Processing Ecological Data in R With the mefa Package

Péter Sólymos
University of Alberta

Abstract

The **mefa** is an R package for multivariate data handling in ecology and biogeography. It provides object classes to represent the data coded by samples, taxa and segments (i.e. subpopulations, repeated measures). Supports easy processing of the data along with relational data tables for samples and taxa. An object of class 'mefa' is a project specific compendium of the dataset and can be easily used in further analyses. Methods are provided for extraction, aggregation, conversion, plotting, summary and reporting of 'mefa' objects. Reports can be generated in plain text or LaTeX format. This paper presents worked examples on a variety of ecological analyses.

Keywords: biodiversity, biogeography, data manipulation, ecology, multivariate methods, R.

1. Introduction

Fortunately, many packages are available for the analysis of ecological data in R (R Development Core Team (2008)), e.g. the **ade4** (Thioulouse and Dray (2007)), **labdsv** (Roberts (2007)) and **vegan** (Oksanen *et al.* (2008)) packages among other more standard statistical packages e.g. **MASS** (Venables and Ripley (2002)), **stats** (R Development Core Team (2008)). These, however, often require the multivariate data in the form of a matrix or data frame.

Extensive ecological data sets (with observations of multiple taxa at multiple locations) are not stored in cross tabulated format because most biodiversity data sets are sparse with matrix fill often lower than 30 percent. The conversion between data formats of biodiversity databases and ecological packages might require substantial work. Of course there are many possibilities for general data manupulation in R (Spector (2008)). The **Hmisc** (Harrell Jr and with contributions from many other users (2007)), **reshape** (Wickham (2007)), **simba** (Jurasinski (2007)) and **labdsv** (Roberts (2007)) packages contain functions for converting ecological data in database formats into crosstabulated data matrices, and vice versa.

When the problems are more complex (e.g. a survey spanning across multiple spatial or temporal scales), results are stored in several related data tables. Most multivariate methods require a matrix, whereas response modeling usually require data frame as input. The simultaneous manipulation and checking of the community data matrix and related data frames can be time consuming with the standard tools.

The aim of the **mefa** R package is to provide standardized computational environment for specialist work in ecology and biogeography by bridging the gap between the data and the analysis, and reducing the time spent with data preprocessing. It provides object classes and

methods for convenient manipulation of related data tables and can be used for generating reports in plain text or LaTeX format.

The package name **mefa** is a short for *metafaunistics* indicating that data processing is a critical step prior to data analysis. The *faunistics* part refers to the study of the fauna of some territory or area, while the *meta* part refers to the procedures (data processing and analysis) beyond the data collection part of the scientific endeavor. Of course, the package intend to be more general than covering only faunistics. It can be useful when dealing with almost all kinds of organisms that can be counted, including microorganisms, fungi and plants as well.

Compared to previous versions (< 2.0, Sólymos (2008)), the package has been extensively rewritten to enhance efficiency and speed, and with a focus on methods, but the data model remained the same. Stable version of the package is available via the Comprehensive R Archive Network (CRAN, `http://cran.r-project.org`), developmental version is available at the R-Forge (`http://r-forge.r-project.org/`).

In this paper I outline the motivation and the general idea behind the package, and I describe the structure of the object classes. A real-world data set is used to demonstrate the methods in the package and the use of objects in further data analysis.

2. Example Dataset

The "Dolina 2007" survey was aimed to discover the soil and litter dwelling macroinvertebrates (land snails, and terrestrial isopods) with special respect to microhabitat characteristics. We have surveyed dolines (sinkholes, karstic depressions) at the Alsó-hegy plateau of the Aggtelek National Park, North Hungary, August, 2007. These dolines were mainly covered by hornbeam and beech, and were 0.5–2 acres in extent. The dolines contained keystone habitat elements i.e. large pieces of coarse woody debris and different rock formations.

We collected invertebrates from four different microhabitats (**L**itter, live **W**ood, **D**ead wood as an equivalent for coarse woody debris, and **R**ock[1]) with at least three replicates per microhabitat per doline. We collected 1 L of litter per replicate (`"quadrat"` method for short) and employed a time restricted search (`"time"` method, five minutes per replicate) in a 1 m radius around the litter sample location. Litter samples were taken from the bottom of live trees, dead trees and rocks (Kemencei *et al.* (2008), Vilisics *et al.* (2008)).

Here I will use the data of land snails from the first dolina only, that is provided with the package (dataset `dol.count`). Samples are named after the microhabitat, the applied method and a number for replication. Snails were identified to species and categorized according to extent of shell decay. Four letter short names were used to refer to species. Live animals and fresh shells constituted the `"fresh"` group. Whitened, decayed and broken shells constituted the `"broken"` group.

A data frame containing microsite characteristics of the samples and the applied methods is provided in the dataset `dol.samp`. Characteristics (nomenclature, taxonomy, adult shell dimension) of the snail species are in the dataset `dol.taxa` and is based on Kerney *et al.* (1983). The full dataset along with the code how it was extracted is available on the Dataverse

[1]Boldface letters indicate the notation used for first characters of sample names. Second characters stand for sampling method, and third is the replicate number. The first quadrat and timed samples from the rock microhabitat are `RQ1` and `RT1`, respectively.

Network (Sólymos and Kemencei (2008)).

3. Motivation

Load the package and the example data set (showing only the first 16 rows out of 297)[2]:

```
R> library(mefa)

This is mefa 2.1-0

R> data(dol.count)
R> dol.count[1:16, ]

   samp        taxa count        segm
1   LQ1        aacu     1       fresh
2   LQ1        amin     1      broken
3   LQ1        amin     2       fresh
4   LQ1        dper     1      broken
5   LQ1        ffau     1      broken
6   LQ1        ppyg    13       fresh
7   LQ1        ppyg    15      broken
8   LT1 zero.pseudo     0 zero.pseudo
9   LQ2        aacu     1       fresh
10  LQ2        amin     1      broken
11  LQ2        dper     1       fresh
12  LQ2        ppyg     5       fresh
13  LQ2        ppyg     8      broken
14  LQ2        pvic     1      broken
15  LT2        amin     1       fresh
16  LT2        pvic     1      broken
```

This is a typical format for biodiversity datasets, where the `samp` column represents the observational units (samples). In planned ecological field experiments and surveys these units are expected to be comparable in terms of sampling effort. However, when data came from unplanned field observations, samples are often not comparable but refer to an observation, or set of individuals collected from a given location, by a given person in a given time.

The `taxa` column refers to the taxonomic identity of the individuals found in a given sample. The taxonomic resolution might vary due to expert knowledge and development stage of the individuals. When an actual sample did not contain individuals, it is convenient to refer to this situation by a pseudo-species (here, it is named as `"zero.pseudo"`) indicating that 0 individuals were found in the sample coded as `LT1` (first replicate from the litter microhabitat collected by timed search).

The `count` column contains the outcome of the field experiment, the number of individuals of a given taxa (i.e. a species) that were found in a given sample. It is zero when the sample

[2]The package demo (`demo(mefa, package = "mefa")`) and vignette (`vignette("mefa", package = "mefa")`) can help to go through the procedures presented in this paper.

contained no individuals. This notation is only necessary, if we are interested in zero samples (i.e. indication of very low abundances). Functions in the package accepts non-integer values, too.

The `segm` column is used to distinguish segments (subpopulations) within individuals of the same species (nested into samples and species). Here we use `"fresh"` and `"broken"` segments to classify individuals based on shell decay. Other common examples for such segments are when distinguishing between males and females, different life stages or age classes. But it can also be used to identify subsets of the data, e.g. in case of a repeated measures experiment, when samples are nested within subsequent sampling period. As an example, data from museum collections can be used in this way to describe data accumulation trends through time (Sólymos and Fehér (2008)).

This format is ideal for the storage of the data but not adequate for data analysis. Prior to analysis, we have to crosstabulate the data to get a matrix filled with numeric values and with rows as samples and columns as taxa. Besides the functions `table` (in package **base**, for factors, R Development Core Team (2008)), there are some other functions in R to do this. For two-way crosstabulation of such data, there is the function `reshape` in the **reshape** package (Wickham (2007)), the `mama` function in the **simba** package (Jurasinski (2007)) based on the `reshape` function, and the `matrify` function in the **labdsv** package (Roberts (2007)).

Complications may arise, however, when we are dealing with three-way crosstabulation of the data (see `xtabs` in package **stats**, with formula interface, R Development Core Team (2008)) and complex data structures, like the results of a hierarchical sampling design. We have to aggregate the samples into higher level units or the taxa into taxonomic or functional groups. And we might want to extract a subset of the crosstabulated data along with subsetted tables for samples and taxa at the same time. The way to do it with the **mefa** is shown in the next sections.

4. Object Classes

The structure of the above data set with the four column is basically the prototype of an object of class 'stcs'. This is the primary format for database style data sets in the **mefa** package. The four letter acronym comes from the first letters of the column names (**s**amples, **t**axa, **c**ounts, **s**egments). To convert our example into an 'stcs' object we do:

```
R> x1 <- stcs(dol.count)
R> str(x1)
```

```
Classes 'stcs' and 'data.frame':        297 obs. of  4 variables:
 $ samp : Factor w/ 24 levels "DQ1","DQ2","DQ3",..: 1 1 1 1 1 1 1 2 2 2 ...
 $ taxa : Factor w/ 29 levels "aacu","amin",..: 2 2 9 11 14 28 28 2 2 3 ...
 $ count: num  2 2 2 2 1 2 7 1 3 1 ...
 $ segm : Factor w/ 3 levels "fresh","broken",..: 1 2 1 1 1 2 1 2 1 2 ...
 - attr(*, "call")= language stcs(dframe = dol.count)
 - attr(*, "expand")= logi FALSE
 - attr(*, "zero.count")= logi TRUE
 - attr(*, "zero.pseudo")= chr "zero.pseudo"
```

```
R> unique(x1$count)

 [1]  2  1  7  3 13  4  5 15  8 10 16 17 25  0  6  9 19 12 22 26
```

The objects inherits from the data frame class, thus methods available for data frames apply for 'stcs' objects as well. But there are four additional attributes. The `call` attribute is the function call. The `expand` attribute refers to the count column. If the maximum of counts values is one, then each row represent one individual, and the `expand` attribute is `TRUE`. We can achieve this by the `expand` argument in the function call:

```
R> x2 <- stcs(dol.count, expand = TRUE)
R> str(x2)

Classes 'stcs' and 'data.frame':        732 obs. of  4 variables:
 $ samp : Factor w/ 24 levels "DQ1","DQ2","DQ3",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ taxa : Factor w/ 29 levels "aacu","amin",..: 2 2 2 2 9 9 11 11 14 28 ...
 $ count: num  1 1 1 1 1 1 1 1 1 1 ...
 $ segm : Factor w/ 3 levels "broken","fresh",..: 1 1 2 2 2 2 2 2 2 1 ...
 - attr(*, "call")= language stcs(dframe = dol.count, expand = TRUE)
 - attr(*, "expand")= logi TRUE
 - attr(*, "zero.count")= logi TRUE
 - attr(*, "zero.pseudo")= chr "zero.pseudo"

R> sum(x2$count)

[1] 731

R> unique(x2$count)

[1] 1 0
```

As an effect, the data frame is expanded and the count column contain 1 and 0 values. As we see, now the result contains 732 rows instead of 297. The 732 is the sum of the counts (one row for each count) plus one row for the empty sample. This expansion can be done with any data frame by the `inflate` function of the package. If values are non-integers, this cannot be applied.

The `zero.count` attribute refers to the presence or absence of empty samples in the data set. If empty samples are present, taxa and segment indices of the corresponding rows are set to the value stored in the `zero.pseudo` attribute. This can be set by the `zero.pseudo` argument in the function call. Count values of this pseudo species should be 0, the segment value is indifferent (but it can be set as the second element of a character vector supplied as the `zero.pseudo` argument).

Empty samples can be dropped by the `drop.zero = TRUE` argument (some multivariate methods, i.e. dissimilarity or diversity indices, does not allow empty rows to be present):

```
R> x3 <- stcs(dol.count, drop.zero = TRUE)
R> str(x3)
```

```
Classes 'stcs' and 'data.frame':        296 obs. of  4 variables:
 $ samp : Factor w/ 23 levels "DQ1","DQ2","DQ3",..: 7 7 7 7 7 7 7 8 8 8 ...
 $ taxa : Factor w/ 28 levels "aacu","amin",..: 1 2 2 11 15 23 23 1 2 11 ...
 $ count: num  1 1 2 1 1 13 15 1 1 1 ...
 $ segm : Factor w/ 2 levels "fresh","broken": 1 2 1 2 2 1 2 1 2 1 ...
 - attr(*, "call")= language stcs(dframe = dol.count, drop.zero = TRUE)
 - attr(*, "expand")= logi FALSE
 - attr(*, "zero.count")= logi FALSE
 - attr(*, "zero.pseudo")= chr "not.defined"
```

```
R> unique(x3$count)

 [1]  1  2 13 15  5  8  3 10 16 17 25  7  4 12 22 26  6  9 19
```

Now the `zero.count` attribute is `FALSE` and the `zero.count` attribute is `"not.defined"` as a result.

The number of columns in the input data frame may vary from two to four. If two columns are provided, it is assumed that the first column contains sample, while the second taxa names. If three columns are provided, the first two are treated as sample and taxa names, while the third is treated as count if numeric, and segment if character or factor. If four columns are provided, those are assumed to be in the samples, taxa, counts, segments order.

To crosstabulate the data of a 'stcs' object, use:

```
R> m1 <- mefa(x1)
R> m1

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 (non nested) segments:
        fresh, broken,
$samp - table for samples not provided,
$taxa - table for taxa not provided.
```

```
R> m1$xtab["LT1", ]

aacu amin apur bbip bcan ccer clam cort ctri dbre dper druf eful estr ffau hobv
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
iiso mbor mobs odol ogla pinc ppyg pvic tuni vcos vicr vidi
   0    0    0    0    0    0    0    0    0    0    0    0
```

The `mefa` function returns an object of class 'mefa'. The print method for the 'mefa' objects returns basic informations on the dimensions of the data. The `$xtab` list element contains the cross tabulated data. The pseudo species has been removed, thus the row for sample `LT1` is empty. Row and column ordering follows the original internal coding of factors in the 'stcs' object. The `$segm` is a list with length equal to the number of segments, and contains the matrices for each segment, with dimensions being the same as for `$xtab`:

```
R> str(m1$xtab)

 num [1:24, 1:28] 0 0 0 0 0 0 1 1 0 0 ...
 - attr(*, "dimnames")=List of 2
  ..$ samp: chr [1:24] "DQ1" "DQ2" "DQ3" "DT1" ...
  ..$ taxa: chr [1:28] "aacu" "amin" "apur" "bbip" ...

R> str(m1$segm)

List of 2
 $ fresh : num [1:24, 1:28] 0 0 0 0 0 0 1 1 0 0 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ samp: chr [1:24] "DQ1" "DQ2" "DQ3" "DT1" ...
  .. ..$ taxa: chr [1:28] "aacu" "amin" "apur" "bbip" ...
 $ broken: num [1:24, 1:28] 0 0 0 0 0 0 0 0 0 0 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ samp: chr [1:24] "DQ1" "DQ2" "DQ3" "DT1" ...
  .. ..$ taxa: chr [1:28] "aacu" "amin" "apur" "bbip" ...
```

If segments represent e.g. successive sampling periods (years, decades) it can be useful to use the `nested = TRUE` argument to inspect the data accumulation over time. In this case, values in subsequent segments are added up and segment names indicate this also:

```
R> mefa(x1, nested = TRUE)

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 nested segments:
        fresh, fresh-broken,
$samp - table for samples not provided,
$taxa - table for taxa not provided.
```

If we do not want to use segments, use the `segment = FALSE` argument:

```
R> mefa(x1, segment = FALSE)

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 1 (all inclusive) segment,
$samp - table for samples not provided,
$taxa - table for taxa not provided.
```

If the input object for the `mefa` function is a matrix or data frame, segments cannot be specified directly. Segments can, however, be defined indirectly through a 'mefa'→'stcs'→'mefa' loop by the `melt` method (discussed in the examples later).

Besides the count data, we also have a data frame for the samples. Here are the covariates for the 24 samples (microhabitat type and sampling method):

```
R> data(dol.samp)
R> str(dol.samp)

'data.frame':        24 obs. of  2 variables:
 $ microhab: Factor w/ 4 levels "dead.wood","litter",..: 2 2 2 2 2 1 1 1 1 1 ...
 $ method  : Factor w/ 2 levels "time","quadrat": 2 2 1 2 1 2 1 2 1 2 ...
```

And we also have a table containing variables related to the species (three factors, one numeric):

```
R> data(dol.taxa)
R> str(dol.taxa)

'data.frame':        121 obs. of  4 variables:
 $ species: Factor w/ 121 levels "Acanthinula aculeata",..: 3 2 81 82 16 17 92 93 72 32 ..
 $ author : Factor w/ 65 levels "(Alder, 1830)",..: 26 53 42 19 44 47 31 14 47 43 ...
 $ familia: Factor w/ 21 levels "Aciculidae","Bradybaenidae",..: 1 1 14 14 6 6 17 17 17 5
 $ size   : num  3.4 5.5 16 16 2.2 2.3 17 8 12 7.5 ...
```

These two tables, or either only one of them can easily be combined with the crosstabulation:

```
R> mefa(x1, samp = dol.samp)

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 (non nested) segments:
        fresh, broken,
$samp - table for samples provided (2 variables),
$taxa - table for taxa not provided.

R> mefa(x1, taxa = dol.taxa)

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 (non nested) segments:
        fresh, broken,
$samp - table for samples not provided,
$taxa - table for taxa provided (4 variables).

R> m2 <- mefa(x1, samp = dol.samp, taxa = dol.taxa)
R> m2

An object of class 'mefa' containing
```

```
$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 (non nested) segments:
        fresh, broken,
$samp - table for samples provided (2 variables),
$taxa - table for taxa provided (4 variables).

R> str(m2$xtab)

 num [1:24, 1:28] 0 0 0 0 0 0 1 1 0 0 ...
 - attr(*, "dimnames")=List of 2
  ..$ samp: chr [1:24] "DQ1" "DQ2" "DQ3" "DT1" ...
  ..$ taxa: chr [1:28] "aacu" "amin" "apur" "bbip" ...

R> str(m2$samp)

'data.frame':        24 obs. of  2 variables:
 $ microhab: Factor w/ 4 levels "dead.wood","litter",..: 1 1 1 1 1 1 2 2 2 2 ...
 $ method  : Factor w/ 2 levels "time","quadrat": 2 2 2 1 1 1 2 2 2 1 ...

R> str(m2$taxa)

'data.frame':        28 obs. of  4 variables:
 $ species: Factor w/ 121 levels "Acanthinula aculeata",..: 1 4 6 10 14 35 37 38 17 40 ...
 $ author : Factor w/ 65 levels "(Alder, 1830)",..: 42 57 1 41 27 49 41 38 47 15 ...
 $ familia: Factor w/ 21 levels "Aciculidae","Bradybaenidae",..: 18 21 21 4 4 4 4 4 6 21 .
 $ size   : num  2 9 5 18 18 18 17 13 2.3 NA ...
```

The tables are subsetted and ordered according to the row and column names in the crosstabulation. If names do not match, an error message is produced. This can be the case e.g. if the input object for the `mefa` function is a matrix without row or column names. It is not necessary to have names, only if tables for samples and taxa are provided.

In those cases, when the tables for samples and taxa do not contain rows for all the crosstabulated ones, the `xtab.fixed = FALSE` argument can be useful. This results in a 'mefa' object containing samples and taxa that are common in the names (natural join based on intersect of dimnames):

```
R> m2.sub <- mefa(x1, dol.samp[-c(1:5), ], dol.taxa[-c(1:80), ],
+     xtab.fixed = FALSE)
R> m2.sub

An object of class 'mefa' containing

$xtab - 228 individuals of 9 taxa in 19 samples,
$segm - 2 (non nested) segments:
        fresh, broken,
$samp - table for samples provided (2 variables),
$taxa - table for taxa provided (4 variables).
```

```
R> str(m2.sub$xtab)

 num [1:19, 1:9] 4 4 3 1 2 3 0 3 7 13 ...
 - attr(*, "dimnames")=List of 2
  ..$ samp: chr [1:19] "DQ1" "DQ2" "DQ3" "DT1" ...
  ..$ taxa: chr [1:9] "amin" "apur" "estr" "ffau" ...

R> str(m2.sub$samp)

'data.frame':        19 obs. of  2 variables:
 $ microhab: Factor w/ 4 levels "dead.wood","litter",..: 1 1 1 1 1 1 2 4 4 4 ...
 $ method  : Factor w/ 2 levels "time","quadrat": 2 2 2 1 1 1 1 2 2 2 ...

R> str(m2.sub$taxa)

'data.frame':        9 obs. of  4 variables:
 $ species: Factor w/ 121 levels "Acanthinula aculeata",..: 4 6 48 24 51 57 76 79 99
 $ author : Factor w/ 65 levels "(Alder, 1830)",..: 57 1 13 48 42 55 42 53 15
 $ familia: Factor w/ 21 levels "Aciculidae","Bradybaenidae",..: 21 21 12 12 12 12 12 12 1
 $ size   : num  9 5 18 20 15 11 16 15.5 8
```

As we see, the count data matrix is also subsetted, according to the related tables of samples and taxa.

An object of class 'mefa' is a list of length five. First element is the function call (`$call`), second is a matrix with the cross tabulated data (`$xtab`). These two are always present, the remaining three depend on the availability of data and the aims of the study. The third element contains a list of matrices for segments (`$segm`), the fourth and fifth contains data frames with data for samples (`$samp`) and taxa (`$taxa`), respectively. Elements, except for the count data matrix, can be missing. The general structure of an objetc of class 'mefa' follows a relational data model, where dimnames attributes of the main count data matrix, the segment data matrices and the samples/taxa data tables for the rows and columns are identical. The print method for the 'mefa' objects shows the names of these elements to help memorizing the notation. Figure 1 depicting the object structure can be called as:

```
R> mefalogo()
```

5. S3 Methods

5.1. Summary and Graphical Display

Dimension and dimension names can be retrieved by the `dim` and `dimnames` methods:

```
R> dim(m2)
```

```
[1] 24 28  2
```

```
R> dimnames(m2)
```

```
$samp
 [1] "DQ1" "DQ2" "DQ3" "DT1" "DT2" "DT3" "LQ1" "LQ2" "LQ3" "LT1" "LT2" "LT3"
[13] "RQ1" "RQ2" "RQ3" "RT1" "RT2" "RT3" "WQ1" "WQ2" "WQ3" "WT1" "WT2" "WT3"

$taxa
 [1] "aacu" "amin" "apur" "bbip" "bcan" "ccer" "clam" "cort" "ctri" "dbre"
[11] "dper" "druf" "eful" "estr" "ffau" "hobv" "iiso" "mbor" "mobs" "odol"
[21] "ogla" "pinc" "ppyg" "pvic" "tuni" "vcos" "vicr" "vidi"

$segm
[1] "fresh"  "broken"
```

To get a summary of the data, use the `summary` method:

```
R> summary(m2)
```

```
Call:
mefa(xtab = x1, samp = dol.samp, taxa = dol.taxa)

                   Summary
Total sum              731
Matrix fill (%)         26
Number of samples       24
Number of taxa          28
Number of segments       2

Segments:
fresh, broken

        s.rich s.abu  t.occ  t.abu
Min.     0.000  0.00  1.000   1.00
1st Qu.  5.000  9.75  2.000   4.00
Median   7.000 20.50  4.500  10.50
Mean     7.333 30.46  6.286  26.11
3rd Qu. 10.000 44.75 10.000  27.75
Max.    16.000 97.00 23.000 245.00
```

This returns basic characteristics of the data matrix. There are observations of 731 land snail individuals in the object and 26 % of the matrix cells are non zero. There is also a list with segment levels, and summaries of the marginal tables derived from the count data matrix. Four vectors are based on marginal tables, the number of taxa (species) and individuals in the samples (`s.rich` and `s.abu`), and the number of occupied samples and individuals per species (`t.occ` and `t.abu`). The characteristics shown in the summary can be extracted according to their names:

```
R> names(summary(m2))

 [1] "s.rich"     "s.abu"      "t.occ"      "t.abu"      "ntot"
 [6] "mfill"      "nsamp"      "ntaxa"      "nsegm"      "segment"
[11] "call"       "nested"     "drop.zero"  "xtab.fixed"
```

For example to return species richness values for the samples, or matrix fill, use:

```
R> summary(m2)$s.rich

DQ1 DQ2 DQ3 DT1 DT2 DT3 LQ1 LQ2 LQ3 LT1 LT2 LT3 RQ1 RQ2 RQ3 RT1 RT2 RT3 WQ1 WQ2
  5   7   4   7  10   5   5   5   7   0   2   5  10  16  13  11  15  10   9  10
WQ3 WT1 WT2 WT3
  8   5   4   3

R> summary(m2)$mfill

[1] 0.2619048
```

To get summaries for the linked data tables, use e.g. `summary(m2$samp)` or `summary(m2$taxa)`. The first four characteristics of the samples and taxa can be plotted as barchart (`type = "bar"`; Figure 2A) or ranked in decreasing order (`type = "rank"`; Figure 2B) by the `plot` method. Values can be log transformed (`trafo = "log"`; Figure 2B) or normalized by their maxima (`trafo = "ratio"`). Here, only two cases are shown (when second argument is `1` and `4`):

```
R> opar <- par(mfrow = c(1, 2))
R> plot(m2, 1, main = "A")
R> plot(m2, 4, type = "rank", trafo = "log", main = "B")
R> par(opar)
```

The `boxplot` method is provided to retreive these statistics for each segment (Fig. 3). Again, only two cases are shown (when second argument is `2` and `3` to complement the previous figure):

```
R> opar <- par(mfrow = c(1, 2))
R> boxplot(m2, 2, main = "A")
R> boxplot(m2, 3, main = "B")
R> par(opar)
```

The `melt` method refers to the conversion of a 'mefa' object into an 'stcs' representation. The generic function is defined here, but can be found in the **reshape** package as well (transforming between wide and long formatted data), which also contains methods for data frames, arrays, matrices, tables and lists. The method for 'mefa' objects is defined in the **mefa** package. Use the sampling method column in the samples table for segments (`m2$samp$method`), and make an object of class 'stcs' as follows:

```
R> molten <- melt(m2, "method")
R> str(molten)

Classes 'stcs' and 'data.frame':        177 obs. of  4 variables:
 $ samp : Factor w/ 24 levels "DQ1","DQ2","DQ3",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ taxa : Factor w/ 29 levels "aacu","amin",..: 2 9 11 14 28 2 3 9 11 12 ...
 $ count: num  4 2 2 1 9 4 1 20 2 2 ...
 $ segm : Factor w/ 3 levels "time","quadrat",..: 2 2 2 2 2 2 2 2 2 2 ...
 - attr(*, "call")= language melt.mefa(x = m2, segm.var = "method")
 - attr(*, "expand")= logi FALSE
 - attr(*, "zero.count")= logi TRUE
 - attr(*, "zero.pseudo")= chr "zero.pseudo"
```

Then make an object of class 'mefa' using the samples and taxa tables, because those have not changed, only the segments (instead of shell decay stage, now it refers to sampling methods, see segment names in the following example):

```
R> m3 <- mefa(molten, dol.samp, dol.taxa)
R> m3

An object of class 'mefa' containing

$xtab - 731 individuals of 28 taxa in 24 samples,
$segm - 2 (non nested) segments:
        time, quadrat,
$samp - table for samples provided (2 variables),
$taxa - table for taxa provided (4 variables).
```

This kind of melting–refreezing can be useful, if e.g. segments were not defined in the original 'stcs' object, or the 'mefa' object was based on a matrix or data frame (without segments). In this way, we can create segmented representations based on groups of samples (e.g. spatial units, repeated measures) or groups of taxa (e.g. taxonomic or functional groups) to explore their differences.

Here we used melting–refreezing to explore the effects of sampling methods (note, originally shell decay stages were used as segments, and not sampling methods). We can visualize the relationship by the `image` method. This method creates a grid of colored rectangles with colors corresponding to the values in count data table. Segments can be specified as well. The rows and the columns are ordered according to their sums, being the top-left corner with the highest values. Transformations (log, order-scaled values based on quantiles, presence-absence) are also available to best represent the data in the plot. Figure 4 shows differences between the sampling methods compared to the main data matrix.

```
R> opar <- par(mfrow = c(1, 3))
R> image(m3, trafo = "log", sub = "all segments", main = "A")
R> for (i in 1:2) image(m3, segm = i, trafo = "log", sub = dimnames(m3)$segm[i],
+     main = LETTERS[i + 1])
R> par(opar)
```

Methods for objects of class 'stcs' (`summary`, `plot`, `boxplot`, `image`) are also provided for convenience, but those are merely use the crostabulated results to do the same job as the respective method for the 'mefa' objects. Other methods for 'stcs' objects rely on methods for data frames (e.g. `print`, `str`, `dim`, `dimnames`).

5.2. Extraction and Aggregation

If we want to analyze only a subset of the data, the extraction of the 'mefa' objects can be done via square brackets and by indexing the samples, taxa and segments to extract. Indexing can be numeric, or can refer to dimnames:

```
R> ex1 <- m2[1:20, 11:15, "fresh"]
R> dim(ex1)

[1] 20  5  1

R> dim(ex1$samp)

[1] 20  2

R> dim(ex1$taxa)

[1] 5 4
```

As a result, the count data and the samples/taxa tables were subsetted in the same step. If negative numeric values are given, the respective parts of the object are omitted, e.g. the first element in the second dimension can be excluded by `m2[ ,-1]`. Unused factor levels in the `$samp` and `$taxa` tables can be dropped by the argument `drop = TRUE`:

```
R> ex2 <- m2[m2$samp$method == "time"]
R> levels(ex2$samp$method)

[1] "time"    "quadrat"

R> ex3 <- m2[m2$samp$method == "time", drop = TRUE]
R> levels(ex3$samp$method)

[1] "time"
```

The data set can be easily aggregated if we want to calculate statistics, e.g. species richness, on different levels of the sampling hierarchy (like in additive diversity partitioning, Lande (1996), Crist *et al.* (2003)). Aggregation can be done by internal vectors, when the name of the variable refers to a column in the samples or taxa tables. If external vectors (that are not part of the 'mefa' object) are used, the current implementation require a class attribute (e.g. `"factor"`) to be recognized as an external object. Now we create a factor with levels `"large"` and `"small"` for snail species with adult body size larger or equal to, or smaller than 5 millimeters, respectively. Then, aggregate the object according to the internal `"microhab"` (microhabitat type) variable for samples and this external body size factor for taxa:

```
R> size.5 <- as.factor(is.na(m3$taxa$size) | m3$taxa$size < 5)
R> levels(size.5) <- c("large", "small")
R> m4 <- aggregate(m3, "microhab", size.5)
R> t(m4$xtab)

      dead.wood litter live.wood rock
large        53     33        80  210
small        50    112        99   94

R> lapply(m4$segm, t)

$time
      dead.wood litter live.wood rock
large        32     12        16  118
small         5      0         1    3

$quadrat
      dead.wood litter live.wood rock
large        21     21        64   92
small        45    112        98   91
```

From these results it is clear, that the proportion of size categories differ among sampling methods and microhabitats. We will further analyze this in the next chapters.

The methods provided for the object classes in the **mefa** package are reviewed in Table 1.

5.3. Writing Reports

If the count data as the result of a field experiment is given, along with information about the sampling locations, making a report is easy with the `report` method. This writes a text file into the specified or the working directory. Contents of this text file can be further used e.g. by copy-pasting it into a word processor. But it is more straightforward to use the `tex = TRUE` argument to write a formatted LaTeX file and use the LaTeX function `input` to include the results into the main document. A sample document on how to use the function in a **Sweave** document (Leisch (2002)) can be viewed by calling `mefadocs("SampleReport")`.

Now we write the dataset into a file after extracting the ten randomly selected species only, retaining all samples and both segments:

```
R> set.seed(1234)
R> m5 <- m2[, sample(1:dim(m2)[2], 10)]
R> report(m5, "report.tex", tex = TRUE, segment = TRUE, taxa.name = 1,
+     author.name = 2, drop.redundant = 1)
```

The meaning of the arguments used is: use the `m5` 'mefa' object to rwite a report into the file `"report.tex"` by usint LaTeX formatting. Use segments, taxa names are in the first, author names are in the second column of the taxa table. We also sa implicitly, that use all columns in the sample table same order as is, to generate sample information (this can be specified

by the `samp.var` argument not used here). Redundant levels of these sample informations are dropped here (see help page `?report.mefa` for further details). The result is shown in Figure 5 with appropriate LaTeX formatting. The formatting rules can be modified via the `tex.control` argument, but by default, it is turned off.

6. Data Analysis

6.1. Single Taxon Response Modeling

Here we use the data of the land snail species *Aegopinella minor* (`"amin"`) and the samples data table to investigate the effect of microhabitat and sampling method on the abundance of this species. The response is the `"amin"` column of the `m2$xtab` matrix, for the data argument of the Poisson GLM, we simply provide the `m2$xamp` table:

```
R> mod.amin <- glm(m2$xtab[, "amin"] ~ ., data = m2$samp, family = poisson)
R> summary(mod.amin)

Call:
glm(formula = m2$xtab[, "amin"] ~ ., family = poisson, data = m2$samp)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8037  -0.8167  -0.2117   0.3163   2.4082

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         0.6475     0.2858   2.266  0.02346 *
microhablitter     -0.3483     0.3770  -0.924  0.35559
microhablive.wood   0.3448     0.3170   1.088  0.27668
microhabrock        0.6633     0.2985   2.222  0.02630 *
methodquadrat       0.6758     0.2281   2.963  0.00305 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 49.377  on 23  degrees of freedom
Residual deviance: 28.515  on 19  degrees of freedom
AIC: 106.99

Number of Fisher Scoring iterations: 5
```

The model output indicates that the species was more frequent in dead wood and rock microhabitats than in live wood and litter, further, most of the individuals were collected by the quadrat method.

6.2. Modeling the Marginal Sum as Response

The distribution of the number of snails in the samples is skewed and likely overdispersed, thus we use a negative binomial model to analyze the relationship between land snail abundances and sample covariates. The negative binomial model is implemented in the **MASS** package (Venables and Ripley (2002)). As part of the maximum likelihood fit, a shape parameter (`theta`) is also estimated in order to model the data as gamma mixture of Poisson distributions. We use the `summary` method to get the vector of the number of individuals, and the samples table again for covariates using their interaction as well:

```
R> library(MASS)
R> mod.abu <- glm.nb(summary(m2)$s.abu ~ .^2, data = m2$samp)
R> summary(mod.abu)

Call:
glm.nb(formula = summary(m2)$s.abu ~ .^2, data = m2$samp, init.theta = 4.73055585509537,
    link = log)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.40783  -0.64639  -0.06349   0.21741   1.70730

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                        2.5123     0.3122   8.046 8.54e-16 ***
microhablitter                    -1.1260     0.5013  -2.246  0.02469 *
microhablive.wood                 -0.7777     0.4762  -1.633  0.10245
microhabrock                       1.1849     0.4198   2.823  0.00476 **
methodquadrat                      0.5787     0.4279   1.352  0.17622
microhablitter:methodquadrat       1.8267     0.6441   2.836  0.00457 **
microhablive.wood:methodquadrat    1.6756     0.6237   2.687  0.00722 **
microhabrock:methodquadrat        -0.1650     0.5812  -0.284  0.77643
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(4.7306) family taken to be 1)

    Null deviance: 94.216  on 23  degrees of freedom
Residual deviance: 26.815  on 16  degrees of freedom
AIC: 198.21

Number of Fisher Scoring iterations: 1


              Theta:  4.73
          Std. Err.:  1.71

 2 x log-likelihood:  -180.214
```

The abundance of snails in the samples was positively associated with dead wood and rock microhabitats. The interaction of sampling method and microhabitat is proved to be significant, indicating that the collecting efficiency of snails varied among microhabitats. The number of snails collected by the quadrat method was higher for the litter and live wood microhabitats, while in the dead wood microhabitat, the timed search resulted in more snails.

We can take advantage of the segments in the analysis as well. We now investigate, whether the proportion of fresh shells (including living animals) differ among microhabitats and sampling methods. First, we make a two-column matrix with the number of fresh shells per samples and the total number of shells (fresh and broken) collected per samples. Than, assuming that decay status (fresh or broken) of a single shell follows a Bernoulli process, we use this matrix in a binomial GLM (logistic regression) with number of trials equal to the total number of species per sample:

```
R> prop.fr <- cbind(summary(m2[, , "fresh"])$s.abu, summary(m2)$s.abu)
R> mod.fr <- glm(prop.fr ~ .^2, data = m2$samp, family = binomial)
R> summary(mod.fr)

Call:
glm(formula = prop.fr ~ .^2, family = binomial, data = m2$samp)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.21602  -0.54213  -0.07867   0.30911   2.13176

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -0.14518    0.24141  -0.601 0.547573
microhablitter                  -0.03714    0.49154  -0.076 0.939771
microhablive.wood               -0.49081    0.47772  -1.027 0.304230
microhabrock                    -1.09526    0.30840  -3.551 0.000383 ***
methodquadrat                   -0.35559    0.31373  -1.133 0.257036
microhablitter:methodquadrat    -0.16278    0.55175  -0.295 0.767977
microhablive.wood:methodquadrat  0.29843    0.53561   0.557 0.577402
microhabrock:methodquadrat       1.17404    0.38609   3.041 0.002359 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 33.970  on 22  degrees of freedom
Residual deviance: 13.825  on 15  degrees of freedom
AIC: 117.20

Number of Fisher Scoring iterations: 4
```

It is clear that the proportion of fresh shells is not constant among microhabitats and sampling methods. The proportion was lowest in the rock microhabitats, although the quadrat method

resulted more fresh shells from the rock microhabitats than the timed search method (due to the significant microhabitat × method interaction).

6.3. Multivariate Response Modeling

We employ a nonparametric multivariate analysis of variance to assess the effects of covariates on community composition. The method is implemented in the `adonis` function (McArdle and Anderson (2001), Anderson (2001)) of the **vegan** package (Oksanen *et al.* (2008)). The method is based on a distance matrix calculated from the community data by the Bray–Curtis index of dissimilarity, by default. Thus, the input data matrix should not contain rows with zero sum. We have to remove those samples from the count data matrix and the samples table as well. This is most easily done by the `as.mefa` method with using the argument `drop.zero = TRUE`:

```
# This requires vegan package
library(vegan)
m6 <- as.mefa(m2, drop.zero = TRUE)
m6.ado <- adonis(m6$xtab ~ .^2,
    data = m6$samp, permutations = 100)
m6.ado
```

The results revealed that 18 % of the total variation in community composition was due to sampling method (see also Figure 4), and 24.6 % was explained by microhabitats. Their interaction was also significant, explaining 15.4 % of the total variation. But the unexplained variance remained relatively high (42 %).

We can visualize this relationship by constrained (canonical) correspondence analysis (CCA). For this, use the `cca` function from the **ade4** package (Thioulouse and Dray (2007), Dray and Dufour (2007)). We base the analysis only on the matrix of the `"fresh"` segment (note that here we use it as data frame[3]) and use the samples table for environmental constraints:

```
# This requires ade4 package
library(ade4)
m2.cca <- ade4::cca(data.frame(m2$segm[["fresh"]]), m2$samp, scan = FALSE)
plot.pcaiv(m2.cca)
```

The compound results of the CCA results show environmental variables loadings and correlations (top and middle left), projection of the axes of the taxa analysis into the CCA (lower left), species scores (bottom middle), eigenvalues bar chart (bottom right), and a biplot of site scores superimposed with their predictions by environmental variables (main graph). Most of the species tend to be associated with dead wood and rock microsites, and samples separates well according to sampling method (see middle letters in the main graph).

6.4. Analyzing Multiple Subsets

The modular structure of the 'mefa' objects enables easy processing of the data in a loop. Now we use hierarchical cluster analysis based on multiple subsets of the data to assess the effects of

[3]We have to use the package name in the call as `ade4::cca`, and the `plot.pcaiv` function written in full, because the object class 'cca' is defined in both the **vegan** and `ade4` packages.

shell decay stages and sampling methods on the community composition. We first make a list with four subsets of the Dolina dataset, referring to combinations of timed search and quadrat method, and fresh and broken shells. We also aggregate the samples over microhabitats:

```
R> m.list <- list()
R> n1 <- rep(c("time", "quadrat"), each = 2)
R> n2 <- rep(c("fresh", "broken"), 2)
R> n3 <- paste(n1, n2, sep = ".")
R> for (i in 1:4) {
+     m.list[[n3[i]]] <- aggregate(m2[m2$samp$method == n1[i],
+         , n2[i]], "microhab")
+ }
```

Then we do the clustering on each elements of the list object `m.list` using Euclidean distance and Ward's method:

```
R> opar <- par(mfrow = c(2, 2))
R> for (i in 1:4) {
+     tmp <- hclust(dist(m.list[[i]]$xtab), "ward")
+     plot(tmp, main = LETTERS[i], sub = names(m.list)[i], xlab = "")
+ }
R> par(opar)
```

Figure 6 shows that the dendrograms for the fresh and broken segments of the quadrat method are congruent, while that of for the timed search are not. For the fresh shells, the dead wood communities resembled the rock community. While for the broken shells only, dead wood communities were more similar to live wood and litter communities than to the rock.

7. Conclusions

The **mefa** package provides standardized environment and a and convenient tool for ecologists and biogeographers working on biodiversity data sets in R. The object classes ('stcs' and 'mefa') and S3 methods provide a coherent framework for data preprocessing of structured data sets. Based on 'mefa' objects, reports can be generated in plain text or LaTeX format. It is presented how the objects can be directly used in further analyses for a variety of ecological problems.

8. Acknowledgements

Initial development of the **mefa** package was motivated by complex spatial sampling designs for landscape scale invertebrate surveys, especially in the "Dolina 2007" experiment. I would like to thank to Zita Kemencei, Ferenc Vilisics, Elisabeth Hornung, Roland Farkas and Zoltán Fehér for discussions and feedback, and providing their data to improve the package. The helpful reviews of two anonymous referees considerably improved the earlier version of the manuscript and the package itself. During the preparation of this manuscript, I was supported

by a postdoctoral fellowship from the NSERC and the Alberta Biodiversity Monitoring Institute (Edmonton, Alberta, Canada).

References

Anderson MJ (2001). "A new method for non-parametric multivariate analysis of variance." *Austral Ecology*, **26**, 32–46.

Crist TO, Veech JA, Gering JC, Summerville KS (2003). "Partitioning Species diversity across landscapes and regions: A hierarchical analysis of α, β, and γ diversity." *American Naturalist*, **162**, 734–743.

Dray S, Dufour AB (2007). "The ade4 Package: Implementing the Duality Diagram for Ecologists." *Journal of Statistical Software*, **22**(4), 1–20. ISSN 1548-7660. URL http://www.jstatsoft.org/v22/i04.

Harrell Jr FE, with contributions from many other users (2007). ***Hmisc**: Harrell Miscellaneous*. R package version 3.4-3, URL http://biostat.mc.vanderbilt.edu/s/Hmisc,http://biostat.mc.vanderbilt.edu/twiki/pub/Main/RS/sintro.pdf,http://biostat.mc.vanderbilt.edu/twiki/pub/Main/StatReport/summary.pdf,http://biostat.mc.vanderbilt.edu/trac/Hmisc.

Jurasinski G (2007). ***simba**: A Collection of Functions for Similarity Calculation of Binary Data*. R package version 0.2-5, URL http://www.r-project.org.

Kemencei Z, Sólymos P, Farkas R, Páll-Gergely B, Vilisics F, Hornung E (2008). "Key habitat structures shelter land snail assemblages against unfavourable environmental conditions." In J Stadler, F Schöppe, M Frenzel (eds.), "EURECO-GFOE 2008 Proceedings," p. 614. Gesellschaft für Ökologie.

Kerney MP, Cameron RAD, Jungbluth JH (1983). *Landschnecken Nord- und Mitteleuropas*. P. Parey, Hamburg–Berlin.

Lande R (1996). "Statistics and partitioning of species diversity, and similarity among multiple communities." *Oikos*, **76**, 5–13.

Leisch F (2002). "**Sweave**: Dynamic Generation of Statistical Reports Using Literate Data Analysis." In W Härdle, B Rönz (eds.), "Compstat 2002 — Proceedings in Computational Statistics," pp. 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9, URL http://www.stat.uni-muenchen.de/~leisch/Sweave.

McArdle BH, Anderson MJ (2001). "Fitting multivariate models to community data: A comment on distance-based redundancy analysis." *Ecology*, **82**, 290–297.

Oksanen J, Kindt R, Legendre P, O'Hara B, Simpson GL, Sólymos P, Stevens MHH, Wagner H (2008). ***vegan**: Community Ecology Package*. R package version 1.15-0, URL http://www.r-project.org,http://vegan.r-forge.r-project.org/.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Roberts DW (2007). ***labdsv**: Ordination and Multivariate Analysis for Ecology*. R package version 1.3-1, URL http://ecology.msu.montana.edu/labdsv/R.

Sólymos P (2008). "**mefa**: an R Package for Handling and Reporting Count Data." *Community Ecology*, **9**, 125–127.

Sólymos P, Fehér Z (2008). "The **mefa** Package: a Tool for Reproducible Data Processing in Biogeography." *IBS Newsletter*, **6**, 9–13.

Sólymos P, Kemencei Z (2008). "Methodological study data set of land snails from the Dolina 2007 project." The Dataverse Network, URL hdl:1902.1/12060.

Spector P (2008). *Data Manipulation with R*. Springer, New York. ISBN 978-0-387-74730-9.

Thioulouse J, Dray S (2007). "Interactive Multivariate Data Analysis in R with the **ade4** and **ade4TkGUI** Packages." *Journal of Statistical Software*, **22**(5), 1–14. ISSN 1548-7660. URL http://www.jstatsoft.org/v22/i05.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Vilisics F, Nagy A, Sólymos P, Farkas R, Kemencei Z, Páll-Gergely B, Kisfali M, Hornung E (2008). "Data on the Terrestrial Isopoda Fauna of the Alsó-hegy, Aggtelek National Park, Hungary." *Folia Faunistica Slovaca*, **13**, 9–12. URL http://zoology.fns.uniba.sk/ffs/interface/00053-Vilisics-et-al-2008.pdf.

Wickham H (2007). ***reshape**: Flexibly Reshape Data*. R package version 0.8.0.

Affiliation:

Péter Sólymos
Department of Mathematical and Statistical Sciences
University of Alberta
Edmonton, Alberta, T6G 2G1, Canada
E-mail: solymos@ualberta.ca
and
Department of Ecology
Faculty of Veterinary Science, Szent István University
Rottenbiller Str. 50, 1077 Budapest, Hungary

Class	Method	Description
`stcs`	`is.stcs`	evaluates if an object is of class 'stcs'.
	`as.stcs`	coerces to class 'stcs'.
	`summary`	does the same as `summary.mefa`, just for convenience.
	`plot`	does the same as `plot.mefa`, just for convenience.
	`boxplot`	does the same as `boxplot.mefa`, just for convenience.
	`image`	does the same as `image.mefa`, just for convenience.
`mefa`	`is.mefa`	evaluates if an object is of class 'mefa'.
	`as.mefa`	coerces to class 'mefa'.
	`print`	print basic characteristics of the object.
	`summary`	print basic summaries on the data.
	`plot`	graphical display of basic summaries of the data based on the main data matrix.
	`boxplot`	graphical display of basic summaries of the data based on data matrices for segments.
	`image`	graphical display of the values in the main data matrix or data matrix for a segment.
	`aggregate`	aggregate (sum) the values in the matrices (main data and segments).
	`[`	extract an object (data matrices and relational data frames) based on indexing for rows (samples) columns (taxa) and segments.
	`melt`	convert an object of class 'mefa' into an object of class 'stcs'.
	`report`	writes data from an object of class 'mefa' into a file.
	`dim`	return dimension of the 'mefa' object.
	`dimnames`	return names for rows (samples) columns (taxa) and segments in the 'mefa' object.

Table 1: Description of methods provided for the object classes.

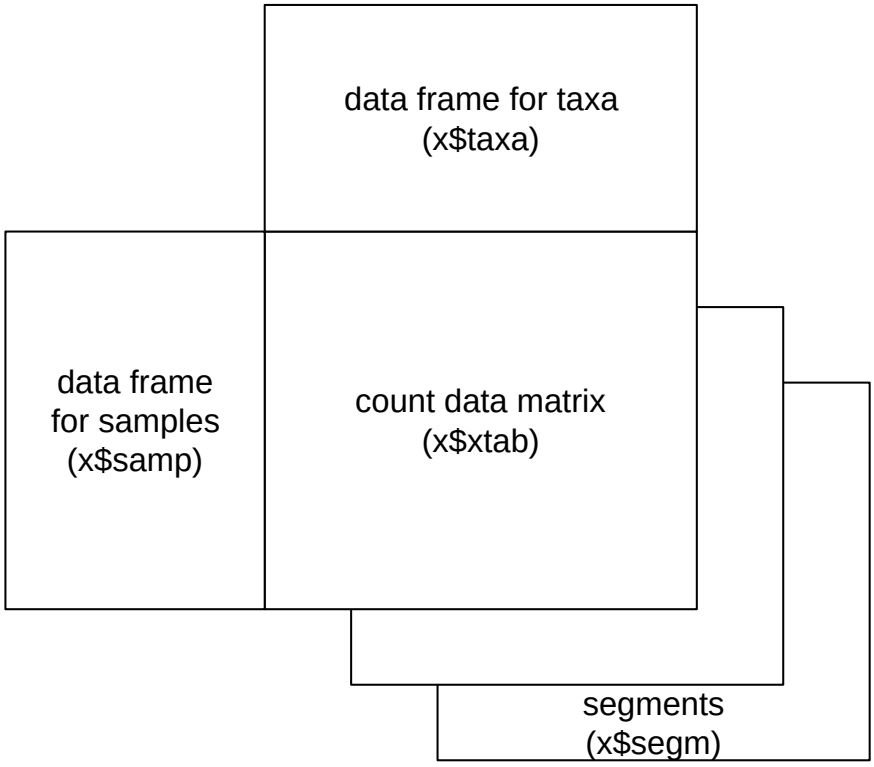

Figure 1: Representation of the relational data model in an object of class 'mefa'.

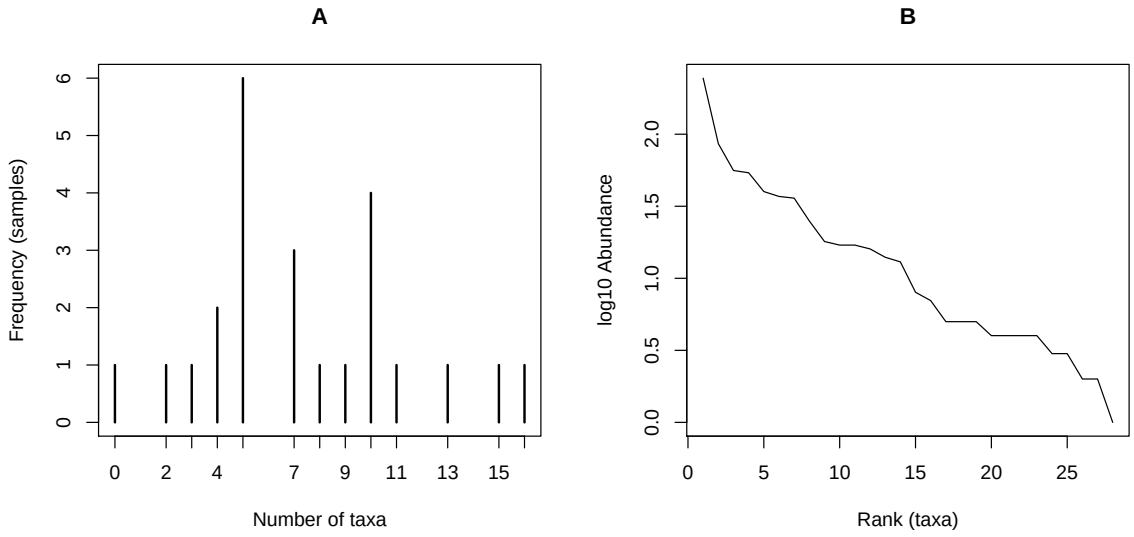

Figure 2: Plotting an object of class 'mefa': barcharts for species richness (A), and ranked log abundances for species (B).

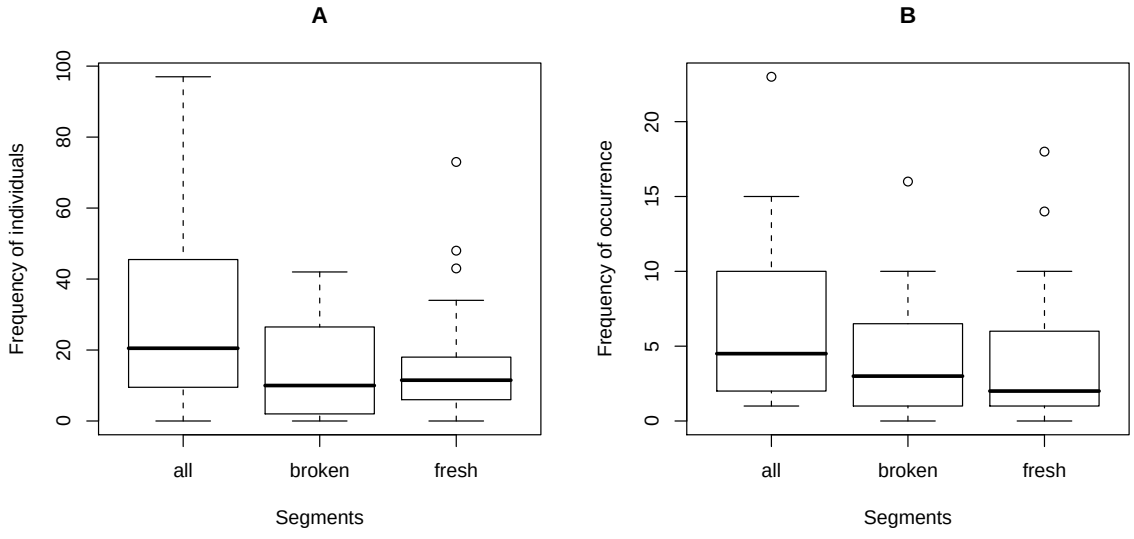

Figure 3: Boxplots to depict differences among segments in abundances for samples (A), frequency of occurrences for taxa (B).

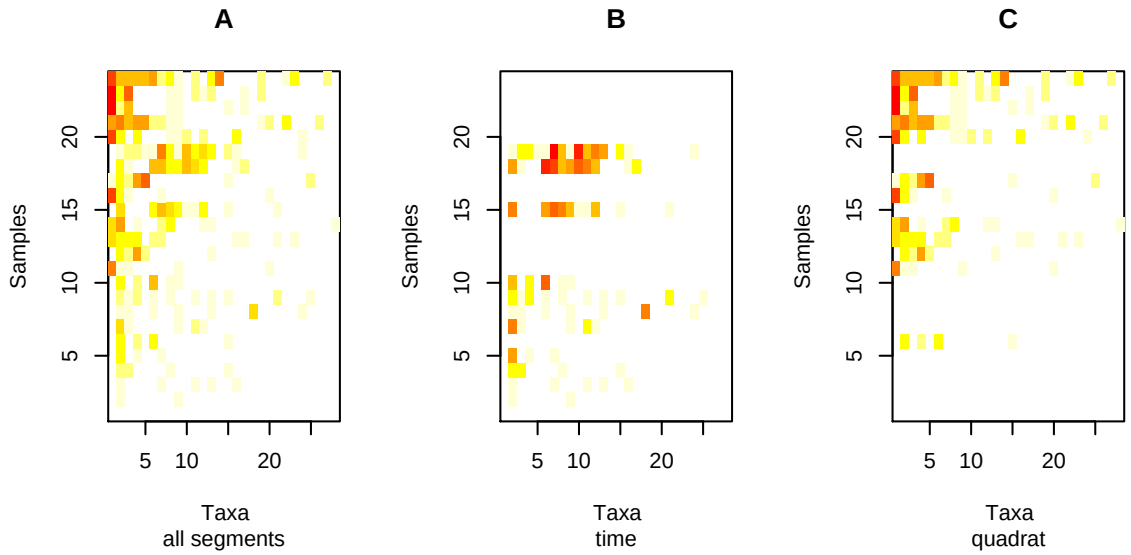

Figure 4: Levelplot representation of an object of class 'mefa'. Intensity of red coloring refers to increasing log abundances in the cells of the main data matrix (A) and the segments for the time restricted search (B), and the quadrat (C) methods.

***Acanthinula aculeata* (O. F. Muller, 1774)**

litter, quadrat (fresh: 2). **rock,** quadrat (broken: 2).

***Balea biplicata* (Montagu, 1803)**

dead.wood, time (fresh: 1, broken: 1). **litter,** quadrat (broken: 1). **live.wood:** time (fresh: 2, broken: 3); quadrat (broken: 1). **rock:** time (fresh: 2, broken: 3); quadrat (fresh: 6, broken: 5).

***Bulgarica cana* (Held, 1836)**

dead.wood, time (broken: 1). **live.wood,** quadrat (broken: 1). **rock,** time (broken: 1).

***Chilostoma faustinum* (Rossmassler, 1835)**

litter, quadrat (broken: 1). **live.wood:** time (broken: 2); quadrat (fresh: 1, broken: 1). **rock:** time (fresh: 1, broken: 5); quadrat (fresh: 6, broken: 19).

***Daudebardia brevipes* (Draparnaud, 1805)**

rock, quadrat (fresh: 3, broken: 1).

***Euomphalia strigella* (Draparnaud, 1801)**

dead.wood: time (fresh: 1); quadrat (broken: 2). **live.wood:** time (broken: 3); quadrat (broken: 1). **rock:** time (fresh: 3); quadrat (fresh: 1, broken: 3).

***Helicodonta obvoluta* (O. F. Muller, 1774)**

dead.wood: time (broken: 3); quadrat (fresh: 7, broken: 1). **rock:** time (fresh: 12, broken: 1); quadrat (fresh: 4, broken: 9).

***Isognomostoma isognomostomos* (Schroter, 1784)**

live.wood, quadrat (fresh: 2). **rock,** time (fresh: 1, broken: 14).

***Oxychilus glaber* (Rossmassler, 1838)**

rock, quadrat (fresh: 2, broken: 2).

***Vallonia costata* (O. F. Muller, 1774)**

rock, quadrat (fresh: 2).

Figure 5: Report generated from a 'mefa' object by the `report` method.

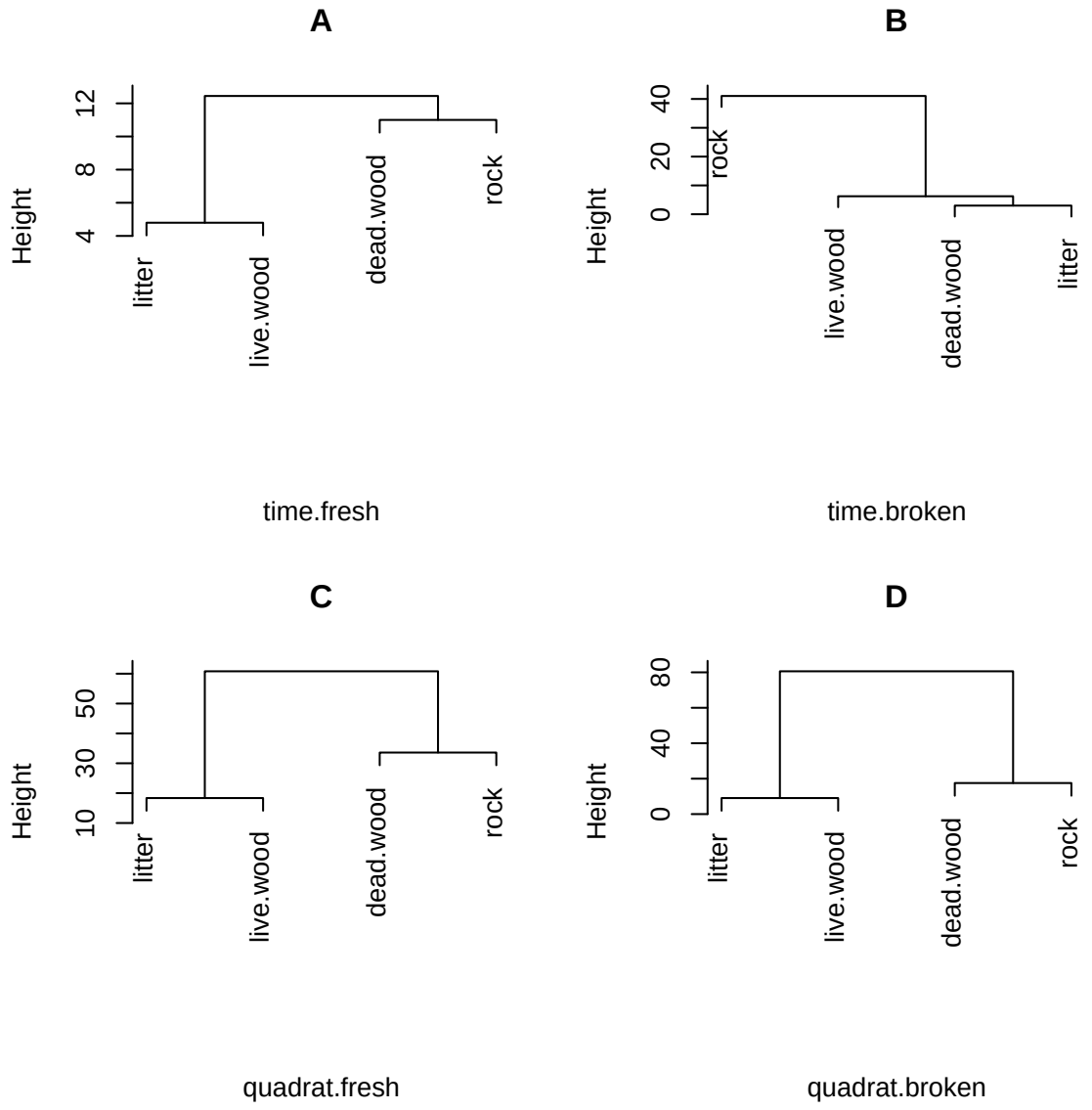

Figure 6: Hierarchical cluster analysis of multiple subsets of the Dolina dataset. Subsets are extracted to represent combinations of shell decay stages (A, C: fresh; B, D: broken) and sampling methods (A–B: timed search; C–D: quadrat method).