Using the R package “compositions”

K. Gerald van den Boogaart

Version 0.9, 1. June 2005

Abstract

“compositions” is a package for the the analysis of (e.g. chemical) compositions. Compositions are typically vectors of positive (or non negative) numbers, where often the sum is either to a constant like 100%, in case of full compositions, or meaningless, in case of subcompositions where meaningless parts have been removed from the full composition. The package and this document can be retrieved from “http://www.stat.boogaart.de/compositions”

1 License

2 Introduction to the basic classes

The package supports four different multivariate scales intended to model multivariate measurements of amounts, e.g. amounts of geochemical trace elements at different locations. The scales are represented by four different classes. In all cases it is assumed that the amounts are nonnegative. The classes differ by the assumption whether or not the total amount is meaningful for the problem and whether the geometry of the differences is a relative (log-scale) distance or a absolute (Euclidean) distance. Under some circumstances and for some datasets one or the other choice might we imperative, while in other situations two or more of the approaches might be equally valid. The four different classes are

- `"rplus"`: The total amount is meaningful and data is analyzed in real (non relative) geometry.
  This approach is mainly equivalent to analyze the data “as is” with classical multivariate methods. This approach is inappropriate for many examples of datasets of amounts due many reasons including heteroskedastisity, strong skewness and external or artificial multiplicative errors on the whole dataset.

- `"rcomp"`: The total amount is meaningless or the individual amounts are part of a whole (in equal units) and the data should be analyzed in real (non relative) geometry.
  This class represents the classical view of compositions as a part of the mathematical simplex (the set of vectors of nonnegative numbers summing to 1). Also a widely used approach, it has some traps and can lead to wrong interpretations [Chayes(1960)] [Aitchison(1986)] and has to be used with great care.

- "acomp": The total amount is meaningless or the individual amounts are part of a whole (in equal units) and the data should be analyzed in a relative geometry.
  This class is based on the logistic approach of compositional data introduced by John Aitchison [Aitchison(1982)] [Aitchison(1986)] [Aitchison(1997)] that has greatly evolved in the past years [Aitchison(2002)] [Aitchison and Greenacre(2002)] [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn] [Buccianti et al.(1999)Buccianti, Pawlowsky-Glahn, Barceló-Vidal, and Jarauta-Bragulat] [Egozcue, Pawlowsky-Glahn, Mateu-Figueras, and Barceló-Vidal(2003)Egozcue and others] [Pawlowsky-Glahn and Egozcue(2001)] [Pawlowsky-Glahn and Egozcue(2002)] [von Eynatten et al.(2002)von Eynatten, Pawlowsky-Glahn, and Egozcue]. This approach can be seen as the modern approach to compositional data analysis. However under some circumstances the approach has been criticized in favor of the more classical "rcomp" approach [Rehder and Zier(2002)] [Shurtz(2003)]. For a deeper understanding of the "acomp" approach the reader is referred to in [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn].

- "aplus": The total amount is meaningful and the data should be analyzed in relative geometry.
  This approach evolved from mixing the ideas of compositional data analysis by John Aitchison in the view of [Pawlowsky-Glahn(2003)] with the assumption of a meaningful total. It is quite near to a simple log-transform approach, which is quite common for geochemical data. However we try to stay more consistent in the concept and try to allow to analyze the original data in a log geometry rather than just log transformed, to keep the relation to the original measurements.

An auxiliary class "rmult" is used to model simple vector valued data in a classical fashion. It is mainly used internally although in theory it provides a nice interface to multivariate data analysis.

3 The generic concept of the package

The package is based on the concept that the right type of analysis is given by the intention of the user (e.g. to plot) and the type of the data (e.g. acomp). Let us illustrate this with an example dataset from the package:

```
> library(compositions)

Attaching package 'compositions':


        The following object(s) are masked from package:stats :

         cor cov dist var


        The following object(s) are masked from package:base :

         %*%

> data(SimulatedAmounts)
> comps   <- acomp(sa.lognormals)  # View data as compositions
> plot(comps)                       # produces Ternary diagrams
> amounts <- aplus(sa.lognormals)  # View data as amounts
> plot(amounts)                     # Produces Scatterplotmatrix in log-scale
```

Depending on the type of the data assigned by the constructors `acomp`, `aplus`, `rcomp`, `rplus`, `rmult` a different plot function called `plot.`*ClassName* will be invoked an plots the data in a fashion most feasible for the given datatype. This principle used used all over the package.

```
> mean(comps)
       Cu         Zn         Pb 
0.08918175 0.23949922 0.67131903 
attr(,"class")
[1] "acomp"
> mean(amounts)
       Cu        Zn        Pb 
 3.018042  8.105008 22.718430 
attr(,"class")
[1] "aplus"
> dat <- comps
```

To keep a maximum of similarity we can apply the same instructions to a dataset of a different type to perform a similar task with methods applicable to the other data type. Thus you should try afterwards the same instructions with

```
> dat <- rcomp(sa.lognormals)
> dat <- rplus(sa.lognormals)
> dat <- aplus(sa.lognormals)
> dat <- acomp(sa.lognormals)
```

and with a dataset of more variables

```
> dat <- acomp(sa.lognormals5)
> dat <- rcomp(sa.lognormals5)
> dat <- rplus(sa.lognormals5)
> dat <- aplus(sa.lognormals5)
```

although only the compositional aspects are explained here in detail.

4 Statistical Graphics

4.1 Ternary diagrams

The first steps in a data analysis should always be plots. The classical plot for compositional data is the ternary diagram. This package also contains advanced treatment for high dimensional compositions.

```
> plot(dat)
```

A ternary diagram has 3 not perpendicular axes. Each corner of a ternary diagram is associated one part of the composition. The location of a point in a ternary diagram has two main interpretations. Any composition on a line parallel to the axis opposite to a corner has the same portion of that component. The portion corresponds to the relative distance to the line to the axis on the distance of the corner to the axis. A second interpretation that all points on a straight line through one of the corners have equal relative portions of the two remaining components. This portion is given by the relative portions represented by the point, where the line crosses the opposite axis.

Several informations can be added to ternary diagrams:

```
plot(mean(dat),pch=20,add=T,col="red")      # The geometric mean
ellipses(mean(dat),var(dat),col="red",r=2) # a 2 sigma region
straight(mean(dat),princomp(dat)$Loadings) # some lines
```

Ellipses and straight lines are drawn here in Aitchison geometry [Pawlowsky-Glahn and Egozcue(2001)]. For a more classical approach with ellipses and straight lines looking straight and round you need to use `rcomp` instead.

However the ternary diagram can only display compositions of three parts. In case of more parts a scatter plot matrix like matrix of ternary diagrams is displayed which selects two components against some sort of margin of the rest:

```
plot(acomp(sa.lognormals5))
plot(acomp(sa.lognormals5),margin="rcomp")
plot(acomp(sa.lognormals5),margin="Cu")
```

4.2 Area plots

To visualize the amounts in a composition by areas we can use piecharts or stacked barplots:

```
barplot(dat)
barplot(acomp(dat[1:10,]))# a subset only
barplot(mean(dat))        # the mean only
barplot(dat-mean(dat))    # relative changes against the mean
pie(mean(dat))            # Only one composition at a time can be drawn
```

The piechart is not part of the package.

4.3 Boxplots

An basic principle in compositions is, that while the individual quantities are influenced by everything, relative portions of two components are meaningful and relatively easy to interpret, since the effect of the other components, which could eventually extrude the two parts, is removed. The boxplot function shows a matrix displaying this relative amounts of all pairs in the dataset.

```
boxplot(dat)        # ratios of amounts
boxplot(rcomp(dat)) # relative amounts
boxplot(rcomp(dat),dots=T) # plot datavalues too
```

The acomp-method of boxplot displays the ratios of the amounts in log geometry, which is typically leading to nice symmetric boxplots. The rcomp-method of boxplot simply displays the relative portion itself, which is more easy to understand but typically shows extreme skewness. This display can be seen as a display of the one dimensional minimal subcompositions.

5 Descriptive Statistics

Various descriptive statistics can be easily computed:

```
mean(dat)           # mean (geometric mean)
var(dat)            # variance in the clr-euclidean space structure
sd(dat)             # !! classical componentwise standarddeviation
mvar(dat)           # metric variance = trace of var(dat)
msd(dat)            # metric standard = mvar(dat) / (D-1)
variation(dat)      # the variation matrix (i.e. var(log(x_i/x_j)))
```

```
summary(dat)          # summaries of all log(x_i/x_j)
# cov(dat1,dat2)      # covariance in clr euclidean space structure
# cor(dat1,dat2)      # correlation in clr euclidean space structure
```

While descriptive statistics and there meaning are well known for classical multivariate datasets, their definition and their interpretation seem to be subject to ongoing research. The summaries provided follow two different general approaches: For `mean`, `var`, `mvar`, `msd` the data is interpreted as a multivariate vector in the geometry associated to the class. In this geometry the mean, variance, generalized variance or mean standard deviation is taken. While the mean of vectors is a vector again, the result is again a composition and thus given as a composition. The spread informations are informations on (squared) distances and thus given in terms of the dimensionless (squared) distances of the simplex. Naively they can be interpreted just as classical means and variances. The mean is a measure of location, while variances and standard deviations are measures of spread. A dataset with more spread has a larger variance. The `var` gives the spread of the vector as a matrix usual for multivariate quantities. However chosen unit axes represent the individual portions and are thus not perpendicular leading to a singular matrix. The `mvar` is a generalized variance of the vector giving the mean squared distance to the mean. The `msd` gives the square route of the mean square distance in arbitrary directions and can thus be interpreted like a classical standard deviation. To get a deeper understanding one must understand the Euclidean space structure known as Aitchison geometry which is explained in [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn] and later publications of the Gerona group. For detailed documentations the reader is referred to the help.

The other summaries given by `variation` and `summary` are based on the idea that a composition is represented by a set of (not unrelated) univariate quantities given by the subcompositions of each pair of two components, like used in the boxplots. All informations are provided in parallel for all the resulting univariate simplices in log-ratio geometry.

For a more classical view understanding the compositions as a multivariate vector of individual portions one can use the `rcomp`-class:

```
mean(rcomp(dat))
var(rcomp(dat))
sd(rcomp(dat))
summary(rcomp(dat))
```

6 Computation in the four scales

6.1 Computing with total mass

When we get a compositional dataset it is often not closed to a sum of due to multiple reasons. For each class can find out the total sum for each case by the `totals` command.

```
> totals(acomp(amounts))
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> totals(aplus(amounts))
 [1]  89.86653  51.27057  54.28445  18.31066  31.19352  21.82243  39.43808
...
```

The constructor “acomp” closes the data to 1 (or the amount given by the total command line option) and thus totals can not be retrieved afterwards. The total sum can be reconstituted by the optional total= parameter of the constructors:

```
> (mass <- totals(aplus(amounts)))
 [1]  89.86653  51.27057  54.28445  18.31066  31.19352  21.82243  39.43808
...
> amounts
              Cu          Zn         Pb
 [1,]  8.8043262  35.1671810  45.895025
...
> dat
               Cu          Zn         Pb
 [1,] 0.097971136 0.391326782 0.51070208
...
> acomp(dat,total=100)   # Give portions in %
              Cu         Zn        Pb
 [1,]  9.7971136 39.1326782 51.070208
> aplus(dat,total=mass)  # give each the right total mass again
              Cu          Zn         Pb
 [1,]  8.8043262  35.1671810  45.895025
...
```

6.2 Subcompositions, Marginal compositions and Grouping

There are various possibilities to transform a composition to a composition of less elements. For this chapter we use a higher dimensional example

```
> (dat5 <- acomp(sa.lognormals5))
               Cu          Zn         Pb           Cd           Co
 [1,] 0.424629320 0.419450330 0.15016391 3.024494e-03 2.731947e-03
...
```

The most simple concept is that of a subcomposition [Aitchison(1986)]: Take only some of the parts and use them as a compositions:

```
> acomp(dat5,c("Pb","Cd","Co"))
             Pb           Cd           Co
 [1,] 0.9630809 1.939768e-02 1.752142e-02
...
```

```
> acomp(dat5,1:3)
               Cu          Zn         Pb
 [1,] 0.427087826 0.421878850 0.15103332
...
```

The parts selected can be given either by names or column numbers. Both methods can not be mixed. It is a major property of the “acomp” approach to be consistent with taking subcompositions.

Another approach is that marginal compositions taking some interesting components and the “rest”. The various approaches differ in how to make a “rest”. The approach of taking just the sum of the rest is consistent with “rcomp”-approach and computed by

```
> rcompmargin(dat5,c("Cd","Cu"))
               Cd          Cu          +
 [1,] 3.024494e-03 0.424629320 0.5723462
...
```

This approach often leads badly readable ternary diagrams since the rest is often nearly everything. A sophisticated approach is that of taking the geometric mean of the rest, which is consistent with the “acomp”-approach:

```
> acompmargin(dat5,c("Cd","Cu"))
               Cd        Cu          *
 [1,] 6.258330e-03 0.8786497 0.11509202
...
```

This approach as been proposed by Vera Pawlowsky-Glahn (as cited in the help in this package). You can distinguish the margins, when selected implicitly in plot functions by the symbol “+” or “*” to name them.

The most advanced concept is that of grouping parts together and to represent each group by some mean amount. The conceptual approach of seeing the groups of parts just as components of the original material for themself leads to grouping by adding the parts in the groups. This approach is consistent with the “`rcomp`” approach and computed by

```
> dat5
              Cu          Zn         Pb           Cd           Co
 [1,] 0.424629320 0.419450330 0.15016391 3.024494e-03 2.731947e-03
...
> groupparts(rcomp(dat5),Cparts=c("Cu","Cd","Co"),Zparts=c("Zn"),Pparts=c("Pb"))
          Cparts      Zparts     Pparts
 [1,] 0.430385760 0.419450330 0.15016391
...
```

An aggregation approach more consistent with the relative geometry of “acomp” is that of taking geometric means instead of sums:

```
> groupparts(dat5,Cparts=c("Cu","Cd","Co"),Zparts=c("Zn"),Pparts=c("Pb"))
            Cparts      Zparts     Pparts
 [1,] 0.0259834679 0.717242525 0.25677401
...
```

This approach seams to change everything and to be difficult to understand . However it is linearly consistent with taking subcompositions and changing units and so on and can not lead to false conclusion because the sequence of data treatment. The approach is simplification of the approach in [Egozcue, J.J. and V. Pawlowsky-Glahn (2005)], which proposes a reweighting of the geometric means to achieve isometry of the transformation. However full isometry can not be achieved when things are seen as compositions afterwards. The `groupparts` function also exists for the classes “`rplus`” using sums and “`aplus`” using compositions.

6.3 Transformations

All the underlying spaces of the four classes can be mapped into a classical coordinate based vectorspace by some transformations. The package provides all the transformations defined for the Aitchison simplex `alr` (additive log ration)[Aitchison(1986)], `clr` (centered log ratio)[Aitchison(1986)] and `ilr` (isometric log ratio) [Egozcue, Pawlowsky-Glahn, Mateu-Figue The concept of transformations is discussed in detail in [Pawlowsky-Glahn(2003)] and further in [Pawlowsky-Glahn and Mateu-Figueras(2005)].

```
> dat
               Cu          Zn         Pb
 [1,] 0.097971136 0.391326782 0.51070208
...
attr(,"class")
[1] "acomp"
> clr(dat)
                Cu            Zn           Pb
 [1,] -1.011994527  0.3728755437  0.639118984
...
attr(,"class")
[1] "rmult"
> clr.inv(clr(dat))
               Cu          Zn         Pb
 [1,] 0.097971136 0.391326782 0.51070208
...attr(,"class")
[1] "acomp"
> ilr(dat)
                [,1]        [,2]
   [1,] -1.239435107 -0.188262542
...
attr(,"class")
[1] "rmult"
> ilr.inv(ilr(dat)) # No rownames

               [,1]        [,2]       [,3]
   [1,] 0.097971136 0.391326782 0.51070208
...attr(,"class")
[1] "acomp"
> alr(dat)
              Cu           Zn
 [1,] -1.6511135 -0.266243440
...
attr(,"class")
[1] "rmult"
> alr.inv(alr(dat)) # No last colname
               Cu          Zn
 [1,] 0.097971136 0.391326782 0.51070208
...
attr(,"class")
[1] "acomp"
```

Similar transformations are newly defined for in the compositions package for the other geometries . You can find more details in the help for `cpt`, `ipt`, `ilt`, and `iit`. Inverses for all these transforms are given by *xxx*`.inv` where xxx stands for the name of the transform. For all scales a dimension preserving (injective, isometric) transform is given by the generic functions `cdt` (centered default transform) and the isometric (bijective, isometric) transform is given by `idt` (isometric default transform):

```
> cdt(dat)   # clr, cpt, ilt, iit for acomp, rcomp, aplus, rplus
                Cu            Zn           Pb
 [1,] -1.011994527  0.3728755437  0.639118984
...
attr(,"class")
```

```
[1] "rmult"
> idt(dat)  # ilr, ipt, ilt, iit for acomp, rcomp, aplus, rplus
                [,1]         [,2]
   [1,] -1.239435107 -0.188262542
...
attr(,"class")
[1] "rmult"
```

6.4 Operations

The composition library is based on the idea of a Euclidean space structure in each of the scale levels and provides overloaded operators for $x + y$, $x - y$, $\alpha * x$, skalar product and linear mappings (matrix multiplication) in these spaces. These mathematical operations are mainly intresting to implement own novel analysis methods or to understand the technical background of the package.

For the real compositions (class "rcomp") and real amounts (class "rplus") the space structure is given by the enclosing R^D space. A problem arises from the fact, that all values in R^D are allowed values in the scale and therefore some operations leave the space and result in a "rmult"-object, which represents the R^D space for the library. Aitchison compositions (class "acomp") form a vector space[Aitchison(1986)] [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn], when we use Aitchisons perturbation as addition:

$$(x+y)_i := \left(\frac{x_i y_i}{\sum_{j=1}^D} x_j y_j \right)_{i=1,\ldots,D}$$

and Aitchisons power transform as scalar multiplication:

$$(\alpha * x)_i := \left(\frac{x_i^\alpha}{\sum_{j=1}^D} x_j^\alpha \right)_{i=1,\ldots,D}$$

The neutral element of the space (i.e. the 0) is given by `rep(1/D,D)`. For a deeper understanding of these space structures of the class "acomp" the reader is referred to [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn].

For the amounts in relative scale (class "aplus") the vector space structure is given by similar operations without closing the data to 1:

$$(x+y)_i := (x_i y_i)_{i=1,\ldots,D}$$

$$(\alpha * x)_i := (x_i^\alpha)_{i=1,\ldots,D}$$

The helper class "rmult" defines classical operations on coordinates:

$$(x+y)_i := (x_i + y_i)_{i=1,\ldots,D}$$

$$(\alpha * x)_i := (\alpha x_i)_{i=1,\ldots,D}$$

The neutral element of the space (i.e. the 0) is given by `rep(1,D)`.

In the library these operations can be applied to individual objects like:

```
> acomp(c(1,2,3)) + 3* acomp(c(10,1,1))
[1] 0.995024876 0.001990050 0.002985075
attr(,"class")
[1] "acomp"
```

to datasets as a whole operating with an individual object or a whole dataset of the same size again (like R typical parallel operation of vectors)

```
2*dat - (dat+dat)  # Naturally a dataset of zeros = c(1/3,1/3,1/3)
```

This style of operations tries make a dataset of compositions behave like a R-vector of vectors, which allows all the parallel operations on vectors using the operations defined in the space. This allows operations like

```
 (dat - mean(dat))/msd(dat)
```

to be meaning full, which is just an isotropic scaling. Furthermore the vector spaces are equipped with a scalar product and a norm [Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and which – according to the original definition of "%*%" in R – are computed by the %*% operator:

```
> acomp(c(1,2,1))%*%acomp(c(1/2,1,2))
[1] 0
```

For datasets the operator does not behave like matrix multiplication but like a componentwise scalar product:

```
> dat %*% dat   # scalar products
 [1]  1.57164217  8.79155615  6.15968662  0.70970049  3.02654905  0.68458123
...
> norm(dat)^2   # = (x,x)
 [1]  1.57164217  8.79155615  6.15968662  0.70970049  3.02654905  0.68458123
...
> mean( dat%*% dat) - mean(dat)%*%mean(dat)  # ML-variance estimator
[1] 2.049732
> mvar(dat)* (nrow(dat)-1)/nrow(dat)
[1] 2.049732
```

Furthermore like usually in S and R the %a matrix multiplication operator, when one element is a vector from one of the scales and the other is a square matrix of the right dimension. Like usually in

```
> matrix(1:9,ncol=3) %*% c(1,0,0)
> c(1,0,0) %*% matrix(1:9,ncol=3)
```

the vectors are treated as columns on the right side of the multiplication and as rows on the left side. The matrix describes a linear mapping with respect to a fixed coordinate system. Depending on the frame size of the matrix the coordinates can either be the coordinate system the idt-transform or cdt-transform of the respective scale. As an example we scale the dataset to unit variance matrix:

```
> var( powerofpsdmatrix(var(dat),-1/2) %*% (dat-mean(dat)) )
           [,1]       [,2]       [,3]
[1,]  0.6666667 -0.3333333 -0.3333333
[2,] -0.3333333  0.6666667 -0.3333333
[3,] -0.3333333 -0.3333333  0.6666667
```

This matrix is of unit variance in the simplex plane. The off diagonal elements correspond to spurious correlation described by [Chayes(1960)]. The powerofpsdmatrix is a convenience function in the package to compute powers, inverses and square roots of singular (positive semidefinite) matrices.

7 Multivariate Methods

The central idea of the package – following the coordinate approach of [Pawlowsky-Glahn(2003)] and [Pawlowsky-Glahn and Mateu-Figueras(2005)] – is to transform the data by one of transforms into a classical multivariate dataset, to apply classical multivariate statistics and to back transform or interpreted the results afterwards in the original space.

7.1 Principle Component Analysis

The package augments the standard principle component analysis with specific interpretations in the given scale.

```
> (pc <- princomp(dat))
Call:
princomp.acomp(x = dat)

Standard deviations:
   Comp.1    Comp.2
1.3604382 0.4460269

 3  variables and  60 observations.
Mean (compositional):
        Cu         Zn         Pb
0.08918175 0.23949922 0.67131903
attr(,"class")
[1] "acomp"
+Loadings (compositional):
              Cu        Zn        Pb
Comp.1 0.5533583 0.5570883 1.8895534
Comp.2 0.4207858 1.7307697 0.8484445
attr(,"class")
[1] "acomp"
-Loadings (compositional):
             Cu        Zn        Pb
Comp.1 1.312246 1.3034604 0.3842932
Comp.2 1.725060 0.4193976 0.8555428
attr(,"class")
[1] "acomp"
> pc$Loadings              # The loadings as compositional vector
              Cu        Zn        Pb
Comp.1 0.5533583 0.5570883 1.8895534
Comp.2 0.4207858 1.7307697 0.8484445
attr(,"class")
[1] "acomp"
> pc$loadings              # The loadings in clr-space
Loadings:
   Comp.1 Comp.2
Cu -0.412 -0.705
Zn -0.405  0.709
Pb  0.816


               Comp.1 Comp.2
SS loadings     1.000  1.000
Proportion Var  0.333  0.333
Cumulative Var  0.333  0.667
> plot(pc)                 # screeplot
> plot(pc,type="variance") # other screeplot
> plot(pc,type="biplot")   # biplot
> plot(pc,type="loadings") # loadings as compositions
> plot(pc,type="relative") # loadings of log-ratios
> ? plot.princomp.acomp    # help
```

A detailed course in interpretation of the results goes far beyond the scope of this software introduction. Not all possibilities have been discussed in literature until now. However references are [Aitchison and Greenacre(2002)], [Pawlowsky-Glahn and Egozcue(2001)], [Pawlowsky-Glahn(2003)], [Pawlowsky-Glahn and Mateu-Figueras(2005)].

7.2 Cluster Analysis

The package does not contain any special routine for cluster analysis, however due to its generic distance computation typical hclust usage is done in the selected geometry and automatically consistent with the selected approach:

```
hc <- hclust(dist(dat,method="euclidean"),linkage="avarage")
# The other distance types '"euclidean"', '"maximum"', '"manhattan"',
# '"canberra"', and '"minkowski"' are also meaningfull here.
plot(hc)  # plot.hclust showing the dendrogramm
plot(dat,col=cutree(hc,4),pch=20)  # show 4 clusters in colors
```

This cluster analysis is automatically based on a meaningful distance computed with the specified method in the cdt (see help) transform. At this time to compute a kmeans-clustering should be done manually in Euclidean coordinates (which is are explained in the help topic ilr) :

```
means <-  acomp(t(sapply(split(dat,factor(cutree(hc,4))),mean)))
km <- kmeans(ilr(dat),ilr(means))
plot(dat,col=km$cluster)
plot(ilr.inv(km$centers),add=T,col=1:4,pch=20)
```

7.3 Discrimination analysis

R provides multiple methods of discrimination analysis and more might follow. A direct support for discrimination analysis is not provided, since we can directly apply standard methods to isometricly transformed data: to apply the standard methods to compositional datasets:

```
library(MASS)
library(mda)
# split a dataset into training and validation part:
selection   <- sample(nrow(sa.groups),floor(nrow(sa.groups)*0.7))
trainset    <- acomp(sa.groups[selection,])
traingroups <- sa.groups.area[selection]
testset     <- acomp(sa.groups[-selection,])
testgroups  <- sa.groups.area[-selection]

# Linear Discrimination analysis
discr <- lda(traingroups~.,data=idt(acomp(trainset,c("clay","sand","gravel"))))
discr
table(traingroups,predict(discr)$class)
predict(discr,newdata=idt(acomp(testset,c("clay","sand","gravel"))))
table(testgroups,predict(discr,newdata=idt(acomp(testset,c("clay","sand","gravel"))))$class)
barplot(ilr.inv(t(discr$scaling))) # Visualise the discrimination functions
plot(acomp(sa.groups),col=predict(discr,idt(acomp(sa.groups,c("clay","sand","gravel"))))$clas
plot(acomp(sa.groups),col=sa.groups.area,add=T)

# Quadratic Discrimination analysis
discr <- qda(traingroups~.,data=idt(acomp(trainset,c("clay","sand","gravel"))))
discr
table(traingroups,predict(discr)$class)
predict(discr,newdata=idt(acomp(testset,c("clay","sand","gravel"))))
table(testgroups, predict(discr,newdata=idt(acomp(testset,c("clay","sand","gravel"))))$class
plot(acomp(sa.groups),col=predict(discr,idt(acomp(sa.groups,c("clay","sand","gravel"))))$clas
plot(acomp(sa.groups),col=sa.groups.area,add=T)

# Flexible discrimination analysis
discr <- fda(traingroups~.,data=idt(acomp(trainset,c("clay","sand","gravel"))))
discr
table(traingroups,predict(discr))
predict(discr,newdata=idt(acomp(testset,c("clay","sand","gravel"))))
table(testgroups, predict(discr,idt(acomp(testset,c("clay","sand","gravel")))))
plot(acomp(sa.groups),col=predict(discr,idt(acomp(sa.groups,c("clay","sand","gravel")))),pch=
plot(acomp(sa.groups),col=sa.groups.area,add=T)
```

7.4 Linear Models

Linear models can use any of the given scales as regressors or as response. However we decided not to introduce special routines for that since one retains much more flexibility by using standard methods in conjunction with transformations. However this means that the user has to be aware of backtransforming. In case of a compositional response this could like

```
> y   <- acomp(sa.groups) # A dataset with usefull regressor
> x   <- sa.groups.area   # The (here categorial) regressor
> (mylm <- lm(ilr(y)~X,data=data.frame(X=x)))

Call:
lm(formula = ilr(y) ~ X, data = data.frame(X = x))

Coefficients:
             [,1]      [,2]
(Intercept)  -2.00642  -0.07032
XMiddle       1.83599   0.88136
XUpper        0.47219  -1.09672

> summary(manova(mylm))
           Df Pillai approx F num Df den Df    Pr(>F)
X           2  1.260   48.505      4    114 < 2.2e-16 ***
Residuals  57
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> ilr.inv(coefficients(mylm))
                   [,1]      [,2]       [,3]
  (Intercept) 0.04102279 0.4556668 0.50331037
  xMiddle     0.79781245 0.1570356 0.04515198
  xUpper      0.40384157 0.1042933 0.49186510
attr(,"class")
[1] "acomp"
```

```
> plot(ilr.inv(resid(mylm)),col=x)
> plot(y,col=x)
> plot( ilr.inv(predict(mylm)),add=T,pch=20,col=x)
> ellipses
> ilr.inv(predict(mylm,newdata=data.frame(X=factor(levels(x)))))
          [,1]      [,2]      [,3]
  1 0.04102279 0.4556668 0.5033104
  2 0.25768467 0.5633886 0.1789267
  3 0.05315796 0.1524882 0.7943539
attr(,"class")
[1] "acomp"
>
```

Similarly we can introduce the composition as regressors

```
> x <- acomp(sa.lognormals5,c("Cd","Zn","Pb","Co"))
> y <- sa.lognormals5[,"Cu"]
> (mylm <- lm(y~idt(X) , data=list(y=y,X=x)))

Call:
lm(formula = y ~ idt(X), data = list(y = y, X = x))

Coefficients:
(Intercept)      idt(X)1      idt(X)2      idt(X)3
      2.791       -1.721        2.154       -1.609
> anova(mylm)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
idt(X)     3 759.74  253.25  18.034 2.595e-08 ***
Residuals 56 786.41   14.04
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> plot(predict(mylm),resid(mylm))
> predict(mylm,newdata=list(X=acomp(x[1:3,])))
```

A combination of all this is also possible:

```
> x <- acomp(sa.groups5,c("Pb","Co","Cd"))
> y <- aplus(sa.groups5,c("Cu","Zn"))
> k <- sa.groups5.area
> plot(y,col=k)
> (mylm <- lm( idt(Y)~k+idt(X),data=list(X=x,Y=y,k=k)))
Call:
lm(formula = idt(Y) ~ k + idt(X), data = list(X = x, Y = y, k = k))

Coefficients:
             Cu        Zn
(Intercept)   1.97369   4.46492
kMiddle      -0.24318  -1.75248
kUpper       -0.76350  -2.21039
idt(X)1      -0.02325  -0.01573
idt(X)2      -0.55759  -0.57721
```

```
> plot(ilt.inv(predict(mylm)),add=T,col=k,pch=20)
> summary(manova(mylm))
```

8 Conclusions

Without conclusion, but opening to further steps like barycentric coordinates (`endpointCoordinates`) or simulation (dlnorm.acomp,rnorm.acomp,runif.acomp, rDirichlet.acomp) I ask the reader to make his own experiences with compositional data analysis.

References

[Aitchison(1982)] Aitchison, J., 1982. The statistical analysis of compositional data (with discussion). Journal of the Royal Statistical Society, Series B (Statistical Methodology) 44 (2), 139–177.

[Aitchison(1984)] Aitchison, J., 1984. Reducing the dimensionality of compositional data sets. Mathematical Geology 16 (6), 617–636.

[Aitchison(1986)] Aitchison, J., 1986. The Statistical Analysis of Compositional Data. Monographs on Statistics and Applied Probability. Chapman & Hall Ltd., London (UK), 416 p.

[Aitchison(1997)] Aitchison, J., 1997. The one-hour course in compositional data analysis or compositional data analysis is simple. In: Pawlowsky-Glahn, V. (Ed.), Proceedings of IAMG'97 — The third annual conference of the International Association for Mathematical Geology. Vol. I, II and addendum. International Center for Numerical Methods in Engineering (CIMNE), Barcelona (E), pp. 3–35.

[Aitchison(2002)] Aitchison, J., 2002. Simplicial inference. In: Viana, M. A. G., Richards, D. S. P. (Eds.), Algebraic Methods in Statistics and Probability. Vol. 287 of Contemporary Mathematics Series. American Mathematical Society, Providence, Rhode Island (USA), pp. 1–22.

[Aitchison and Greenacre(2002)] Aitchison, J., Greenacre, M., 2002. Biplots for compositional data. Applied Statistics 51 (4), 375–392.

[Barceló-Vidal et al.(2001)Barceló-Vidal, Martín-Fernández, and Pawlowsky-Glahn] Barceló-Vidal, C., Martín-Fernández, J. A., Pawlowsky-Glahn, V., 2001. Mathematical foundations of compositional data analysis. In: Ross, G. (Ed.), Proceedings of IAMG'01 — The sixth annual conference of the International Association for Mathematical Geology. Vol. CD-. p. 20 p, electronic publication.

[Billheimer et al.(2001)Billheimer, Guttorp and Fagan] Billheimer, D. and Guttorp, P. and Fagan, W.F. (2001). Statistical interpretation of species composition. *Journal of the American Statistical Association 96*(456), 1205–1214.

[Buccianti et al.(1999)Buccianti, Pawlowsky-Glahn, Barceló-Vidal, and Jarauta-Bragulat] Buccianti, A., Pawlowsky-Glahn, V., Barceló-Vidal, C., Jarauta-Bragulat, E., 1999. Visualization and modeling of natural trends in ternary diagrams: a geochemical case study. In: Lippard, S. J., Næss, A., Sinding-Larsen, R. (Eds.), Proceedings of IAMG'99 — The fifth annual conference of the International Association for Mathematical Geology. Vol. I and II. Tapir, Trondheim (N), pp. 139–144.

[Chayes(1960)] Chayes, F., 1960. On correlation between variables of constant sum. Journal of Geophysical Research 65 (12), 4185–4193.

[Egozcue, Pawlowsky-Glahn, Mateu-Figueras, and Barceló-Vidal(2003)Egozcue and others] Egozcue, J. J., V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal (2003). Isometric logratio transformations for compositional data analysis. *Mathematical Geology 35*(3), 279–300.

[Egozcue, J.J. and V. Pawlowsky-Glahn (2005)] Egozcue, J.J. and V. Pawlowsky-Glahn (2005) Groups of Parts and their Balances in Compositional Data Analysis, *Mathematical Geology*, in press

[Martín-Fernández et al.(2003)Martín-Fernández, Barceló-Vidal, and Pawlowsky-Glahn] Martín-Fernández, J., Barceló-Vidal, C., Pawlowsky-Glahn, V., 4 2003. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Math. Geol. 35 (3).

[Otero et al.(2005)Otero, Tolosana-Delgado, Soler, Pawlowsky-Glahn and Canals] Otero, N., Tolosana-Delgado, R., Soler, A., Pawlowsky-Glahn, V. and Canals, A. (2005). Relative vs absolute analysis of compositions: a comparative analysis in surface waters of a Mediterranean river. *Water Research* (in press).

[Pawlowsky-Glahn and Egozcue(2001)] Pawlowsky-Glahn, Vera and Egozcue, Juan José (2001). Geometric approach to statistical analysis on the simplex. *Stochastic Environmental Research and Risk Assessment (SERRA) 15*(5), 384–398.

[Pawlowsky-Glahn and Egozcue(2002)] Pawlowsky-Glahn, V., Egozcue, J. J., 2002. BLU estimators and compositional data. Mathematical Geology 34 (3), 259–274.

[Pawlowsky-Glahn(2003)] Pawlowsky-Glahn, Vera (2003). Statistical modelling on coordinates. In: Thió-Henestrosa and Martín-Fernàndez(2003) *Compositional Data Analysis Workshop – CoDaWork'03, Proceedings.* Universitat de Girona, ISBN 84-8458-111-X, http://ima.udg.es/Activitats/CoDaWork03/.

[Pawlowsky-Glahn and Mateu-Figueras(2005)] Pawlowsky-Glahn, V., G. Mateu-Figueras (2005) The Statistical Analysis on Coordinates in Constrained Spaces, in International Statistical Institute. Session (55th :, 2005 : Sydney, N.S.W.) (2005) Abstract book : 55th session of the International Statistical Institute (ISI), 5-12 April 2005, Sydney Convention & Exhibition Centre, Sydney, Australia. ISBN 1-877040-28-2

[Pearson(1897)] Pearson, K., 1897. Mathematical contributions to the theory of evolution. on a form of spurious correlation which may arise when indices are used in the measurement of organs. Proceedings of the Royal Society of London LX, 489–502.

[R Development Core Team(2004)] R Development Core Team (2004). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-00-3.

[Rehder and Zier(2002)] Rehder, S. and Zier, U. (2002), Some remarks about transformations. In: Bayer, Burger, Skala (Eds.), Proceedings of IAMG'02 — The eight annual conference of the International Association for Mathematical Geology. Vol. I and II. Berlin (D), pp. 423–428.

[Shurtz(2003)] Shurtz, Robert F., 2003. Compositional geometry and mass conservation. Mathematical Geology 35 (8), 972–937.

[von Eynatten et al.(2003)von Eynatten, Barceló-Vidal, and Pawlowsky-Glahn] von Eynatten, H., Barceló-Vidal, C., Pawlowsky-Glahn, V., 2003. Modelling compositional change: the example of chemical weathering of granitoid rocks. Mathematical Geology 35 (in press).

[von Eynatten et al.(2002)von Eynatten, Pawlowsky-Glahn, and Egozcue] von Eynatten, H., Pawlowsky-Glahn, V., Egozcue, J. J., 2002. Understanding perturbation on the simplex: a simple method to better visualise and interpret compositional data in ternary diagrams. Mathematical Geology 34 (3), 249–257.