCA must be computed on a table of dimensionally homogeneous variables. The reason is that it is the sum of the variances of the variables that is partitioned into eigenvalues. Variables must be in the same physical units to produce a meaningful sum of variances (the units of a variance is the square of the units of the variable from which it was computed), or they must be dimensionless, which is the case for standardized or log-transformed variables.

- The data matrix must not be transposed since covariances or correlations among objects are meaningless.
- Covariances and correlations are defined for quantitative variables. However, PCA is very robust to variations in the precision of data. Since a Pearson correlation coefficient on semi-quantitative data is equivalent to a Spearman's correlation, a PCA on such variables yields an ordination where the relationship among variables is estimated using that measure.
- PCA can be applied to binary data. Gower (1966, *in* Legendre and Legendre 1998) has shown that, with binary descriptors, PCA positions the objects, in the multidimensional space, at distances that are the square roots of complements of simple matching coefficients S_1 (i.e. $\sqrt{1-S_1}$) times a constant which is the square root of the number of binary variables.
- Species presence–absence data can be subjected to a Hellinger or chord transformation prior to PCA. The justification is that the Hellinger and chord distances computed on presence–absence data are both equal to $\sqrt{2}\sqrt{1-\text{Ochiai similarity}}$, so PCA after Hellinger or chord transformation preserves the Ochiai distance among objects in scaling type 1 plots. We also know that $\sqrt{1-\text{Ochiai similarity}}$ is a metric distance (Legendre and Legendre 1998, Table 7.2) which is appropriate for the analysis of community composition presence–absence data.
- Avoid the mistake of interpreting the relationships among variables based on the proximities of the apices (tips) of the vector arrows instead of their angles in biplots.

5.3.5 PCA Using Function `PCA()`

For someone who wants a quick assessment of the structure of his or her data, a quick way is to use functions **`PCA()`** and **`biplot.PCA()`**. Here is how they work (example on the Doubs environmental data).

```
# PCA on the environmental data set using PCA and biplot.PCA
# ***********************************************************

source("PCA.R")   # In the working directory or give path

# PCA; scaling 1 is the default for biplots
env.PCA.PL1 <- PCA(env, stand=TRUE)
biplot.PCA (env.PCA.PL1)
abline(h=0, lty=3)
abline(v=0, lty=3)
```

```
# PCA; scaling 2 in the biplot
env.PCA.PL2 <- PCA(env, stand=TRUE)
biplot.PCA (env.PCA.PL2 ,scaling=2)
abline(h=0, lty=3)
abline(v=0, lty=3)

  # The graphs may be mirror images of those obtained with vegan.
  # This is unimportant since the choice of the sign of the
  # principal components, made within the PCA functions, is
  # arbitrary.
```

5.4 Correspondence Analysis

5.4.1 Introduction

For a long time, CA has been one of the favourite tools for the analysis of species presence–absence or abundance data. The raw data are first transformed into a matrix $\overline{Q}$ of cell-by-cell contributions to the Pearson χ^2 statistic, and the resulting table is submitted to a singular value decomposition to compute its eigenvalues and eigenvectors. The result is an ordination, where it is the χ^2 distance (D_{16}) that is preserved among sites instead of the Euclidean distance D_1. The χ^2 distance is not influenced by the double zeros. Therefore, CA is a method adapted to the analysis of species abundance data without pre-transformation. Note that the data submitted to CA must be frequencies or frequency-like, dimensionally homogeneous and non-negative; that is the case of species counts or presence–absence data.

For technical reasons not developed here, CA ordination produces one axis fewer than min[*n*,*p*]. As in PCA, the orthogonal axes are ranked in decreasing order of the variation they represent, but instead of the total variance of the data, the variation is measured by a quantity called the total inertia (sum of squares of all values in matrix $\overline{Q}$, see Legendre and Legendre 1998, eq. 9.32). Individual eigenvalues are always smaller than 1. To know the amount of variation represented along an axis, one divides the eigenvalue of this axis by the total inertia of the species data matrix.

In CA, both the objects and the species are generally represented as points in the same joint plot. As in PCA, two scalings of the results are most useful in ecology. They are explained here for data matrices where objects are rows and species are columns:

- *CA scaling 1*: rows are at the centroids of columns. This scaling is the most appropriate if one is primarily interested in the ordination of *objects (sites)*.

In the multidimensional space, the χ^2 distance is preserved among objects. Interpretation: (1) the distances among objects in the reduced space approximate their χ^2 distances. Thus, object points that are close to one another are likely to be relatively similar in their species relative frequencies. (2) Any object found near the point representing a species is likely to contain a high contribution of that species. For presence–absence data, the object is more likely to possess the state "1" for that species.

- *CA scaling 2*: columns are at the centroids of rows. This scaling is the most appropriate if one is primarily interested in the ordination of *species*. In the multidimensional space, the χ^2 distance is preserved among species. Interpretation: (1) the distances among species in the reduced space approximate their χ^2 distances. Thus, species points that are close to one another are likely to have relatively similar relative frequencies along the objects. (2) Any species that lies close to the point representing an object is more likely to be found in that object, or to have a higher frequency there than in objects that are further away in the joint plot.

The Kaiser–Guttman criterion and the broken stick model, explained in Sect. 5.3.2.3, can be applied to CA axes for guidance as to the number of axes to retain. Our application below deals with the raw fish abundance data.

5.4.2 CA Using Function `cca()` of Package `vegan`

5.4.2.1 Running the Analysis and Drawing the Biplots

The procedure below closely resembles the one applied for PCA. First, let us run the analysis and draw the Kaiser–Guttman and broken stick plots (Fig. 5.5):

```
# CA of the raw species dataset (original species abundances)
# ***********************************************************

# Compute CA
spe.ca <- cca(spe)
spe.ca
summary(spe.ca)            # default scaling 2
summary(spe.ca, scaling=1)

 # The first axis has a large eigenvalue. In CA, values over 0.6
 # indicate a very strong gradient in the data. What proportion
 # of the total inertia does the first axis account for?
 # Note that the eigenvalues are the same in both scalings.
 # The scaling affects the eigenvectors but not the eigenvalues.
```

```
# Plot eigenvalues and % of variance for each axis
(ev2 <- spe.ca$CA$eig)
evplot(ev2)

 # Here the broken stick rule is more conservative than the
 # other.
 # The first axis is extremely dominant, as can be seen from
 # the bar plots as well as the numerical results.
```

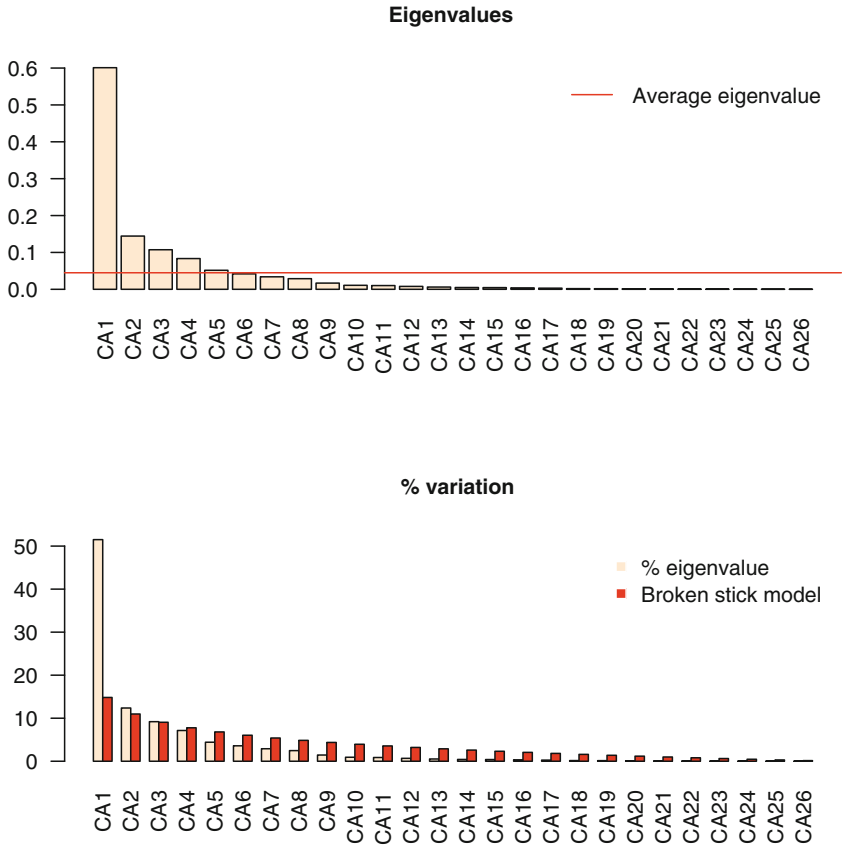

Fig. 5.5 Kaiser–Guttman and broken-stick plots to help assess the number of interpretable axes in CA. Application to the Doubs fish raw abundance data

It is time to draw the CA biplots of this analysis. Let us compare the two scalings (Fig. 5.6).

```
# CA biplots
# **********

par(mfrow=c(1,2))

# Scaling 1: sites are centroids of species
plot(spe.ca, scaling=1, main="CA fish abundances - biplot
  scaling 1")

# Scaling 2 (default): species are centroids of sites
plot(spe.ca, main="CA fish abundances - biplot scaling 2")
```

Hint *Here you could also produce a clustering and overlay its result on the CA plot.*

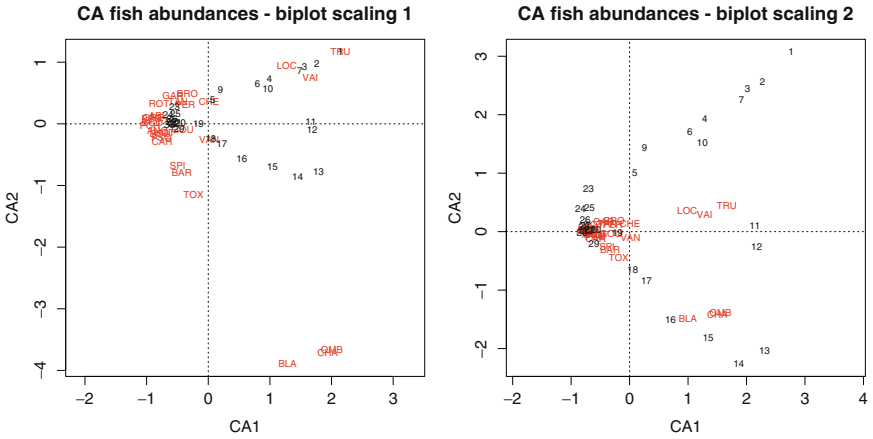

Fig. 5.6 CA biplots of the Doubs fish abundance data

The first axis opposes the lower section of the stream (sites 19–30) to the upper portion. This is clearly a strong contrast, which explains why the first eigenvalue is so high. Many species appear close to sites 19–30, indicating that they are more abundant downstream. Many of them are actually absent from the upper part of the river. The second axis contrasts the ten upstream sites to the intermediate ones. Both groups of sites, which display short gradients on their own, are associated with characteristic species. The scaling 2 plot shows how small groups of species are distributed among the sites. One can see that the grayling (OMB), the bullhead (CHA) and the varione (BLA) are found in the intermediate group of sites (11–18),

while the brown trout (`TRU`), the Eurasian minnow (`VAI`) and the stone loach (`LOC`) are found in a longer portion of the stream (approximately sites 1–18).

Observe how scalings 1 and 2 produce different plots. Scaling 1 shows the sites at the (weighted) centre of mass of the species. This is appropriate to interpret site proximities and find gradients or groups of sites. The converse is true for the scaling 2 biplot, where one can look for groups or replacement series of species. In both cases, care should be taken for the interpretation of species close to the origin of the graph. This proximity could mean either that the species is at its optimum in the mid-range of the ecological gradients represented by the axes, or that it is present everywhere along the gradient.

5.4.2.2 Passive (*Post Hoc*) Explanation of Axes Using Environmental Variables

Although there are means of incorporating explanatory variables directly in the ordination process (canonical ordination, see Chap. 6), one may be interested in interpreting a simple ordination by means of external variables. This can be done in **`vegan`** by means of the function **`envfit()`**. According to its author, Jari Oksanen, "**`envfit`** *finds vectors or factor averages of environmental variables. [...] The projections of points onto vectors have maximum correlation with corresponding environmental variables, and the factors show the averages of factor levels*".

The result is an object containing coordinates of factor levels (points) or arrowheads (quantitative variables) that can be used to project these variables into the ordination diagram (Fig. 5.7):

```
# A posteriori projection of environmental variables in a CA
# The last plot produced (CA scaling 2) must be active

spe.ca.env <- envfit(spe.ca, env)
plot(spe.ca.env)
# This has added the environmental variables to the last biplot
drawn

  # Does this new information help interpret the biplot?
```

Hint *This is a* post hoc *interpretation of ordination axes. Compare with Chap. 6.*

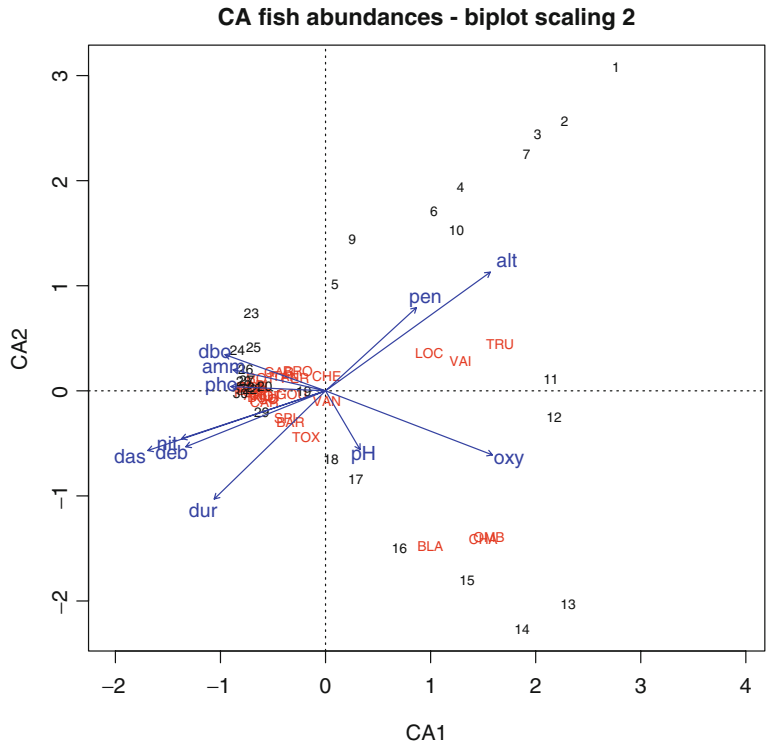

Fig. 5.7 CA biplot (scaling 2) of the Doubs fish abundance data with *a posteriori* projection of environmental variables

`envfit()` also proposes permutation tests to assess the significance of the r^2 of each explanatory variable regressed on the two axes of the biplot. But this is not, by far, the best way to test the effect of explanatory variables on a table of response variables. We explore this topic in Chap. 6.

5.4.2.3 Reordering the Data Table on the Basis of an Ordination Axis

A CA result is sometimes used to reorder the data table according to the first ordination axis. A compact form of ordered table is provided by the **vegan** function **vegemite()** already used in Chap. 4, which can use the information provided by an ordination computed in **vegan**:

```
# Species data table ordered after the CA result
# **********************************************

vegemite(spe, spe.ca)

  # The left-right and up-down orderings in this ordered
  # table depends on the (arbitrary) orientation of the
  # ordination axes.
  # Observe that the ordering is not optimal since it is done
  # only on the basis of the first axis. Therefore, sites
  # 1 to 10 and 11 to 18 (separated along axis 2) and their
  # corresponding characteristic species are interspersed.
```

5.4.3 CA Using Function CA()

As in the case of PCA, we propose a simple CA function: **CA()**. Here is how to use it on the fish data.

```
# CA using CA() function
# **********************

source("CA.R") # Function in the working directory or give path
spe.CA.PL <- CA(spe)
biplot(spe.CA.PL, cex=1)

# Ordering of the data table following the first CA axis
# The table is transposed, as in vegemite() output
summary(spe.CA.PL)
t(spe[order(spe.CA.PL$F[,1]),order(spe.CA.PL$V[,1])])
```

Hints *The use of matrices F and V to reorder the table relates to the symbolism used by Legendre and Legendre (1998, Section 9.4) to explain the mathematics of correspondence analysis. Using Vhat instead of F and Fhat instead of V (i.e. using the scaling 2 projection) would have produced the same ordered table.*

Argument `cex` *of the* **biplot()** *function is here to adapt the size of the symbols and the site and species names to the plot. The default is* `cex=2`*. Smaller values produce smaller symbols and characters. They may be useful for plots containing many sites and species.*

5.4.4 Arch Effect and Detrended Correspondence Analysis

Long environmental gradients often support a succession of species. Since the species that are controlled by environmental factors tend to have unimodal distributions, a long gradient may encompass sites that, at both ends of the gradient, have no species in common; thus, their distance reaches a maximum value (or their similarity is 0). But if one looks at either side of the succession, contiguous sites continue to grow more different from each other. Therefore, instead of a linear trend, the gradient is represented on a pair of CA axes as an arch. Several detrending techniques have been proposed to counter this effect and straighten up gradients in ordination diagrams, leading to detrended correspondence analysis (DCA):

- Detrending by segments combined with nonlinear rescaling: axis I is divided into an arbitrary number of segments and, within each one, the mean of the object scores along axis 2 is made equal to zero. The number of segments has a large influence on the result. The DCA results presented in the literature suggest that the scores along the second axis are essentially meaningless. The authors of this book strongly warn against the use of this form of DCA as an ordination technique; however, it may be used to estimate the "gradient length" of the first ordination axis, expressed in standard deviation units of species turnover. A gradient length larger than 4 indicates that some species have a unimodal response along the axis (ter Braak and Šmilauer 2002).
- Detrending by polynomials: another line of reasoning about the origin of the arch effect leads to the observation that when an arch occurs, the second axis can be seen as quadratically related to the first (i.e. it is the first axis to the power 2). This explains for the parabolic shape of the scatter of points. Hence, a solution is to make the second axis not only linearly, but also quadratically independent from the first. Although intuitively attractive, this method of detrending has to be applied with caution because it actually imposes a constraining model on the data.

DCA by segments is available in package **`vegan`** (function **`decorana()`**). In the output of this function, the gradient length of the axes is called "Axis lengths".

Given all its problems (see discussion in Legendre and Legendre 1998, pp. 465–472), we do not describe this method further here.

An even more extreme effect of the same kind exists in PCA. It is called the horseshoe effect because, in the case of strong gradients, the sites of both ends bend inwards and appear closer than other pairs. This is due to the fact that PCA considers double zeros as resemblances. Consequently, sites located at opposite ends of an ecological gradient, having many double zeros, "resemble" each other on this respect. The Hellinger or chord transformation of the species data partly alleviates this problem.

5.4.5 Multiple Correspondence Analysis

Multiple correspondence analysis (MCA) is the counterpart of PCA for the ordination of a table of categorical variables, i.e. a data frame in which all variables are factors. It is implemented in the function **`mca()`** of the package **`MASS`** and, with more options, in the function **`MCA()`** of the package **`FactoMineR`**.

5.5 Principal Coordinate Analysis

5.5.1 Introduction

PCA as well as CA impose the distance preserved among objects: the Euclidean distance (and several others with pre-transformations) for PCA and the χ^2 distance for CA. If one wishes to ordinate objects on the basis of another distance measure, more appropriate to the problem at hand, then PCoA is the method of choice. It provides a Euclidean representation of a set of objects whose relationships are measured by any similarity or distance measure chosen by the user. For example, if the coefficient is Gower's index S_{15}, which can combine descriptors of many mathematical types into a single measure of resemblance, then the ordination represents the relationships among the objects based upon these many different variables. This would not be possible with PCA or CA.

Like PCA and CA, PCoA produces a set of orthogonal axes whose importance is measured by eigenvalues. Since it is based on an association matrix, it can directly represent the relationships either among objects (if the association matrix was in Q mode) or variables (if the association matrix was in R mode). If it is necessary to project variables, e.g. species, on a PCoA ordination of the objects, the variables can be related *a posteriori* to the ordination axes using correlations or weighted averages and drawn on the ordination plot. In the case of Euclidean association measures, PCoA behaves in a Euclidean manner. For instance, computing a Euclidean distance among sites and running a PCoA yields the same results as running a PCA on a covariance matrix of the same data and looking at the scaling 1 ordination results. But if the association coefficient used is non-Euclidean, then

PCoA may react by producing several negative eigenvalues in addition to the positive ones (and a null eigenvalue in-between). The axes corresponding to negative eigenvalues cannot be represented on real ordination axes since they are complex. In most applications, this does not affect the representation of the objects on the several first principal axes, but it can lead to problems if the largest negative eigenvalues are of the same magnitude in absolute value as the first positive ones.

There are technical solutions to this problem, which consist in adding a constant to either the squared distances among objects (Lingoes correction) or to the distances themselves (Cailliez correction) (Gower and Legendre 1986). In the function **`cmdscale()`** presented below, the Cailliez correction is obtained with the argument `add=TRUE`.

One can avoid complex axes by keeping the eigenvectors with their original Euclidean norm (vector length = 1) instead of dividing each one by the square root of its eigenvalue, as is usual in the PCoA procedure. This workaround is used in the MEM spatial analysis presented in Chap. 7. It should not be used for routine ordination by PCoA since eigenvectors that have not been rescaled to $\sqrt{\text{eigenvalue}}$ cannot be used to produce plots that preserve the original distances among the objects.

The ordination axes of a PCoA can be interpreted like those of a PCA or CA: proximity of objects represents similarity in the sense of the association measure used.

5.5.2 *Application to the Doubs Data Set Using* ***`cmdscale`*** *and* ***`vegan`***

As an example, let us compute a matrix of Bray–Curtis dissimilarities among sites, and subject this matrix to PCoA. In **`vegan`**, there is a way to project weighted averages of species abundances on a PCoA plot, by means of function **`wascores()`** (Fig. 5.8). Since species are projected as weighted averages of their contributions to the sites, their interpretation with respect to the sites is done as in CA.

```
# PCoA on a Bray-Curtis dissimilarity matrix of fish species
# ***********************************************************
spe.bray <- vegdist(spe)
spe.b.pcoa <- cmdscale(spe.bray, k=(nrow(spe)-1), eig=TRUE)

# Plot of the sites and weighted average projection of species
ordiplot(scores(spe.b.pcoa)[,c(1,2)], type="t",
 main="PCoA with species")
abline(h=0, lty=3)
abline(v=0, lty=3)
```

```
# Add species
spe.wa <- wascores(spe.b.pcoa$points[,1:2], spe)
text(spe.wa, rownames(spe.wa), cex=0.7, col="red")
```

Hint *Observe the use of two* **vegan** *functions,* **ordiplot()** *and* **scores()**, *to produce the ordination plot.* **vegan** *is a world in itself and often requires special functions to handle its own results.*

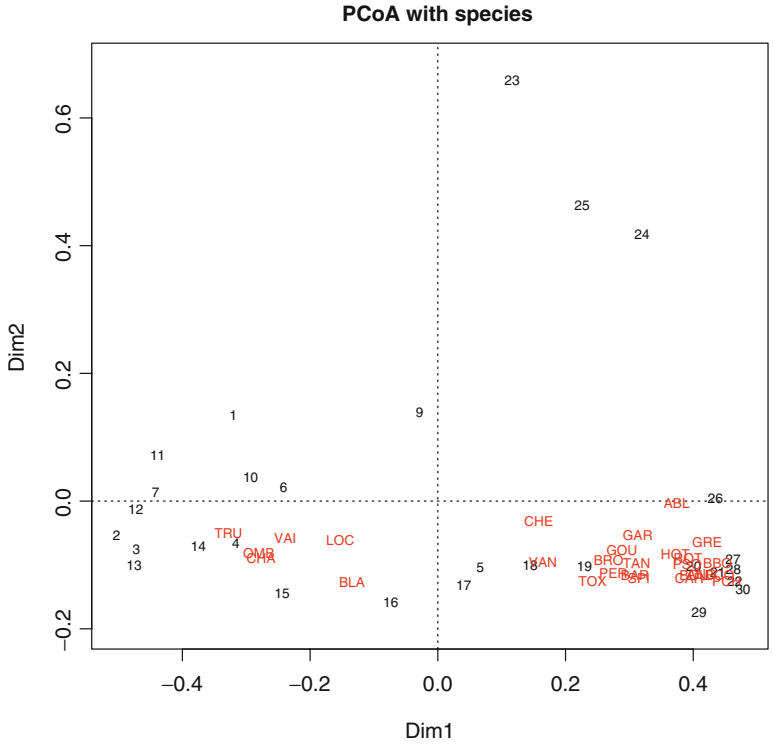

Fig. 5.8 PCoA biplot of a Bray–Curtis dissimilarity matrix of the raw Doubs fish abundance data. *A posteriori* projection of the species as weighted averages. The relationships between species and sites are interpreted as in CA

5.5.3 Application to the Doubs Data Set Using `pcoa()`

There is another way to achieve double projection. It is based on correlations of the environmental variables with the PCoA ordination axes (see Legendre and Legendre 1998, p. 431). If a PCoA of a matrix of Euclidean distances and scaling 1 is computed, this method produces vectors corresponding to what would be obtained in a scaling 1 PCA biplot of the same data. This representation is available in functions **`pcoa()`** and **`biplot.pcoa()`**, both available in packages **`ape`** and **`PCNM`**.

Here is how these functions work. In our example, PCoA is run on a Euclidean distance matrix computed on a Hellinger-transformed species abundance matrix; the result of these two operations is a Hellinger distance matrix. In such a case, it is actually better (simpler and faster) to run a PCA directly on the transformed species data, but here the idea is to allow a comparison with the PCA run presented in Sect. 5.3.3. Two biplots are proposed, with projection of the raw and standardized species abundances. Compare the result below (Fig. 5.9) with the biplot of the PCA scaling 1 result.

```
# PCoA and projection of species vectors using function pcoa()
# ************************************************************

spe.h.pcoa <- pcoa(dist(spe.h))

# Biplots
par(mfrow=c(1,2))
# First biplot: Hellinger-transformed species data
biplot.pcoa(spe.h.pcoa, spe.h, dir.axis2=-1)
abline(h=0, lty=3)
abline(v=0, lty=3)
# Second biplot: standardized Hellinger-transformed species data
spe.std <- apply(spe.h, 2, scale)
biplot.pcoa(spe.h.pcoa, spe.std, dir.axis2=-1)
abline(h=0, lty=3)
abline(v=0, lty=3)

 # How does this result compare with that of the PCA?
```

Hints *For projection of species data onto a PCoA plot, it is important to use the species data with the same transformation (if any) as the one used to compute the dissimilarity matrix. The standardization proposed here as an alternative may help better visualize the variables if they have very different variances.*

The argument `dir.axis2=-1` *reverses axis 2 to make the result directly comparable with the PCA result in Fig. 5.4, scaling 1. Remember that the signs of ordination axes are arbitrary.*

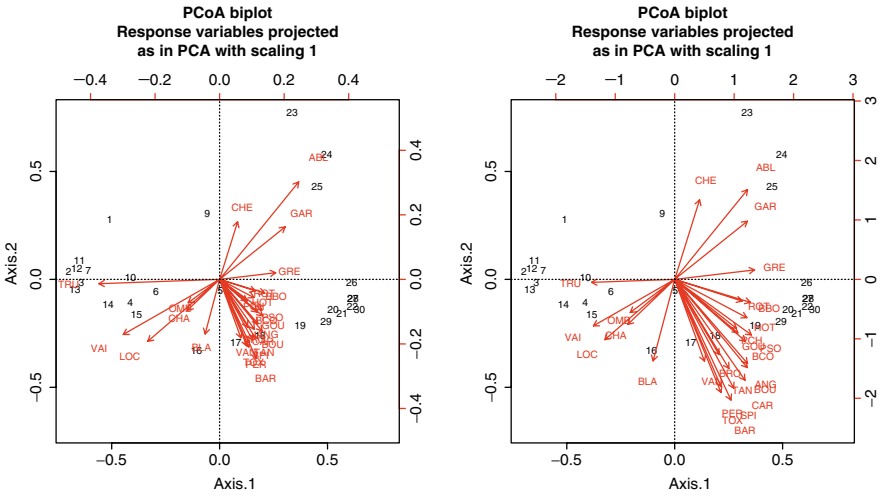

Fig. 5.9 PCoA biplots of the fish data obtained with functions **`pcoa()`** and **`biplot.pcoa()`**. *Left*: Hellinger-transformed raw species variables. *Right*: standardized Hellinger-transformed species. The bottom and left-hand scales are for the objects, the top and right-hand scales are for the species

As mentioned above, PCoA should actually be reserved to situations where no Euclidean measure is available or selected. With Jaccard and Sørensen dissimilarity matrices computed by **`ade4`**, for example, the ordination is fully Euclidean. In other cases, however, such as Bray–Curtis dissimilarities computed with **`vegan`**, the dissimilarity matrices may not be Euclidean (see Sect. 3.3.5). This results in PCoA producing some negative eigenvalues. Lingoes and Cailliez corrections are available in the function **`pcoa()`**. This function provides the eigenvalues along with a broken stick comparison in its output. Compare the examples below:

```
# Comparison of PCoA results with Euclidean and non-Euclidean
# dissimilarity matrices
# ************************************************************

# PCoA on a Hellinger distance matrix
is.euclid(dist(spe.h))
summary(spe.h.pcoa)
spe.h.pcoa$values
```

```
# PCoA on a Bray-Curtis dissimilarity matrix
is.euclid(spe.bray)
spe.bray.pcoa <- pcoa(spe.bray)
spe.bray.pcoa$values     # Observe eigenvalues 18 and following

# PCoA on the square root of a Bray-Curtis dissimilarity matrix
is.euclid(sqrt(spe.bray))
spe.braysq.pcoa <- pcoa(sqrt(spe.bray))
spe.braysq.pcoa$values  # Observe the eigenvalues

# PCoA on a Bray-Curtis dissimilarity matrix with Lingoes
correction
spe.brayl.pcoa <- pcoa(spe.bray, correction = "lingoes")
spe.brayl.pcoa$values   # Observe the eigenvalues

# PCoA on a Bray-Curtis dissimilarity matrix with Cailliez
correction
spe.brayc.pcoa <- pcoa(spe.bray, correction = "cailliez")
spe.brayc.pcoa$values   # Observe the eigenvalues

 # If you want to choose the analysis displaying the highest
 # proportion of variation on axes 1+2, which solution will you
 # select among those above?
```

5.6 Nonmetric Multidimensional Scaling

5.6.1 Introduction

If the researcher's priority is not to preserve the exact distances among objects in an ordination plot, but rather to represent as well as possible the ordering relationships among objects in a small and specified number of axes, nonmetric multidimensional scaling (NMDS) may be the solution. Like PCoA, NMDS can produce ordinations of objects from any distance matrix. The method can cope with missing distances, as long as there are enough measures left to position each object with respect to a few others. NMDS is not an eigenvalue technique, and it does not maximize the variability associated with individual axes of the ordination. As a result, plots may arbitrarily be rotated, centred or inverted. The procedure goes as follows (very schematically; for details see Legendre and Legendre 1998, p. 445 *et seq.*):

- Specify the number *m* of axes (dimensions) sought.
- Construct an initial configuration of the objects in the *m* dimensions, to be used as a starting point of an iterative adjustment process. This is a tricky step, since

the end-result may depend on the starting configuration. A PCoA ordination may be a good starting point. Otherwise, try many independent runs with random initial configurations.
- An iterative procedure tries to position the objects in the requested number of dimensions in such a way as to minimize a stress function (scaled from 0 to 1), which measures how far the distances in the reduced-space configuration are from being monotonic to the original distances in the association matrix.
- The adjustment goes on until the stress value can no more be lowered, or until it reaches a predetermined value (tolerated lack-of-fit).
- Most NMDS programs rotate the final solution using PCA for easier interpretation.

For a given and small number of axes (e.g. $m = 2$ or 3), NMDS often achieves a less deformed representation of the distance relationships among objects than a PCoA in the same number of dimensions. But NMDS is a computer-intensive technique exposed to the risk of suboptimal solutions in the iterative process. Indeed, the objective stress function to minimize often reaches a local minimum larger than the true minimum.

5.6.2 Application to the Fish Data

NMDS can be performed in **R** with the elegant function **`metaMDS()`** from the **`vegan`** package. **`metaMDS()`** accepts raw data or distance matrices. Let us apply it to the fish abundances using the Bray–Curtis index. **`metaMDS()`** uses random starts and iteratively tries to find the best possible solution. Species points are added to the ordination plot using **`wascores()`**. See Fig. 5.10.

If one must use a distance matrix with missing values, NMDS can be computed with the function **`isoMDS()`**. An initial configuration must be provided in the form of a matrix positioning the sites (argument `y`) in the number of dimensions specified for the analysis (argument `k`). To reduce the risk of reaching a local minimum, we suggest to use the function **`bestnmds()`** of the package **`labdsv`**. This function, which is a wrapper for **`isoMDS()`**, computes the analysis a user-specified number of times (argument `itr`) with internally produced random initial configurations. The solution with the smallest stress value is retained by the function.

```
# NMDS applied to the fish species - Bray-Curtis distance matrix
# ************************************************************

spe.nmds <- metaMDS(spe, distance="bray")
spe.nmds
```

```
spe.nmds$stress
plot(spe.nmds, type="t", main=paste("NMDS/Bray - Stress =",
  round(spe.nmds$stress,3)))

  # How does this result compare with those of PCA, CA and PCoA?
```

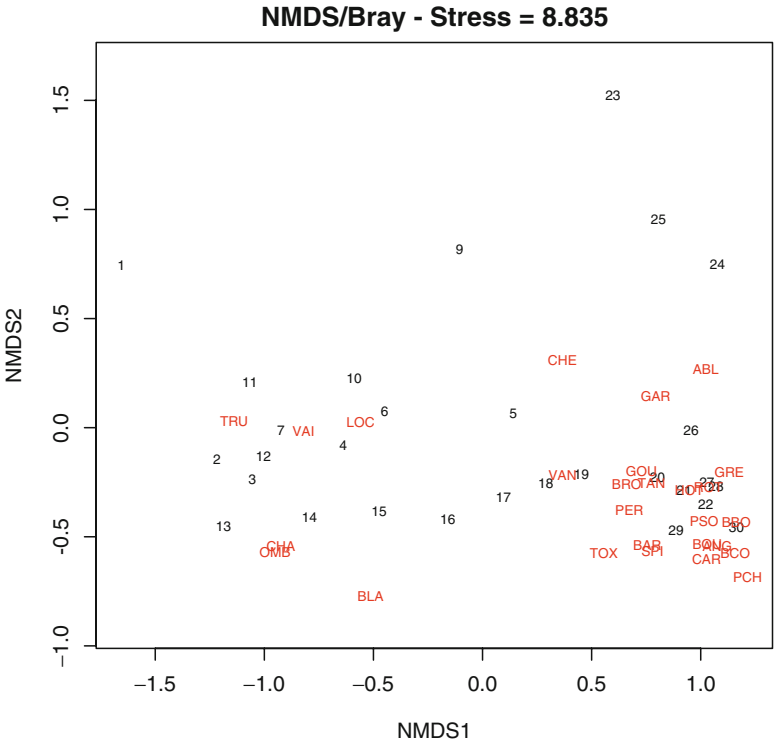

Fig. 5.10 NMDS biplot of a Bray–Curtis dissimilarity matrix of the fish abundance data. Species added using weighted averages. The relationships between species and sites are interpreted as in CA

A useful way to assess the appropriateness of an NMDS result is to compare, in a *Shepard diagram,* the distances among objects in the ordination plot with the original distances. In addition, the goodness-of-fit of the ordination is measured as the R^2

of either a linear or a non-linear regression of the NMDS distances on the original ones. All this is possible in **R** using **vegan**'s functions **stressplot()** and **goodness()**(Fig. 5.11):

```
# Shepard plot and goodness of fit
# ********************************

par(mfrow=c(1,2))
stressplot(spe.nmds, main="Shepard plot")
gof = goodness(spe.nmds)
plot(spe.nmds, type="t", main="Goodness of fit")
points(spe.nmds, display="sites", cex=gof*2)
```

Hint *See how the goodness-of-fit of individual sites is represented using the results of the* **goodness()** *analysis by way of the* `cex` *argument of the* **points()** *function. Poorly fitted sites have larger bubbles.*

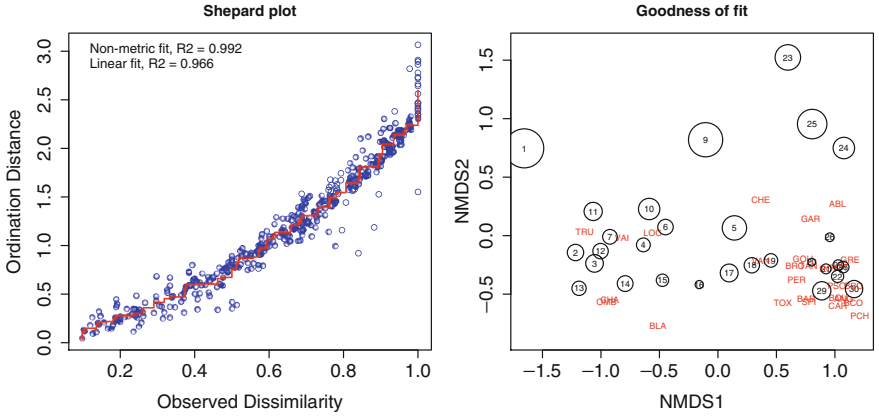

Fig. 5.11 Shepard and goodness-of-fit diagrams of the NMDS result presented in Fig. 5.10

As with the other ordination methods, it is possible to add information coming from a clustering result to an NMDS ordination plot. For instance, compute a Ward clustering of the Bray–Curtis matrix, extract four groups and colourize the sites according to them:

```
# Add colours from a clustering result to an NMDS plot
# ***************************************************

# Ward clustering of Bray-Curtis dissimilarity matrix
# and extraction of four groups
spe.bray.ward <- hclust(spe.bray, "ward")
spe.bw.groups <- cutree(spe.bray.ward, k=4)
grp.lev <- levels(factor(spe.bw.groups))

# Combination with NMDS result
sit.sc <- scores(spe.nmds)
p <- ordiplot(sit.sc, type="n",
 main="NMDS/Bray + clusters Ward/Bray")
for (i in 1:length(grp.lev)) {
 points(sit.sc[spe.bw.groups==i,], pch=(14+i), cex=2, col=i+1)
 }
text(sit.sc, row.names(spe), pos=4, cex=0.7)
# Add the dendrogram
ordicluster(p, spe.bray.ward, col="dark grey")
legend(locator(1), paste("Group",c(1:length(grp.lev))),
 pch=14+c(1:length(grp.lev)), col=1+c(1:length(grp.lev)),
 pt.cex=2)
```

5.7 Handwritten Ordination Function

To conclude this chapter, let us dive into the bowels of an ordination method…

The Code It Yourself corner #3

Legendre and Legendre (1998) provide the algebra necessary to program the ordination methods seen above directly "from scratch", i.e., using the matrix algebra functions implemented in ***R.*** *While it is not the purpose of this book to do this for all methods, we provide an example that could stimulate the interest among users. After all, numerical ecology is a living science, and anybody could one day stumble upon a situation for which no ready-made function exists. The researcher may then be interested in developing his or her own method and write the functions to implement it.*

The example below is based on the algebraic development presented in Legendre and Legendre (1998), Section 9.1. It is presented in the form of a function, the kind that any user could write for her or his own use. The steps are the following for a PCA on a covariance matrix. To obtain a PCA on a correlation matrix, one has to standardize the data before using the function, or implement this as an option in the function itself.

1. *Compute the covariance matrix S of the original or centred data matrix.*
2. *Compute the eigenvectors and eigenvalues of S (eqs. 9.1 and 9.2).*

3. *Extract matrix U of the eigenvectors and compute matrix F of the principal components (eq. 9.4) for the scaling 1 biplot.*
4. *Computation of matrices U2 and G for the scaling 2 biplot.*
5. *Output of the results.*

```
# A simple function to perform PCA

   myPCA <- function(Y) {
   Y.mat <- as.matrix(Y)
   object.names <- rownames(Y)
   var.names <- colnames(Y)

# Centre the data (will be needed to compute F)
   Y.cent <- scale(Y.mat, center=TRUE, scale=FALSE)

# Covariance matrix S
   Y.cov <- cov(Y.cent)

# Eigenvectors and eigenvalues of S (eq. 9.1 and 9.2)
   Y.eig <- eigen(Y.cov)

# Copy the eigenvectors to matrix U (used to represent
# variables in scaling 1 biplots)
   U <- Y.eig$vectors
   rownames(U) <- var.names

# Compute matrix F (used to represent objects
# in scaling 1 plots)
   F <- Y.cent%*%U       # eq. 9.4
   rownames(F) <- object.names

# Compute matrix U2 (to represent variables in scaling 2
# plots) Legendre and Legendre 1998, unnumbered equation p. 397)
   U2 <- U%*%diag(Y.eig$values^0.5)
   rownames(U2) <- var.names

# Compute matrix G (to represent objects in scaling 2 plots)
# Legendre and Legendre 1998, unnumbered equation p. 404)
   G <- F%*%diag(Y.eig$values^0.5)
   rownames(G) <- object.names

# Output of a list containing all the results
   result <- list(Y.eig$values,U,F,U2,G)
   names(result) <- c("eigenvalues","U", "F", "U2", "G")
   result
   }
```

This function should give exactly the same results as the function **`PCA()`** *used in Section 5.3.5. Now try it on the Hellinger-transformed fish species data and compare the results.*

To make your function active, either **save it in a file** *(called for instance* **`myPCA.R`***) and source it, or (less elegant) copy the whole code directly into your* ***R*** *console.*

```
# PCA on fish species using hand-written function
fish.PCA <- myPCA(spe.h)
summary(fish.PCA)
# Eigenvalues
fish.PCA$eigenvalues
# Eigenvalues expressed as percentages
round(100*fish.PCA$eigenvalues/sum(fish.PCA$eigenvalues),2)
# Alternate computation of total variation (denominator)
round(100*fish.PCA$eigenvalues/sum(diag(cov(spe.h))),2)
# Cumulative eigenvalues expressed as percentages
round(cumsum(100*fish.PCA$eigenvalues/sum(fish.PCA$eigenvalues))
,2)

# Biplots
par(mfrow=c(1,2))
# Scaling 1 biplot
biplot(fish.PCA$F, fish.PCA$U)
# Scaling 2 biplot
biplot(fish.PCA$G, fish.PCA$U2)
```

Now you could plot other pairs of axes, for instance axes 1 and 3. Compared to CA or PCoA, the code above is rather straightforward. But nothing prevents you from trying to program another method. You can also display the code of the **`CA()`** *and* **`pcoa()`** *functions and interpret them with the manual in hand.*