Detecting outliers in compositional data using Invariant Coordinate Selection

Supplementary material

Last update: 2023-01-05

Contents

This document presents the **R** codes used to obtain the computational results included in the paper "Detecting outliers in compositional data using Invariant Coordinate Selection". To cite this work, please use:

Anne Ruiz-Gazen, Christine Thomas-Agnan, Thibault Laurent, and Camille Mondon (2022). Detecting outliers in compositional data using Invariant Coordinate Selection, *TSE WP*, **22-1320**.

Packages needed:

```
library(codareg) # to use the data set on the market shares
library(compositions) # compositions data
library(ICS) # ICS
library(ICSOutlier) # outliers detection
library(plotrix) # matrix plot
library(RColorBrewer) # palette colors
library(zoo) # time series
```

Note that all packages can be installed from CRAN excepted **codareg** that can be installed from github:

```
devtools::install_github("tibo31/codareg")
```

1 Toy Examples

This section contains the **R** codes that allow to reproduce results and figures presented in Section 6.1 in the article.

1.1 First example

To obtain the first example:

- we simulate two samples from a bivariate Gaussian distribution.
    - the first sample of size 90 with $\mu_1 = (0, 0)$ and $\Sigma_1 = \begin{pmatrix} 0.04 & 0.02 \\ 0.02 & 0.04 \end{pmatrix}$
    - the second sample of size 10 with $\mu_2 = (\frac{2}{\sqrt{2}} \log 2, \frac{-1}{\sqrt{6}} \log 2)$ and $\Sigma_2 = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.05 \end{pmatrix}$

We define the contrast matrix:

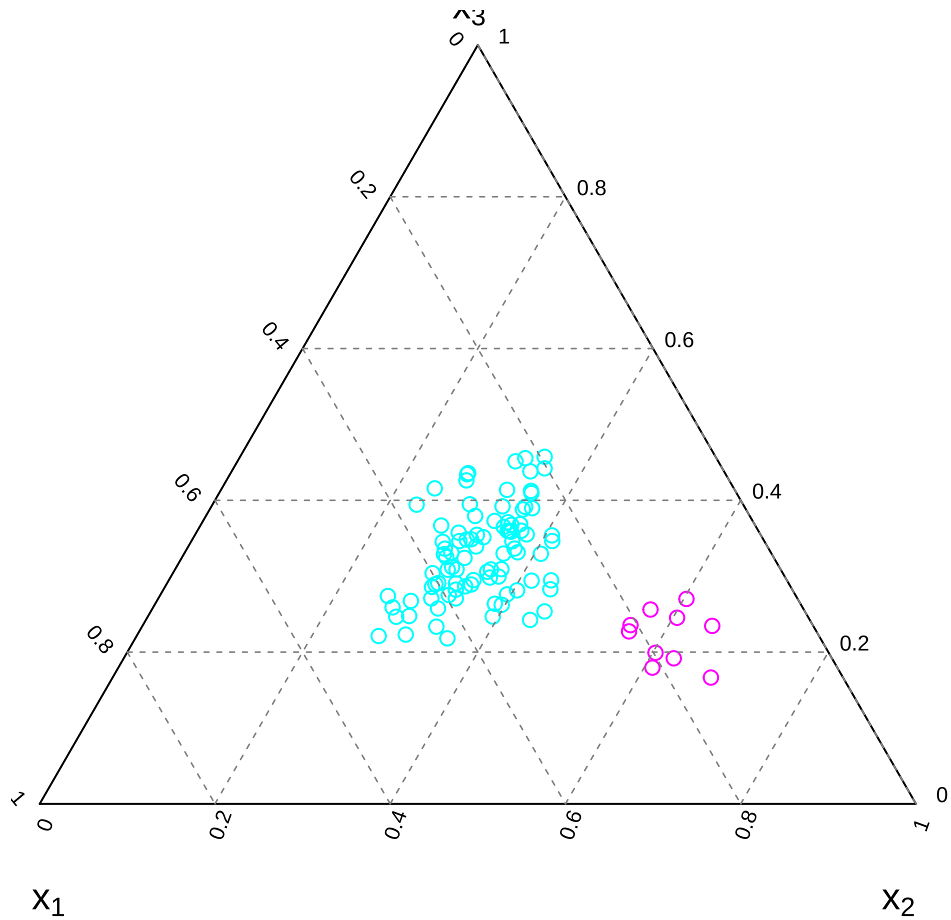
```
V_contrast <- compositions::ilrBase(D=3)
```

```
# group 1
mu_1 <- c(0, 0)
sigma_1 <- matrix(c(0.04, 0.02, 0.02, 0.04), nrow = 2)
set.seed(2)
Y_ilr_1 <- mvtnorm::rmvnorm(90,
                mean = mu_1, sigma = sigma_1)
Y_s_1 <- as(compositions::ilrInv(Y_ilr_1, V = V_contrast), "matrix")
# group 2
mu_2 <- c(2 / sqrt(2) * log(2), -1 / sqrt(6) * log(2))
sigma_2 <- matrix(c(0.05, 0, 0, 0.05), nrow = 2)
Y_ilr_2 <- mvtnorm::rmvnorm(10,
                            mean = mu_2, sigma = sigma_2)
Y_s_2 <- as(compositions::ilrInv(Y_ilr_2, V = V_contrast), "matrix")
```

- we transform those two samples into simplex vectors using an ilr inverse transformation ilr^{-1} (contrast matrix is the one used by default in package **compositions**) and we obtain (Figure 1 on the left in the article) :

```
# representation
A <- c(0, 0)
B <- c(1, 0)
C <- c(0.5, sqrt(3) / 2)

Y_simplex_x_g1 <- Y_s_1[, 1] * A[1] + Y_s_1[, 2] * B[1] + Y_s_1[, 3] * C[1]
Y_simplex_y_g1 <- Y_s_1[, 1] * A[2] + Y_s_1[, 2] * B[2] + Y_s_1[, 3] * C[2]

Y_simplex_x_g2 <- Y_s_2[, 1] * A[1] + Y_s_2[, 2] * B[1] + Y_s_2[, 3] * C[1]
Y_simplex_y_g2 <- Y_s_2[, 1] * A[2] + Y_s_2[, 2] * B[2] + Y_s_2[, 3] * C[2]

# pdf(file = "figures/toy_data/toy_data_simplex.pdf", width = 4, height = 4)
op <- par(oma = c(.1, .1, .1, .1), mar = c(0, 0.7, .5, 0.7))

plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
     type = "n", xlab = "", ylab = "", asp = 1, main = "",
     ylim = c(-0.12, sqrt(3)/2), cex.main = 2)

text(c(0.01, 0.98, 1/2-0.01), c(0-0.15, -0.15, sqrt(3)/2+0.0),
     c(expression(x[1]), expression(x[2]), expression(x[3])), pos = 3,
     cex = 1.2)
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

points(Y_simplex_x_g1, Y_simplex_y_g1, col = "cyan", cex = 1)
points(Y_simplex_x_g2, Y_simplex_y_g2, col = "magenta", cex = 1)

# XR lines
for (k in 1:5) {
  lines(c(k/10, 1-k/10), c(k/5 * C[2], k/5 * C[2]), lty = 2,
        lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
}
```

```r
# Left lines
for (k in 1:5) {
  lines(c(k/5, 0.5 + k * 1/10), c(0, C[2] - k/5 * C[2]), lty = 2,
        lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
}
# Right lines
for (k in 1:5) {
  lines(c(k * 1 / 10, k/5), c(k/5 * C[2], 0), lty = 2, lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
}
for (k in 0:5) {
    text(1.03-k/10, k/5 * C[2] -0.02, 2 * k/10, pos = 3, cex = 0.7)
    text(k/5, 0, 2 * k/10, pos = 1, cex = 0.7, srt = 70)
      text(k * 1 / 10, k/5 * C[2], 1 - 2 * k/10, pos = 2, cex = 0.7, srt = -50)
}
```




```r
par(op)
# dev.off()
```

The parameters of the mean and the variance of the distributions of these simplex vectors are :

```r
compositions::ilrInv(mu_1)
```

```
##           1           2           3
## "0.3333333" "0.3333333" "0.3333333"
## attr(,"class")
## [1] "acomp"
```

```
V_contrast %*% sigma_1 %*% t(V_contrast)
```

```
##             1            2            3
## 1  0.03821367 -0.013333333 -0.024880339
## 2 -0.01333333  0.015119661 -0.001786328
## 3 -0.02488034 -0.001786328  0.026666667
```

```
compositions::ilrInv(mu_2)
```

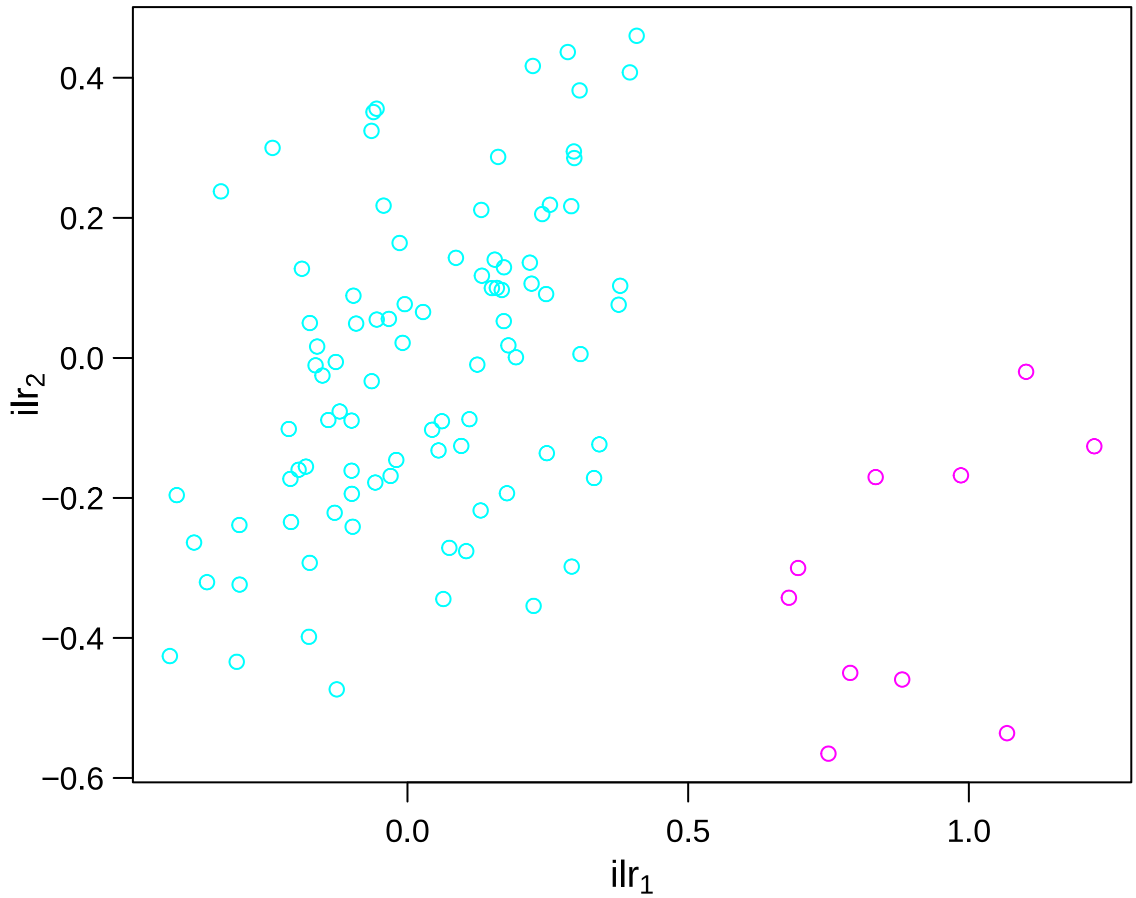
```
##           1           2           3
## "0.1559038" "0.6236150" "0.2204812"
## attr(,"class")
## [1] "acomp"
```

```
V_contrast %*% sigma_2 %*% t(V_contrast)
```

```
##             1           2           3
## 1  0.03333333 -0.01666667 -0.01666667
## 2 -0.01666667  0.03333333 -0.01666667
## 3 -0.01666667 -0.01666667  0.03333333
```

- we go back to the ilr space considering the following contrast matrix: $V_v = \begin{pmatrix} -\frac{1}{2}\sqrt{\frac{2}{3}} & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{2}{3}} & 0 \\ -\frac{1}{2}\sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{2}} \end{pmatrix}$ and we plot the data (figure 1 on the middle) :

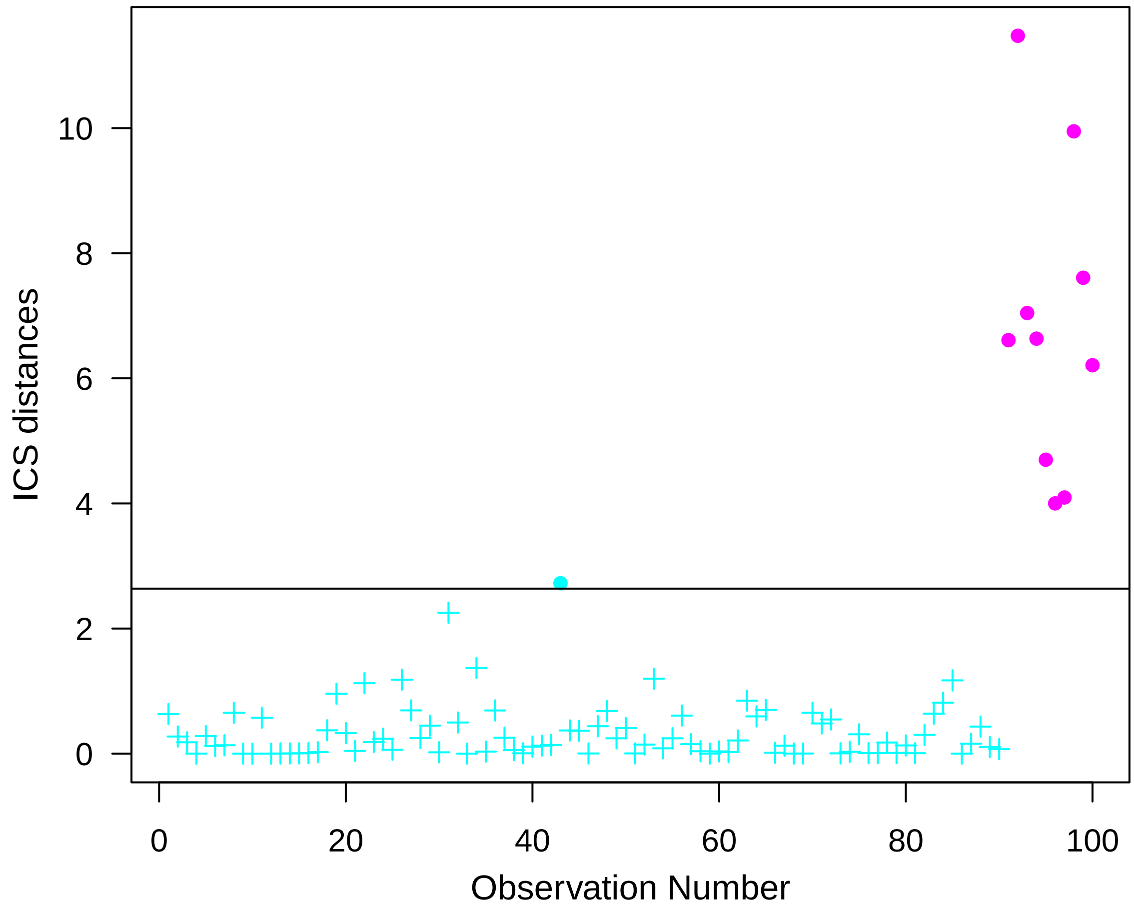
```
# The data in the simplex
Y_s <- rbind(Y_s_1, Y_s_2)

# Transformation
my_ilr_v <- compositions::ilr(Y_s, V = V_contrast)

# Representation of the data
# pdf(file = "figures/toy_data/toy_data_ilr.pdf", width = 4, height = 4)
op <- par(oma = c(0,0,0,0), mar = c(3.3, 3.3, 1.5, 0.7), #  c(1, 1.5, 1.4, 1), mar = c(3.3, 3.3, 1.5, 0
          las = 1, mgp = c(2.1, 0.75, 0))
plot(my_ilr_v[1:90, 1], my_ilr_v[1:90, 2],
     col = "cyan", xlab = expression(ilr[1]), ylab = expression(ilr[2]),
     xlim = range(my_ilr_v[, 1]), ylim = range(my_ilr_v[, 2]),
     main = "", cex.main = 2, cex.lab = 1.2)
points(my_ilr_v[91:100, 1], my_ilr_v[91:100, 2], col = "magenta")
```




```
par(op)
# dev.off()
```

- We use package `ICSOutlier` to detect outliers (figure 1 on the right):

```
my_ics <- ics2(my_ilr_v)
icsOutlier <- ics.outlier(my_ics, level.dist = 0.1, mDist = 50, ncores = 1)
# pdf(file = "figures/toy_data/toy_data_ics.pdf", width = 4, height = 4)
op <- par(oma = c(0,0,0,0), mar = c(3.3, 3.3, 1.5, 0.7),
          las = 1, mgp = c(2.25, 1, 0))
plot(icsOutlier@ics.distances, main = "",
     pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 3),
     col = c(rep("cyan", 90),
             rep("magenta", 10)),
     xlab = "Observation Number", ylab = "ICS distances", cex.main = 2, cex.lab = 1.1)
abline(h = icsOutlier@ics.dist.cutoff)
```




```
# dev.off()
```

- To get the eigenvectors :
  - we compute the centered data in coordinate space

```
mean_v <- apply(my_ilr_v, 2, mean)
my_ilr_v_centre <- my_ilr_v - matrix(mean_v, byrow = T,
                                     nrow(my_ilr_v), ncol(my_ilr_v))
```

  - we run the ics and take the inverse of the unmixed matrix (which is the matrix that allows to reconstruct the original data)

```
ics.ilr.v_centre <- ics(my_ilr_v_centre, stdB = "Z", stdKurt = F)
V_star_centre <- solve(ics.ilr.v_centre@UnMix)
```

  - to print the eigenvalues

```
ics.ilr.v_centre@gKurt
```

```
## [1] 1.5725463 0.8096538
```

  - to print the eigenvectors in coordinate space and in the simplex

```
a_1_ilr <- V_star_centre[, 1]
a_2_ilr <- V_star_centre[, 2]
cbind(a_1_ilr, a_2_ilr)
```

```
##          a_1_ilr   a_2_ilr
## [1,]  0.31018157 0.1268776
## [2,] -0.09883097 0.2152115
```

```
a_1 <- compositions::ilrInv(a_1_ilr, V = V_contrast)
a_2 <- compositions::ilrInv(a_2_ilr, V = V_contrast)
cbind(a_1, a_2)
```

```
##         a_1       a_2
## 1 0.2736791 0.2762211
## 2 0.4243758 0.3305089
## 3 0.3019451 0.3932701
```

- to plot the projections (Figure 2 in the article)

```
# Representation of the data
# pdf(file = "figures/toy_data/toy_data_eigen_ilr_non_centered.pdf", width = 8, height = 8)
op <- par(oma = c(0, 1.5, 1.4, 1), mar = c(3.3, 3.3, 1.5, 0.7),
          las = 1, mgp = c(2.25, 1, 0), mfrow = c(2, 2))
# projection on a1*
plot(my_ilr_v[1:90, 1], my_ilr_v[1:90, 2],
   col = "cyan", xlab = expression(ilr[1]), ylab = expression(ilr[2]),
   xlim = range(c(my_ilr_v)),
   ylim = range(c(my_ilr_v)),
   main = expression("Projection on: " ~ paste(a[1], "*")),
   asp = 1, pch = 16, cex = 0.4, cex.main = 1.4, cex.lab= 1.3)
points(my_ilr_v[91:100, 1], my_ilr_v[91:100, 2],
       col = "magenta", pch = 16, cex = 0.4)
# a1 and a2
my_a_1 <- a_1_ilr[2] / a_1_ilr[1]
abline(mean_v[2] - my_a_1 * mean_v[1], my_a_1, lty = 2, col = rgb(0.5, 0.5, 0.5))
my_a_2 <- a_2_ilr[2] / a_2_ilr[1]
abline(mean_v[2] - my_a_2 * mean_v[1], my_a_2, lty = 2, col = rgb(0.5, 0.5, 0.5))

for(k in 91:100) {
  x_0 <- a_1_ilr[1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[1]
  y_0 <- a_1_ilr[2] * ics.ilr.v_centre@Scores[k, 1] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "magenta", lwd = 1, lty = 3)
}

for(k in 1:90) {
  x_0 <- a_1_ilr[1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[1]
  y_0 <- a_1_ilr[2] * ics.ilr.v_centre@Scores[k, 1] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "cyan", lwd = 1, lty = 3)
}

points(a_1_ilr[1] * ics.ilr.v_centre@Scores[, 1] + mean_v[1],
       a_1_ilr[2] * ics.ilr.v_centre@Scores[, 1] + mean_v[2],
       col = c(rep("cyan", 90), rep("magenta", 10)),
       pch = 16, cex = 0.9)

points(a_1_ilr[1] + mean_v[1], a_1_ilr[2] + mean_v[2], pch = 15,
```

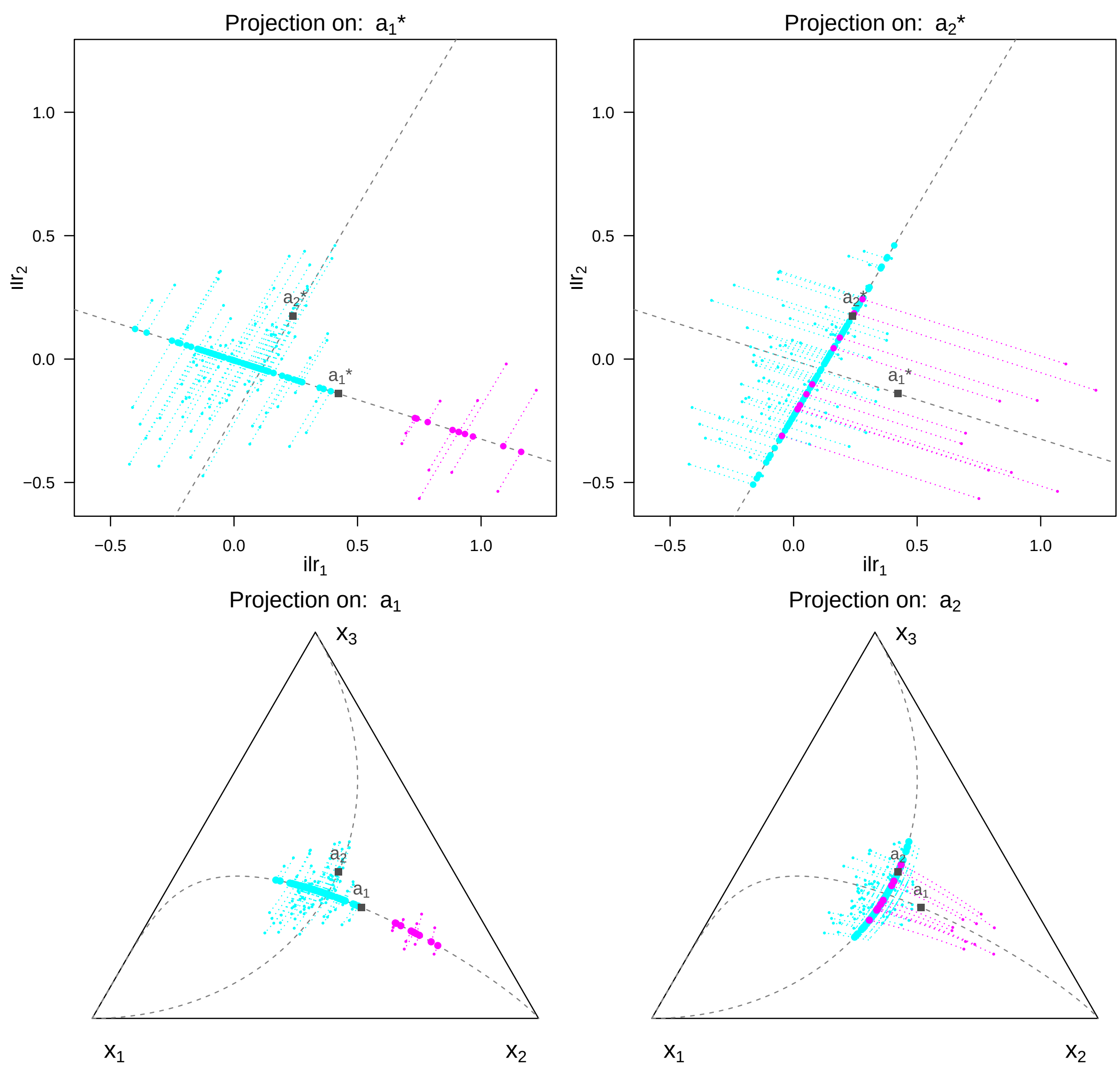
```
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a_1_ilr[1] + mean_v[1], a_1_ilr[2] + mean_v[2], expression("" ~ paste(a[1], "*")),
     pos = 3, col = rgb(0.3, 0.3, 0.3), cex = 1.1)
points(a_2_ilr[1] + mean_v[1], a_2_ilr[2] + mean_v[2], pch = 15,
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a_2_ilr[1] + mean_v[1], a_2_ilr[2] + mean_v[2], expression("" ~ paste(a[2], "*")),
     pos = 3, col = rgb(0.3, 0.3, 0.3), cex = 1.1)

# projection on a2*
plot(my_ilr_v[1:90, 1], my_ilr_v[1:90, 2],
   col = "cyan", xlab = expression(ilr[1]), ylab = expression(ilr[2]),
   xlim = range(c(my_ilr_v)),
   ylim = range(c(my_ilr_v)),
   main = expression("Projection on: " ~ paste(a[2], "*")),
   asp = 1, pch = 16, cex = 0.4, cex.main = 1.4, cex.lab = 1.3)
points(my_ilr_v[91:100, 1], my_ilr_v[91:100, 2],
       col = "magenta", pch = 16, cex = 0.4)

# a1 and a2
my_a_1 <- a_1_ilr[2] / a_1_ilr[1]
abline(mean_v[2] - my_a_1 * mean_v[1], my_a_1, lty = 2, col = rgb(0.5, 0.5, 0.5))
my_a_2 <- a_2_ilr[2] / a_2_ilr[1]
abline(mean_v[2] - my_a_2 * mean_v[1], my_a_2, lty = 2, col = rgb(0.5, 0.5, 0.5))

for(k in 91:100) {
  x_0 <- a_2_ilr[1] * ics.ilr.v_centre@Scores[k, 2] + mean_v[1]
  y_0 <- a_2_ilr[2] * ics.ilr.v_centre@Scores[k, 2] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "magenta", lty = 3)
}

for(k in 1:90) {
  x_0 <- a_2_ilr[1] * ics.ilr.v_centre@Scores[k, 2] + mean_v[1]
  y_0 <- a_2_ilr[2] * ics.ilr.v_centre@Scores[k, 2] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "cyan", lty = 3)
}
points(a_2_ilr[1] * ics.ilr.v_centre@Scores[, 2] + mean_v[1],
       a_2_ilr[2] * ics.ilr.v_centre@Scores[, 2] + mean_v[2],
       col = c(rep("cyan", 90), rep("magenta", 10)),
       pch = 16, cex = 0.9)

points(a_1_ilr[1] + mean_v[1], a_1_ilr[2] + mean_v[2], pch = 15,
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a_1_ilr[1] + mean_v[1], a_1_ilr[2] + mean_v[2],
     expression("" ~ paste(a[1], "*")),
     pos = 3, col = rgb(0.3, 0.3, 0.3), cex = 1.1)
```

```r
points(a_2_ilr[1] + mean_v[1], a_2_ilr[2] + mean_v[2], pch = 15,
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a_2_ilr[1] + mean_v[1], a_2_ilr[2] + mean_v[2],
     expression("" ~ paste(a[2], "*")),
     pos = 3, col = rgb(0.3, 0.3, 0.3), cex =1.1)

####  Results in the simplex

A <- c(0, 0)
B <- c(1, 0)
C <- c(0.5, sqrt(3) / 2)

Y_a1 <- ics.ilr.v_centre@Scores[, 1] * a_1 +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_simplex_x_a1 <- Y_a1[, 1] * A[1] + Y_a1[, 2] * B[1] + Y_a1[, 3] * C[1]
Y_simplex_y_a1 <- Y_a1[, 1] * A[2] + Y_a1[, 2] * B[2] + Y_a1[, 3] * C[2]

Y_a2 <- ics.ilr.v_centre@Scores[, 2] * a_2 +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_simplex_x_a2 <- Y_a2[, 1] * A[1] + Y_a2[, 2] * B[1] + Y_a2[, 3] * C[1]
Y_simplex_y_a2 <- Y_a2[, 1] * A[2] + Y_a2[, 2] * B[2] + Y_a2[, 3] * C[2]

# lines generated by a1
lines_a1 <- seq(-100, 100, length.out = 1000) * a_1 +
  compositions::ilrInv(mean_v, V = V_contrast)
lines_simplex_x_a1 <- lines_a1[, 1] * A[1] + lines_a1[, 2] * B[1] +
  lines_a1[, 3] * C[1]
lines_simplex_y_a1 <- lines_a1[, 1] * A[2] + lines_a1[, 2] * B[2] +
  lines_a1[, 3] * C[2]

# lines generated by a2
lines_a2 <- seq(-100, 100, length.out = 1000) * a_2  +
  compositions::ilrInv(mean_v, V = V_contrast)
lines_simplex_x_a2 <- lines_a2[, 1] * A[1] + lines_a2[, 2] * B[1] +
  lines_a2[, 3] * C[1]
lines_simplex_y_a2 <- lines_a2[, 1] * A[2] + lines_a2[, 2] * B[2] +
  lines_a2[, 3] * C[2]

# a1 in the simplex
a_1_simplex <- a_1 + compositions::ilrInv(mean_v, V = V_contrast)
a1_simplex_x <- a_1_simplex[1] * A[1] + a_1_simplex[2] * B[1] +
  a_1_simplex[3] * C[1]
a1_simplex_y <- a_1_simplex[1] * A[2] + a_1_simplex[2] * B[2] +
  a_1_simplex[3] * C[2]

# a2 in the simplex
a_2_simplex <- a_2 + compositions::ilrInv(mean_v, V = V_contrast)
a2_simplex_x <- a_2_simplex[1] * A[1] + a_2_simplex[2] * B[1] +
  a_2_simplex[3] * C[1]
a2_simplex_y <- a_2_simplex[1] * A[2] + a_2_simplex[2] * B[2] +
  a_2_simplex[3] * C[2]

# ICS 1
```

```r
plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
 type = "n", xlab = "", ylab = "", asp = 1, main = "",
 ylim = c(-0.1, sqrt(3)/2), cex.main = 1)

   title(expression("Projection on: " ~ paste(a[1], "")), line = +0.4, cex.main = 1.4)

text(c(0.05, 0.95, 1/2+0.07), c(0-0.12, -0.12, sqrt(3)/2-0.05),
 c(expression(x[1]), expression(x[2]), expression(x[3])), pos = 3, cex = 1.5)
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

points(Y_simplex_x_g1, Y_simplex_y_g1, col = "cyan", pch = 16, cex = 0.4)
points(Y_simplex_x_g2, Y_simplex_y_g2, col = "magenta", pch = 16, cex = 0.4)

lines(lines_simplex_x_a1, lines_simplex_y_a1, lty = 2, col = rgb(0.5, 0.5, 0.5))
lines(lines_simplex_x_a2, lines_simplex_y_a2, lty = 2, col = rgb(0.5, 0.5, 0.5))

 for(k in 91:100) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

  lines(seq_simplex_x, seq_simplex_y, col = "magenta", lwd = 1, lty = 3)
 }

 for(k in 1:90) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]
```

```
  lines(seq_simplex_x, seq_simplex_y, col = "cyan", lwd = 1, lty = 3)
 }

points(Y_simplex_x_a1, Y_simplex_y_a1,
   pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 16),
   col = c(rep("cyan", 90), rep("magenta", 10)), cex = 1)

points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3,
     col = rgb(0.3, 0.3, 0.3), cex = 1.1)

points(a2_simplex_x, a2_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a2_simplex_x, a2_simplex_y, expression(a[2]), pos = 3, col = rgb(0.3, 0.3, 0.3),
     cex = 1.1)

# ICS 2

plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
 type = "n", xlab = "", ylab = "", asp = 1,
 main = "",
 ylim = c(-0.1, sqrt(3)/2), cex.main = 1)

title(expression("Projection on: " ~ paste(a[2], "")), line = 0.4, cex.main = 1.4)
text(c(0.05, 0.95, 1/2+0.07), c(0-0.12, -0.12, sqrt(3)/2-0.05),
 c(expression(x[1]), expression(x[2]), expression(x[3])), pos = 3, cex = 1.5)
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

points(Y_simplex_x_g1, Y_simplex_y_g1,
   col = "cyan", pch = 16, cex = 0.4)

points(Y_simplex_x_g2, Y_simplex_y_g2,
   col = "magenta", pch = 16, cex = 0.4)

lines(lines_simplex_x_a1, lines_simplex_y_a1, lty = 2, col = rgb(0.5, 0.5, 0.5))
lines(lines_simplex_x_a2, lines_simplex_y_a2, lty = 2, col = rgb(0.5, 0.5, 0.5))

 for(k in 91:100) {
  x_0 <- V_star_centre[1, 2] * ics.ilr.v_centre@Scores[k, 2]
  y_0 <- V_star_centre[2, 2] * ics.ilr.v_centre@Scores[k, 2]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]
```

```
  lines(seq_simplex_x, seq_simplex_y, col = "magenta", lwd = 1, lty = 3)
 }

 for(k in 1:90) {
  x_0 <- V_star_centre[1, 2] * ics.ilr.v_centre@Scores[k, 2] + mean_v[1]
  y_0 <- V_star_centre[2, 2] * ics.ilr.v_centre@Scores[k, 2] + mean_v[2]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

  lines(seq_simplex_x, seq_simplex_y, col = "cyan", lwd = 1, lty = 3)
 }

points(Y_simplex_x_a2, Y_simplex_y_a2,
   pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 16),
   col = c(rep("cyan", 90), rep("magenta", 10)), cex = 1)

points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3, col = rgb(0.3, 0.3, 0.3))

points(a2_simplex_x, a2_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
text(a2_simplex_x, a2_simplex_y, expression(a[2]), pos = 3, col = rgb(0.3, 0.3, 0.3))
```




```
par(op)
#   dev.off()
```

– We identify the outliers (Figure 3 in the article):

```
#   pdf("figures/confidence_toy_example_nc.pdf", width = 10, height = 5)
# Representation of the data
op <- par(oma = c(1, 1.5, 1.4, 1), mar = c(3.3, 3.3, 1.5, 0.7),
          las = 1, mgp = c(2.25, 1, 0), mfrow = c(1, 2))
# projection on a1*
plot(my_ilr_v[1:90, 1], my_ilr_v_centre[1:90, 2],
   col = "cyan", xlab = expression(ilr[1]), ylab = expression(ilr[2]),
   xlim = range(c(my_ilr_v_centre, V_star_centre)),
   ylim = range(c(my_ilr_v_centre, V_star_centre)),
   main = "Outliers area in the ilr space", asp = 1, pch = 16, cex = 0.4)

# confidence interval
x_line <- seq(-100, 100, length.out = 10000)
```

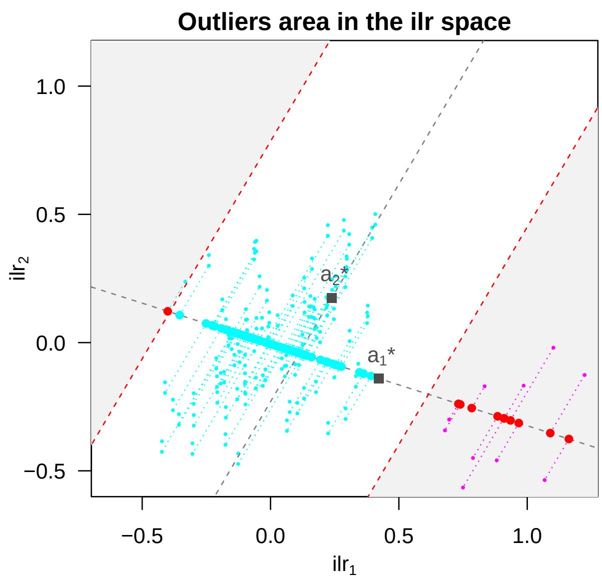
```r
# left extreme limit
x_0 <- V_star_centre[1, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[1]
y_0 <- V_star_centre[2, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[2]
b <- V_star_centre[2, 2] / V_star_centre[1, 2]
a <- y_0 - b * x_0
y_line_moins = a + b * x_line
#  right extreme limit
x_0 <- -V_star_centre[1, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[1]
y_0 <- -V_star_centre[2, 1] *  sqrt(icsOutlier@ics.dist.cutoff) + mean_v[2]
a <- y_0 - b * x_0
y_line_plus = a + b * x_line

x_line_poly_1 <- c(-100, x_line, -100)
y_line_poly_1 <- c(100, y_line_plus, 100)
polygon(x_line_poly_1, y_line_poly_1, col = rgb(0.95, 0.95, 0.95), border = NA)

x_line_poly_2 <- c(100, x_line, 100)
y_line_poly_2 <- c(-100, y_line_moins, -100)
polygon(x_line_poly_2, y_line_poly_2,  col = rgb(0.95, 0.95, 0.95), border = NA)

# axis
# a1 and a2
my_a_1 <- a_1_ilr[2] / a_1_ilr[1]
abline(mean_v[2] - my_a_1 * mean_v[1], my_a_1, lty = 2, col = rgb(0.5, 0.5, 0.5))
# a1 and a2
my_a_2 <- a_2_ilr[2] / a_2_ilr[1]
abline(mean_v[2] - my_a_2 * mean_v[1], my_a_2, lty = 2, col = rgb(0.5, 0.5, 0.5))

# points
points(my_ilr_v[91:100, 1], my_ilr_v[91:100, 2], col = "magenta", pch = 16, cex = 0.4)
points(my_ilr_v[1:90, 1], my_ilr_v[1:90, 2], col = "cyan", pch = 16, cex = 0.4)

for(k in 91:100) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "magenta", lwd = 1, lty = 3)
}

for(k in 1:90) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1] + mean_v[2]
  x_1 <- my_ilr_v[k, 1]
  y_1 <- my_ilr_v[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)
  lines(x_seq, y_seq, col = "cyan", lwd = 1, lty = 3)
}

points(V_star_centre[1, 1] * ics.ilr.v_centre@Scores[, 1] + mean_v[1],
```

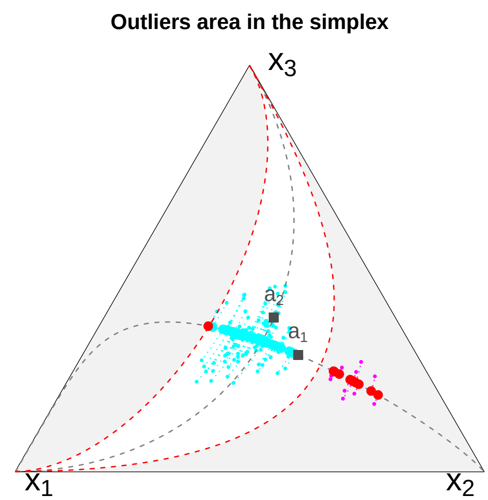
```
       V_star_centre[2, 1] * ics.ilr.v_centre@Scores[, 1] + mean_v[2],
       col = ifelse(icsOutlier@outliers, "red",
                    c(rep("cyan", 90), rep("magenta", 10))),
       pch = 16, cex = 0.9)

# confidence interval
x_line <- seq(-100, 100, length.out = 10000)

# left extreme limit
x_0 <- V_star_centre[1, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[1]
y_0 <- V_star_centre[2, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[2]
b <- V_star_centre[2, 2] / V_star_centre[1, 2]
a <- y_0 - b * x_0
y_line_moins = a + b * x_line
lines(x_line, y_line_moins, col = "red", lty = 2)
# right extreme limit
x_0 <- -V_star_centre[1, 1] * sqrt(icsOutlier@ics.dist.cutoff) + mean_v[1]
y_0 <- -V_star_centre[2, 1] *  sqrt(icsOutlier@ics.dist.cutoff) + mean_v[2]
a <- y_0 - b * x_0
y_line_plus = a + b * x_line
lines(x_line, y_line_plus, col = "red", lty = 2)

# a1
points(V_star_centre[1, 1] + mean_v[1], V_star_centre[2, 1] + mean_v[2], pch = 15,
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(V_star_centre[1, 1] + mean_v[1], V_star_centre[2, 1] + mean_v[2],
     expression("" ~ paste(a[1], "*")),
     pos = 3, col = rgb(0.3, 0.3, 0.3))
# a2
points(V_star_centre[1, 2] + mean_v[1], V_star_centre[2, 2] + mean_v[2], pch = 15,
       col = rgb(0.3, 0.3, 0.3), cex = 1)
text(V_star_centre[1, 2] + mean_v[1], V_star_centre[2, 2] + mean_v[2],
     expression("" ~ paste(a[2], "*")),
   pos = 3, col = rgb(0.3, 0.3, 0.3))

#### Results in the simplex

A <- c(0, 0)
B <- c(1, 0)
C <- c(0.5, sqrt(3) / 2)
a_1 <- compositions::ilrInv(V_star_centre[, 1], V = V_contrast)
a_2 <- compositions::ilrInv(V_star_centre[, 2], V = V_contrast)

Y_a1 <- ics.ilr.v_centre@Scores[, 1] * a_1 +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_simplex_x_a1 <- Y_a1[, 1] * A[1] + Y_a1[, 2] * B[1] + Y_a1[, 3] * C[1]
Y_simplex_y_a1 <- Y_a1[, 1] * A[2] + Y_a1[, 2] * B[2] + Y_a1[, 3] * C[2]

Y_a2 <- ics.ilr.v_centre@Scores[, 2] * a_2 +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_simplex_x_a2 <- Y_a2[, 1] * A[1] + Y_a2[, 2] * B[1] + Y_a2[, 3] * C[1]
Y_simplex_y_a2 <- Y_a2[, 1] * A[2] + Y_a2[, 2] * B[2] + Y_a2[, 3] * C[2]
```

```r
# line generated by a1
lines_a1 <- seq(-100, 100, length.out = 1000) * a_1  +
  compositions::ilrInv(mean_v, V = V_contrast)
lines_simplex_x_a1 <- lines_a1[, 1] * A[1] + lines_a1[, 2] * B[1] + lines_a1[, 3] * C[1]
lines_simplex_y_a1 <- lines_a1[, 1] * A[2] + lines_a1[, 2] * B[2] + lines_a1[, 3] * C[2]

# line generated by  a2
lines_a2 <- seq(-100, 100, length.out = 1000) * a_2 +
  compositions::ilrInv(mean_v, V = V_contrast)
lines_simplex_x_a2 <- lines_a2[, 1] * A[1] + lines_a2[, 2] * B[1] + lines_a2[, 3] * C[1]
lines_simplex_y_a2 <- lines_a2[, 1] * A[2] + lines_a2[, 2] * B[2] + lines_a2[, 3] * C[2]

# a1 in the simplex
a_1_c <- a_1 + compositions::ilrInv(mean_v, V = V_contrast)
a1_simplex_x <- a_1_c[1] * A[1] + a_1_c[2] * B[1] + a_1_c[3] * C[1]
a1_simplex_y <- a_1_c[1] * A[2] + a_1_c[2] * B[2] + a_1_c[3] * C[2]

# a2 in the simplex
a_2_c <- a_2 + compositions::ilrInv(mean_v, V = V_contrast)
a2_simplex_x <- a_2_c[1] * A[1] + a_2_c[2] * B[1] + a_2_c[3] * C[1]
a2_simplex_y <- a_2_c[1] * A[2] + a_2_c[2] * B[2] + a_2_c[3] * C[2]

#  confidence interval
ic_1 <- compositions::ilrInv(cbind(x_line_poly_1, y_line_poly_1), V = V_contrast)
ic_1_x <- ic_1[, 1] * A[1] + ic_1[, 2] * B[1] + ic_1[, 3] * C[1]
ic_1_y <- ic_1[, 1] * A[2] + ic_1[, 2] * B[2] + ic_1[, 3] * C[2]

ic_2 <- compositions::ilrInv(cbind(x_line_poly_2, y_line_poly_2), V = V_contrast)
ic_2_x <- ic_2[, 1] * A[1] + ic_2[, 2] * B[1] + ic_2[, 3] * C[1]
ic_2_y <- ic_2[, 1] * A[2] + ic_2[, 2] * B[2] + ic_2[, 3] * C[2]

# ICS 1
plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
 type = "n", xlab = "", ylab = "", asp = 1, main = "Outliers area in the simplex",
 ylim = c(0, sqrt(3)/2), cex.main = 1)

text(c(0.05, 0.95, 1/2+0.07), c(0-0.08, -0.08, sqrt(3)/2-0.05),
 c(expression(x[1]), expression(x[2]), expression(x[3])), pos = 3, cex = 1.5)
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

polygon(ic_1_x, ic_1_y, col = rgb(0.95, 0.95, 0.95), border = NA)
polygon(ic_2_x, ic_2_y,  col = rgb(0.95, 0.95, 0.95), border = NA)

points(Y_simplex_x_g1, Y_simplex_y_g1, col = "cyan", pch = 16, cex = 0.4)

points(Y_simplex_x_g2, Y_simplex_y_g2, col = "magenta", pch = 16, cex = 0.4)

lines(lines_simplex_x_a1, lines_simplex_y_a1, lty = 2, col = rgb(0.5, 0.5, 0.5))
lines(lines_simplex_x_a2, lines_simplex_y_a2, lty = 2, col = rgb(0.5, 0.5, 0.5))
```

```r
 for(k in 91:100) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

  lines(seq_simplex_x, seq_simplex_y, col = "magenta", lwd = 1, lty = 3)
 }

 for(k in 1:90) {
  x_0 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  y_0 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  x_1 <- my_ilr_v_centre[k, 1]
  y_1 <- my_ilr_v_centre[k, 2]
  x_seq <- seq(x_0, x_1, length.out = 100)
  y_seq <- seq(y_0, y_1, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x_seq, y_seq), V = V_contrast) +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

  lines(seq_simplex_x, seq_simplex_y, col = "cyan", lwd = 1, lty = 3)
 }

points(Y_simplex_x_a1, Y_simplex_y_a1,
   pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 16),
   col = ifelse(icsOutlier@outliers, "red",
                    c(rep("cyan", 90), rep("magenta", 10))), cex = 1)

seq_simplex <- compositions::ilrInv(cbind(x_line, y_line_moins), V = V_contrast)

seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

lines(seq_simplex_x, seq_simplex_y, col = "red", lwd = 1, lty = 2)

seq_simplex <- compositions::ilrInv(cbind(x_line, y_line_plus), V = V_contrast)
```

```
seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

lines(seq_simplex_x, seq_simplex_y, col = "red", lwd = 1, lty = 2)

text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3, col = rgb(0.3, 0.3, 0.3))
text(a2_simplex_x, a2_simplex_y, expression(a[2]), pos = 3, col = rgb(0.3, 0.3, 0.3))
points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
points(a2_simplex_x, a2_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)
```






```
#    dev.off()
```

1.2 Second example

To obtain the data presented in the second example:

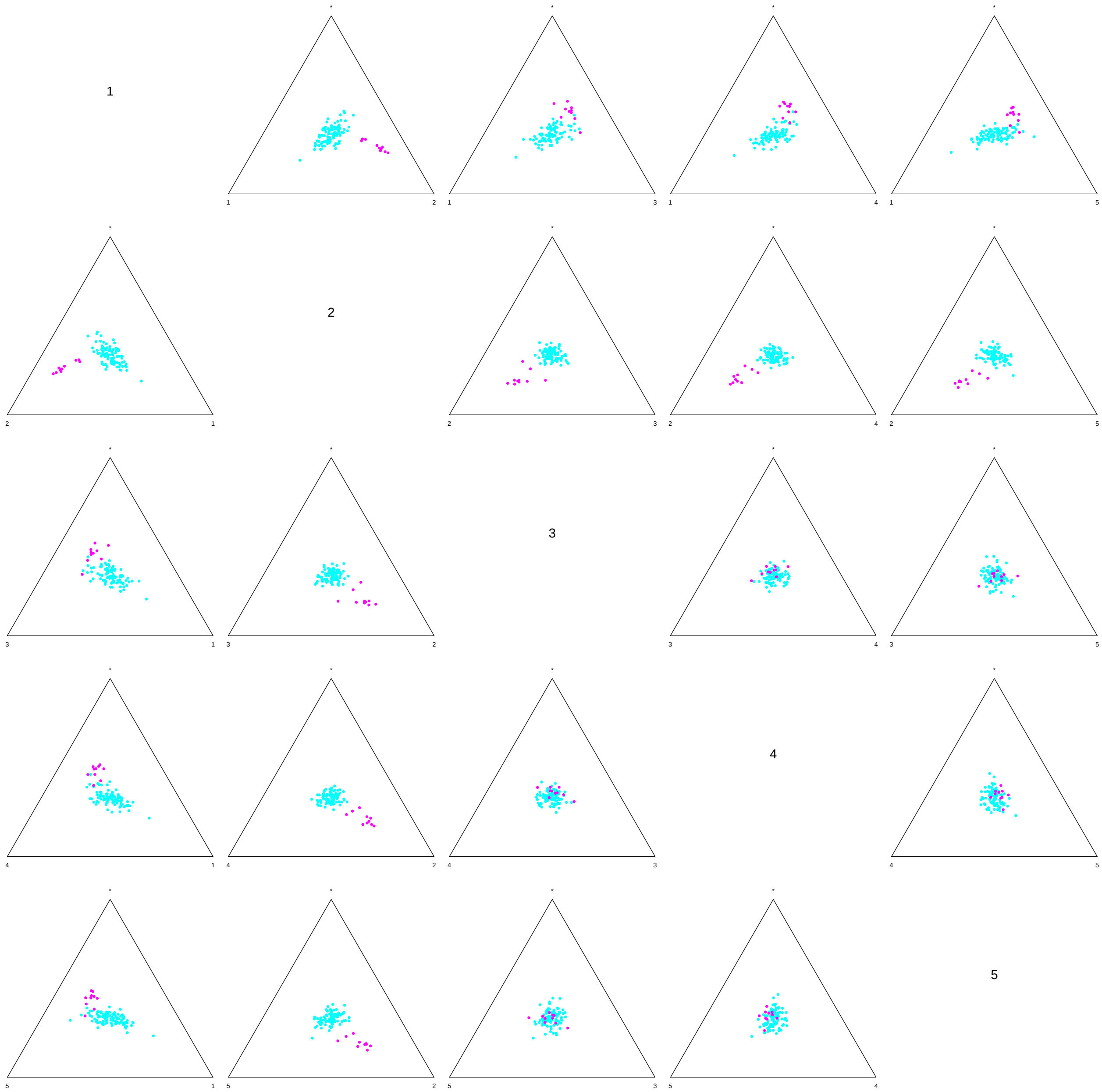
- we simulate two bivariate Gaussian vectors in $\mathbb{R}^2$.
  - the first sample of size 90 with $\mu_1 = (0, 0, \ldots, 0)$ and $\Sigma_1 = \begin{pmatrix} 0.04 & 0.02 & \ldots & 0.02 \\ 0.02 & 0.04 & \ldots & 0.02 \\ \vdots & \vdots & \vdots & \vdots \\ 0.02 & 0.02 & \ldots & 0.04 \end{pmatrix}$
  - the second sample of size 10 with $\mu_2 = (\frac{2}{\sqrt{2}} \log 2, \frac{-1}{\sqrt{6}} \log 2, 0, \ldots, 0)$ and $\Sigma_2 = \begin{pmatrix} 0.05 & 0 & 0 & \ldots & 0 \\ 0 & 0.05 & 0 & \ldots & 0 \\ 0 & 0 & 0.04 & \ldots & 0.02 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0.02 & \ldots & 0.04 \end{pmatrix}$

```
D <- 20
V_contrast <- compositions::ilrBase(D = D)
# group 1
mu_1 <- rep(0, D - 1)
sigma_1 <- matrix(0.02, nrow = D - 1, ncol = D - 1)
```

```r
  diag(sigma_1) <- 0.04
  # group 2
  mu_2 <- c(2 / sqrt(2) * log(2), -1 / sqrt(6) * log(2), rep(0, D - 3))
  sigma_2 <- sigma_1
  sigma_2[1:2, 1:2] <- matrix(c(0.05, 0, 0, 0.05), nrow = 2)
  if (D > 3) {
    sigma_2[1:2, 3:(D-1)] <- 0
    sigma_2[3:(D-1), 1:2] <- 0
  }
  set.seed(123)
    # group 1
  Y_ilr_1 <- mvtnorm::rmvnorm(90,
                mean = mu_1, sigma = sigma_1)
  Y_s_1 <- as(compositions::ilrInv(Y_ilr_1, V = V_contrast), "matrix")
  # group 2
  Y_ilr_2 <- mvtnorm::rmvnorm(10,
                mean = mu_2, sigma = sigma_2)
  Y_s_2 <- as(compositions::ilrInv(Y_ilr_2, V = V_contrast), "matrix")
```

- we transform those two samples in the simplex using an ilr inverse transformation ilr^{-1} (the contrast matrix is the one used by default in package **compositions**) and we make the representation (not represented in the article):

```r
Y <- rbind(Y_s_1, Y_s_2)
require(compositions)
# pdf(file = "figures/big_data_simplex.pdf", width = 20, height = 20)
op <- par(oma = c(1, 1, 1.5, 1), mar = c(0, 0.7, 1.5, 0.7))
plot(acomp(Y, 1:5),  pch = 16,
     col = c("cyan", "magenta")[c(rep(1, 90), rep(2, 10))])
```




```
# dev.off()
```

The parameters of the mean and the variance in the simplex are :

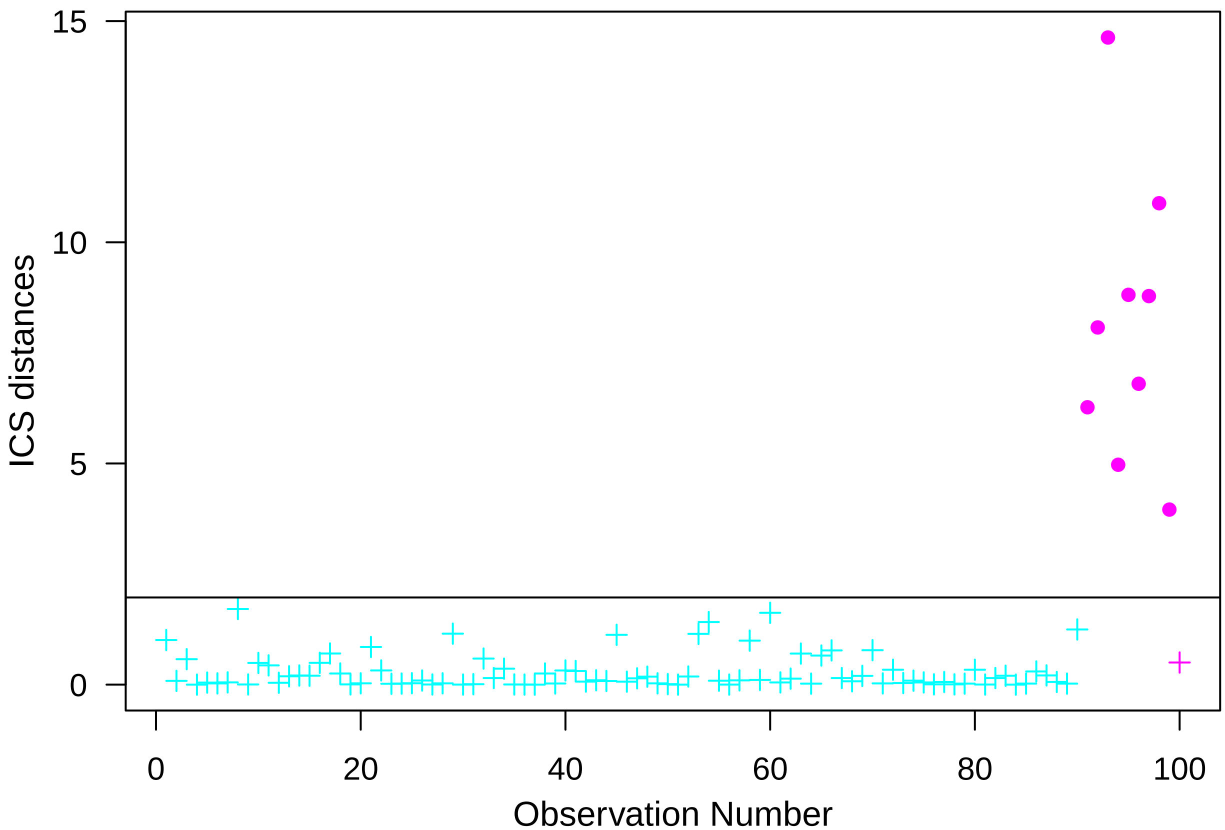
```
compositions::ilrInv(mu_1)
V_contrast %*% sigma_1 %*% t(V_contrast)
compositions::ilrInv(mu_2)
V_contrast %*% sigma_2 %*% t(V_contrast)
```

We transform the data in the ilr space:

```
my_ilr_v <- ilr(Y, V = V_contrast)
```

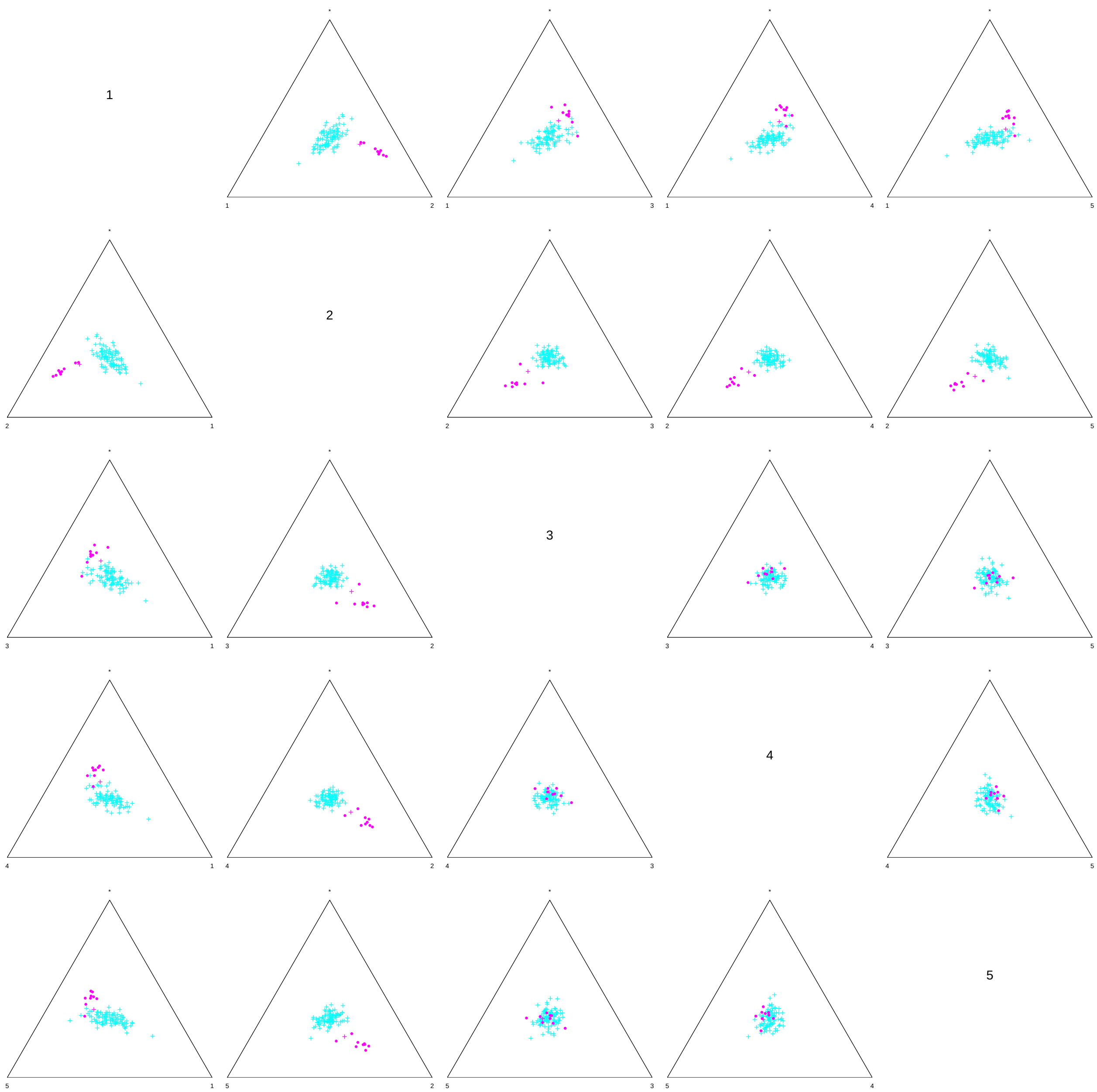
- We use the package `ICSOutlier` to detect outliers (Figure 4):

```
require("ICSOutlier")
my_ics <- ics2(my_ilr_v)
icsOutlier <- ics.outlier(my_ics,
          level.dist = 0.1  , mDist = 50, ncores = 1)
# pdf(file = "figures/big_data/big_data_ics.pdf", width = 4, height = 4)
op <- par(oma = c(0, 0, 0, 0), mar = c(3.3, 3.3, 1, 0.7),
          las = 1, mgp = c(2.2, 1, 0))
plot(icsOutlier@ics.distances, main = "",
         pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 3),
     col = c(rep("cyan", 90),
             rep("magenta", 10)),
     xlab = "Observation Number", ylab = "ICS distances", cex.lab = 1.1)
abline(h = icsOutlier@ics.dist.cutoff)
```




```
# dev.off()
```

```
Y <- rbind(Y_s_1, Y_s_2)
require(compositions)
# pdf(file = "figures/big_data/big_data_with_outliers_a.pdf", width = 8, height = 8)
op <- par(oma = c(1, 1, 1.5, 1), mar = c(0, 0.7, 1.5, 0.7))
plot(acomp(Y, 1:5),
         pch = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, 16, 3),
     col = c(rep("cyan", 90), rep("magenta", 10)))
```




```
# dev.off()
```

- To get the eigenvectors :
  - we compute the centered data in coordinate space

```
mean_v <- apply(my_ilr_v, 2, mean)
my_ilr_v_centre <- my_ilr_v - matrix(mean_v, byrow = T,
                                     nrow(my_ilr_v), ncol(my_ilr_v))
ics.ilr.v_centre <- ics(my_ilr_v_centre, stdB = "Z", stdKurt = F)
V_star_centre <- solve(ics.ilr.v_centre@UnMix)
```

  - we compute the eigenvalues

```
ics.ilr.v_centre@gKurt
```

```
##  [1] 1.2400340 1.1613024 1.1309494 1.1135123 1.0795038 1.0503802 1.0155113
##  [8] 1.0072303 0.9592308 0.9457001 0.9331142 0.9075813 0.8840836 0.8682124
```

```
## [15] 0.8640172 0.8499717 0.8391100 0.8262162 0.7804047
```

– Then we go back to the simplex and print the eigenvectors

```
a_1 <- compositions::ilrInv(V_star_centre[, 1], V = V_contrast)
a_2 <- compositions::ilrInv(V_star_centre[, 2], V = V_contrast)
cbind(a_1, a_2)
```

```
##           a_1        a_2
## 1  0.04064691 0.05786777
## 2  0.06327009 0.05615647
## 3  0.05012660 0.05219352
## 4  0.04819536 0.05226582
## 5  0.05193086 0.05046730
## 6  0.04941494 0.05192770
## 7  0.04858286 0.05162847
## 8  0.05022844 0.04702526
## 9  0.04881702 0.04690670
## 10 0.05136570 0.04767055
## 11 0.05084744 0.04927472
## 12 0.04953715 0.04909180
## 13 0.05031566 0.04904769
## 14 0.04897820 0.04730043
## 15 0.04997317 0.04970391
## 16 0.05070258 0.04738739
## 17 0.04818029 0.05067836
## 18 0.05001876 0.04699503
## 19 0.04925115 0.04971886
## 20 0.04961682 0.04669224
```

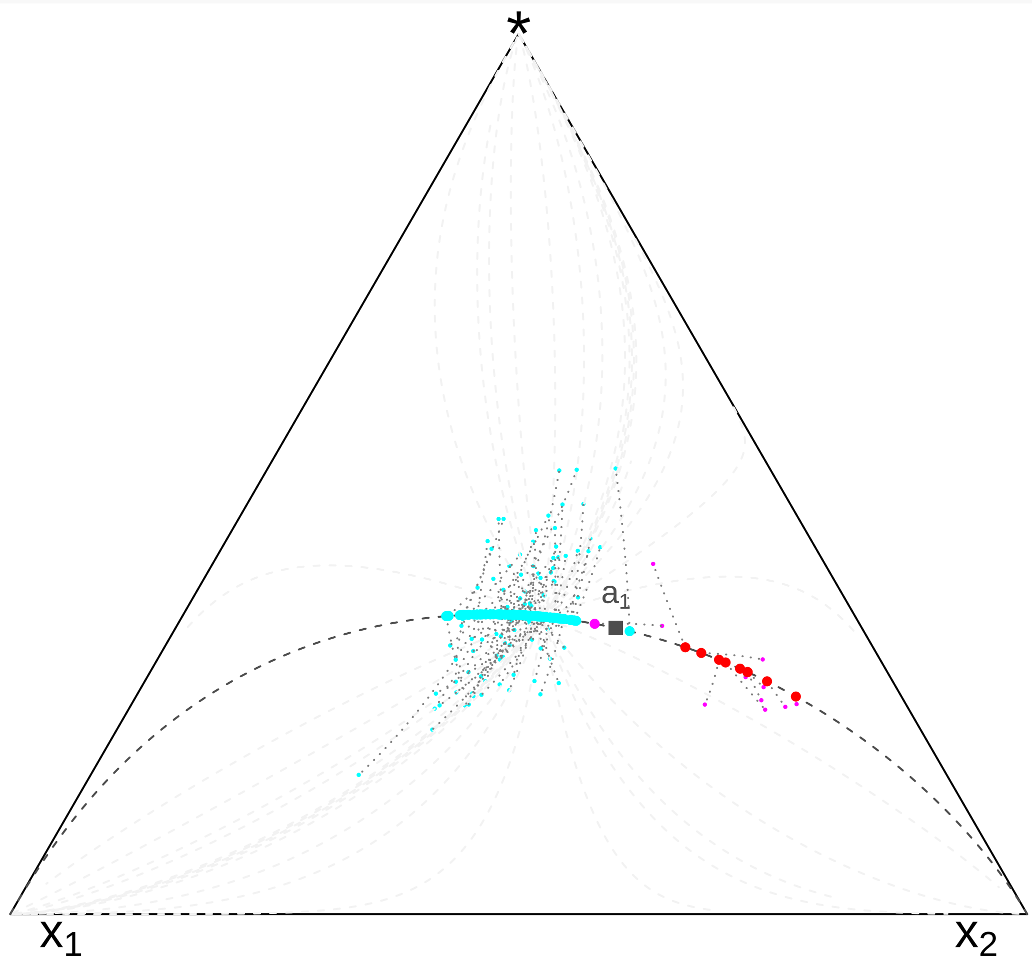
– We compute the scores multiplied by the a_1 ICS component transformed in the simplex. We represent them in the ternary diagram with vertices x_1, x_2 and the rest of the components (Figure 5 in the article)

```
# representation
A <- c(0, 0)
B <- c(1, 0)
C <- c(0.5, sqrt(3) / 2)
Y_s <- as(compositions::ilrInv(my_ilr_v, V = V_contrast), "matrix")
Y_s <- acompmargin(Y_s, c(1, 2))

Y_simplex_x <- Y_s[, 1] * A[1] + Y_s[, 2] * B[1] + Y_s[, 3] * C[1]
Y_simplex_y <- Y_s[, 1] * A[2] + Y_s[, 2] * B[2] + Y_s[, 3] * C[2]

#   pdf(file = "figures/big_data_a1.pdf", width = 6, height = 6)
op <- par(oma = c(0, 0, 0, 0), mar = c(0, 0.7, .5, 0.7))

Y_v <- ics.ilr.v_centre@Scores[, 1] * a_1  +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_a1 <- acompmargin(Y_v, c(1, 2))
Y_simplex_x_V <- Y_a1[, 1] * A[1] + Y_a1[, 2] * B[1] + Y_a1[, 3] * C[1]
Y_simplex_y_V <- Y_a1[, 1] * A[2] + Y_a1[, 2] * B[2] + Y_a1[, 3] * C[2]

lines_a1 <- seq(-100, 100, length.out = 1000) * a_1  +
  compositions::ilrInv(mean_v, V = V_contrast)
Y_a1_lines <- acompmargin(lines_a1, c(1, 2))

lines_simplex_x_V <- Y_a1_lines[, 1] * A[1] + Y_a1_lines[, 2] * B[1] +
```

```
  Y_a1_lines[, 3] * C[1]
lines_simplex_y_V <- Y_a1_lines[, 1] * A[2] + Y_a1_lines[, 2] * B[2] +
  Y_a1_lines[, 3] * C[2]

a_1_nc <- a_1 + compositions::ilrInv(mean_v, V = V_contrast)
a_1_3d <- acompmargin(a_1_nc, c(1, 2))
a1_simplex_x <- a_1_3d[1] * A[1] + a_1_3d[2] * B[1] + a_1_3d[3] * C[1]
a1_simplex_y <- a_1_3d[1] * A[2] + a_1_3d[2] * B[2] + a_1_3d[3] * C[2]

# ICS 1
plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
 type = "n", xlab = "", ylab = "", asp = 1, main = "",
 ylim = c(-0.1, sqrt(3)/2), cex.main = 2)

points(Y_simplex_x, Y_simplex_y, col = c(rep("cyan", 90), rep("magenta", 10)),
       cex = 0.3, pch = 16)

text(c(0.05, 0.95, 1/2), c(0-0.06, -0.06, sqrt(3)/2-0.04),
 c(expression(x[1]), expression(x[2]), "*"), pos = 3, cex = c(1.5, 1.5, 2))
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

   for(k in 2:19) {
 v_k <- compositions::ilrInv(V_star_centre[, k], V = V_contrast)
 lines_vk <- seq(-100, 100, length.out = 1000) * v_k  +
 compositions::ilrInv(mean_v, V = V_contrast)
 Y_vk_lines <- acompmargin(lines_vk, c(1, 2))

 lines_simplex_x_V_k <- Y_vk_lines[, 1] * A[1] + Y_vk_lines[, 2] * B[1] +
   Y_vk_lines[, 3] * C[1]
 lines_simplex_y_V_k <- Y_vk_lines[, 1] * A[2] + Y_vk_lines[, 2] * B[2] +
   Y_vk_lines[, 3] * C[2]

 lines(lines_simplex_x_V_k, lines_simplex_y_V_k, lty = 2, col = rgb(0.95, 0.95, 0.95))
   }

 for(k in 1:100) {
  xo_1 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_2 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_3 <- V_star_centre[3, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_4 <- V_star_centre[4, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_5 <- V_star_centre[5, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_6 <- V_star_centre[6, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_7 <- V_star_centre[7, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_8 <- V_star_centre[8, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_9 <- V_star_centre[9, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_10 <- V_star_centre[10, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_11 <- V_star_centre[11, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_12 <- V_star_centre[12, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_13 <- V_star_centre[13, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_14 <- V_star_centre[14, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_15 <- V_star_centre[15, 1] * ics.ilr.v_centre@Scores[k, 1]
```

```
xo_16 <- V_star_centre[16, 1] * ics.ilr.v_centre@Scores[k, 1]
xo_17 <- V_star_centre[17, 1] * ics.ilr.v_centre@Scores[k, 1]
xo_18 <- V_star_centre[18, 1] * ics.ilr.v_centre@Scores[k, 1]
xo_19 <- V_star_centre[19, 1] * ics.ilr.v_centre@Scores[k, 1]

xd_1 <- my_ilr_v_centre[k, 1]
xd_2 <- my_ilr_v_centre[k, 2]
xd_3 <- my_ilr_v_centre[k, 3]
xd_4 <- my_ilr_v_centre[k, 4]
xd_5 <- my_ilr_v_centre[k, 5]
xd_6 <- my_ilr_v_centre[k, 6]
xd_7 <- my_ilr_v_centre[k, 7]
xd_8 <- my_ilr_v_centre[k, 8]
xd_9 <- my_ilr_v_centre[k, 9]
xd_10 <- my_ilr_v_centre[k, 10]
xd_11 <- my_ilr_v_centre[k, 11]
xd_12 <- my_ilr_v_centre[k, 12]
xd_13 <- my_ilr_v_centre[k, 13]
xd_14 <- my_ilr_v_centre[k, 14]
xd_15 <- my_ilr_v_centre[k, 15]
xd_16 <- my_ilr_v_centre[k, 16]
xd_17 <- my_ilr_v_centre[k, 17]
xd_18 <- my_ilr_v_centre[k, 18]
xd_19 <- my_ilr_v_centre[k, 19]

seq_simplex <- compositions::ilrInv(cbind(
  seq(xo_1, xd_1, length.out = 100),
  seq(xo_2, xd_2, length.out = 100),
  seq(xo_3, xd_3, length.out = 100),
  seq(xo_4, xd_4, length.out = 100),
  seq(xo_5, xd_5, length.out = 100),
  seq(xo_6, xd_6, length.out = 100),
  seq(xo_7, xd_7, length.out = 100),
  seq(xo_8, xd_8, length.out = 100),
  seq(xo_9, xd_9, length.out = 100),
  seq(xo_10, xd_10, length.out = 100),
  seq(xo_11, xd_11, length.out = 100),
  seq(xo_12, xd_12, length.out = 100),
  seq(xo_13, xd_13, length.out = 100),
  seq(xo_14, xd_14, length.out = 100),
  seq(xo_15, xd_15, length.out = 100),
  seq(xo_16, xd_16, length.out = 100),
  seq(xo_17, xd_17, length.out = 100),
  seq(xo_18, xd_18, length.out = 100),
  seq(xo_19, xd_19, length.out = 100)),
   V = V_contrast) + compositions::ilrInv(mean_v, V = V_contrast)

seq_simplex <- acompmargin(seq_simplex, c(1, 2))

seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
  seq_simplex[, 3] * C[1]
seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
  seq_simplex[, 3] * C[2]
```

```
  lines(seq_simplex_x, seq_simplex_y, col = rgb(0.5, 0.5, 0.5), lwd = 1, lty = 3)
 }

lines(lines_simplex_x_V, lines_simplex_y_V, lty = 2, col = rgb(0.3, 0.3, 0.3))
points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.3, 0.3, 0.3), cex = 1)

points(Y_simplex_x_V, Y_simplex_y_V,
   pch = 16, col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff,
                          "red", c(rep("cyan", 90), rep("magenta", 10))),
   cex = 0.7)

text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3, col = rgb(0.3, 0.3, 0.3))
```




```
par(op)
#   dev.off()
```

2 Market Share Example

This section contains the **R** codes that allow to reproduce results and figures presented in Section 6.2 in the article.

We first import the data:

```r
library(codareg)
data(BDDSegX)
Y_s <- BDDSegX[, c("S_A", "S_B", "S_C", "S_D", "S_E")]
colnames(Y_s) <- c("A", "B", "C", "D", "E")
```

We have 5 components: A-B-C-D-E segment market share (in terms of sales volumes) simulated from a model (fitted on real data). We transform the data in the ilr space:

```r
V_contrast <- compositions::ilrBase(D = 5)
my_ilr_v <- ilr(Y_s, V = V_contrast)
```

We use package `ICSOutlier` to detect outliers:

```r
require("ICSOutlier")
my_ics <- ics2(my_ilr_v)
icsOutlier <- ics.outlier(my_ics,
          level.dist = 0.05, mDist = 50, ncores = 1)
```

To get the eigenvectors :

- we compute the centered data in coordinate space

  ```r
  mean_v <- apply(my_ilr_v, 2, mean)
  my_ilr_v_centre <- my_ilr_v - matrix(mean_v, byrow = T,
                                       nrow(my_ilr_v), ncol(my_ilr_v))
  ```

- we run the ics and take the inverse of the unmixed matrix

  ```r
  ics.ilr.v_centre <- ics(my_ilr_v_centre, stdB = "Z", stdKurt = F)
  V_star_centre <- solve(ics.ilr.v_centre@UnMix)
  ```

- we compute the eigenvalues

  ```r
  ics.ilr.v_centre@gKurt
  ```

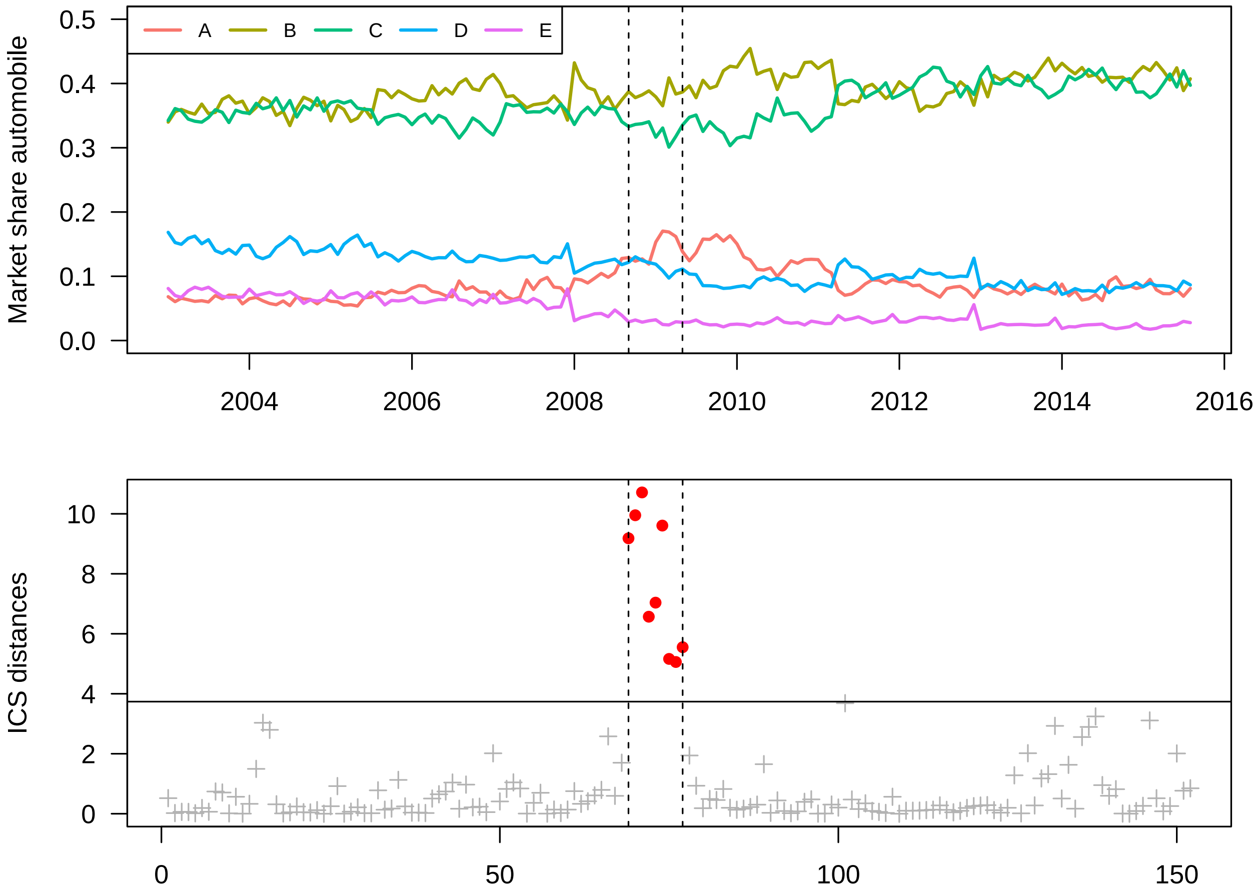
  ```
  ## [1] 1.1443544 0.9823742 0.9295889 0.6523845
  ```

- Then we go back to the simplex and print the eigenvectors

  ```r
  a_1_ilr <- V_star_centre[, 1]
  a_2_ilr <- V_star_centre[, 2]
  a_1 <- compositions::ilrInv(a_1_ilr, V = V_contrast)
  a_2 <- compositions::ilrInv(a_2_ilr, V = V_contrast)
  cbind(a_1, a_2)
  ```

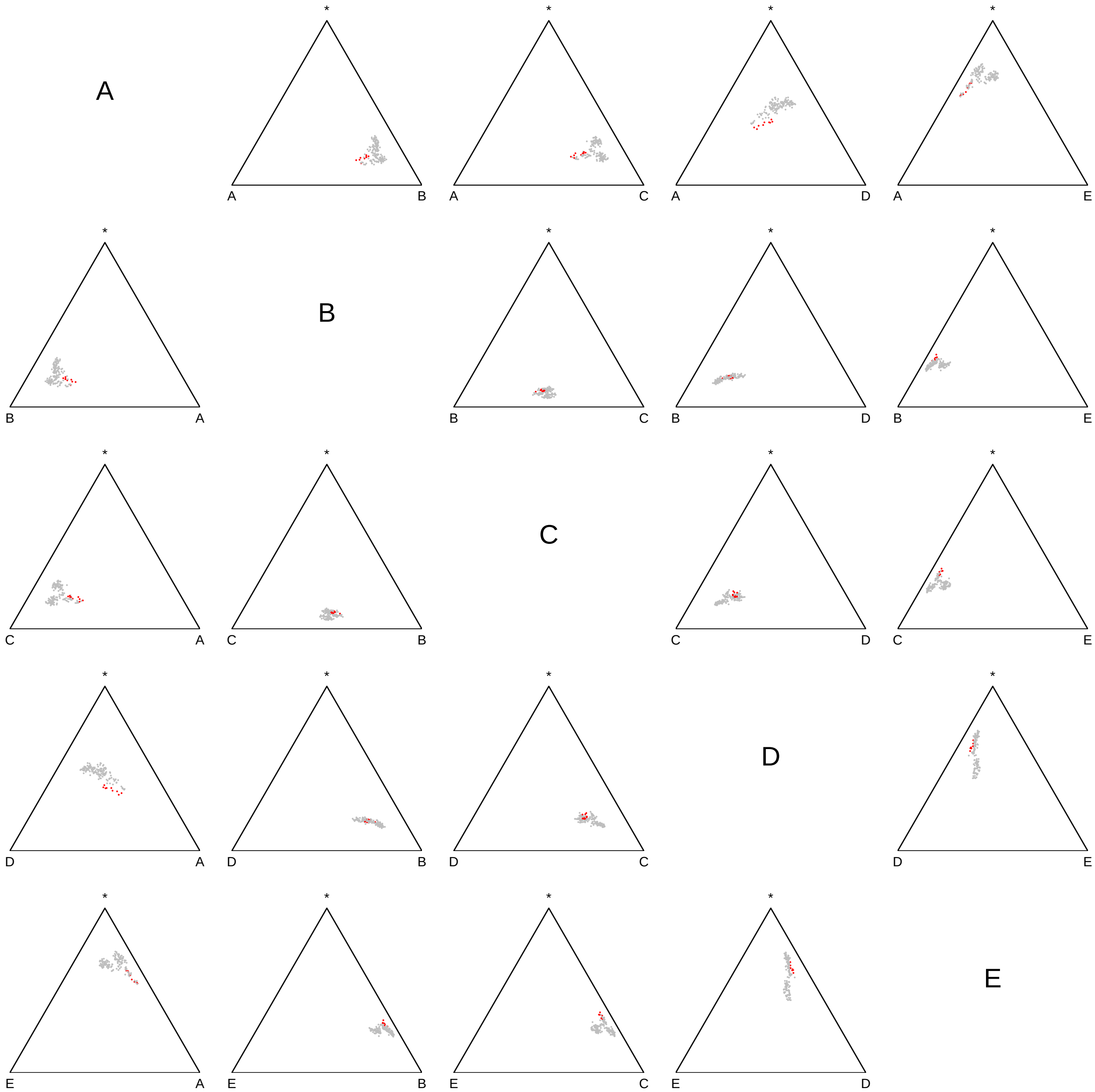
  ```
  ##         a_1       a_2
  ## 1 0.2331270 0.2326111
  ## 2 0.1965259 0.1952448
  ## 3 0.1922815 0.1884713
  ## 4 0.2021838 0.1892105
  ## 5 0.1758819 0.1944623
  ```

To get Figure 6 in the article



To get Figure 7 in the article :

```
# representation
A <- c(0, 0)
B <- c(1, 0)
C <- c(0.5, sqrt(3) / 2)

# pdf(file = "figures/ilr_to_ternary_sima_extra.pdf", width = 10, height = 10)
op <- par(oma = c(1, 1, 1.5, 1), mar = c(0, 0.7, 1.5, 0.7))
plot(acomp(Y_s), margin = "acomp", pch = 16, cex = 0.3,
     col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff, "red", "grey"))
```




```
par(op)
# dev.off()
```

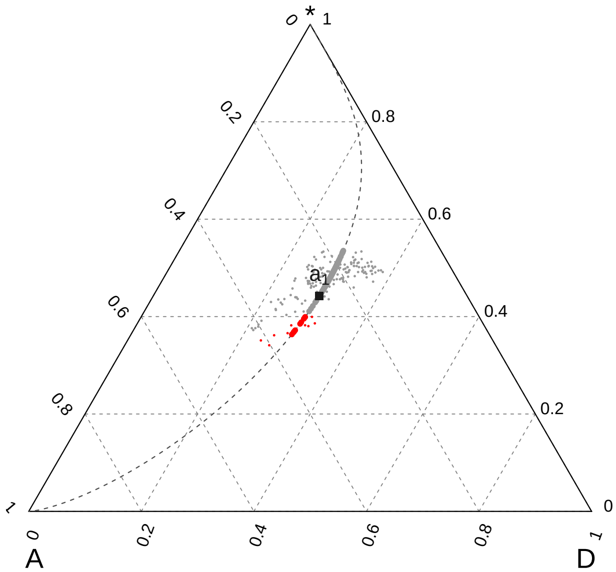
Figure 8 in the article: French Market automobile shares example: projection of the data on the first ICS axis a1 in the sub-ternary diagram defined by A, D, and the amalgamation of others components (left). Zoom on the interesting part of the ternary diagram (right)

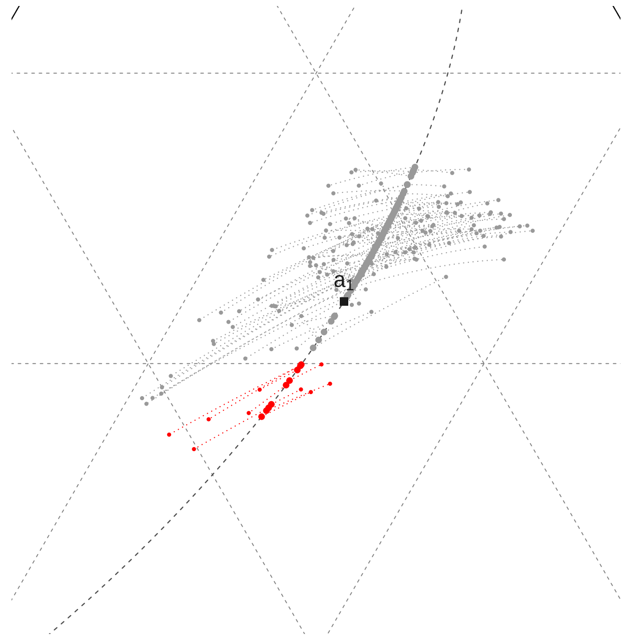
```
     # representation
    A <- c(0, 0)
    B <- c(1, 0)
    C <- c(0.5, sqrt(3) / 2)

    # original data
    Y_s_comp <- acomp(Y_s)
    Y_s_r <- acompmargin(Y_s_comp, c(1, 4))
```

```r
    Ys_simplex_x <- Y_s_r[, 1] * A[1] + Y_s_r[, 2] * B[1] + Y_s_r[, 3] * C[1]
    Ys_simplex_y <- Y_s_r[, 1] * A[2] + Y_s_r[, 2] * B[2] + Y_s_r[, 3] * C[2]

# pdf(file = "figures/market_share/market_data_a1_c.pdf", width = 12, height = 6)
    op <- par(oma = c(0, 1, 0, 1), mar = c(0, 0.7, 1.5, 0.7), mfrow = c(1, 2))

    Y_v <- ics.ilr.v_centre@Scores[, 1] * a_1  +
      compositions::ilrInv(mean_v, V = V_contrast)
    Y_a1 <- acompmargin(Y_v, c(1, 4))
    Y_a1 <- acompmargin(Y_v, c(1, 4))
    Y_simplex_x_V <- Y_a1[, 1] * A[1] + Y_a1[, 2] * B[1] + Y_a1[, 3] * C[1]
    Y_simplex_y_V <- Y_a1[, 1] * A[2] + Y_a1[, 2] * B[2] + Y_a1[, 3] * C[2]

    lines_a1 <- seq(-100, 100, length.out = 1000) * a_1  +
      compositions::ilrInv(mean_v, V = V_contrast)
    Y_a1_lines <- acompmargin(lines_a1, c(1, 4))
    Y_a1_lines <- acompmargin(lines_a1, c(1, 4))

    lines_simplex_x_V <- Y_a1_lines[, 1] * A[1] + Y_a1_lines[, 2] * B[1] +
      Y_a1_lines[, 3] * C[1]
    lines_simplex_y_V <- Y_a1_lines[, 1] * A[2] + Y_a1_lines[, 2] * B[2] +
      Y_a1_lines[, 3] * C[2]

    a_1_c <- a_1  + compositions::ilrInv(mean_v, V = V_contrast)
    a_1_3d <- acompmargin(a_1_c, c(1, 4))
    a_1_3d <- acompmargin(a_1_c, c(1, 4))
    a1_simplex_x <- a_1_3d[1] * A[1] + a_1_3d[2] * B[1] + a_1_3d[3] * C[1]
    a1_simplex_y <- a_1_3d[1] * A[2] + a_1_3d[2] * B[2] + a_1_3d[3] * C[2]

    # NO ZOOM
    plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
      type = "n", xlab = "", ylab = "", asp = 1, main = "",
      cex.main = 2, ylim = c(-0.1, sqrt(3)/2))

    # XR lines
    for (k in 0:5) {
      lines(c(k/10, 1-k/10), c(k/5 * C[2], k/5 * C[2]), lty = 2,
            lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
      text(1.03-k/10, k/5 * C[2]-0.02,  2 * k/10, pos = 3, cex = 0.9)
    }

    # Left lines
    for (k in 0:5) {
      lines(c(k/5, 0.5 + k * 1/10), c(0, C[2] - k/5 * C[2]),
            lty = 2, lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
      text(k/5, 0-0.02, 2 * k/10, pos = 1, cex = 0.9, srt = 70)
    }
    # Right lines
    for (k in 0:5) {
      lines(c(k * 1 / 10, k/5), c(k/5 * C[2], 0), lty = 2,
            lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
      text(k * 1 / 10-0.01, k/5 * C[2], 1 - 2 * k/10,
           pos = 2, cex = 0.9, srt = -50)
```

```r
}

points(Ys_simplex_x, Ys_simplex_y, pch = 16,
      col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff,
      "red", rgb(0.6, 0.6, 0.6)), cex = 0.3)

text(c(0.01, 0.99, 1/2), c(0-0.12, -0.12, sqrt(3)/2-0.02),
 c("A", "D", "*"), pos = 3, cex = c(1.5, 1.5, 1.5))
lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

lines(lines_simplex_x_V, lines_simplex_y_V, lty = 2, col = rgb(0.3, 0.3, 0.3))

points(Y_simplex_x_V, Y_simplex_y_V,
   pch = 16, col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff,
                         "red", rgb(0.6, 0.6, 0.6)), cex = 0.7)

points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.1, 0.1, 0.1), cex = 1)
text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3, col = rgb(0.1, 0.1, 0.1),
     cex = 1.2)

# with ZOOM
plot(rbind(A, B, C), xaxt = "n", yaxt = "n", frame = F,
 type = "n", xlab = "", ylab = "", asp = 1, main = "",
 cex.main = 2, # ylim = c(-0.1, sqrt(3)/2)
 ylim = c(C[2] * 0.23, C[2] * 0.63), xlim = c(0.4, 0.6))

# XR lines
for (k in 0:5) {
  lines(c(k/10, 1-k/10), c(k/5 * C[2], k/5 * C[2]), lty = 2,
        lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
  text(1.03-k/10, k/5 * C[2], paste0("B+C+E=", 2 * k/10),
       pos = 3, cex = 0.7)
}

# Left lines
for (k in 0:5) {
  lines(c(k/5, 0.5 + k * 1/10), c(0, C[2] - k/5 * C[2]), lty = 2,
        lwd = 0.8, col = rgb(0.5, 0.5, 0.5))
}
# Right lines
for (k in 0:5) {
  lines(c(k * 1 / 10, k/5), c(k/5 * C[2], 0), lty = 2, lwd = 0.8,
        col = rgb(0.5, 0.5, 0.5))
}
text(0.42, 0.65, "A=0.2", pos = 2, cex = 0.7, srt = -60)
text(0.57, 0.6, "D=0.2", pos = 2, cex = 0.7, srt = 65)

points(Ys_simplex_x, Ys_simplex_y, pch = 16,
      col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff,
      "red", rgb(0.6, 0.6, 0.6)), cex = 0.5)
```

```
  for(k in 1:nrow(my_ilr_v)) {
  xo_1 <- V_star_centre[1, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_2 <- V_star_centre[2, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_3 <- V_star_centre[3, 1] * ics.ilr.v_centre@Scores[k, 1]
  xo_4 <- V_star_centre[4, 1] * ics.ilr.v_centre@Scores[k, 1]
  xd_1 <- my_ilr_v_centre[k, 1]
  xd_2 <- my_ilr_v_centre[k, 2]
  xd_3 <- my_ilr_v_centre[k, 3]
  xd_4 <- my_ilr_v_centre[k, 4]

  x1_seq <- seq(xo_1, xd_1, length.out = 100)
  x2_seq <- seq(xo_2, xd_2, length.out = 100)
  x3_seq <- seq(xo_3, xd_3, length.out = 100)
  x4_seq <- seq(xo_4, xd_4, length.out = 100)

  seq_simplex <- compositions::ilrInv(cbind(x1_seq, x2_seq, x3_seq, x4_seq),
                                      V = V_contrast)  +
    compositions::ilrInv(mean_v, V = V_contrast)

  seq_simplex <- acompmargin(seq_simplex, c(1, 4))

  seq_simplex_x <- seq_simplex[, 1] * A[1] + seq_simplex[, 2] * B[1] +
    seq_simplex[, 3] * C[1]
  seq_simplex_y <- seq_simplex[, 1] * A[2] + seq_simplex[, 2] * B[2] +
    seq_simplex[, 3] * C[2]

  lines(seq_simplex_x, seq_simplex_y,
        col = ifelse(icsOutlier@ics.distances[k] > icsOutlier@ics.dist.cutoff,
                     "red", rgb(0.6, 0.6, 0.6)), lwd = 1, lty = 3)
 }

lines(c(0, 1), c(0, 0), col = "black")
lines(c(0, 0.5), c(0, sqrt(3)/2), col = "black")
lines(c(1, 0.5), c(0, sqrt(3)/2), col = "black")

lines(lines_simplex_x_V, lines_simplex_y_V, lty = 2, col = rgb(0.3, 0.3, 0.3))

points(Y_simplex_x_V, Y_simplex_y_V,
   pch = 16, col = ifelse(icsOutlier@ics.distances > icsOutlier@ics.dist.cutoff,
                "red", rgb(0.6, 0.6, 0.6)), cex = 0.8)

points(a1_simplex_x, a1_simplex_y, pch = 15, col = rgb(0.1, 0.1, 0.1), cex = 1)
text(a1_simplex_x, a1_simplex_y, expression(a[1]), pos = 3, col = rgb(0.1, 0.1, 0.1),
     cex = 1.2)
par(op)
title("")
```






```
# dev.off()
```