FABIA: Factor Analysis for Bicluster Acquisition
— *Examples how to use FABIA* —

Sepp Hochreiter

Institute of Bioinformatics, Johannes Kepler University Linz
Altenberger Str. 69, 4040 Linz, Austria
hochreit@bioinf.jku.at

Version 0.1.3, December 30, 2009

Contents

1 Example 1

Now we demonstrate how to use the package `fabia` on an example with output and plots. The data has been constructed using the package `fabia` by:

```
> n=200
> l=100
> p=4
> dat <- make_fabi_data_blocks(n = n,l= l,p = p,f1 = 5,f2 = 5,of1 = 5,of2 = 10,
+ sd_noise = 3.0,sd_z_noise = 0.2,mean_z = 2.0, sd_z = 1.0,sd_l_noise = 0.2,
+ mean_l = 3.0,sd_l = 1.0)
> X <- dat[[1]]
> ZC <- dat[[3]]
> LC <- dat[[4]]
> gclab <- rep.int(0,l)
> gllab <- rep.int(0,n)
> clab <- as.character(1:l)
> llab <- as.character(1:n)
> for (i in 1:p){
 +   for (j in ZC[i]){
 +       clab[j] <- paste(as.character(i),"_",clab[j],sep="")
 +   }
 +   for (j in LC[i]){
 +       llab[j] <- paste(as.character(i),"_",llab[j],sep="")
 +   }
 +   gclab[unlist(ZC[i])] <- gclab[unlist(ZC[i])] + p^i
 +   gllab[unlist(LC[i])] <- gllab[unlist(LC[i])] + p^i
 + }
> groups <- gclab
> write.csv(X, file = "datafile.csv")
> write.table(groups, file = "groups.txt")
> write.table(clab, file = "clabels.txt")
> write.table(llab, file = "rlabels.txt")
```

We stored the data to keep the same data in each Sweave run.

1. Load the library:

```
> library(fabia)

+----------------------------+
|............................|
|............................|
|..............########......|  #######    #    ######    ###      #
|..............########......|  #         # #   #     #    #      # #
|.....####.....########......|  #        #   #  #     #    #     #   #
|.....####.....########......|  #####   #     # ######     #    #     #
|.....####...................|  #       ####### #     #    #    #######
|.....####...........###.....|  #       #     # #     #    #    #     #
|....................###.....|  #       #     # ######    ###   #     #
|....................###.....|
|............................|
+----------------------------+
FABIA: Factor Analysis for Bicluster Acquisition
Changes in FABIA:
Version 0.1.1: first version (paper).
Version 0.1.2: centering, mode estimation.
Version 0.1.3: S4 classes, plot, summary, show.

For updates please check http://www.bioinf.jku.at/software/fabia/fabia.html
```

2. Read the data file “datafile.csv” and labels:

```
> X <- read.table("datafile.csv", header = TRUE, sep = ",")
> X <- as.matrix(X[, -1])
> groups <- read.table(file = "groups.txt")
> clab <- read.table(file = "clabels.txt")
> llab <- read.table(file = "rlabels.txt")
> colnames(X) <- 1:ncol(X)
> rownames(X) <- 1:nrow(X)
```

3. Select the model based on the data: 4 biclusters; sparseness 0.1; 400 cycles

```
> res <- fabia(X, 4, 0.1, 400, 1, 1)

Running FABIA on  200 x 100  matrix with
   Number of biclusters --------------- p:  4
   Sparseness factor --------------- alpha:  0.1
   Number of iterations -------------- cyc:  400
   Loading prior parameter ----------- spl:  1
   Factor prior parameter ------------ spz:  1
   Initialization interval -------- random:  1
   Centering ---------------------- center:  2  = median
   Scaling to variance one: --------- norm:  1  = Yes
```

```
   Scaling loadings per iteration -- scale:  0  = No
   Constraint variational parameter -- lap:  1
Cycle: 0
Cycle: 20
Cycle: 40
Cycle: 60
Cycle: 80
Cycle: 100
Cycle: 120
Cycle: 140
Cycle: 160
Cycle: 180
Cycle: 200
Cycle: 220
Cycle: 240
Cycle: 260
Cycle: 280
Cycle: 300
Cycle: 320
Cycle: 340
Cycle: 360
Cycle: 380
```

4. Show information content of the biclusters:

   ```
   > res@avini
   ```

   ```
   [1]  83.39424  84.55786  97.26839  63.55998 328.77310
   ```

5. Give summary:

   First, it gives the number or rows and columns of the original matrix and then the number of clusters. Then for the row cluster the information content is given. Then for each column its information content is given. Then for each cluster a column summary is given by a boxplot. Finally, for each cluster a row summary is given by a boxplot.

   ```
   > summary(res)
   ```

   ```
   An object of class Factorization

   call:
           "fabia"

   Number of rows:  200

   Number of columns:  100

   Number of clusters:  4
   ```

```
Information content of the clusters:
 BC 1    BC 2    BC 3    BC 4  BC sum
83.39   84.56   97.27   63.56  328.77



Information content of the samples:
 Sample 1    Sample 2    Sample 3    Sample 4    Sample 5    Sample 6
     5.79        6.30        8.27        7.07        7.17       12.06
 Sample 7    Sample 8    Sample 9   Sample 10   Sample 11   Sample 12
     7.11        5.76        9.78        9.73       12.59        5.38
Sample 13   Sample 14   Sample 15   Sample 16   Sample 17   Sample 18
     5.13        7.70        8.03        7.81        4.71       10.69
Sample 19   Sample 20   Sample 21   Sample 22   Sample 23   Sample 24
    11.00        6.50        9.58        4.49        6.89       13.77
Sample 25   Sample 26   Sample 27   Sample 28   Sample 29   Sample 30
     8.60        8.20        7.34        8.25        6.45        5.87
Sample 31   Sample 32   Sample 33   Sample 34   Sample 35   Sample 36
     5.40        8.23        6.53        3.68       10.03        3.97
Sample 37   Sample 38   Sample 39   Sample 40   Sample 41   Sample 42
     3.59        2.61        4.12        7.30        9.45       12.05
Sample 43   Sample 44   Sample 45   Sample 46   Sample 47   Sample 48
     5.25        5.79        6.71        6.85        1.46        8.84
Sample 49   Sample 50   Sample 51   Sample 52   Sample 53   Sample 54
     4.94        8.33        8.60        3.72        1.64        6.87
Sample 55   Sample 56   Sample 57   Sample 58   Sample 59   Sample 60
     2.24        8.04        5.70        3.96       10.35        0.95
Sample 61   Sample 62   Sample 63   Sample 64   Sample 65   Sample 66
     1.96        0.96        5.26        7.55        3.67        2.11
Sample 67   Sample 68   Sample 69   Sample 70   Sample 71   Sample 72
     6.39        6.07        6.86        4.16        4.68        9.65
Sample 73   Sample 74   Sample 75   Sample 76   Sample 77   Sample 78
    15.58        9.61        9.81       10.09        6.35        3.28
Sample 79   Sample 80   Sample 81   Sample 82   Sample 83   Sample 84
    13.85        8.00        8.02        6.67        6.84        6.82
Sample 85   Sample 86   Sample 87   Sample 88   Sample 89   Sample 90
    12.98        3.55        9.70        2.13        5.93        3.02
Sample 91   Sample 92   Sample 93   Sample 94   Sample 95   Sample 96
     5.24        3.05        5.62        2.29        3.92        3.93
Sample 97   Sample 98   Sample 99  Sample 100  Sample sum
     4.27        6.35        3.81        2.31      328.77

Column clusters / Factors:
BC 1              BC 2              BC 3              BC 4
Min.   :-1.1566   Min.   :-4.0084   Min.   :-1.3693   Min.   :-5.5285
```

```
 1st Qu.:-0.2591   1st Qu.:-0.0658   1st Qu.:-0.3100   1st Qu.:-0.0311  
 Median :-0.0234   Median : 0.0297   Median :-0.0215   Median : 0.0169  
 Mean   : 0.1373   Mean   :-0.2001   Mean   : 0.1641   Mean   :-0.0896  
 3rd Qu.: 0.0493   3rd Qu.: 0.2514   3rd Qu.: 0.1272   3rd Qu.: 0.2271  
 Max.   : 4.0055   Max.   : 1.1020   Max.   : 3.4562   Max.   : 1.9453  


 Row clusters / Loadings:
 BC 1              BC 2                BC 3              BC 4           
 Min.   :-0.7547   Min.   :-0.78254   Min.   :-0.7855   Min.   :-0.60071  
 1st Qu.:-0.0815   1st Qu.:-0.07069   1st Qu.:-0.0903   1st Qu.:-0.09105  
 Median : 0.0000   Median : 0.00000   Median : 0.0000   Median : 0.00000  
 Mean   : 0.0155   Mean   :-0.00120   Mean   :-0.0278   Mean   :-0.00874  
 3rd Qu.: 0.0901   3rd Qu.: 0.07902   3rd Qu.: 0.0676   3rd Qu.: 0.07569  
 Max.   : 0.7918   Max.   : 0.80556   Max.   : 0.3557   Max.   : 0.54532  
```

6. Plot some statistics:

   ```
   > show(res)
   ```

   The statistics are shown in Figure 1 to Figure 4. They give

   - (1) the information content of biclusters (Figure 1),
   - (2) the information content of samples (Figure 2),
   - (3) boxplots of loadings for each bicluster (row membership, Figure 3),
   - (4) boxplots of factors for each bicluster (column membership, Figure 4).

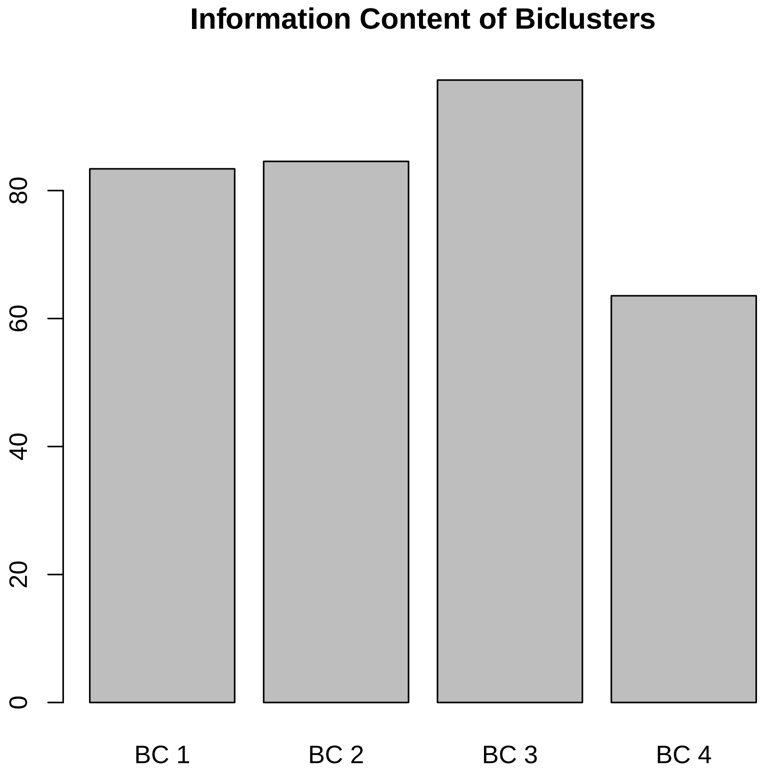

Figure 1: Information content of biclusters. The biclusters are ranked according to their information. More information means larger and less noisy biclusters.

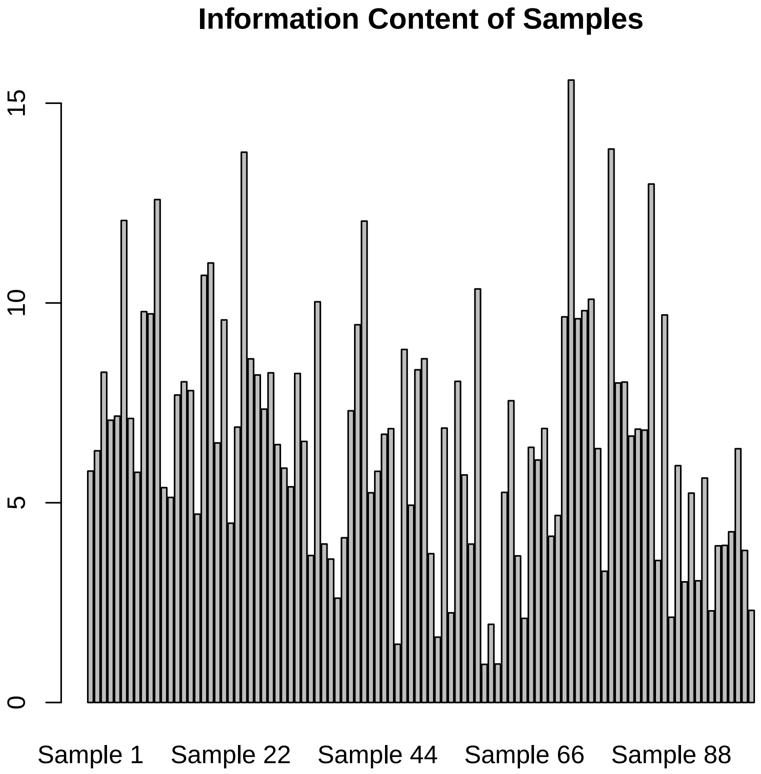

Figure 2: Information content of samples. The samples are ranked according to their information. More information means that the sample participates at more biclusters or has stronger signals.

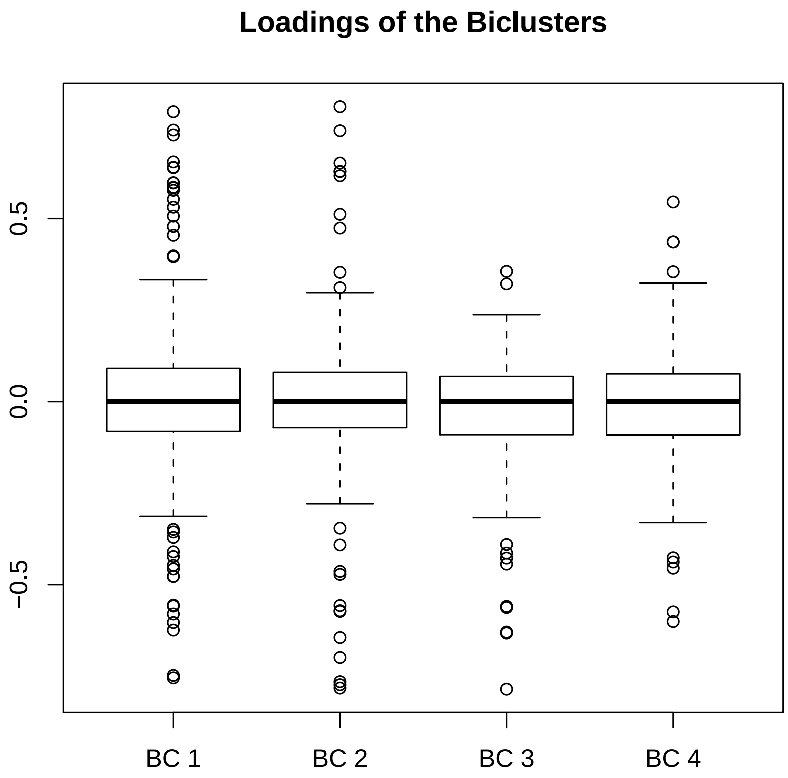

Figure 3: Boxplots of loadings for each bicluster (row membership). Gives the row size of the biclusters.

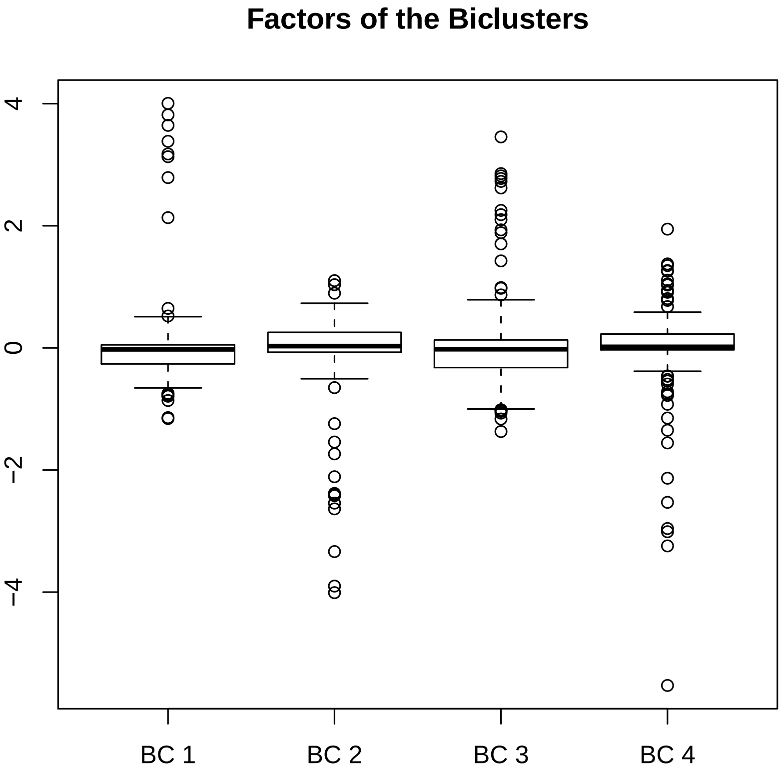

Figure 4: Boxplots of factors for each bicluster (column membership). Gives the column size of the biclusters. Biclusters may be driven by few samples and are not robust.

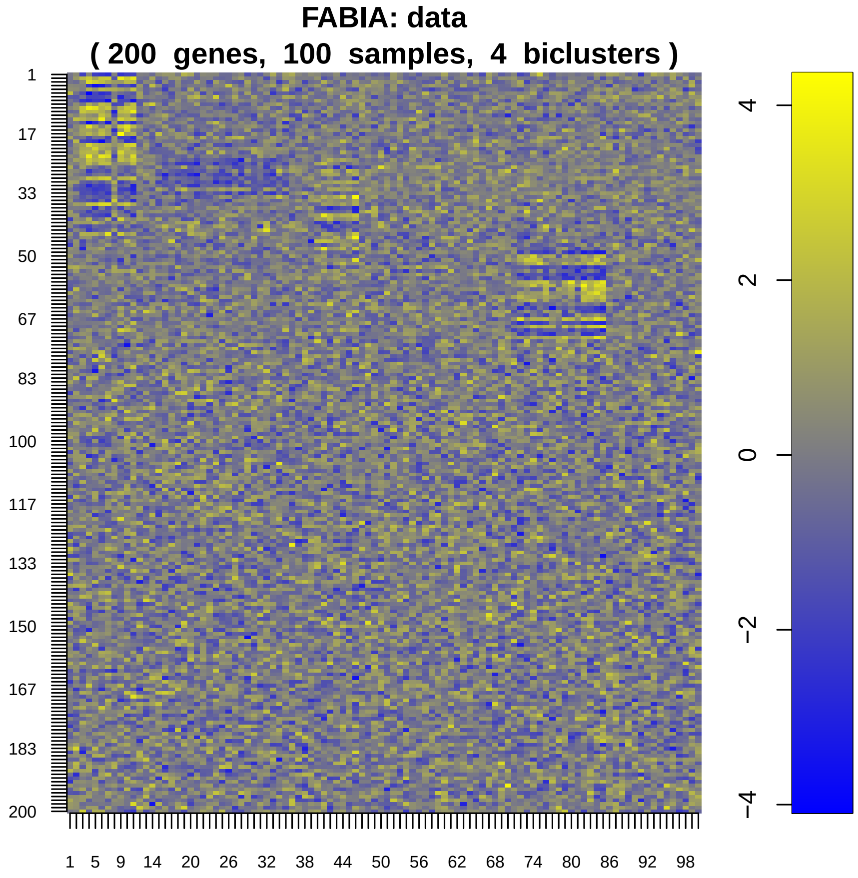

Figure 5: The data.

7. Plot the factorization results:

```
> rr <- extract_plot(res,ti="FABIA")
```

The results are shown in Figure 5 to Figure 11. They give

- (1) the original data (Figure 5),
- (2) the reconstructed data by FABIA (Figure 6),
- (3) the reconstruction error of FABIA (Figure 7),
- (4) the estimated absolute loadings (Figure 8),
- (5) the estimated absolute factors (Figure 9),
- (6) the reconstructed data matrix sorted according to k-means membership applied to factors/columns and loadings/rows (Figure 10),
- (7) the original data matrix sorted according to k-means membership applied to factors/columns and loadings/rows (Figure 11),

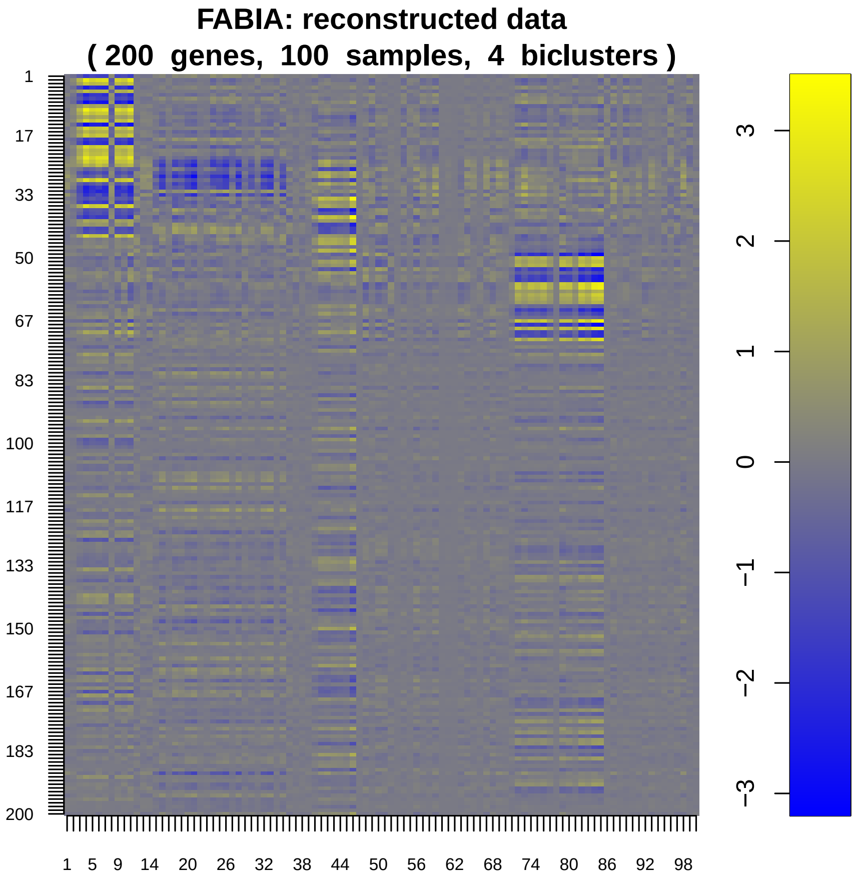

Figure 6: The reconstructed data by FABIA.

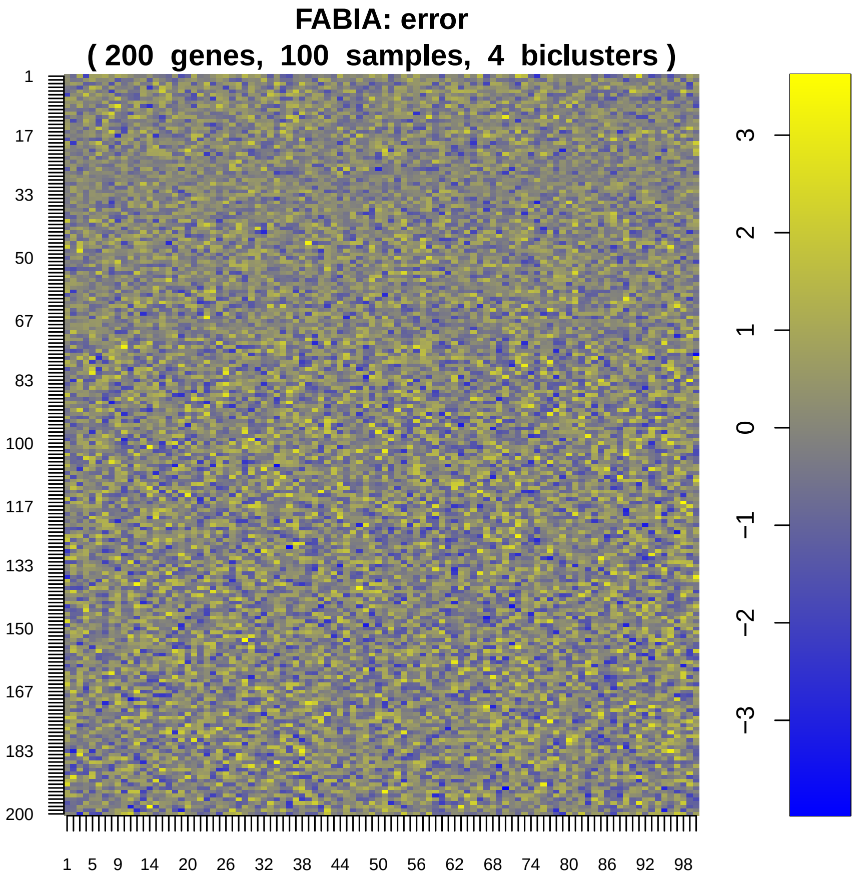

Figure 7: The reconstruction error of FABIA.

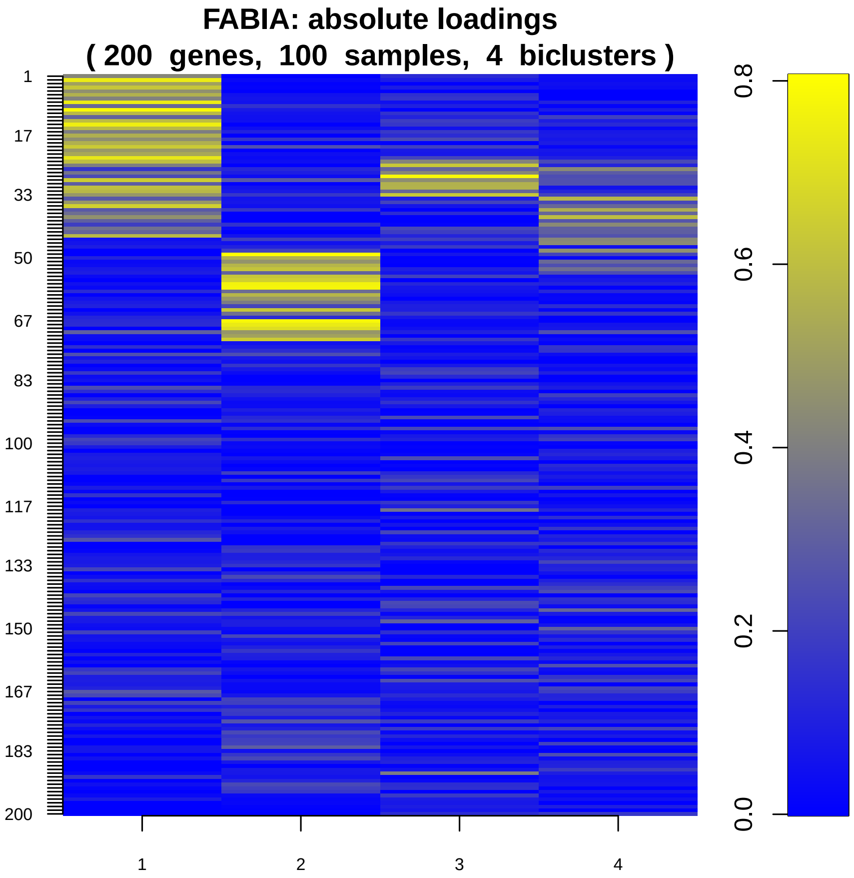

Figure 8: The estimated absolute loadings.

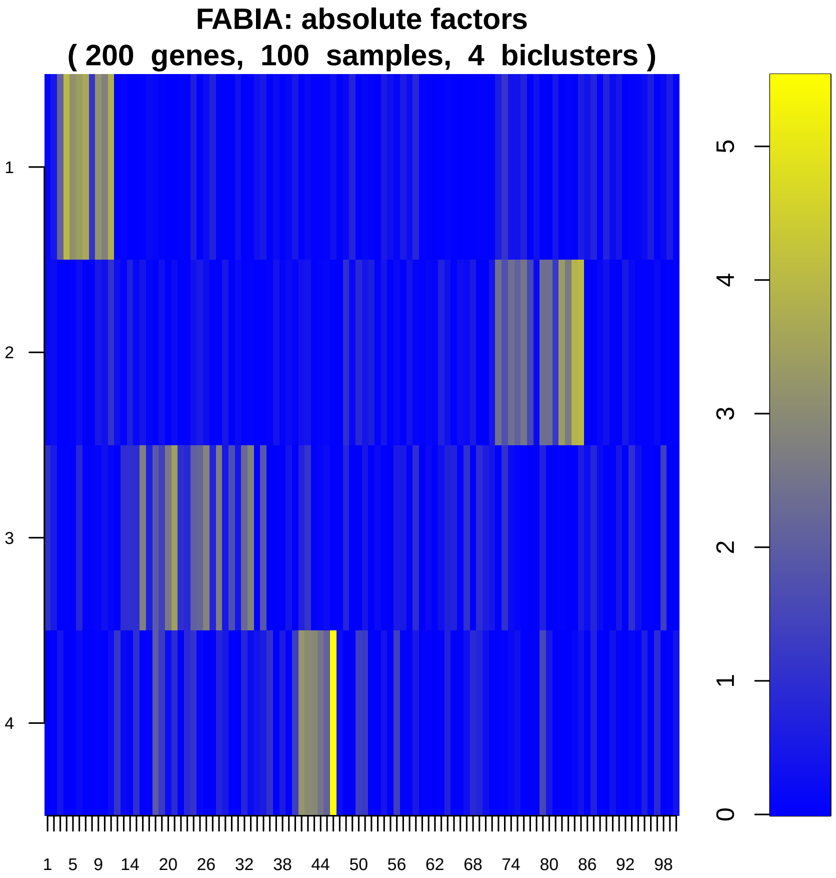

Figure 9: The estimated absolute factors.

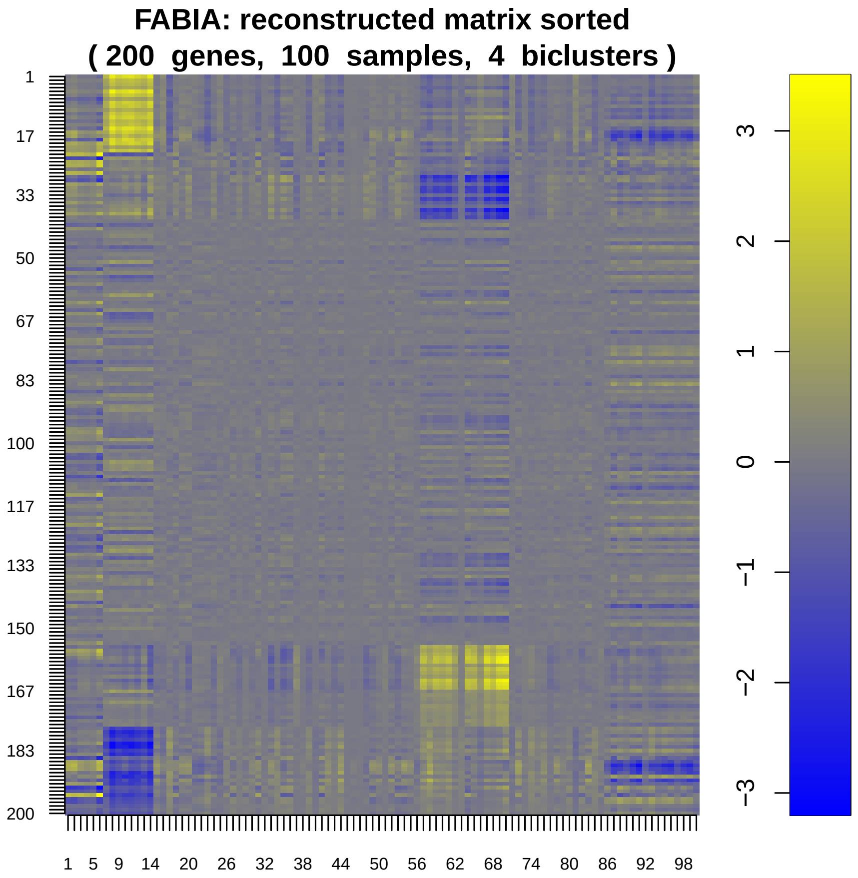

Figure 10: The reconstructed data matrix sorted according to k-means membership applied to factors (columns) and loadings (rows).

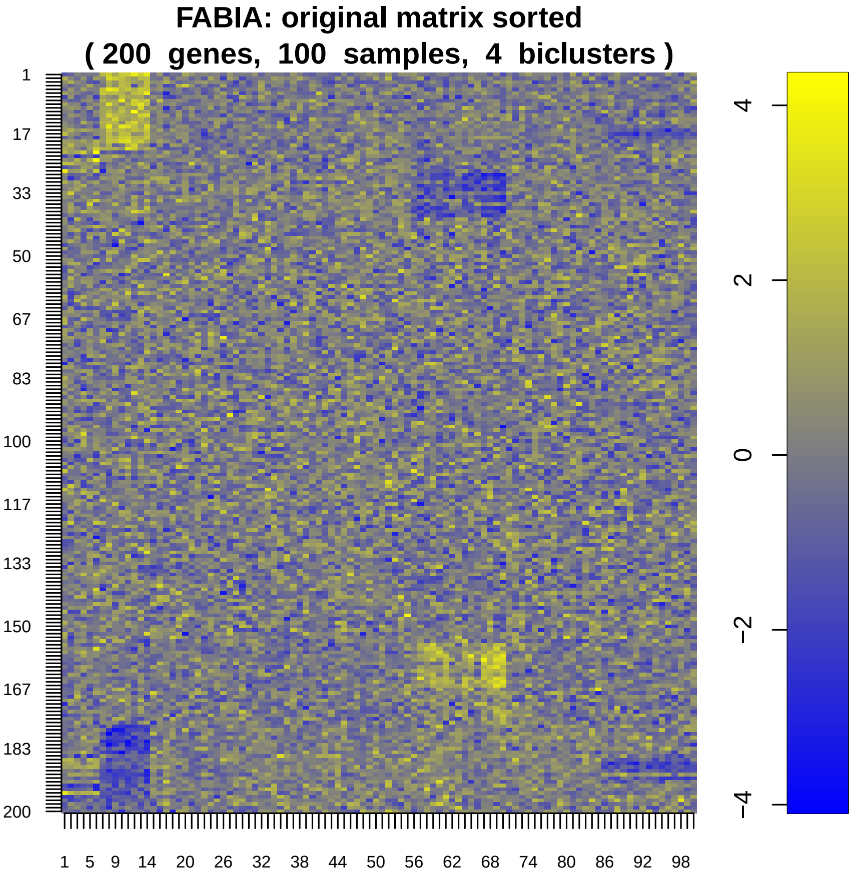

Figure 11: The original data matrix sorted according to k-means membership applied to factors (columns) and loadings (rows).

8. Plot the result as a biplot:

```
> colnames(res@X) <- clab$x
> rownames(res@X) <- llab$x
```

The biclusters will be sorted according to their information content.

Matrix factorization results are presented as *biplots*. A *biplot* allows information on both columns (samples) and rows (genes or variables) of a data matrix (gene expression matrix) to be displayed graphically.

In our plots,

- (1) **indicative row items:** red large circles give the most indicative row items (genes),
- (2) **less indicative row items:** golden small circles are row items (genes) which are less indicative,
- (3) **column items:** squares are the column items (samples),
- (4) **groups of column items:** column items (samples) of the same group have the same color,
- (5) **labels:** all data points are labeled according the column/row position in the original data matrix,
- (6) **true bicluster membership:** If a column/row item belongs to a true bicluster (from data generation) then this is indicated by the bicluster number as a prefix of the label followed by an underscore — note, that an item my belong to more than one bicluster.

If squares/circles have a large deviation from zero in the x-direction then they will be assigned to the bicluster in x direction. If squares/circles have a large deviation from zero in the y-direction then they will be assigned to the bicluster in y direction. Squares/circles with large deviation from zero in $45\,^\circ$ or $135\,^\circ$ belong to both biclusters.

We give the following biplots:

- (1) biplot of bicluster 1 and 2 of the FABIA result (Figure 12),
- (2) biplot of bicluster 1 and 3 of the FABIA result (Figure 13),
- (3) biplot of bicluster 1 and 4 of the FABIA result (Figure 14),
- (4) biplot of bicluster 2 and 3 of the FABIA result (Figure 15),
- (5) biplot of bicluster 2 and 4 of the FABIA result (Figure 16),
- (6) biplot of bicluster 3 and 4 of the FABIA result (Figure 17).

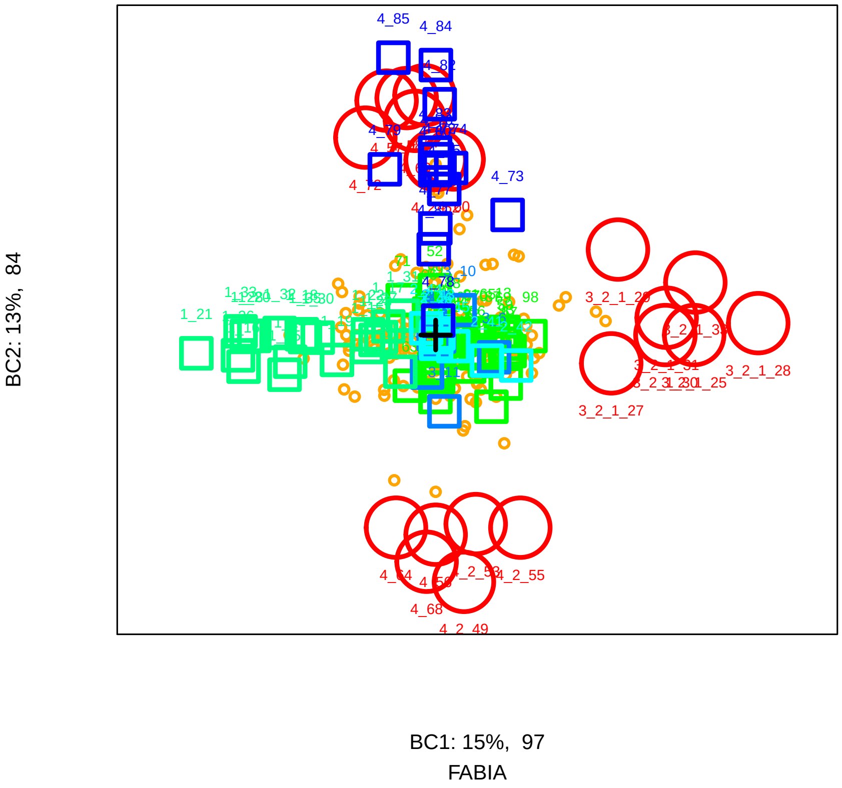

Figure 12: Biplot of bicluster 1 and 2 of the FABIA result, where biclusters are sorted now according to their information content. Red large circles give the most indicative row items. Golden small circles are row items which are less indicative. Squares are the column items, where members of the same group have the same color. All data points are labeled according the column/row position in the original data matrix. If a column/row item belongs to a true bicluster (from data generation) then this is indicated by the bicluster number as a prefix of the label followed by an underscore. Note, that an item my belong to more than one bicluster. If squares/circles have a large deviation from zero in the x-direction then they will be assigned to the bicluster in x direction. If squares/circles have a large deviation from zero in the y-direction then they will be assigned to the bicluster in y direction. Squares/circles with large deviation from zero in 45° or 135° belong to both biclusters. Bicluster 1 and 2 share genes which are large circles on the 45° line or on the 135° line depending on the sign of loadings of the genes for the biclusters. The shared genes can be seen at the reconstructed data in Fig. 6 and the loadings in Fig. 8. Note, that the biclusters are sorted now according to their information content, thus bicluster 1 is extracted bicluster 1 and bicluster 2 is the extracted bicluster 4.

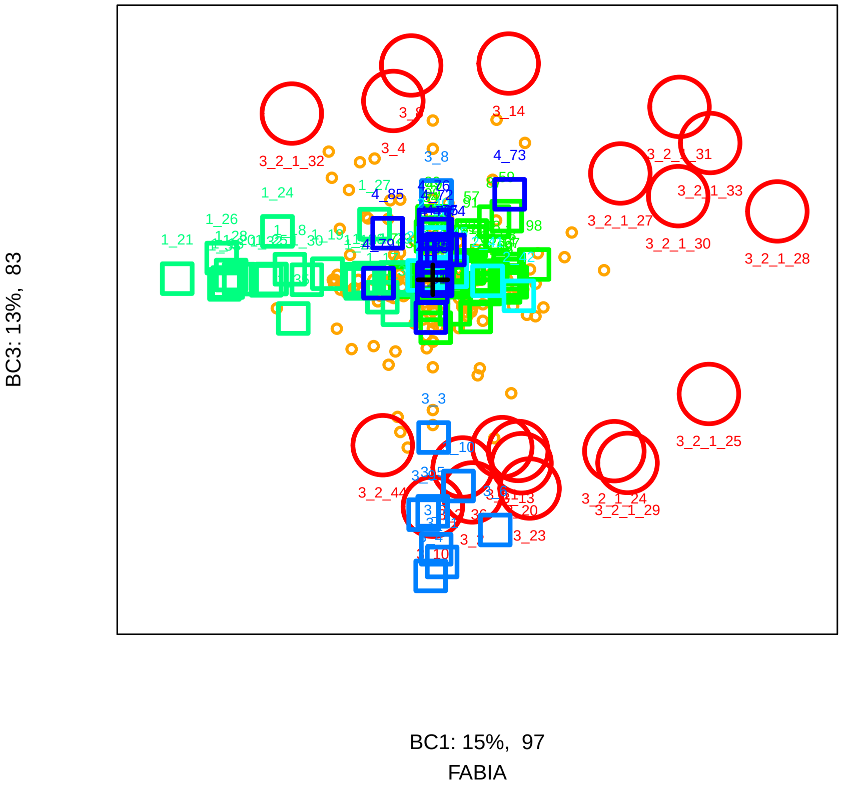

Figure 13: Biplot of bicluster 1 and 3 of the FABIA result. See Fig. 12 for the explanation.

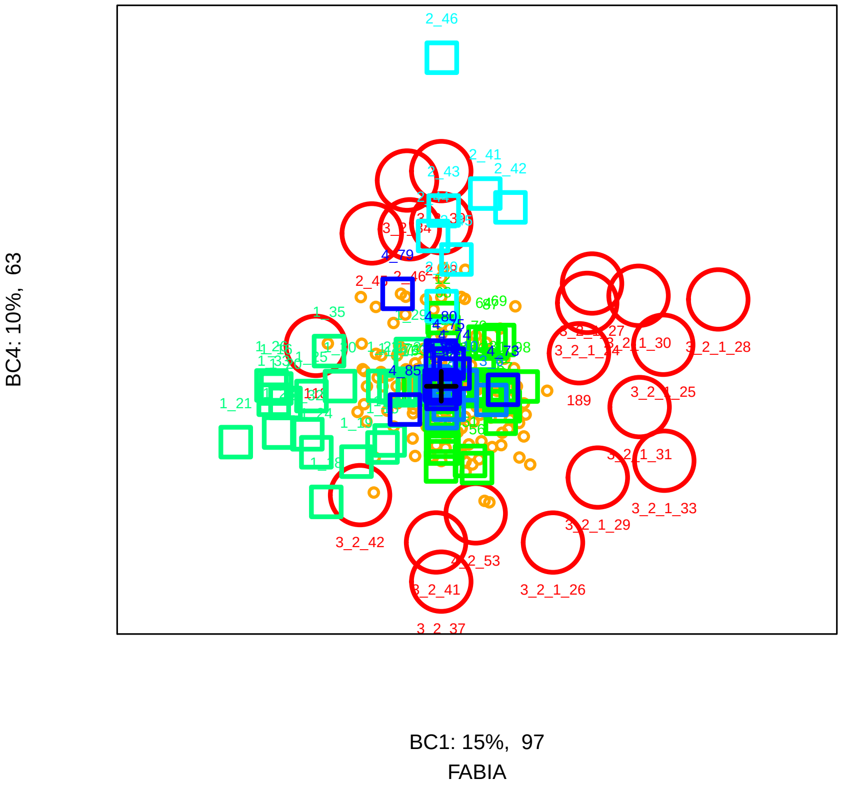

Figure 14: Biplot of bicluster 1 and 4 of the FABIA result. See Fig. 12 for the explanation. Bicluster 1 and 4 share genes which are large circles on the 45 ° line or on the 135 ° line depending on the sign of loadings of the genes for the biclusters. The shared genes can be seen at the reconstructed data in Fig. 6 and the loadings in Fig. 8. Note, that the biclusters are sorted now according to their information content, thus bicluster 1 is the extracted bicluster 1 and bicluster 4 is the extracted bicluster 2.

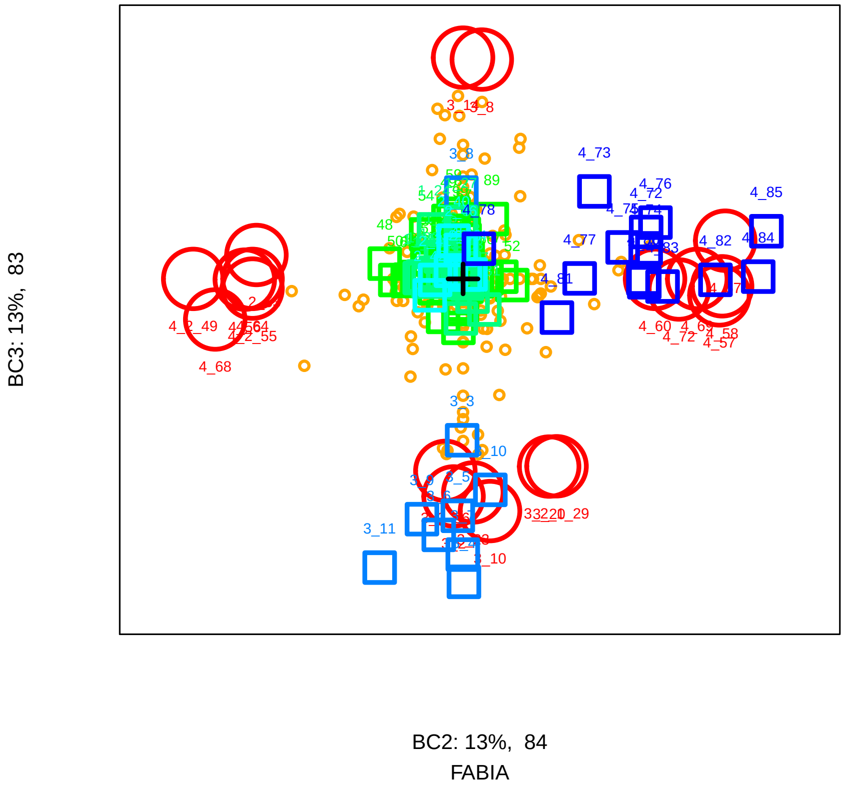

Figure 15: Biplot of bicluster 2 and 3 of the FABIA result. See Fig. 12 for the explanation.

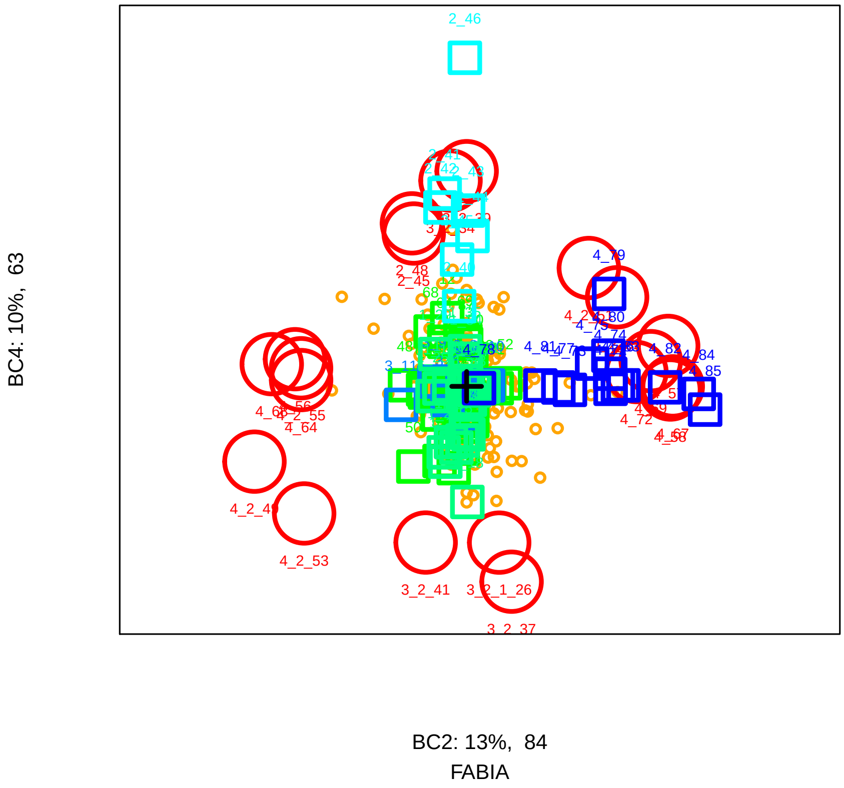

Figure 16: Biplot of bicluster 2 and 4 of the FABIA result. See Fig. 12 for the explanation. Bicluster 2 and 4 share genes which are large circles on the 135 ° line. The shared genes can be seen at the reconstructed data in Fig. 6 and the loadings in Fig. 8. Note, that the biclusters are sorted now according to their information content, thus bicluster 2 is the extracted bicluster 4 and bicluster 4 is the extracted bicluster 2.

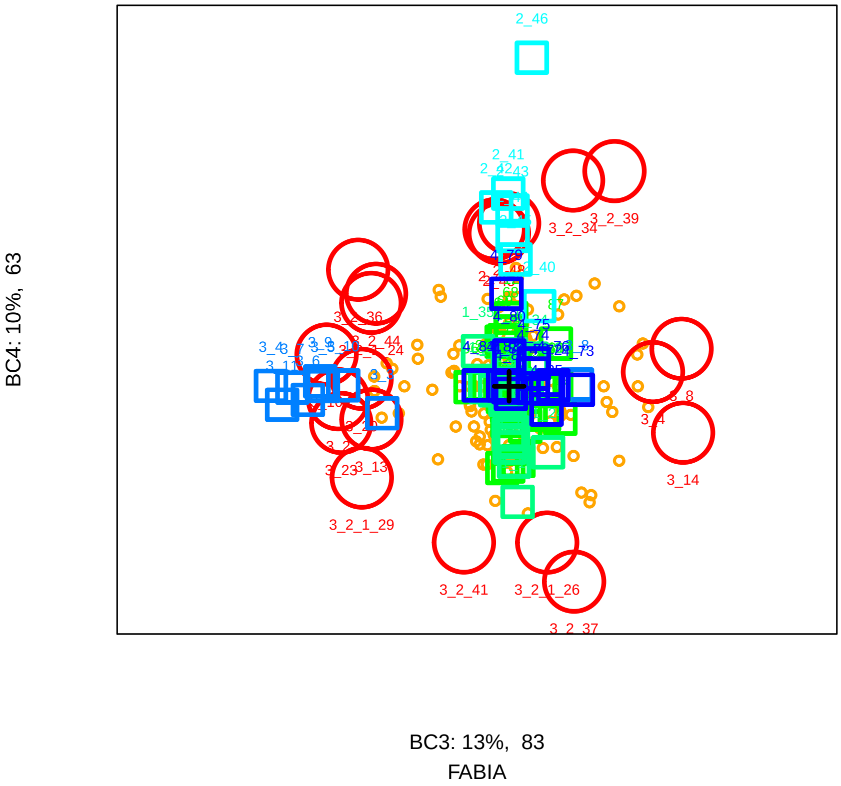

Figure 17: Biplot of bicluster 3 and 4 of the FABIA result. See Fig. 12 for the explanation. Also bicluster 3 and 4 share genes which are large circles on the 45° line or on the 135° line depending on the sign of loadings of the genes for the biclusters. The shared genes can be seen at the reconstructed data in Fig. 6 and the loadings in Fig. 8. Note, that the biclusters are sorted now according to their information content, thus bicluster 3 is the extracted bicluster 3 and bicluster 4 is the extracted bicluster 2.

9. Extract biclusters:

   ```
   > rb <- extract_bic(res)
   ```

   In `bic` the biclusters are extracted according to the largest absolute values of the component i, i.e. the largest values of $\boldsymbol{\lambda}_i$ and the largest values of $\boldsymbol{z}_i$.

   The components of `bic` are:

   - `binp`,
   - `bixv`,
   - `bixn`,
   - `biypv`, and
   - `biypn`.

   They contain following information:

   - `binp` give the size of the bicluster: number observations and number samples.
   - `bixv` gives the values of the extracted observations that have absolute values above a threshold. They are sorted.
   - `bixn` gives the extracted observation names (e.g. gene names).
   - `biypv` gives the values of the extracted samples that have absolute values above a threshold. They are sorted.
   - `biypn` gives the names of the extracted samples (e.g. sample names).

   In `bicopp` the opposite cluster to the biclusters are given. Opposite means that the negative pattern is present.

   The components of opposite clusters `bicopp` are:

   - `binn`,
   - `bixv`,
   - `bixn`,
   - `biypnv`, and
   - `biynn`.

   They contain following information:

   - `binp` give the size of the opposite bicluster: number observations and number samples.
   - `bixv` gives the values of the extracted observations that have absolute values above a threshold. They are sorted.
   - `bixn` gives the extracted observation names (e.g. gene names).
   - `biynv` gives the values of the opposite extracted samples that have absolute values above a threshold. They are sorted.
   - `biynn` gives the names of the opposite extracted samples (e.g. sample names).

That means the samples are divided into two groups where one group shows large positive values and the other group has negative values with large absolute values. That means a observation pattern can be switched on or switched off relative to the average value.

`numn` gives the indices of `bic` with components:

- `numng` = `bix` and
- `numnp` = `biypn`.

`numn` gives the indices of `bicopp` with components:

- `numng` = `bix` and
- `numnn` = `biynn`.

10. List bicluster 1:

    ```
    > rb$bic[1, ]

    $binp
    [1] 28 12

    $bixv
     [1]  0.7918153 -0.7546732 -0.7480349  0.7418612  0.7281603  0.6547531
     [7]  0.6399667  0.6389596 -0.6240919 -0.6034900  0.5979212  0.5971951
    [13]  0.5852676  0.5838222 -0.5801785  0.5771147 -0.5584461 -0.5559548
    [19]  0.5525784  0.5309607  0.5066357  0.4783094 -0.4777657 -0.4773567
    [25] -0.4571823  0.4543615 -0.4474381 -0.4232859

    $bixn
     [1] "10" "14" "8"  "2"  "23" "36" "20" "29" "4"  "31" "24" "13" "15" "11" "32"
    [16] "44" "6"  "19" "17" "22" "3"  "21" "35" "33" "39" "5"  "18" "1"

    $biypv
     [1] 4.0054604 3.8168875 3.6448392 3.3843838 3.1769709 3.1319345 2.7893371
     [8] 2.1336908 0.6475568 0.5227615 0.5112324 0.5020229

    $biypn
     [1] "4"  "11" "7"  "6"  "9"  "5"  "10" "3"  "96" "35" "81" "2"
    ```

11. List bicluster 2:

    ```
    > rb$bic[2, ]

    $binp
    [1] 18 15

    $bixv
     [1]  0.8055587 -0.7825429 -0.7733592 -0.7650996  0.7398992 -0.6990429
     [7]  0.6516050 -0.6444496  0.6286309  0.6284904  0.6168964 -0.5731003
    ```

```
[13] -0.5712088 -0.5571252  0.5115990  0.4739272 -0.4720350 -0.4639124

$bixn
 [1] "49" "67" "58" "57" "68" "69" "56" "72" "55" "64" "53" "60" "52" "50" "71"
[16] "70" "61" "51"

$biypv
 [1] -4.0083639 -3.9008218 -3.3355192 -2.6362816 -2.5422327 -2.4198204
 [7] -2.4134336 -2.3951393 -2.3828737 -2.1094423 -1.7356527 -1.5410969
[13] -1.2403794 -0.6502351 -0.5057733

$biypn
 [1] "85" "84" "82" "83" "76" "72" "74" "79" "80" "75" "73" "77" "81" "52" "71"
```

12. Show bicluster 3:

```
> rb$bic[3, ]

$binp
[1]  8 21

$bixv
[1] -0.7855035 -0.6323874 -0.6294644 -0.5624963 -0.5594361 -0.4439355 -0.4275572
[8] -0.4138484

$bixn
[1] "28" "33" "25" "31" "30" "29" "27" "24"

$biypv
 [1] 3.4561763 2.8548925 2.8172918 2.7746912 2.7271221 2.6195334 2.2525953
 [8] 2.1828300 2.1007515 1.9336496 1.8872969 1.7031376 1.4253472 0.9879875
[15] 0.9763863 0.8680340 0.7882886 0.7342218 0.6836046 0.6107089 0.5105079

$biypn
 [1] "21" "26" "33" "16" "28" "20" "32" "25" "24" "18" "35" "30" "19" "15" "22"
[16] "23" "27" "79" "17" "85" "29"
```

13. List bicluster 4:

```
> rb$bic[4, ]

$binp
[1]  8 16

$bixv
[1] -0.6007108 -0.5744221  0.5453208 -0.4550748 -0.4381924  0.4362393  0.4359989
[8] -0.4266766
```

```
$bixn
[1] "39" "34" "37" "48" "46" "26" "41" "45"

$biypv
 [1] -5.5284726 -3.2427672 -3.0101572 -2.9576651 -2.5290120 -2.1341345
 [7] -1.5557628 -1.3484948 -1.1482656 -0.9229002 -0.7785127 -0.7505664
[13] -0.7178881 -0.5920485 -0.5435078 -0.5166205

$biypn
 [1] "46" "41" "42" "43" "44" "45" "79" "40" "12" "68" "69" "64" "87" "35" "29"
[16] "80"
```

14. Plot bicluster 1:

    Now the biclusters are visualized in the original data. Each bicluster is visualized in the raw data by two plots:

    - Plot1: The data matrix is sorted such that the bicluster appear at the upper left corner. The bicluster is marked by a rectangle.
    - Plot2: Only the bicluster is plotted.

    We have following figures:

    - Plot1 bicluster 1: Fig. 18,
    - Plot2 bicluster 1: Fig. 19,
    - Plot1 bicluster 2: Fig. 20,
    - Plot2 bicluster 2: Fig. 21,
    - Plot1 bicluster 3: Fig. 22,
    - Plot2 bicluster 3: Fig. 23,
    - Plot1 bicluster 4: Fig. 24,
    - Plot2 bicluster 4: Fig. 25.

    ```
    > plotBicluster(X,unlist(rb$bic[1,5]),unlist(rb$bic[1,3]))
    ```

15. Plot bicluster 2:

    ```
    > plotBicluster(X,unlist(rb$bic[2,5]),unlist(rb$bic[2,3]))
    ```

16. Plot bicluster 3:

    ```
    > plotBicluster(X,unlist(rb$bic[3,5]),unlist(rb$bic[3,3]))
    ```

17. Plot bicluster 4:

    ```
    > plotBicluster(X,unlist(rb$bic[4,5]),unlist(rb$bic[4,3]))
    ```

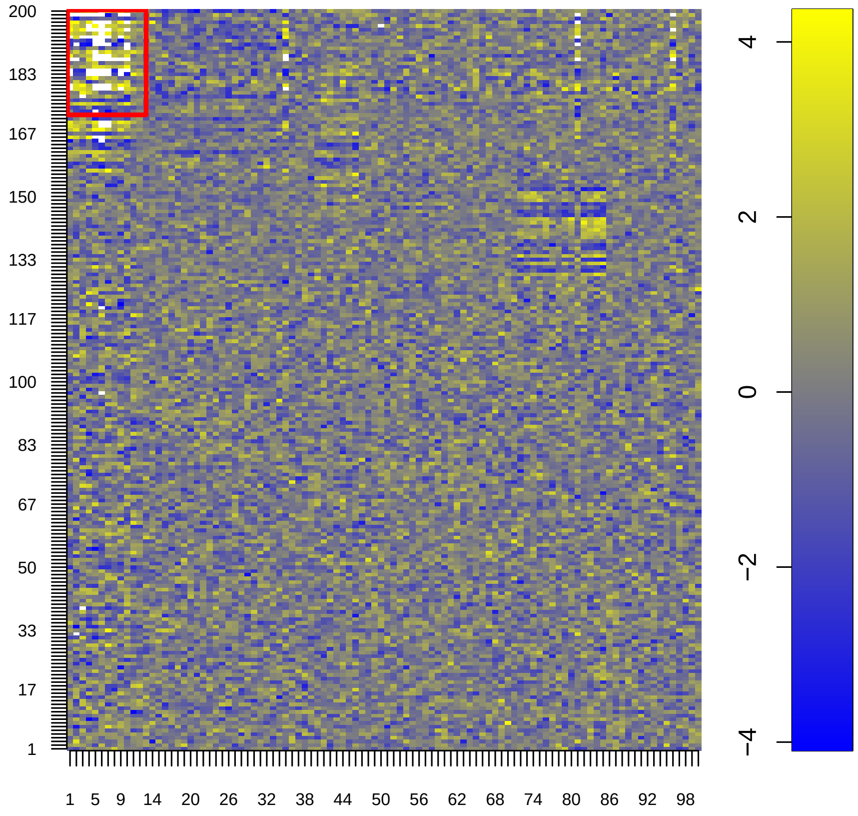

Figure 18: Plot of Bicluster 1. The bicluster is moved to the top left of the original matrix and marked.

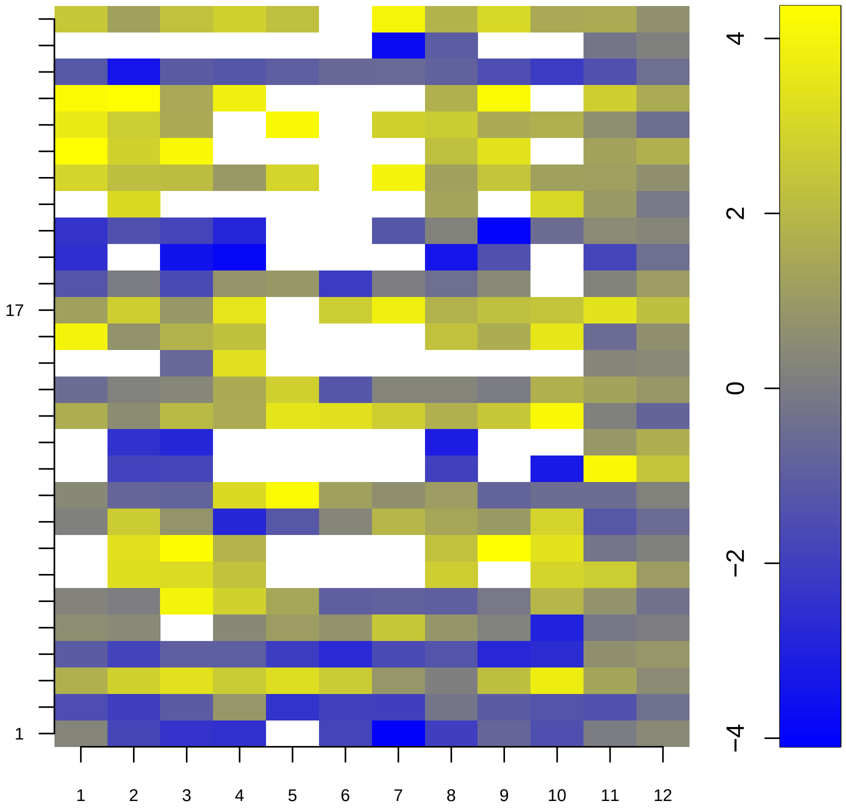

Figure 19: Plot of Bicluster 1. The bicluster, marked region of the previous image, is enlarged.

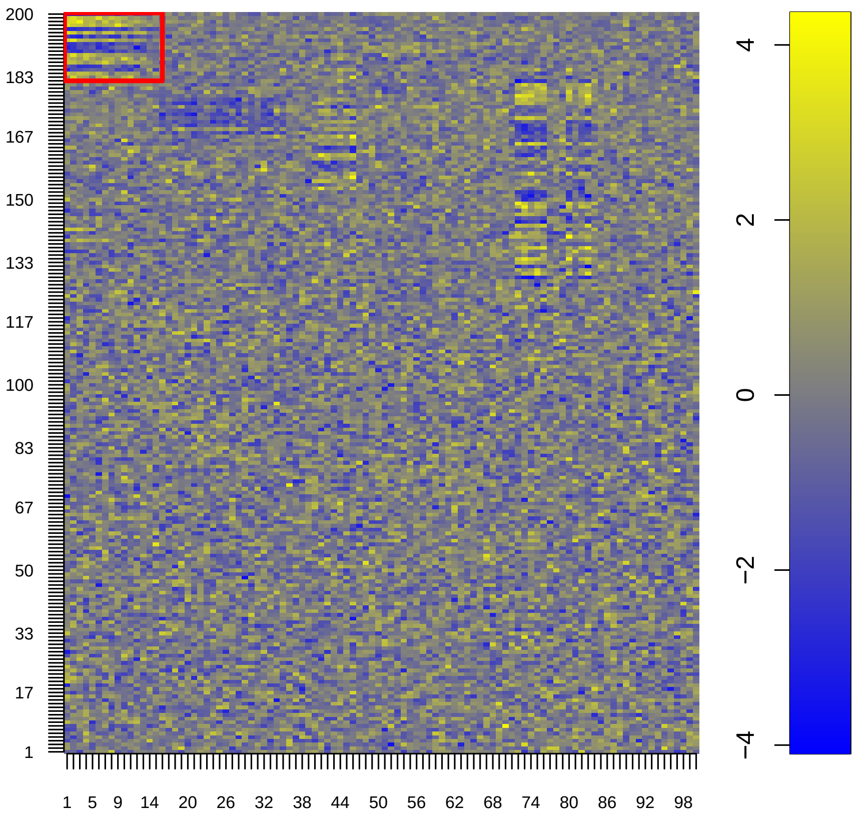

Figure 20: Plot of Bicluster 2. The bicluster is moved to the top left of the original matrix and marked.

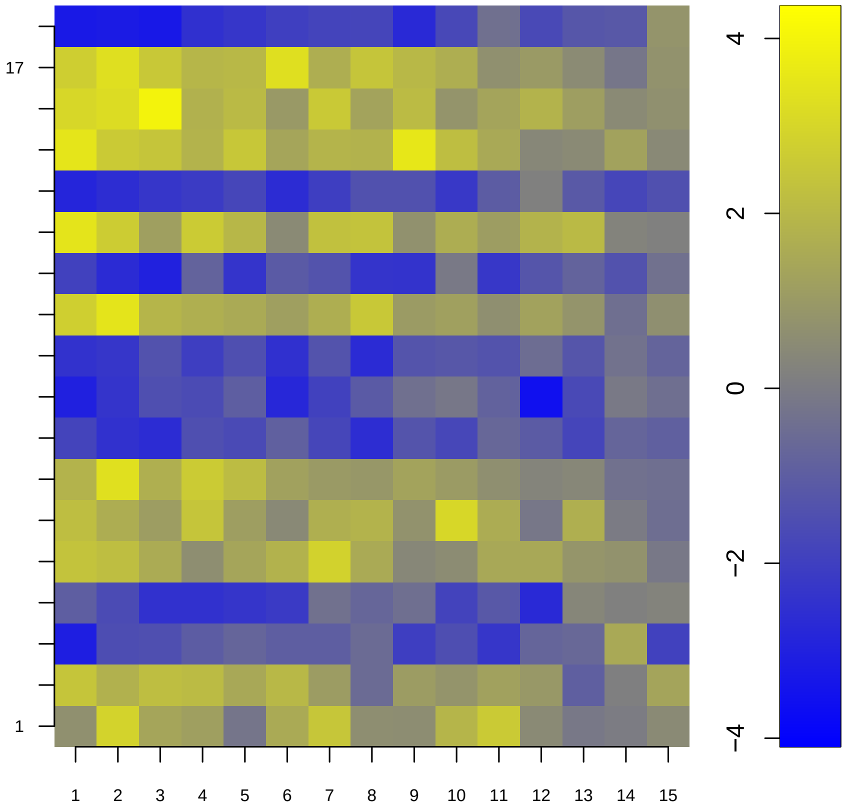

Figure 21: Plot of Bicluster 2. The bicluster, marked region of the previous image, is enlarged.

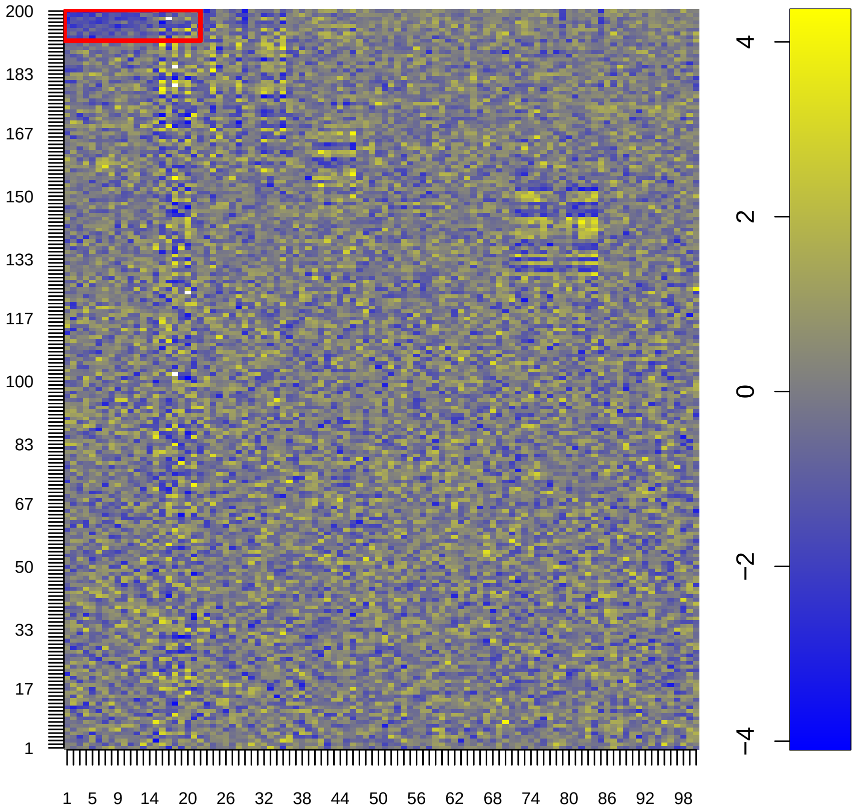

Figure 22: Plot of Bicluster 3. The bicluster is moved to the top left of the original matrix and marked.

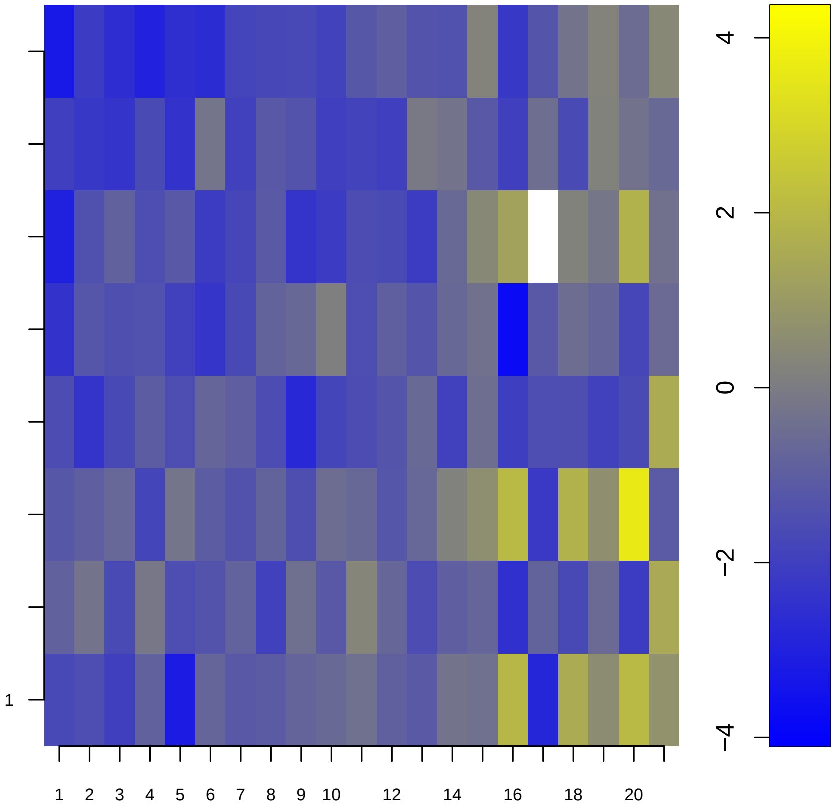

Figure 23: Plot of Bicluster 3. The bicluster, marked region of the previous image, is enlarged.

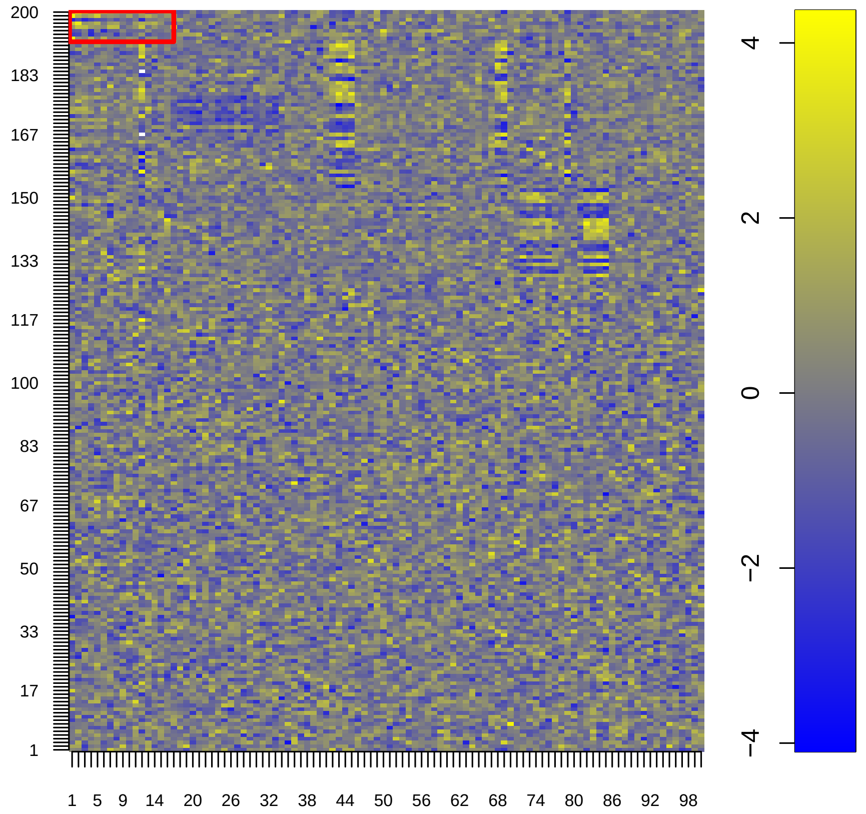

Figure 24: Plot of Bicluster 4. The bicluster is moved to the top left of the original matrix and marked.

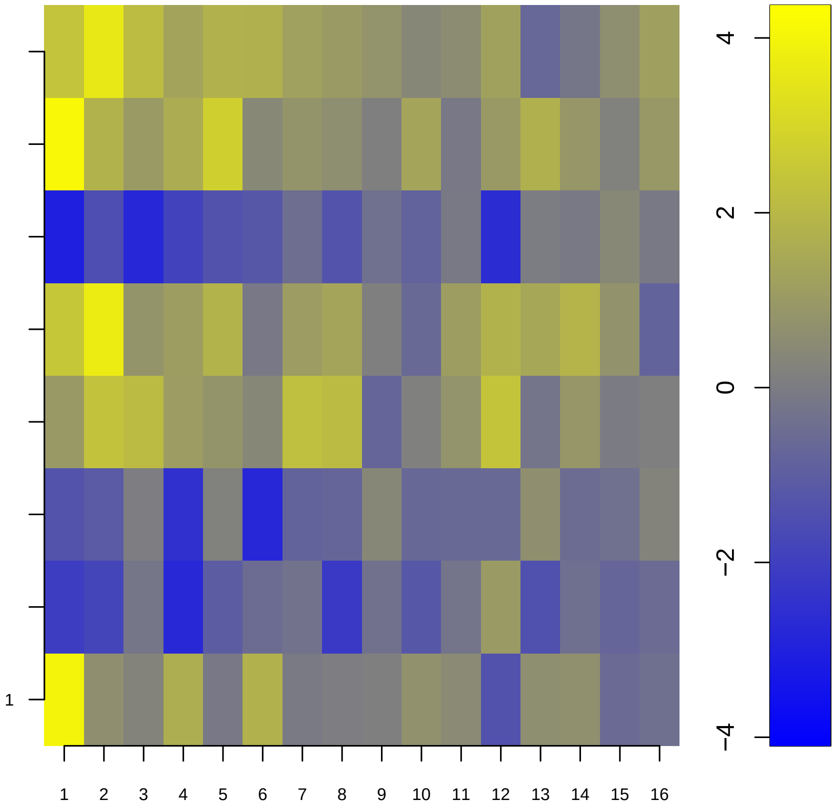

Figure 25: Plot of Bicluster 4. The bicluster, marked region of the previous image, is enlarged.

18. List opposite bicluster 1:

```
> rb$bicopp[1, ]

$binn
[1] 28 16

$bixv
 [1]  0.7918153 -0.7546732 -0.7480349  0.7418612  0.7281603  0.6547531
 [7]  0.6399667  0.6389596 -0.6240919 -0.6034900  0.5979212  0.5971951
[13]  0.5852676  0.5838222 -0.5801785  0.5771147 -0.5584461 -0.5559548
[19]  0.5525784  0.5309607  0.5066357  0.4783094 -0.4777657 -0.4773567
[25] -0.4571823  0.4543615 -0.4474381 -0.4232859

$bixn
 [1] "10" "14" "8"  "2"  "23" "36" "20" "29" "4"  "31" "24" "13" "15" "11" "32"
[16] "44" "6"  "19" "17" "22" "3"  "21" "35" "33" "39" "5"  "18" "1"

$biynv
 [1] -1.1565660 -1.1412940 -0.8615079 -0.7913953 -0.7871622 -0.7658792
 [7] -0.7532986 -0.7442789 -0.6551553 -0.6362984 -0.6301016 -0.6200443
[13] -0.6000115 -0.5882842 -0.5192052 -0.5060870

$biynn
 [1] "73" "8"  "59" "89" "87" "49" "27" "76" "24" "99" "85" "72" "57" "54" "91"
[16] "40"
```

19. Plot of opposite bicluster 1:

    Now the opposite biclusters are visualized in the original data. In some applications the negative row pattern may be important which is given by opposite biclusters.

    - Plot1: The data matrix is sorted such that the opposite bicluster appear at the upper left corner. The opposite bicluster is marked by a rectangle.
    - Plot2: Only the opposite bicluster is plotted.

    We have following figures:

    - Plot1 opposite bicluster 1: Fig. 26,
    - Plot2 opposite bicluster 1: Fig. 27.

```
> plotBicluster(X,unlist(rb$bicopp[1,5]),unlist(rb$bic[1,3]))
```

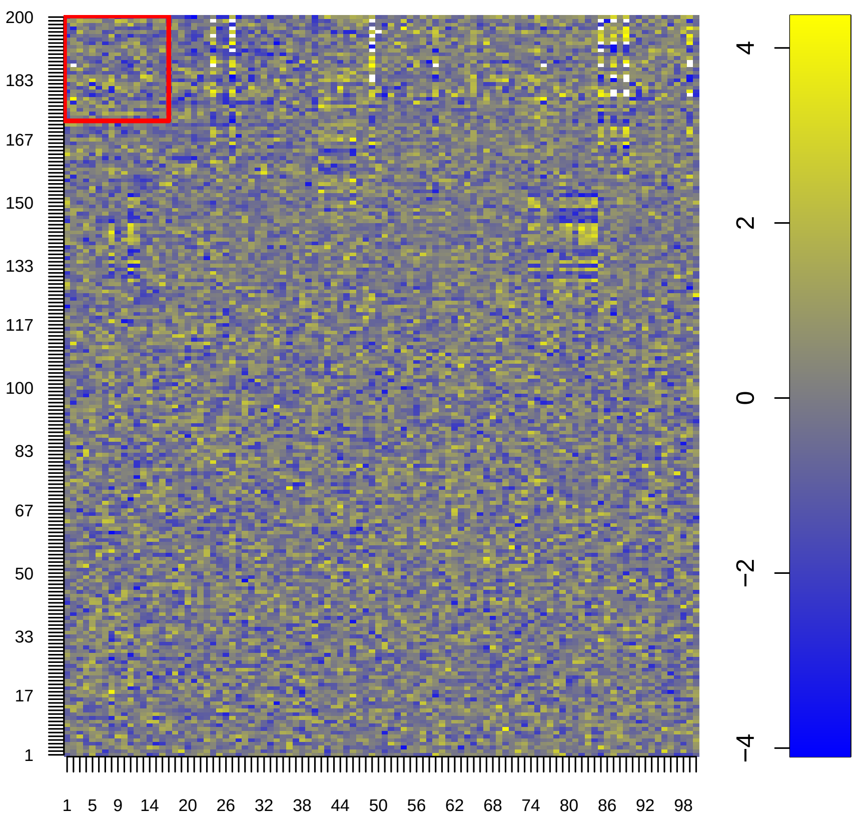

Figure 26: Plot of opposite Bicluster 1. The bicluster is moved to the top left of the original matrix and marked.

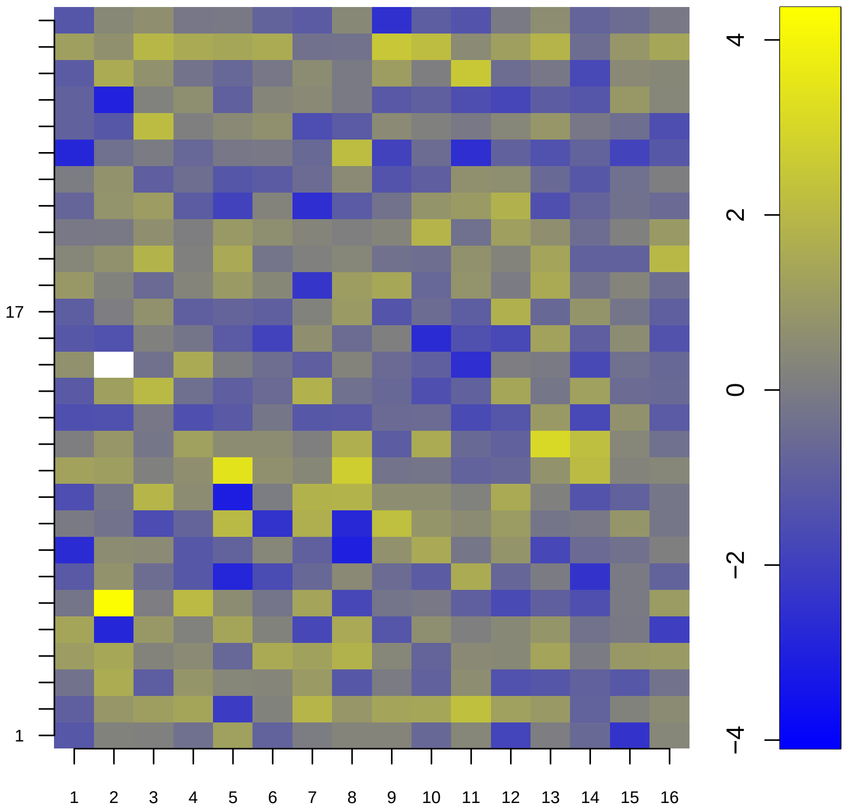

Figure 27: Plot of opposite Bicluster 1. The bicluster, marked region of the previous image, is enlarged.

2 Example 2

Here another example which has been exactly generated as the previous one but with different initialization.

1. Load the library:

```
> library(fabia)
```

2. Read the data file “datafile.csv” and labels:

```
> X <- read.table("datafile1.csv", header = TRUE, sep = ",")
> X <- as.matrix(X[, -1])
> groups <- read.table(file = "groups1.txt")
> clab <- read.table(file = "clabels1.txt")
> llab <- read.table(file = "rlabels1.txt")
> colnames(X) <- 1:ncol(X)
> rownames(X) <- 1:nrow(X)
```

3. Select the model based on the data: 4 biclusters; sparseness 0.1; 400 cycles

```
> res <- fabia(X, 4, 0.1, 400, 1, 1)
```

```
Running FABIA on  200 x 100  matrix with
   Number of biclusters ---------------- p:  4
   Sparseness factor --------------- alpha:  0.1
   Number of iterations -------------- cyc:  400
   Loading prior parameter ----------- spl:  1
   Factor prior parameter ------------ spz:  1
   Initialization interval -------- random:  1
   Centering ---------------------- center:  2  = median
   Scaling to variance one: --------- norm:  1  = Yes
   Scaling loadings per iteration -- scale:  0  = No
   Constraint variational parameter -- lap:  1
Cycle: 0
Cycle: 20
Cycle: 40
Cycle: 60
Cycle: 80
Cycle: 100
Cycle: 120
Cycle: 140
Cycle: 160
Cycle: 180
Cycle: 200
Cycle: 220
Cycle: 240
Cycle: 260
```

```
Cycle: 280
Cycle: 300
Cycle: 320
Cycle: 340
Cycle: 360
Cycle: 380
```

4. Show information content of the biclusters:

```
> res@avini
```

```
[1]  83.70053  55.00468  81.34770 110.41335 330.45634
```

5. Give summary:

   First, it gives the number or rows and columns of the original matrix and then the number of clusters. Then for the row cluster the information content is given. Then for each column its information content is given. Then for each cluster a column summary is given by a boxplot. Finally, for each cluster a row summary is given by a boxplot.

```
> summary(res)
```

```
An object of class Factorization

call:
        "fabia"

Number of rows:  200

Number of columns:  100

Number of clusters:  4

 Information content of the clusters:
  BC 1    BC 2    BC 3    BC 4  BC sum
 83.70   55.00   81.35  110.41  330.46



 Information content of the samples:
  Sample 1    Sample 2    Sample 3    Sample 4    Sample 5    Sample 6
      5.01       11.31        6.45        2.60        9.65        7.48
  Sample 7    Sample 8    Sample 9   Sample 10   Sample 11   Sample 12
      3.92        1.64        5.75       10.04        3.21        4.00
 Sample 13   Sample 14   Sample 15   Sample 16   Sample 17   Sample 18
      3.23        1.37        0.97        8.93       13.74        3.69
 Sample 19   Sample 20   Sample 21   Sample 22   Sample 23   Sample 24
      9.93       10.75       10.53        4.21        3.19        4.02
```

```
Sample 25   Sample 26   Sample 27   Sample 28   Sample 29   Sample 30
     2.01        3.86        1.41        1.74        3.70        6.32
Sample 31   Sample 32   Sample 33   Sample 34   Sample 35   Sample 36
     4.45        9.59       12.25        2.74       10.43        7.17
Sample 37   Sample 38   Sample 39   Sample 40   Sample 41   Sample 42
     9.46       14.02       11.85       12.09        9.88       11.93
Sample 43   Sample 44   Sample 45   Sample 46   Sample 47   Sample 48
    11.47       12.01        1.20        7.16        6.02        5.68
Sample 49   Sample 50   Sample 51   Sample 52   Sample 53   Sample 54
     3.96        4.16        7.14        3.43        4.66        2.69
Sample 55   Sample 56   Sample 57   Sample 58   Sample 59   Sample 60
     2.42        6.97        4.97        7.37        7.67        5.77
Sample 61   Sample 62   Sample 63   Sample 64   Sample 65   Sample 66
     6.31        7.38        4.81        3.36        4.63        4.64
Sample 67   Sample 68   Sample 69   Sample 70   Sample 71   Sample 72
     5.52        1.53        1.01        5.18        9.50        1.45
Sample 73   Sample 74   Sample 75   Sample 76   Sample 77   Sample 78
     7.48       10.12       13.32       11.24       12.21        8.55
Sample 79   Sample 80   Sample 81   Sample 82   Sample 83   Sample 84
    11.85        7.32       11.50       10.23       11.02       12.40
Sample 85   Sample 86   Sample 87   Sample 88   Sample 89   Sample 90
     7.74        7.53        6.79        7.15        5.76        4.59
Sample 91   Sample 92   Sample 93   Sample 94   Sample 95   Sample 96
     3.80        8.57        8.72        1.35        5.51        5.48
Sample 97   Sample 98   Sample 99  Sample 100  Sample sum
     7.71        4.29        5.55        2.54      330.46

Column clusters / Factors:
BC 1              BC 2               BC 3              BC 4
Min.   :-1.1259   Min.   :-1.38922   Min.   :-5.3201   Min.   :-1.2395
1st Qu.:-0.2087   1st Qu.:-0.13620   1st Qu.:-0.0760   1st Qu.:-0.3372
Median :-0.0171   Median :-0.00999   Median : 0.0174   Median :-0.0524
Mean   : 0.1788   Mean   : 0.12864   Mean   :-0.2033   Mean   : 0.1780
3rd Qu.: 0.0681   3rd Qu.: 0.06568   3rd Qu.: 0.1729   3rd Qu.: 0.0502
Max.   : 4.0803   Max.   : 4.89308   Max.   : 0.9322   Max.   : 4.1988


Row clusters / Loadings:
BC 1              BC 2               BC 3              BC 4
Min.   :-0.7055   Min.   :-0.73476   Min.   :-0.7696   Min.   :-0.8073
1st Qu.:-0.0487   1st Qu.:-0.05721   1st Qu.:-0.0861   1st Qu.:-0.0783
Median : 0.0000   Median : 0.00000   Median : 0.0000   Median : 0.0000
Mean   : 0.0119   Mean   :-0.00738   Mean   :-0.0248   Mean   : 0.0144
3rd Qu.: 0.0685   3rd Qu.: 0.04417   3rd Qu.: 0.0506   3rd Qu.: 0.1054
Max.   : 0.7283   Max.   : 0.70033   Max.   : 0.7840   Max.   : 0.8347
```

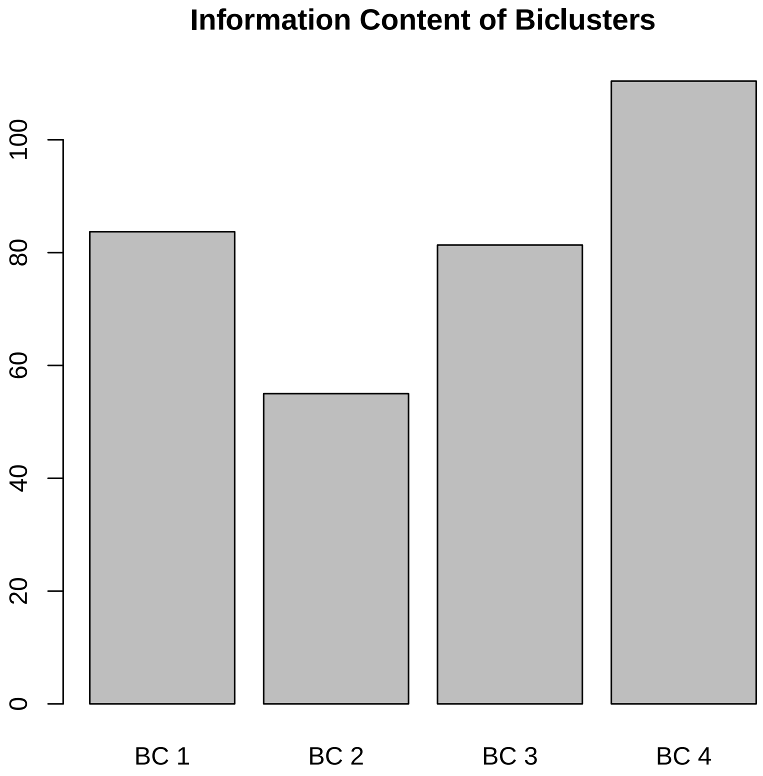

Figure 28: Information content of biclusters. The biclusters are ranked according to their information. More information means larger and less noisy biclusters.

6. Plot some statistics:

   ```
   > show(res)
   ```

   The statistics are shown in Figure 28 to Figure 31. They give

   - (1) the information content of biclusters (Figure 28),
   - (2) the information content of samples (Figure 29),
   - (3) boxplots of loadings for each bicluster (row membership, Figure 30),
   - (4) boxplots of factors for each bicluster (column membership, Figure 31).

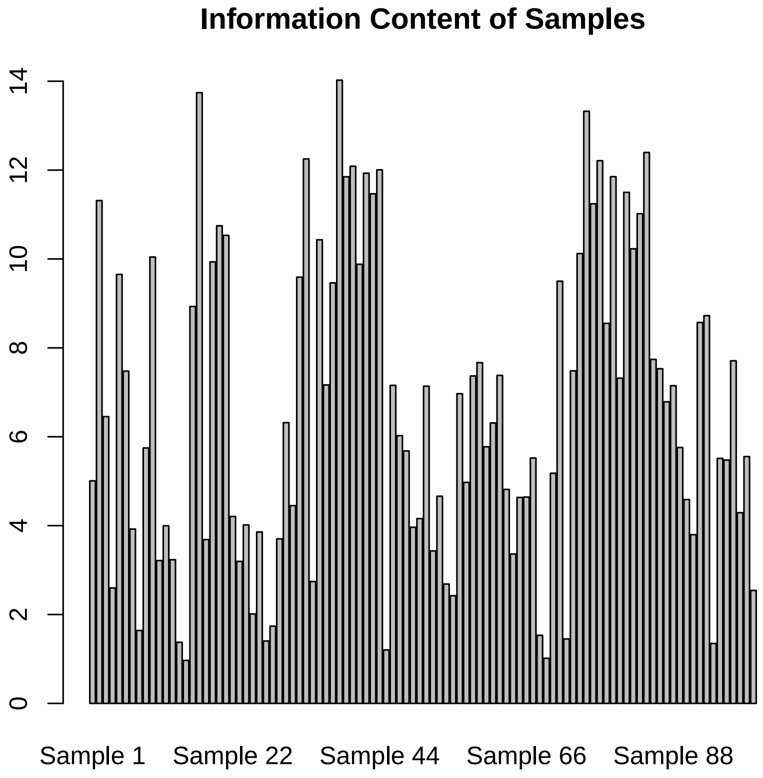

Figure 29: Information content of samples. The samples are ranked according to their information. More information means that the sample participates at more biclusters or has stronger signals.

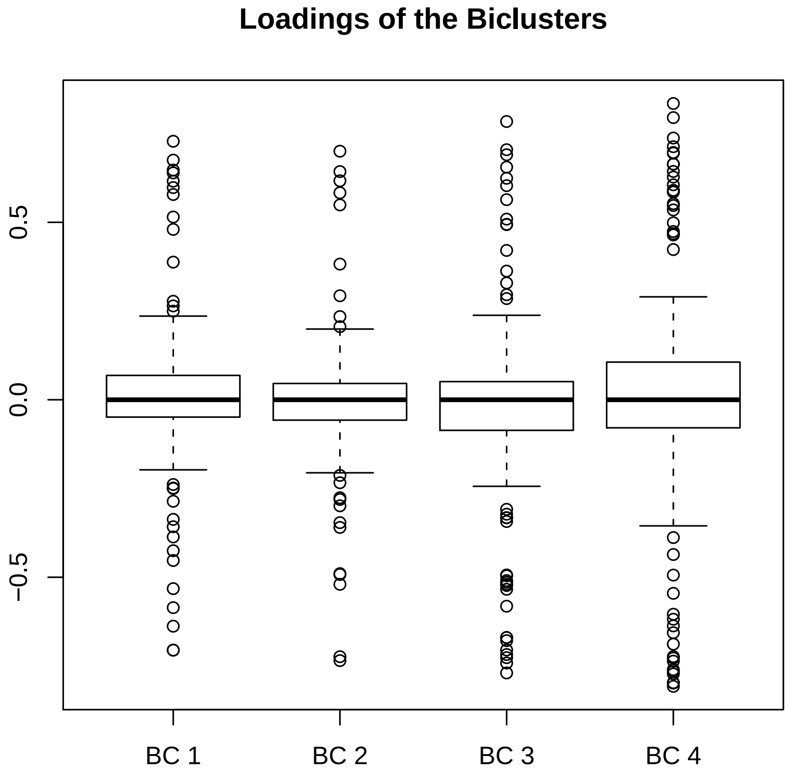

Figure 30: Boxplot of loadings for each bicluster (row membership). Gives the row size of the biclusters.

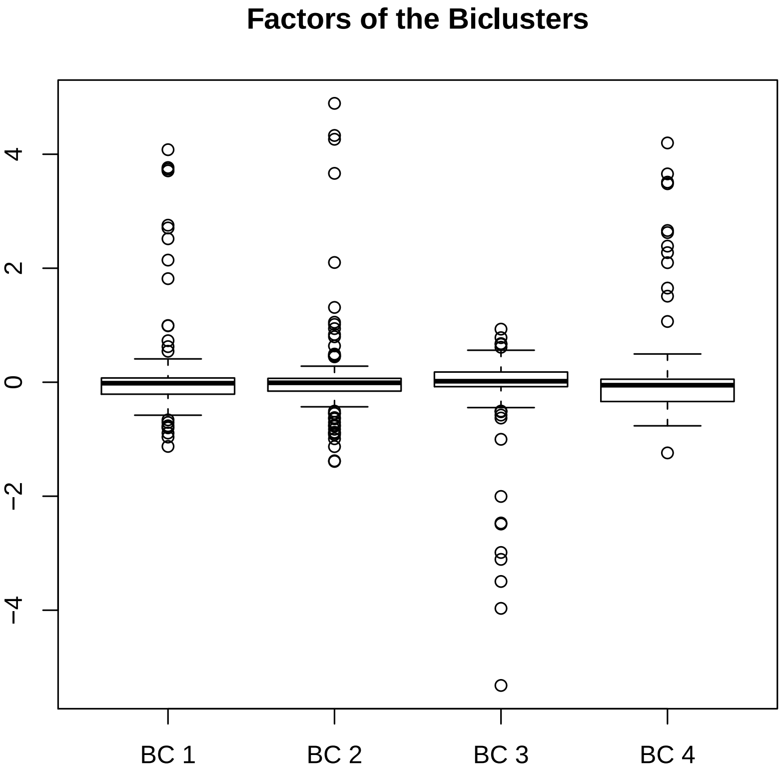

Figure 31: Boxplot of factors for each bicluster (column membership). Gives the column size of the biclusters. Biclusters may be driven by few samples and are not robust.

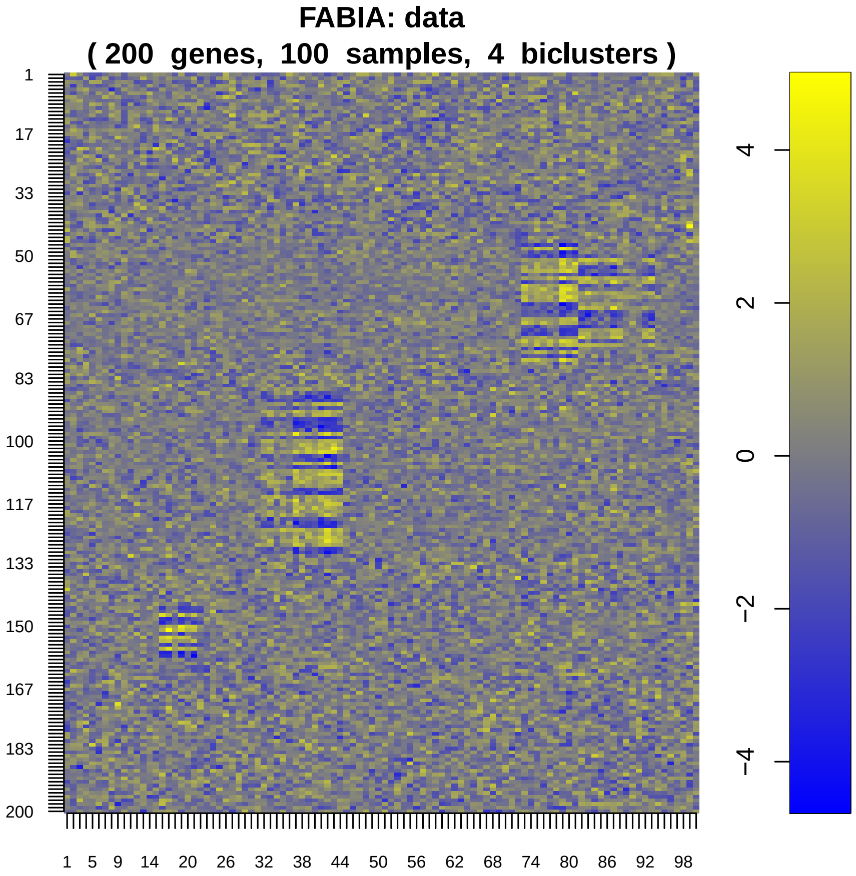

Figure 32: The data.

7. Plot the factorization results:

   ```
   > rr <- extract_plot(res,ti="FABIA")
   ```

   The results are shown in Figure 32 to Figure 38. They give

   - (1) the original data (Figure 32),
   - (2) the reconstructed data by FABIA (Figure 33),
   - (3) the reconstruction error of FABIA (Figure 34),
   - (4) the estimated absolute loadings (Figure 35),
   - (5) the estimated absolute factors (Figure 36),
   - (6) the reconstructed data matrix sorted according to k-means membership applied to factors/columns and loadings/rows (Figure 37),
   - (7) the original data matrix sorted according to k-means membership applied to factors/columns and loadings/rows (Figure 38),

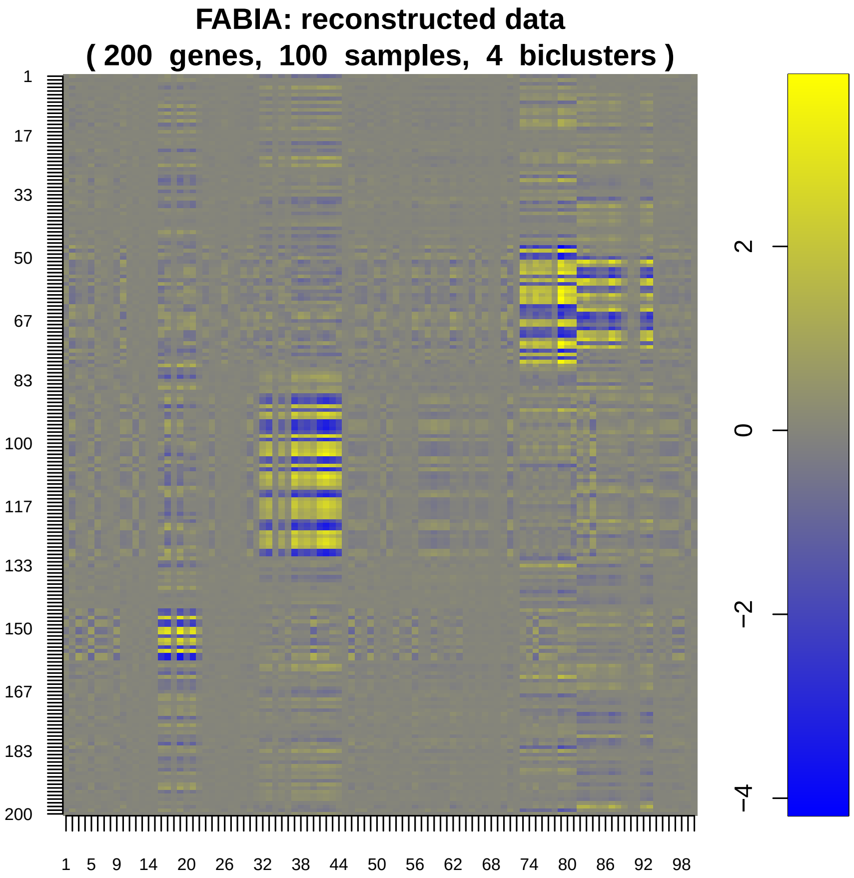

Figure 33: The reconstructed data by FABIA.

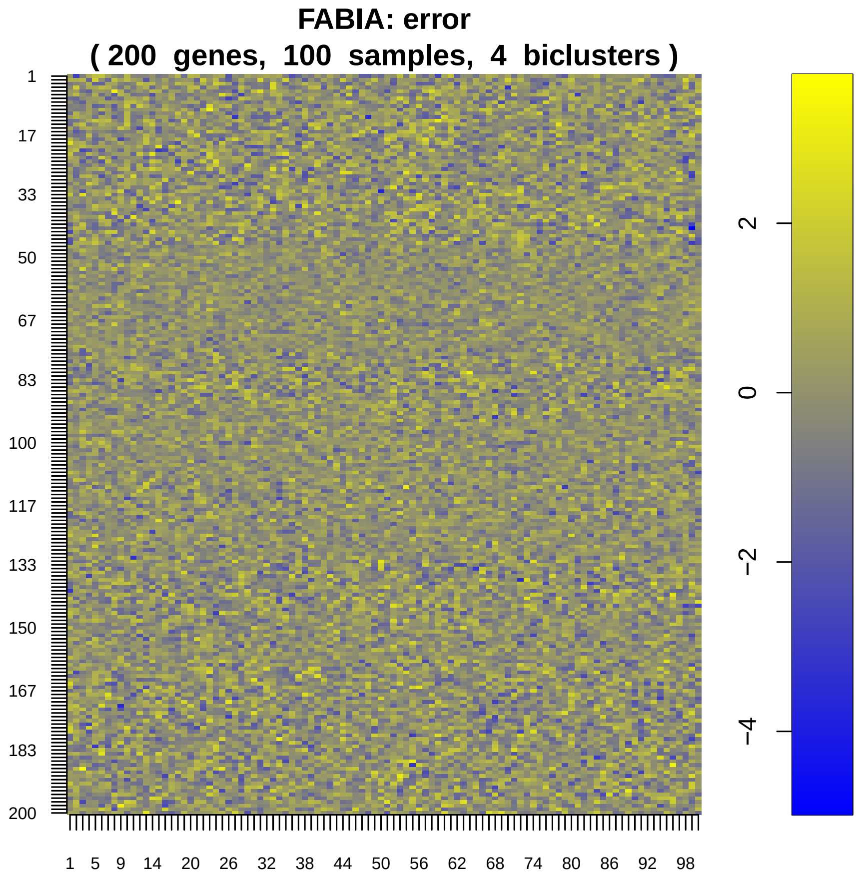

Figure 34: The reconstruction error of FABIA.

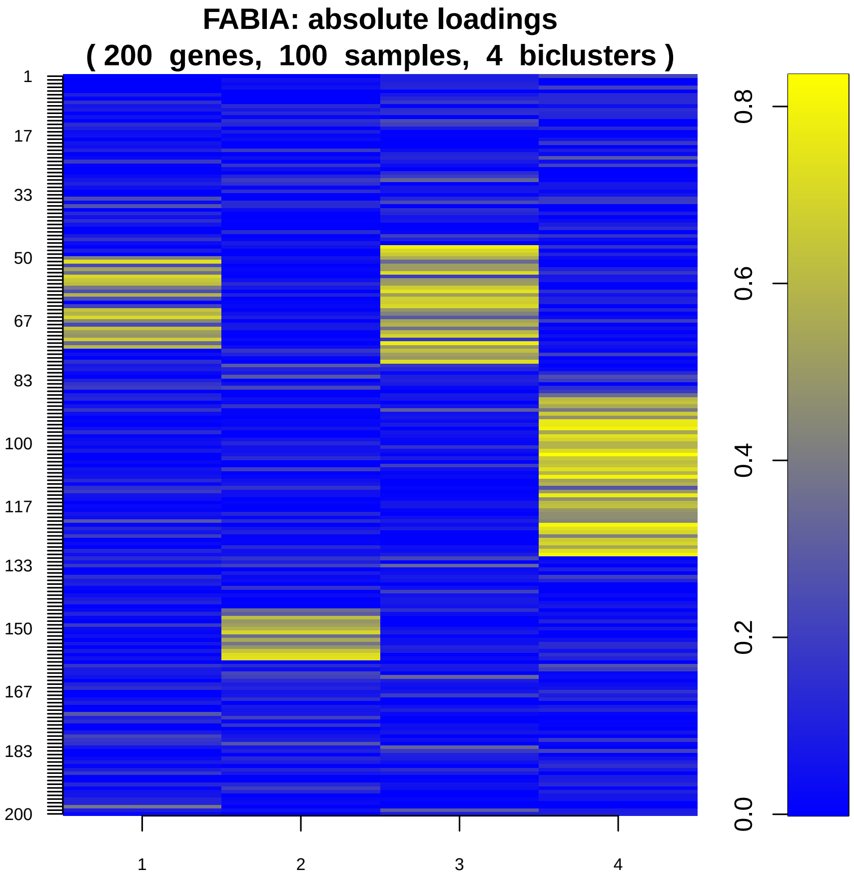

Figure 35: The estimated absolute loadings.

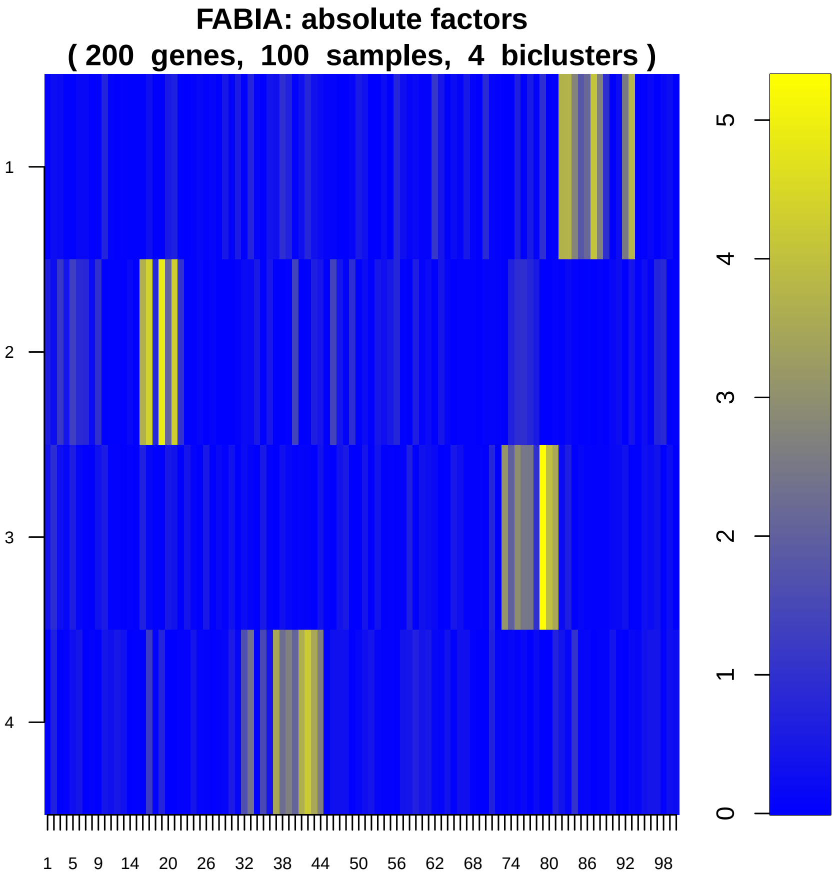

Figure 36: The estimated absolute factors.

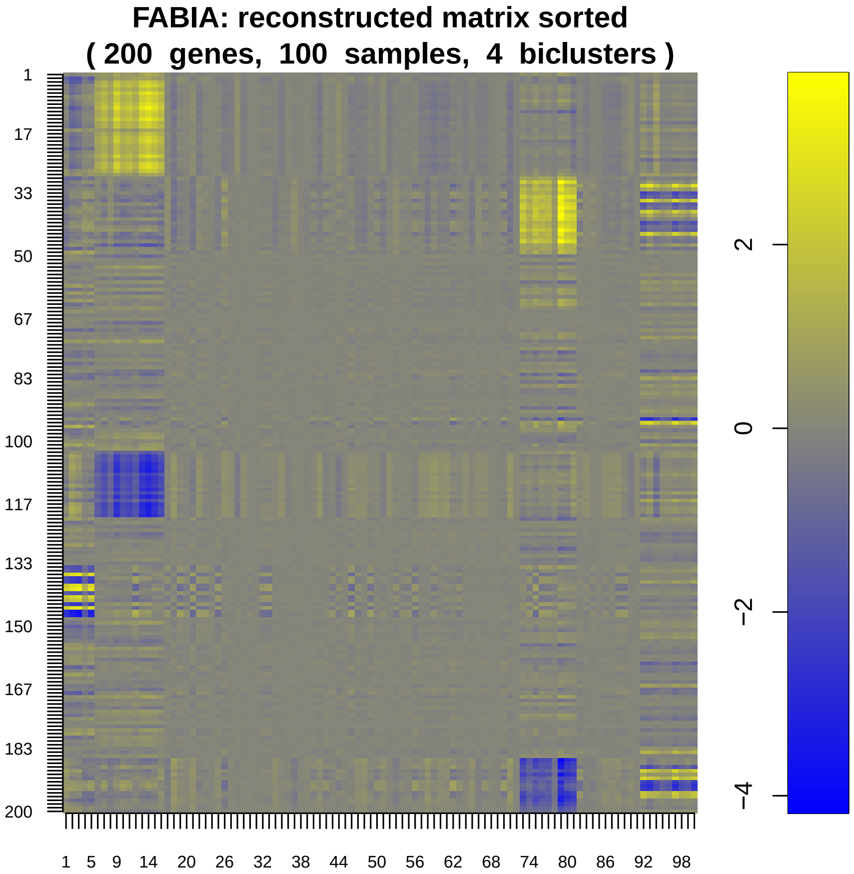

Figure 37: The reconstructed data matrix sorted according to k-means membership applied to factors (columns) and loadings (rows).

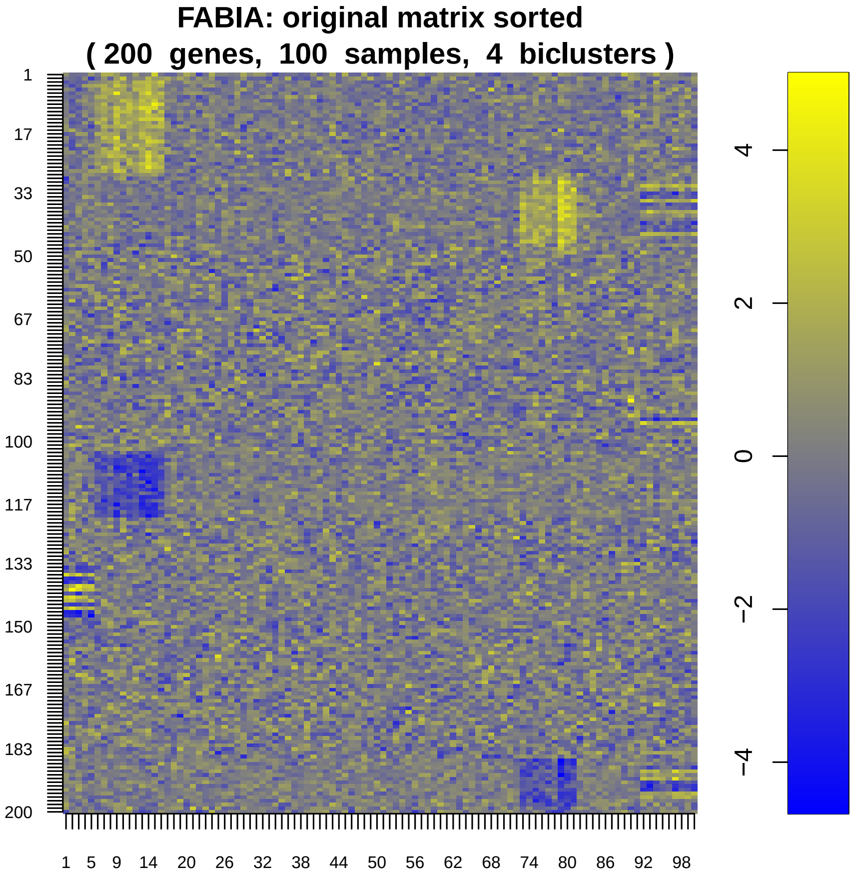

Figure 38: The original data matrix sorted according to k-means membership applied to factors (columns) and loadings (rows).

8. Plot the result as a biplot:

```
> colnames(res@X) <- clab$x
> rownames(res@X) <- llab$x
```

The biclusters will be sorted according to their information content.

Matrix factorization results are presented as *biplots*. A *biplot* allows information on both columns (samples) and rows (genes or variables) of a data matrix (gene expression matrix) to be displayed graphically.

In our plots,

- (1) **indicative row items:** red large circles give the most indicative row items (genes),
- (2) **less indicative row items:** golden small circles are row items (genes) which are less indicative,
- (3) **column items:** squares are the column items (samples),
- (4) **groups of column items:** column items (samples) of the same group have the same color,
- (5) **labels:** all data points are labeled according the column/row position in the original data matrix,
- (6) **true bicluster membership:** If a column/row item belongs to a true bicluster (from data generation) then this is indicated by the bicluster number as a prefix of the label followed by an underscore — note, that an item my belong to more than one bicluster.

If squares/circles have a large deviation from zero in the x-direction then they will be assigned to the bicluster in x direction. If squares/circles have a large deviation from zero in the y-direction then they will be assigned to the bicluster in y direction. Squares/circles with large deviation from zero in $45\,^\circ$ or $135\,^\circ$ belong to both biclusters.

We give the following biplots:

- (1) biplot of bicluster 1 and 2 of the FABIA result (Figure 39),
- (2) biplot of bicluster 1 and 3 of the FABIA result (Figure 40),
- (3) biplot of bicluster 1 and 4 of the FABIA result (Figure 41),
- (4) biplot of bicluster 2 and 3 of the FABIA result (Figure 42),
- (5) biplot of bicluster 2 and 4 of the FABIA result (Figure 43),
- (6) biplot of bicluster 3 and 4 of the FABIA result (Figure 44).

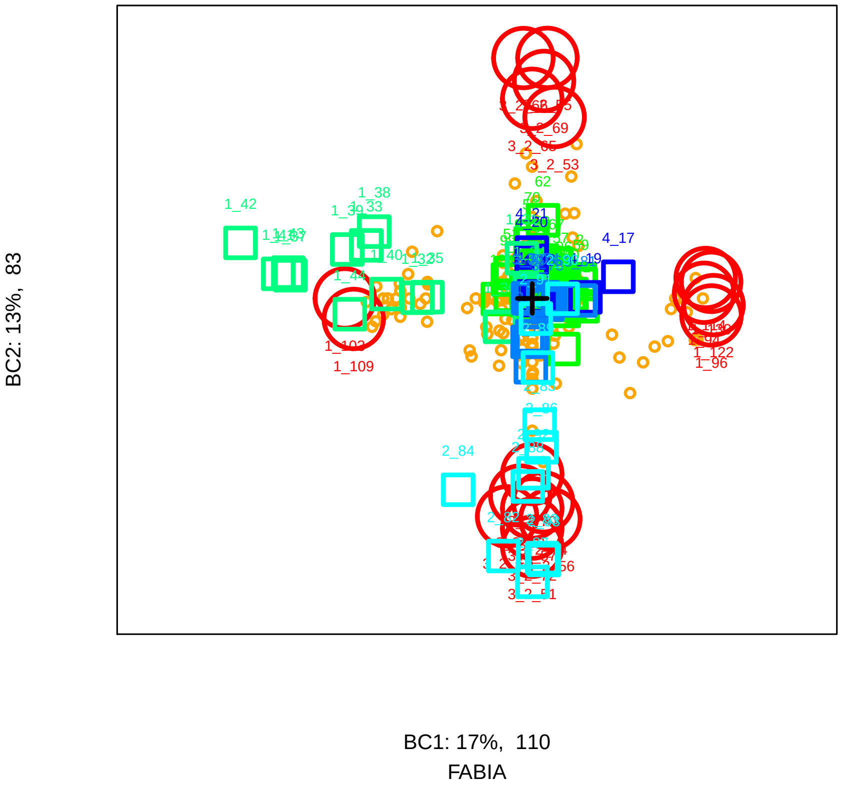

Figure 39: Biplot of bicluster 1 and 2 of the FABIA result, where biclusters are sorted now according to their information content. Red large circles give the most indicative row items. Golden small circles are row items which are less indicative. Squares are the column items, where members of the same group have the same color. All data points are labeled according the column/row position in the original data matrix. If a column/row item belongs to a true bicluster (from data generation) then this is indicated by the bicluster number as a prefix of the label followed by an underscore. Note, that an item my belong to more than one bicluster. If squares/circles have a large deviation from zero in the x-direction then they will be assigned to the bicluster in x direction. If squares/circles have a large deviation from zero in the y-direction then they will be assigned to the bicluster in y direction. Squares/circles with large deviation from zero in $45\,^\circ$ or $135\,^\circ$ belong to both biclusters.

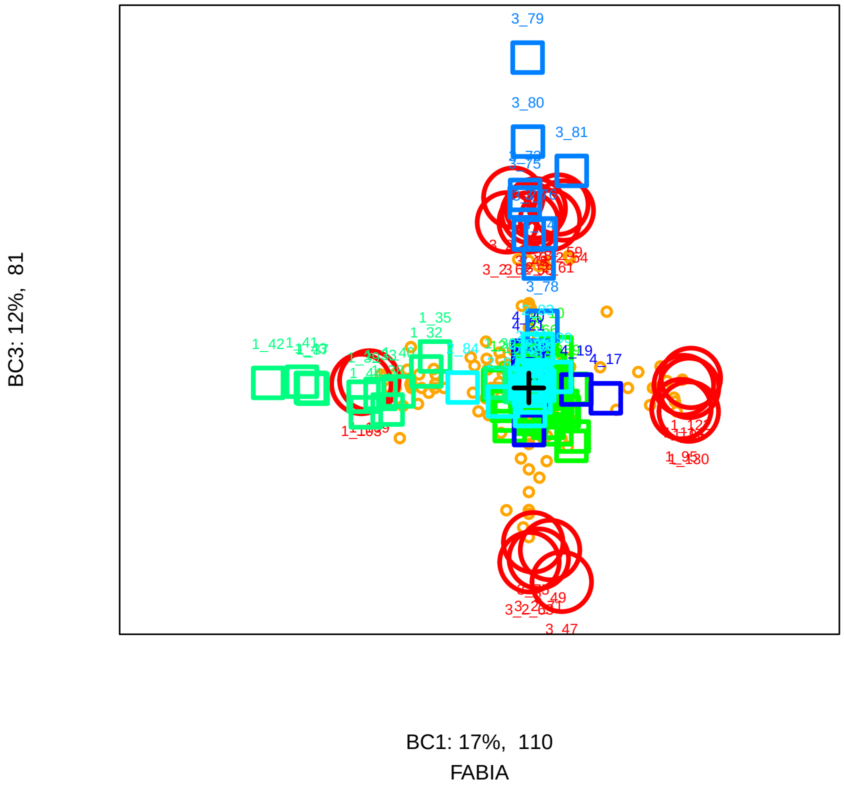

Figure 40: Biplot of bicluster 1 and 3 of the FABIA result. See Fig. 39 for the explanation.

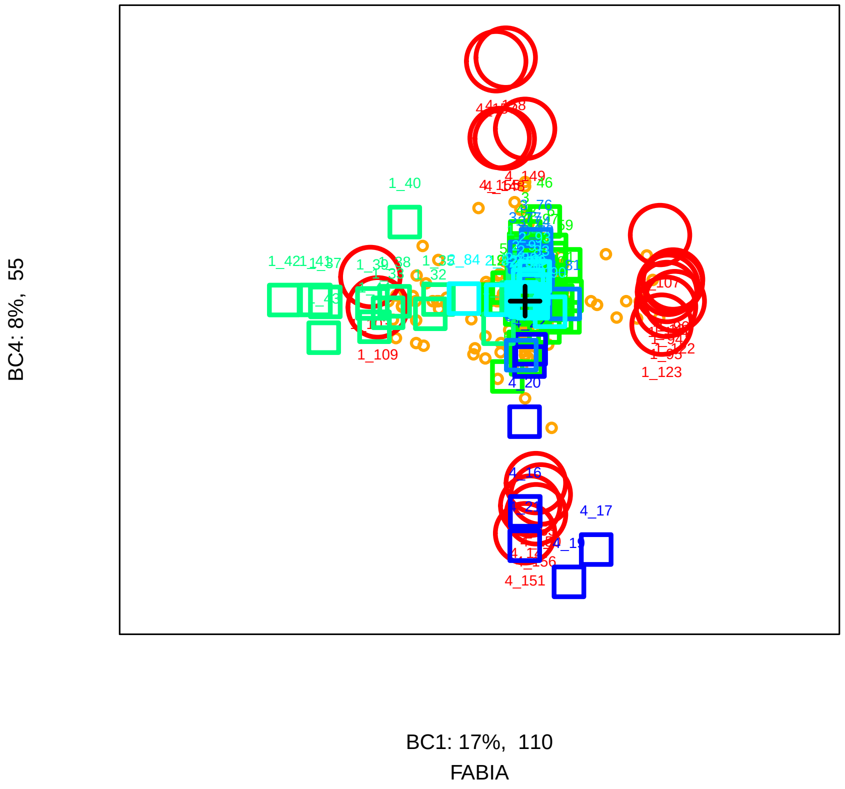

Figure 41: Biplot of bicluster 1 and 4 of the FABIA result. See Fig. 39 for the explanation.

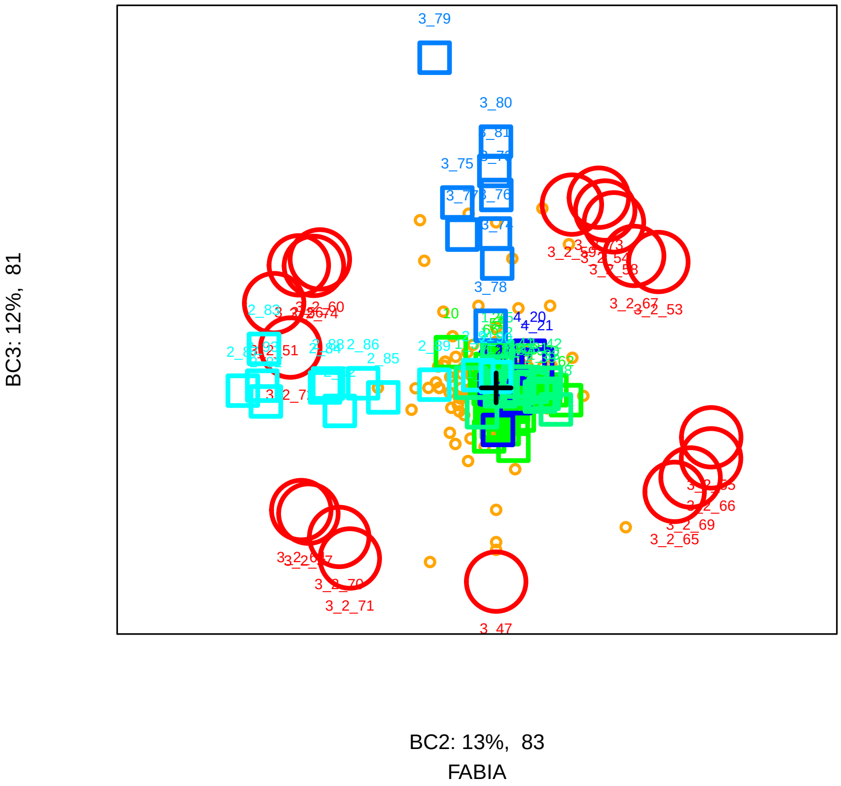

Figure 42: Biplot of bicluster 2 and 3 of the FABIA result. See Fig. 39 for the explanation. This is an nice example where bicluster 2 and 3 share most of the genes. Those shared genes are large circles on the 45° line or on the 135° line depending on the sign of loadings of the genes for the biclusters. The shared genes can be seen at the reconstructed data in Fig. 33 and the loadings in Fig. 35. Note, that the biclusters are sorted now according to their information content, thus bicluster 2 is the extracted bicluster 4 and bicluster 3 is the extracted bicluster 1.

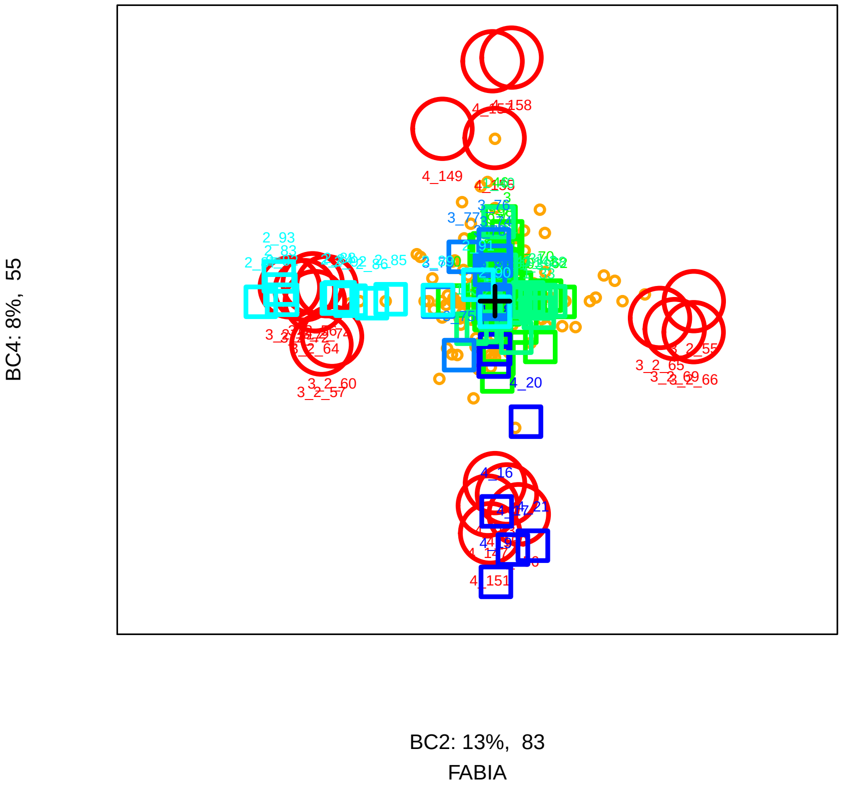

Figure 43: Biplot of bicluster 2 and 4 of the FABIA result. See Fig. 39 for the explanation.

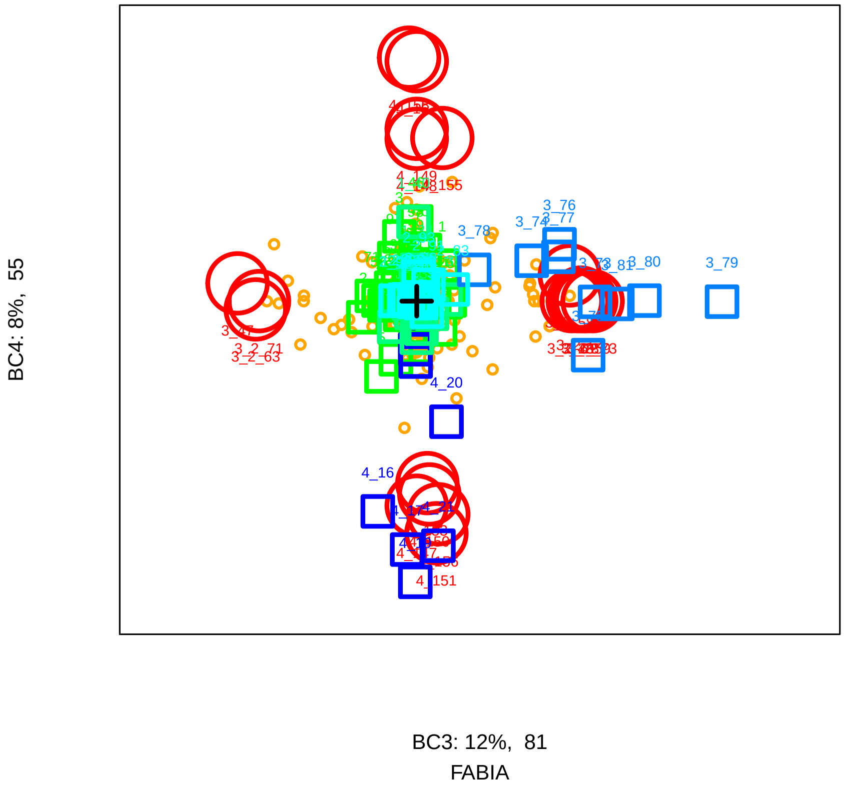

Figure 44: Biplot of bicluster 3 and 4 of the FABIA result. See Fig. 39 for the explanation.

9. Extract biclusters:

   ```
   > rb <- extract_bic(res)
   ```

   In `bic` the biclusters are extracted according to the largest absolute values of the component i, i.e. the largest values of $\boldsymbol{\lambda}_i$ and the largest values of $\boldsymbol{z}_i$.

   The components of `bic` are:

   - `binp`,
   - `bixv`,
   - `bixn`,
   - `biypv`, and
   - `biypn`.

   They contain following information:

   - `binp` give the size of the bicluster: number observations and number samples.
   - `bixv` gives the values of the extracted observations that have absolute values above a threshold. They are sorted.
   - `bixn` gives the extracted observation names (e.g. gene names).
   - `biypv` gives the values of the extracted samples that have absolute values above a threshold. They are sorted.
   - `biypn` gives the names of the extracted samples (e.g. sample names).

   In `bicopp` the opposite cluster to the biclusters are given. Opposite means that the negative pattern is present.

   The components of opposite clusters `bicopp` are:

   - `binn`,
   - `bixv`,
   - `bixn`,
   - `biypnv`, and
   - `biynn`.

   They contain following information:

   - `binp` give the size of the opposite bicluster: number observations and number samples.
   - `bixv` gives the values of the extracted observations that have absolute values above a threshold. They are sorted.
   - `bixn` gives the extracted observation names (e.g. gene names).
   - `biynv` gives the values of the opposite extracted samples that have absolute values above a threshold. They are sorted.
   - `biynn` gives the names of the opposite extracted samples (e.g. sample names).

That means the samples are divided into two groups where one group shows large positive values and the other group has negative values with large absolute values. That means a observation pattern can be switched on or switched off relative to the average value.

`numn` gives the indices of `bic` with components:

- `numng` = `bix` and
- `numnp` = `biypn`.

`numn` gives the indices of `bicopp` with components:

- `numng` = `bix` and
- `numnn` = `biynn`.

10. List bicluster 1:

```
> rb$bic[1, ]

$binp
[1] 14 14

$bixv
 [1]  0.7282878 -0.7055136 -0.7050421  0.6752001  0.6471913  0.6392858
 [7] -0.6378798  0.6158848  0.5978589 -0.5855576  0.5782550 -0.5320636
[13]  0.5149076  0.4799769

$bixn
 [1] "51" "55" "66" "72" "56" "64" "69" "57" "74" "65" "60" "53" "70" "71"

$biypv
 [1] 4.0803425 3.7670183 3.7393711 3.7091438 2.7555222 2.7072182 2.5175424
 [8] 2.1430550 1.8178234 0.9949151 0.9910319 0.7287770 0.6252090 0.5440144

$biypn
 [1] "87" "93" "83" "82" "84" "88" "92" "86" "85" "89" "79" "10" "75" "77"
```

11. List bicluster 2:

```
> rb$bic[2, ]

$binp
[1] 10 12

$bixv
 [1] -0.7347599 -0.7237299  0.7003256  0.6429161  0.6170422  0.5831093
 [7]  0.5489500 -0.5199172 -0.4922033 -0.4897646

$bixn
 [1] "158" "157" "151" "156" "147" "150" "153" "149" "155" "148"
```

```
$biypv
 [1] 4.8930808 4.3301115 4.2630195 3.6650263 2.1011037 1.3127166 1.0532333
 [8] 1.0140897 0.9411019 0.8386498 0.7978445 0.6374875

$biypn
 [1] "19" "17" "21" "16" "20" "5"  "22" "9"  "75" "18" "56" "43"
```

12. Show bicluster 3:

```
> rb$bic[3, ]

$binp
[1] 26 14

$bixv
 [1]  0.7840192 -0.7695602 -0.7415442 -0.7264132 -0.7176893 -0.7049110
 [7]  0.7045732  0.6903743 -0.6785510 -0.6698484 -0.6696130  0.6554305
[13]  0.6241190  0.6032392 -0.5814744  0.5638100 -0.5338614 -0.5232321
[19] -0.5199036 -0.5143169 -0.5093665  0.5089974 -0.4958850  0.4944296
[25] -0.4935683  0.4934973

$bixn
 [1] "47" "73" "59" "78" "54" "48" "63" "71" "61" "58" "62" "49" "75" "70" "68"
[16] "50" "67" "76" "60" "52" "53" "57" "56" "64" "74" "77"

$biypv
 [1] -5.3200551 -3.9675752 -3.4955638 -3.1081751 -2.9868574 -2.4878519
 [7] -2.4697203 -2.0037119 -1.0014696 -0.6284133 -0.5760768 -0.5184633
[13] -0.5080778 -0.5069546

$biypn
 [1] "79" "80" "81" "73" "75" "76" "77" "74" "78" "83" "10" "20" "26" "35"
```

13. List bicluster 4:

```
> rb$bic[4, ]

$binp
[1] 35 12

$bixv
 [1]  0.8347163 -0.8073244 -0.7982483 -0.7968363  0.7950034 -0.7722240
 [7] -0.7653183 -0.7602627  0.7373761 -0.7363439 -0.7278777 -0.7235443
[13]  0.7130559  0.6968159  0.6948011 -0.6883460  0.6641010  0.6639213
[19] -0.6561803  0.6432195 -0.6369467  0.6274656 -0.6182669  0.6057058
[25] -0.6041100  0.5902159  0.5878468  0.5877135  0.5851631  0.5525924
[31]  0.5471646 -0.5453413  0.5349082  0.4986227 -0.4938649
```

```
$bixn
 [1] "103" "122" "96"  "130" "109" "114" "94"  "95"  "98"  "123" "107" "129"
[13] "102" "124" "127" "99"  "92"  "126" "104" "106" "89"  "117" "105" "116"
[25] "88"  "108" "100" "101" "111" "90"  "128" "97"  "110" "118" "113"

$biypv
 [1] 4.198769 3.655922 3.510759 3.483028 2.662345 2.626083 2.388467 2.274455
 [9] 2.097810 1.652436 1.510333 1.066223

$biypn
 [1] "42" "41" "43" "37" "39" "44" "33" "38" "40" "32" "35" "84"
```

14. Plot bicluster 1:

    Now the biclusters are visualized in the original data. Each bicluster is visualized in the raw data by two plots:

    - Plot1: The data matrix is sorted such that the bicluster appear at the upper left corner. The bicluster is marked by a rectangle.
    - Plot2: Only the bicluster is plotted.

    We have following figures:

    - Plot1 bicluster 1: Fig. 45,
    - Plot2 bicluster 1: Fig. 46,
    - Plot1 bicluster 2: Fig. 47,
    - Plot2 bicluster 2: Fig. 48,
    - Plot1 bicluster 3: Fig. 49,
    - Plot2 bicluster 3: Fig. 50,
    - Plot1 bicluster 4: Fig. 51,
    - Plot2 bicluster 4: Fig. 52.

    ```
    > plotBicluster(X,unlist(rb$bic[1,5]),unlist(rb$bic[1,3]))
    ```

15. Plot bicluster 2:

    ```
    > plotBicluster(X,unlist(rb$bic[2,5]),unlist(rb$bic[2,3]))
    ```

16. Plot bicluster 3:

    ```
    > plotBicluster(X,unlist(rb$bic[3,5]),unlist(rb$bic[3,3]))
    ```

17. Plot bicluster 4:

    ```
    > plotBicluster(X,unlist(rb$bic[4,5]),unlist(rb$bic[4,3]))
    ```

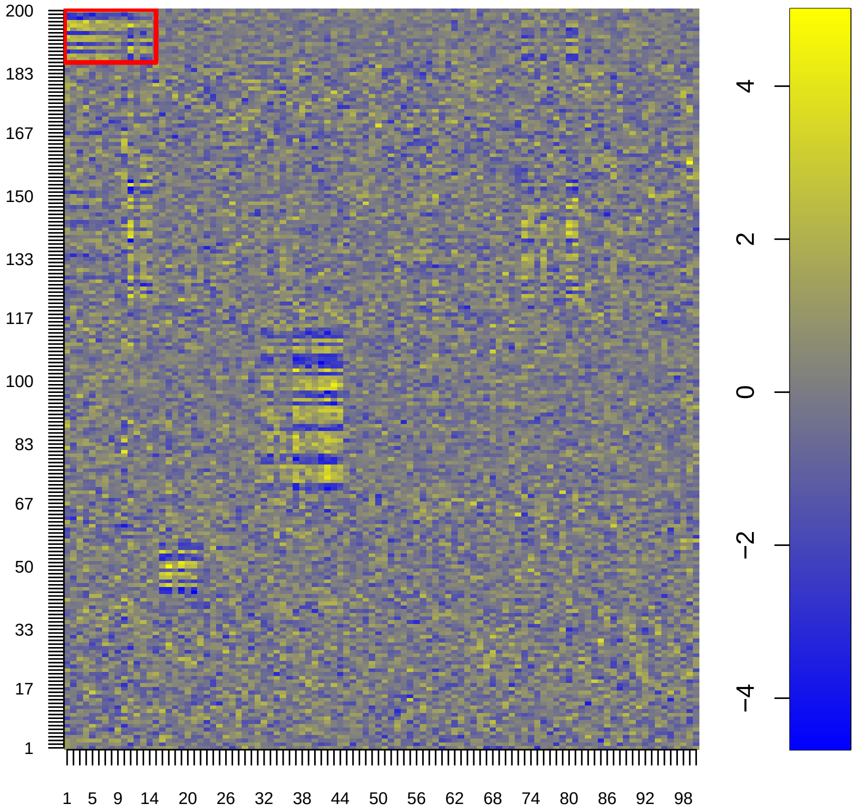

Figure 45: Plot of Bicluster 1. The bicluster is moved to the top left of the original matrix and marked.

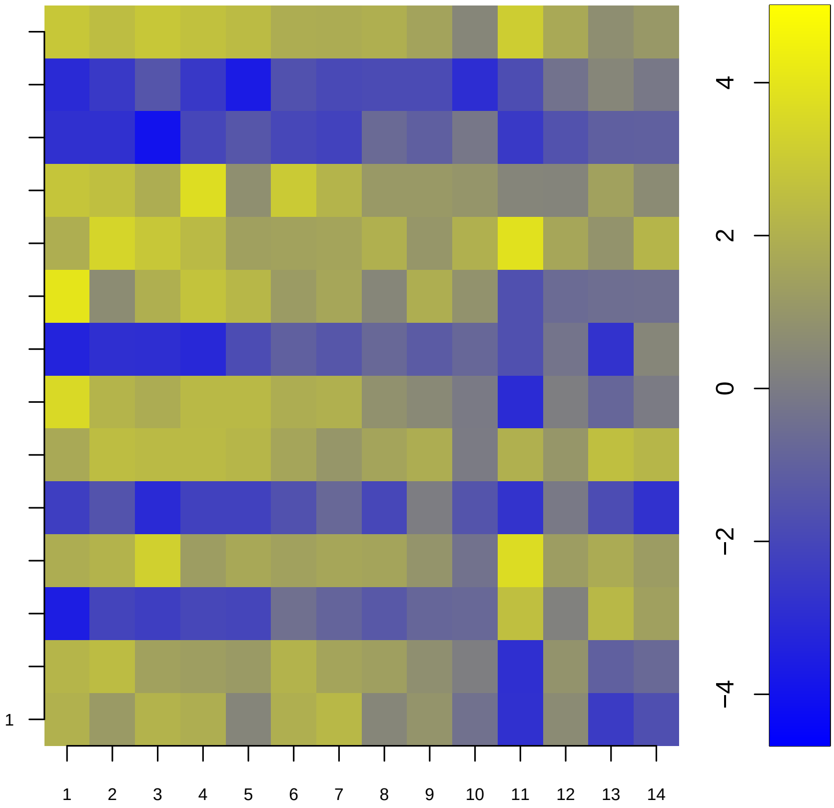

Figure 46: Plot of Bicluster 1. The bicluster, marked region of the previous image, is enlarged.

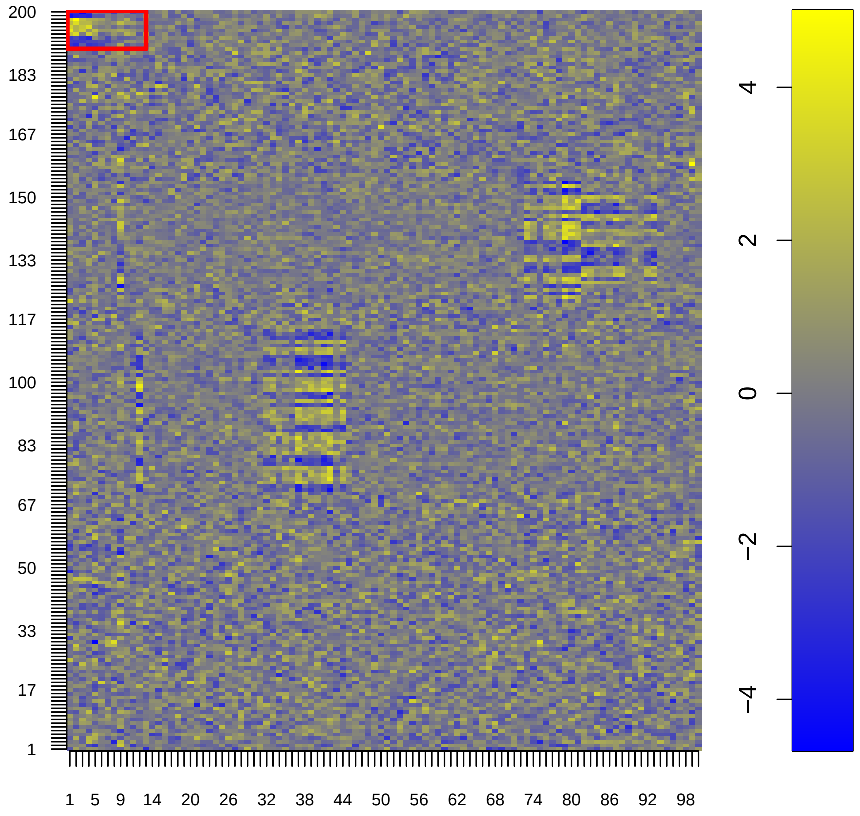

Figure 47: Plot of Bicluster 2. The bicluster is moved to the top left of the original matrix and marked.

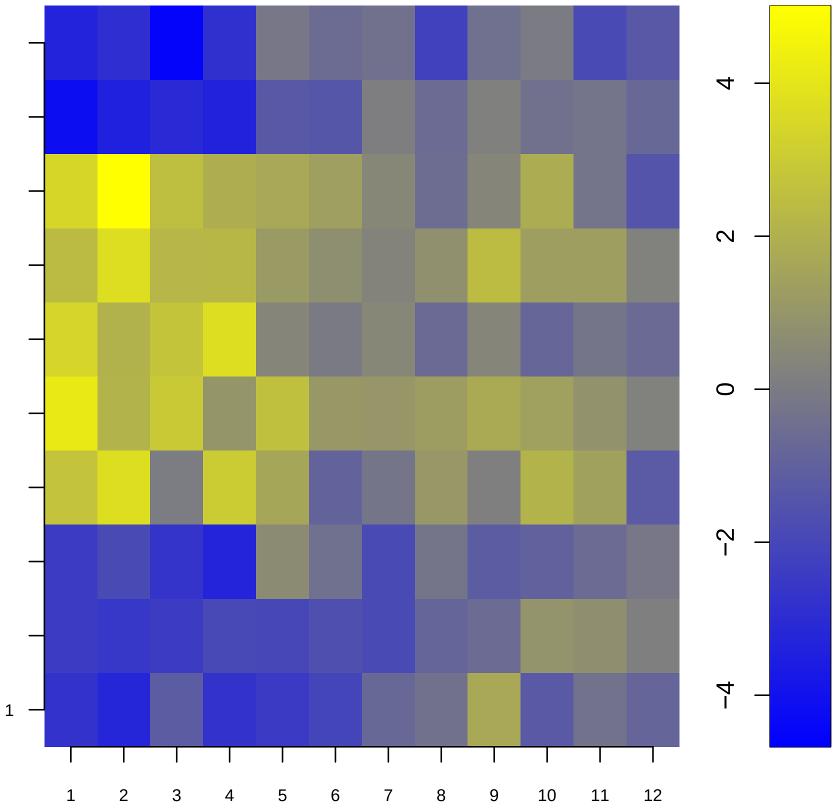

Figure 48: Plot of Bicluster 2. The bicluster, marked region of the previous image, is enlarged.

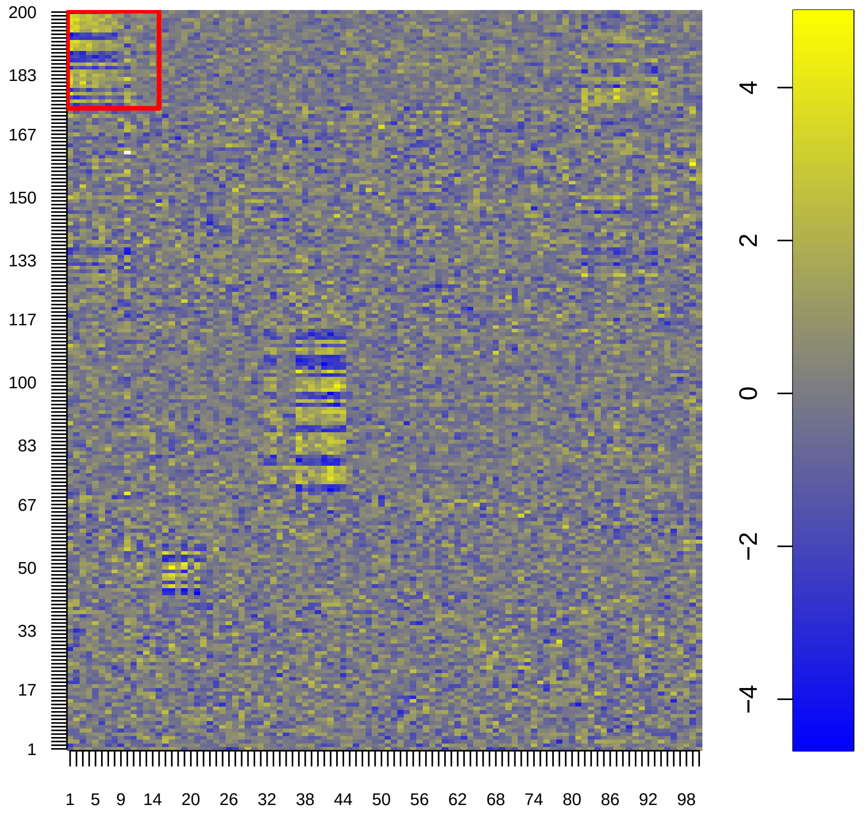

Figure 49: Plot of Bicluster 3. The bicluster is moved to the top left of the original matrix and marked.

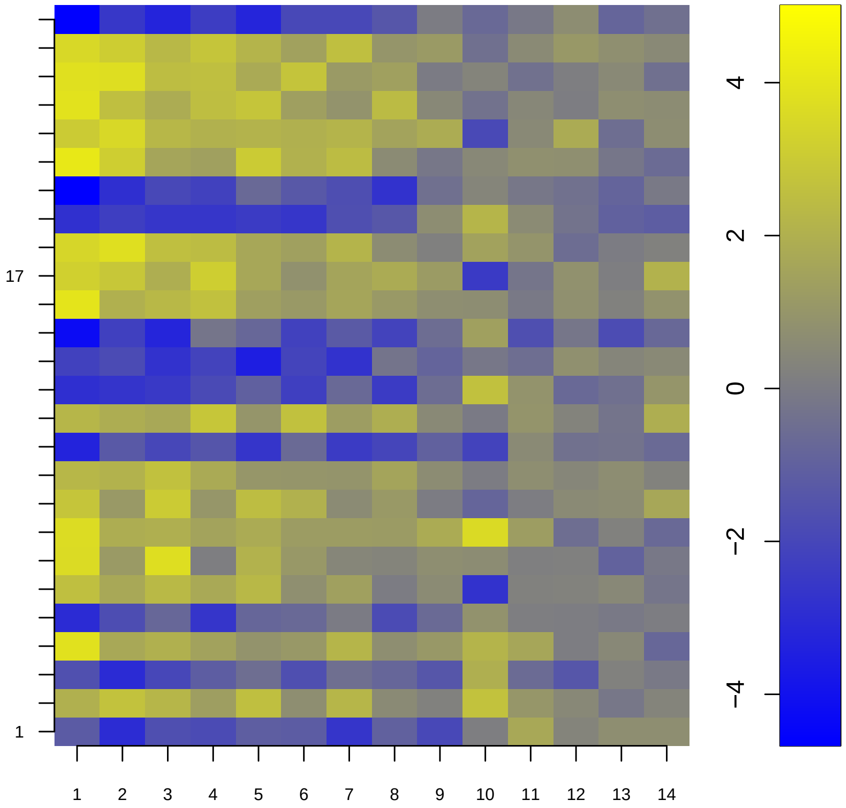

Figure 50: Plot of Bicluster 3. The bicluster, marked region of the previous image, is enlarged.

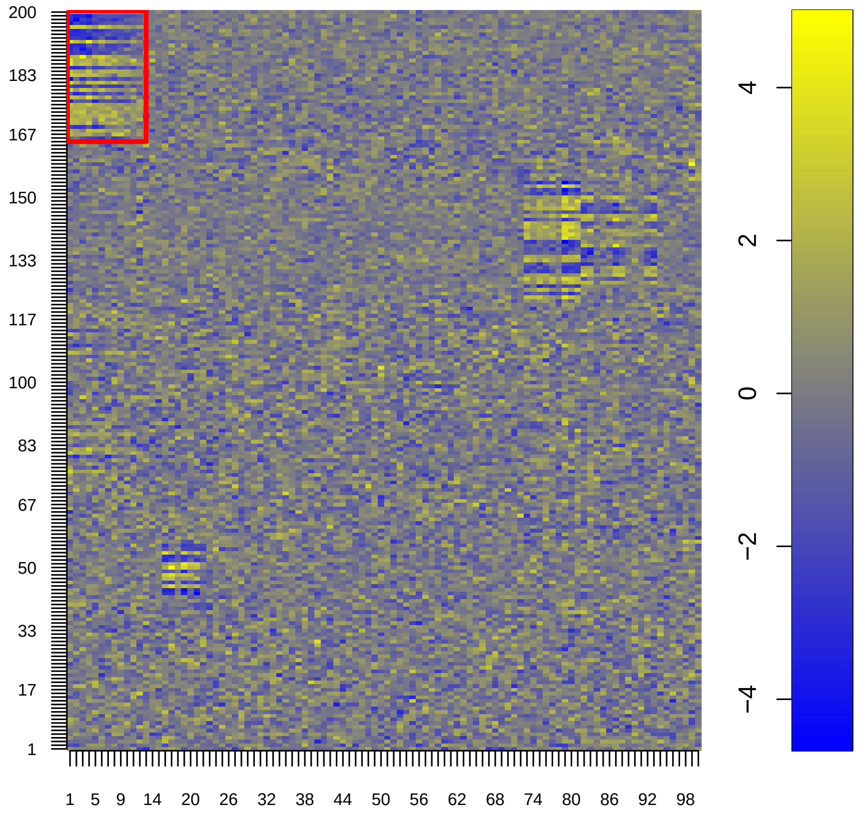

Figure 51: Plot of Bicluster 4. The bicluster is moved to the top left of the original matrix and marked.

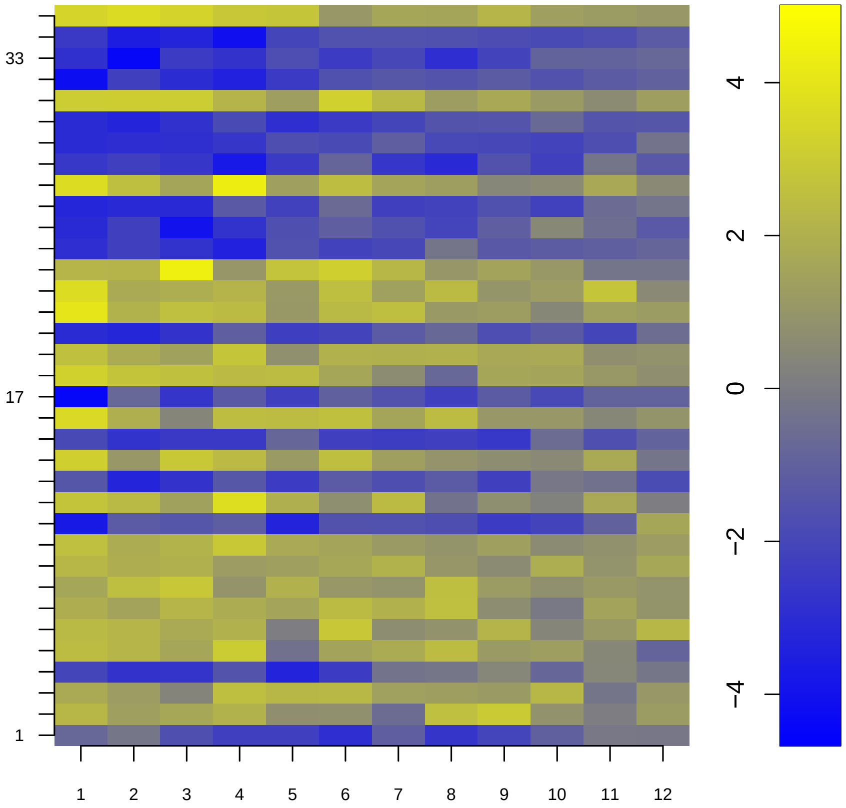

Figure 52: Plot of Bicluster 4. The bicluster, marked region of the previous image, is enlarged.

18. List opposite bicluster 1:

```
> rb$bicopp[1, ]

$binn
[1] 14 12

$bixv
 [1]  0.7282878 -0.7055136 -0.7050421  0.6752001  0.6471913  0.6392858
 [7] -0.6378798  0.6158848  0.5978589 -0.5855576  0.5782550 -0.5320636
[13]  0.5149076  0.4799769

$bixn
 [1] "51" "55" "66" "72" "56" "64" "69" "57" "74" "65" "60" "53" "70" "71"

$biynv
 [1] -1.1258789 -0.9619663 -0.8908189 -0.7996883 -0.7895223 -0.7641011
 [7] -0.7051749 -0.6602593 -0.5784825 -0.5447497 -0.5399782 -0.5005576

$biynn
 [1] "62" "38" "70" "42" "56" "33" "39" "21" "31" "50" "20" "67"
```

19. Plot of opposite bicluster 1:

    Now the opposite biclusters are visualized in the original data. In some applications the negative row pattern may be important which is given by opposite biclusters.

    - Plot1: The data matrix is sorted such that the opposite bicluster appear at the upper left corner. The opposite bicluster is marked by a rectangle.
    - Plot2: Only the opposite bicluster is plotted.

    We have following figures:

    - Plot1 opposite bicluster 1: Fig. 53,
    - Plot2 opposite bicluster 1: Fig. 54.

```
> plotBicluster(X,unlist(rb$bicopp[1,5]),unlist(rb$bic[1,3]))
```

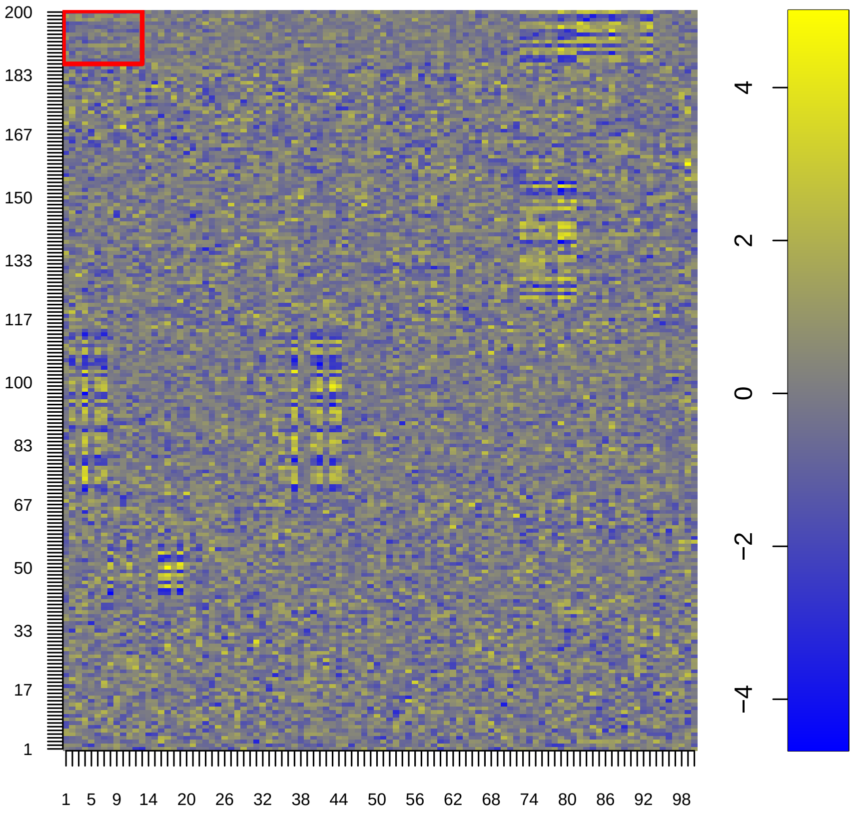

Figure 53: Plot of opposite Bicluster 1. The bicluster is moved to the top left of the original matrix and marked.

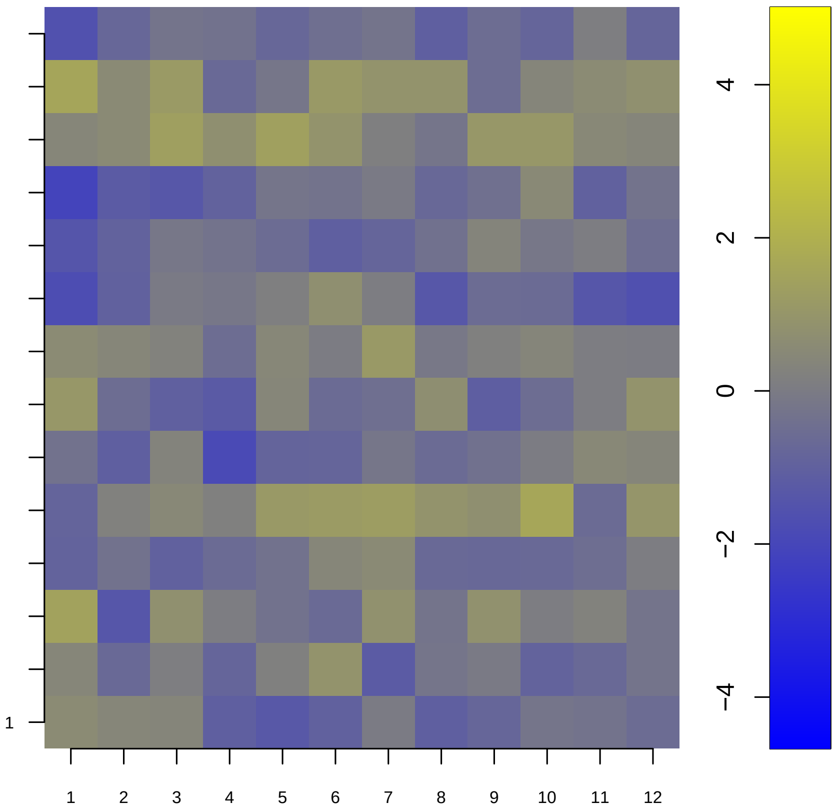

Figure 54: Plot of opposite Bicluster 1. The bicluster, marked region of the previous image, is enlarged.