Functional Multivariate analysis with the tmod package

Tips and tricks

January Weiner january@mpiib-berlin.mpg.de

2015-06-24

Overview

About this presentation

This is an Rmarkdown document; it includes all code necessary to run *every* plot shown in this presentation. You can recreate all the plots or extract all code from the presentation.

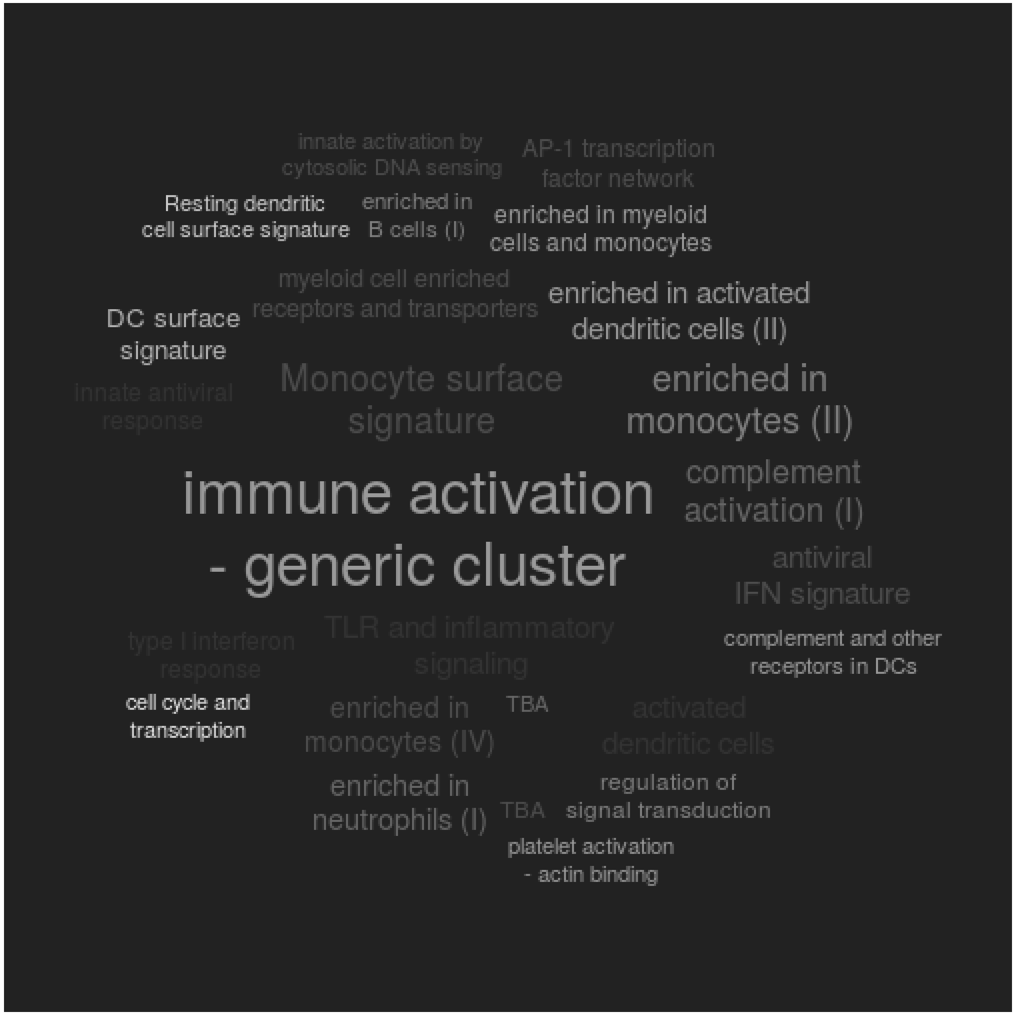
Enrichment tests

are an important tool in functional analysis of gene expression data – it turns unreadable lists of genes into something useful

```
## Loading required package: methods
```

```
## Error in is.data.frame(x): object 'E' not found
```

```
##
## 4178                                                     family with sequence sim
## 20799                                           Fc fragment of IgG, high affinity
## 4122                                              basic leucine zipper transcripti
## 23567                                                                         anky
## 20498
## 20360
## 2513
## 24032                                                                          Gol
## 1337
## 467                                     serpin peptidase inhibitor, clade G (C1
## 18119                                                                          BE
## 14168                                                                      guanyla
## 19820                                                   dehydrogenase/reductase
## 19404                                                         growth factor rece
## 36635                                                    family with sequence si
## 23807                                                      kringle containing tra
## 44719
## 17853                                     guanylate binding protein 1, interf
```


innate activation by cytosolic DNA sensing
AP-1 transcription factor network
Resting dendritic cell surface signature
enriched in B cells (I)
enriched in myeloid cells and monocytes
myeloid cell enriched receptors and transporters
enriched in activated dendritic cells (II)
DC surface signature
Monocyte surface signature
enriched in monocytes (II)
innate antiviral response
immune activation - generic cluster
complement activation (I)
antiviral IFN signature
type I interferon response
TLR and inflammatory signaling
complement and other receptors in DCs
cell cycle and transcription
enriched in monocytes (IV)
TBA
activated dendritic cells
enriched in neutrophils (I)
TBA
regulation of signal transduction
platelet activation - actin binding

Two new approaches will be presented:

- a new statistical test for continuous enrichments
- a method for ordering genes

Multivariate analysis + enrichment = Functional multivariate analysis (FMA)

Combination of multivariate techniques such as PCA and functional enrichment analysis can circumvent the need for analysis of differential expression. A primer on FMA will be presented here.

tmod

We introduce *tmod*, an R package which implements several of the shown approaches, and more.

CERNO test: a variant of Fisher’s exact test

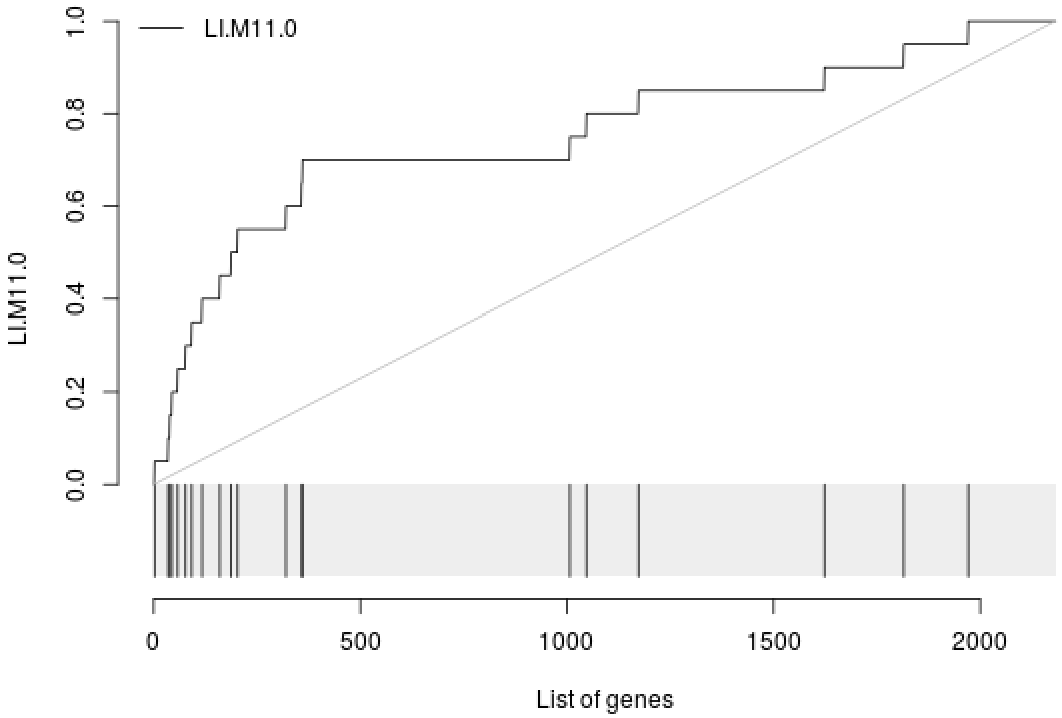
Some enrichment tests (such as the hypergeometric test) rely on arbirtrary tresholds to divide the genes into “differentially expressed” and “background” (or equivalent sets). It is easy to run a statistical test on such a setup, however it is problematic: the number of significantly regulated genes depends on the statistical power, i.e. for example on the number of samples.

Better tests yet are independent of arbitrary thresholds. Examples include

- Randomization approaches (such as GSEA)
- ANOVA-like approaches
- Mann-Whitney U statistic

How does this work?

```r
evidencePlot(l=tt$GENE_SYMBOL, m="LI.M11.0")
```



In an U-test, the U statistic is (almost) the same as the Area Under Curve:

$$r = 1 - \frac{2 \cdot U}{n_1 \cdot n_2} = 1 - 2 \cdot \mathrm{AUC}$$

(r is the effect size for an U-test)

CERNO: Ranks can be treated as probabilities

$$P(rank(g_j) < rank(g_i)) = \frac{rank(g_i)}{N}$$

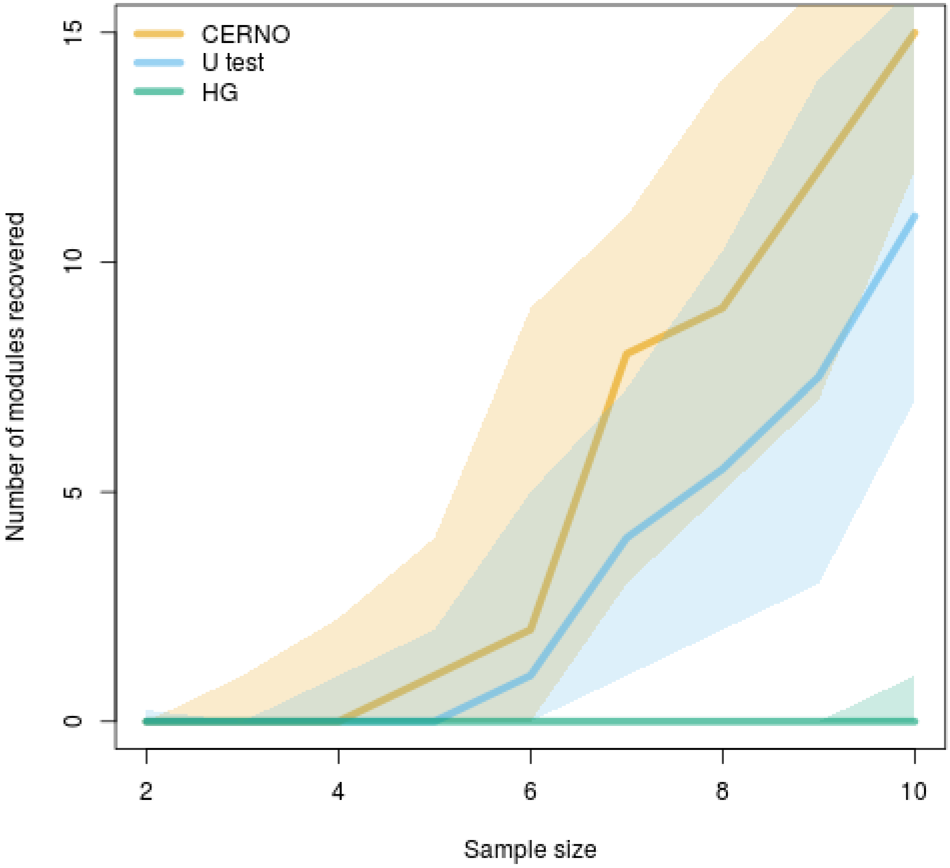
Where N is the total number of genes.

We apply Fisher’s method to ranks

$$\mathbf{CERNO} = -2 \cdot \sum_{i=1}^{N} \ln(\frac{rank(g_i)}{N})$$

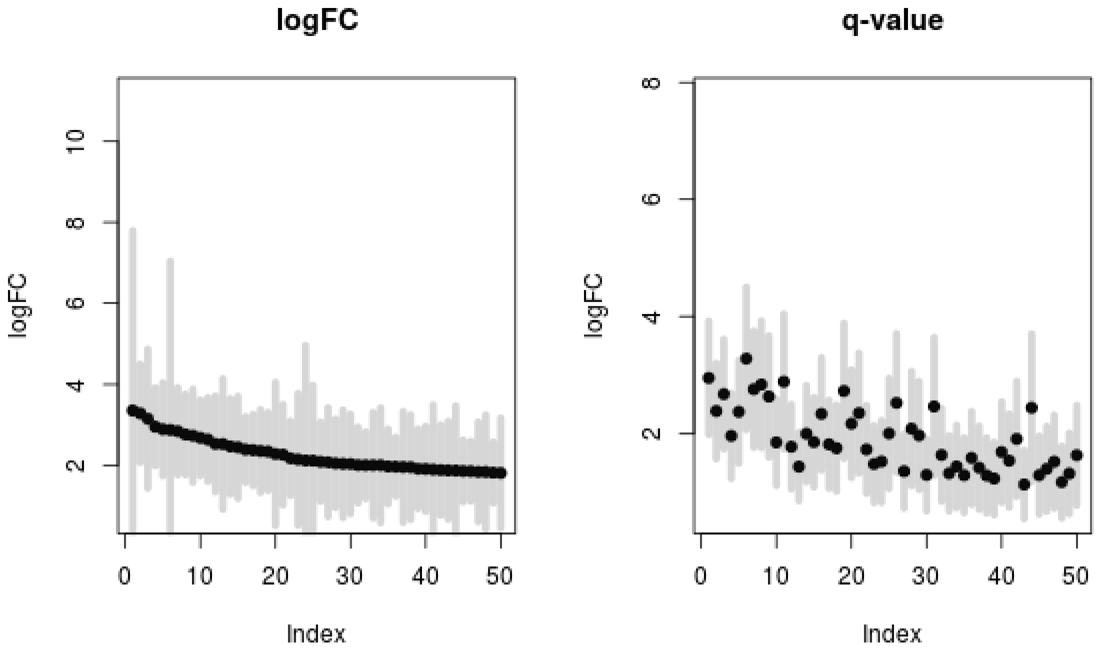
The statistics has a χ^2 distribution with $2 \cdot N$ degrees of freedom.

First, second and third quartiles of number of modules recovered by the different statistical tests in dependence of the sample size in 100 random sample replicates.



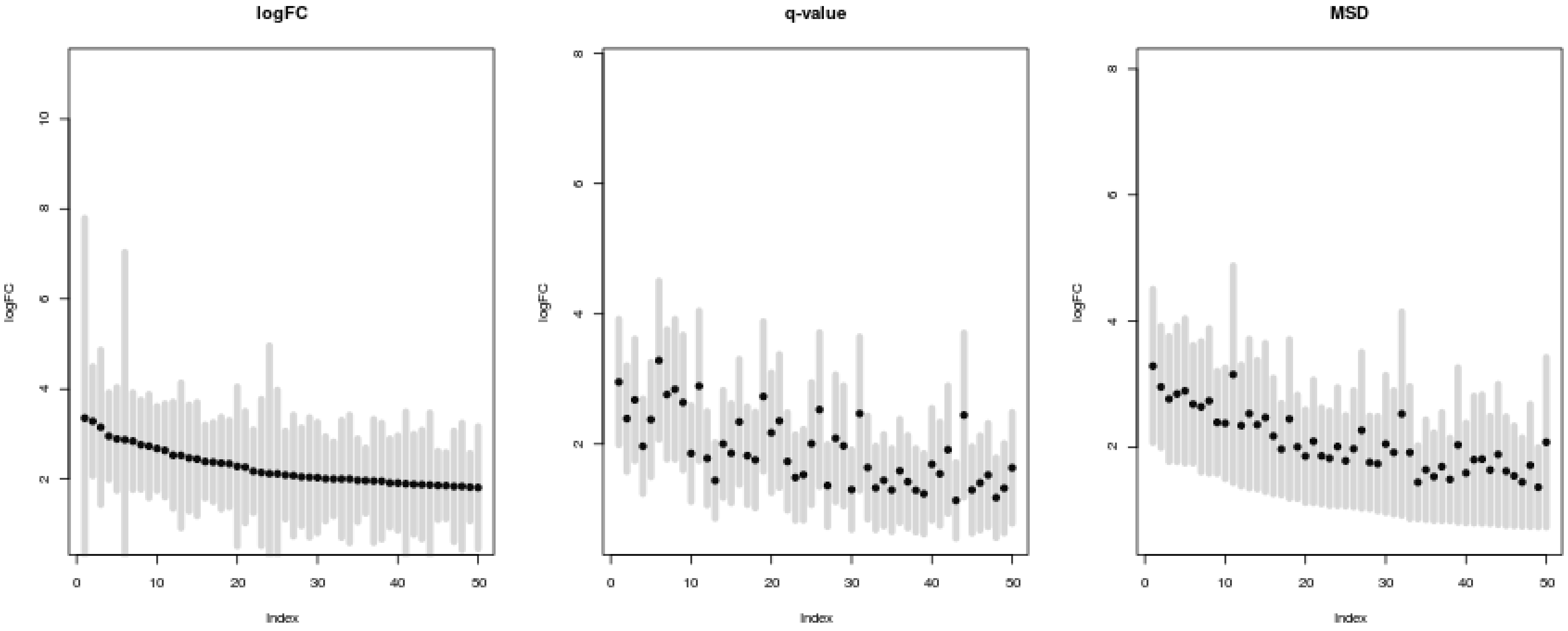
How to order genes?

- Order by p-values (common approach).
  - Genes with strong expression tend to have lower p-values even if log-fold changes are small
- Order by (absolute) log fold change
  - Genes with weak expression (near background) can have huge log fold changes despite lack of significance

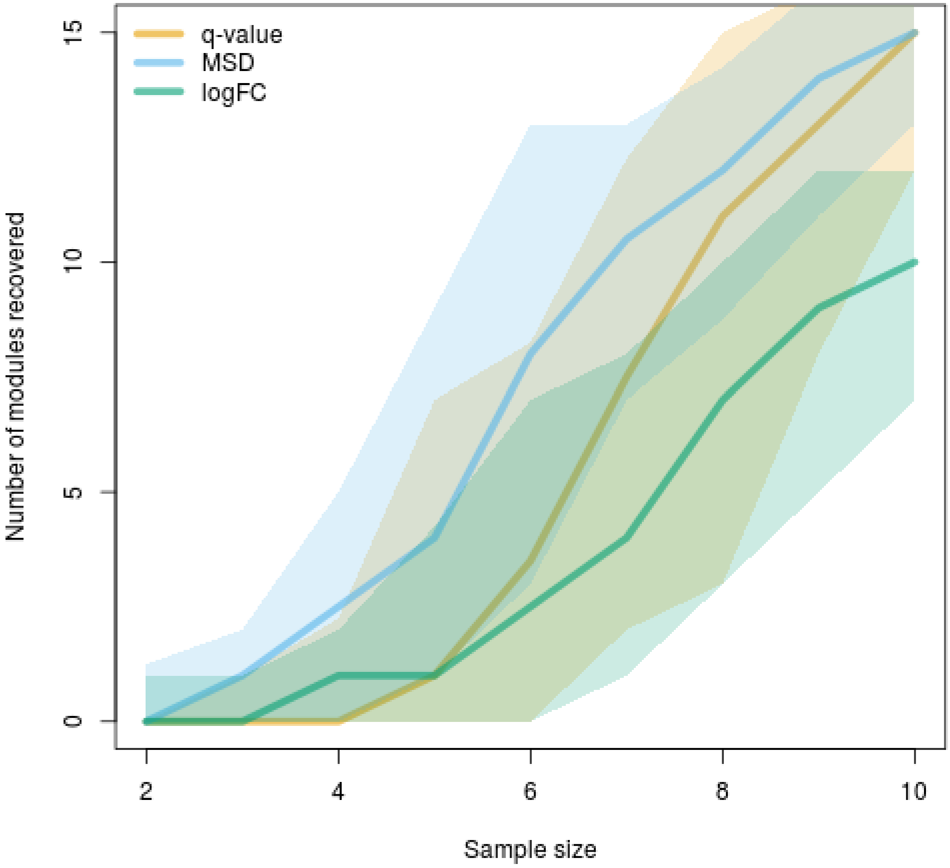

logFC
logFC
2
4
6
8
10
0
10
20
30
40
50
Index
q-value
logFC
2
4
6
8
0
10
20
30
40
50
Index

MSD – Minimal Significant Difference

$$\text{MSD} = \begin{cases} CI.L & \text{if logFC} > 0 \\ -CI.R & \text{if logFC} < 0 \end{cases}$$



First, second and third quartiles of number of modules recovered by the different approaches in dependence of the sample size in 100 random sample replicates.



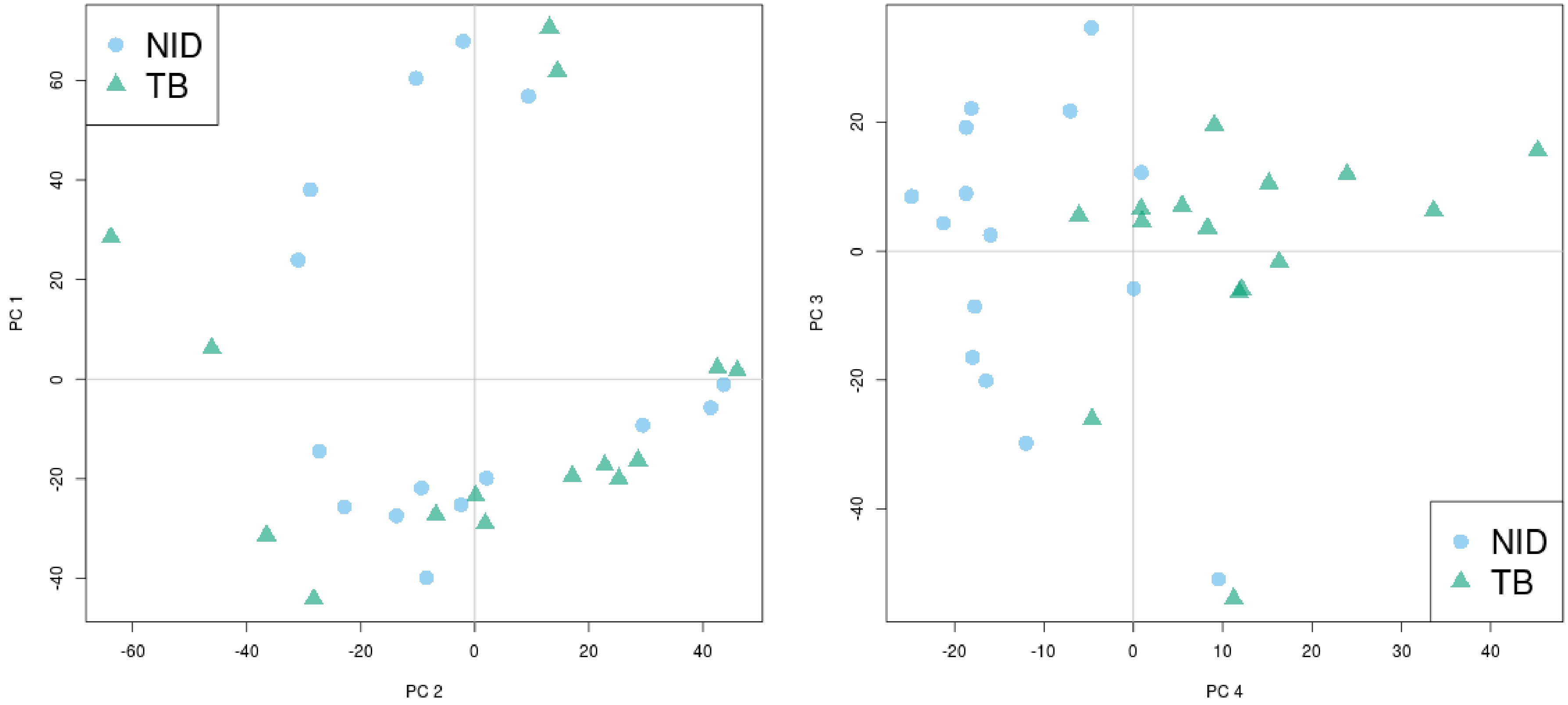
Functional multivariate analysis
(a primer)

Functional Principal Component Analysis (PCA)

In PCA, the $N \times K$ matrix $\mathbf{X}$ of N samples and K variables (e.g. genes) is rotated, which results in a new matrix, $\mathbf{Y}$, with N samples and J principal components (PCs).

Effectively, a $K \times J$ matrix $\mathbf{W}$ is calculated, such that

$$\mathbf{X} \times \mathbf{W} = \mathbf{Y}$$



Question in MFA: *What do these components mean?*

$$\mathbf{X} \times \mathbf{W} = \mathbf{Y}$$

Each column of $\mathbf{X}$ is a principal component. Each row corresponds to one sample.

A value for a given PC j and a given sample n is calculated as

$$y_{n,j} = \sum_{k=1}^{K} w_{j,k} \cdot x_{k,n}$$

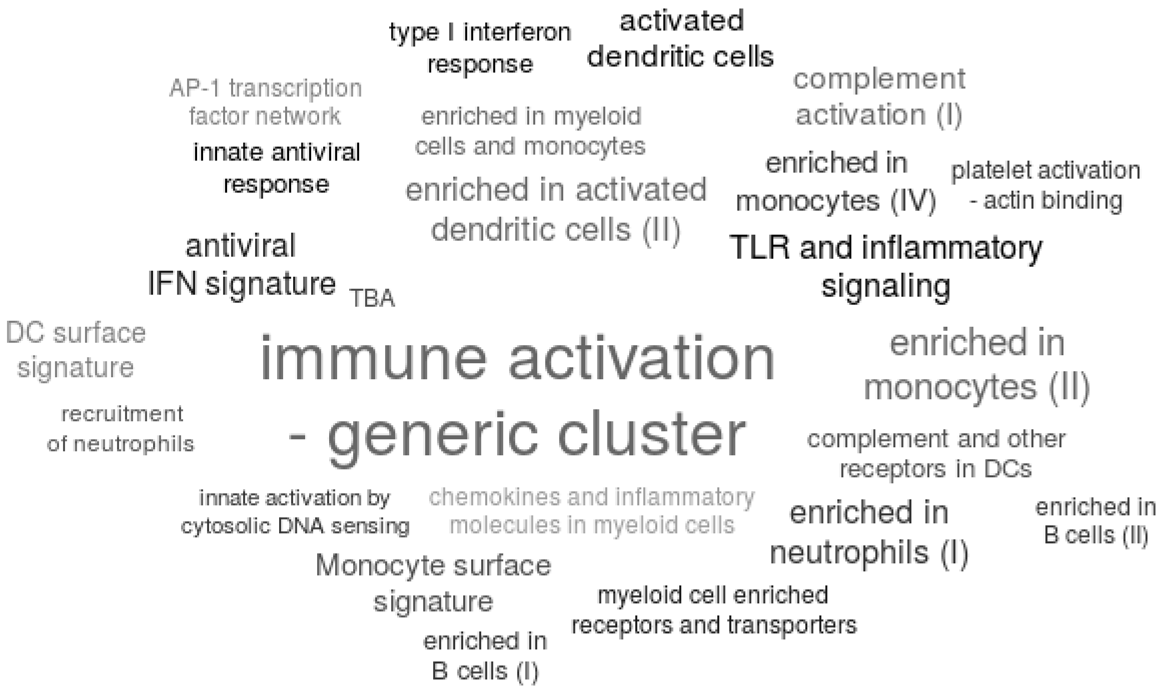
The terms $w_{j,k}$ are variable- (or: gene-) specific *weights* or *loadings* for each component j.

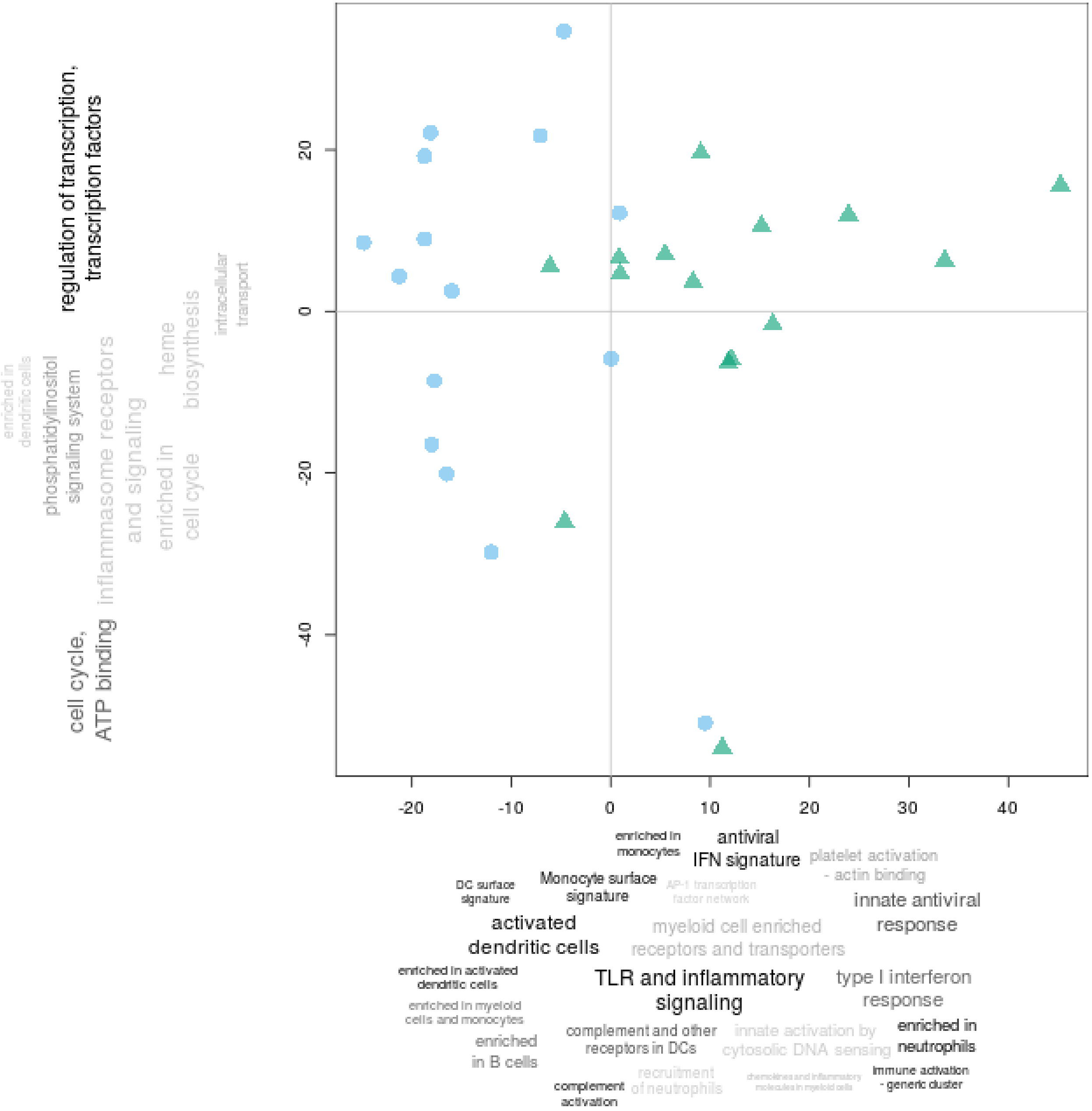
$$y_{(n,j)} = \sum_{k=1}^{K} w_{k,j} \cdot x_{k,n}$$

The larger the absolute value of $w_{k,j}$, the more impact this gene has on the j-th principal component.

We can sort the genes by their weight in a component. Since as a result we get a sorted list of genes, we can apply a continuous enrichment algorithm.

Enrichment in component 4





20
0
-20
-40
-20
-10
0
10
20
30
40
regulation of transcription, transcription factors
intracellular transport
heme biosynthesis
enriched in cell cycle
inflammasome receptors and signaling
phosphatidylinositol signaling system
enriched in dendritic cells
cell cycle, ATP binding
enriched in monocytes
antiviral IFN signature
platelet activation - actin binding
DC surface signature
Monocyte surface signature
AP-1 transcription factor network
innate antiviral response
activated dendritic cells
myeloid cell enriched receptors and transporters
enriched in activated dendritic cells
TLR and inflammatory signaling
type I interferon response
enriched in myeloid cells and monocytes
complement and other receptors in DCs
innate activation by cytosolic DNA sensing
enriched in neutrophils
enriched in B cells
recruitment of neutrophils
chemokines and inflammatory molecules in myeloid cells
immune activation - generic cluster
complement activation

This approach works well also with other multivariate analyses such as independent component analysis (ICA), partial least squares (PLS) or correspondance analysis (CA).

Directly combining multivariate analyses with gene set enrichment allows us to achieve the same results without involving a direct group - to - group comparison. This makes it especially suitable for exploratory analyses.

Serial analysis of enrichment with *tmod*

tmod has been designed as a package for testing the enrichment of blood transcriptional modules. Therefore, *tmod* contains two sets of blood transcriptional module definitions; however, it can be used with any arbitrary gene set definition (e.g. GSEA/MSigDB) or high throughput data type (e.g. metabolomics)

tmod implements HG / U / CERNO tests, functional multivariate analyses, serial analysis / visualization and more.

Availability: http://bioinfo.mpiib-berlin.mpg.de/tmod/

Example: MFA with R and tmod

Data set Egambia: GEO GSE28623.

```
library(tmod)
data(Egambia)
head(Egambia)
```

```
##    GENE_SYMBOL                                      GENE_NAME     EG        NID
## 34    C19orf15            chromosome 19 open reading frame 15  57828  3.2618218
## 36     UNQ9368                                       RTFV9368 643036  1.5671748 4
## 41      ADORA3                          adenosine A3 receptor    140  6.2246027 5
## 44        CDH6 cadherin 6, type 2, K-cadherin (fetal kidney)"   1004  0.8328559
## 52       VASH1                                    vasohibin 1  22846 11.3952226 1
## 62     MAB21L2                     mab-21-like 2 (C. elegans)  10586  5.7530317
##          NID       NID        NID        NID       NID       NID       NID
## 34  4.617986  3.033595  3.1866326  3.6506719  3.787375  3.019342  2.795293  3.020
## 36  4.786995  3.091925  2.2736422  4.1327518  3.934754  3.077131  6.428547  4.655
## 41  6.878103  4.702415  7.6848512  5.2048066  4.836591  4.965997  8.234983  5.072
## 44  2.589377  3.307486  0.7026353  0.8349973  3.951534  2.112500  1.223633  1.477
## 52 11.376962 13.061029 13.0915988 12.0304966 11.980200 12.323327 11.076847 13.187
## 62  7.167419  6.299295  5.8910289  5.4252899  5.265659  6.367774  6.691451  6.351
##           TB        TB        TB        TB        TB        TB        TB
## 34  3.962293  2.080173  3.750405  2.248475  4.148280  4.203384  4.319223
## 36  5.551801  5.021816  5.338259  6.258222  6.383069  5.995486  5.203686
## 41  7.689227  6.004437  5.928957  5.178725  5.661376  6.611350  6.008429
## 44  0.977040  1.174805  1.764985  3.400484  2.486234  1.115761  1.454525
```

```
pca <- prcomp(t(Egambia[,-c(1:3)]), scale.=TRUE)
names(pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
head(pca$x[,1:5])
```

```
##             PC1        PC2        PC3          PC4        PC5
## NID   -27.40722 -13.745073  21.755484  -7.09238147   3.427730
## NID.1  60.44502 -10.323684 -20.091148 -16.49059391  28.610291
## NID.2  67.86063  -2.049866  -5.840748   0.03206066  -4.409519
## NID.3  -5.69328  41.378405 -16.466891 -18.00896080  -5.135397
## NID.4 -25.59922 -22.852422  22.133902 -18.13982412 -13.619485
## NID.5 -39.83765  -8.461881 -50.897837   9.51198824   3.981510
```

```
head(pca$rotation[,1:5])
```

```
##              PC1           PC2          PC3          PC4          PC5
## 34 -0.018481790  0.0024852364 -0.005385593  0.002711500 -0.025291642
## 36  0.003759772 -0.0010608658 -0.018658252  0.041998920  0.006653159
## 41 -0.016103961  0.0002934259 -0.009930961 -0.008813419 -0.011689654
## 44  0.021992983 -0.0073125000  0.005115505  0.015148515  0.001656177
## 52  0.014524608  0.0258657287  0.015226449 -0.009626087 -0.001871263
## 62 -0.001791174  0.0081258050 -0.016757250  0.001834450 -0.009573289
```

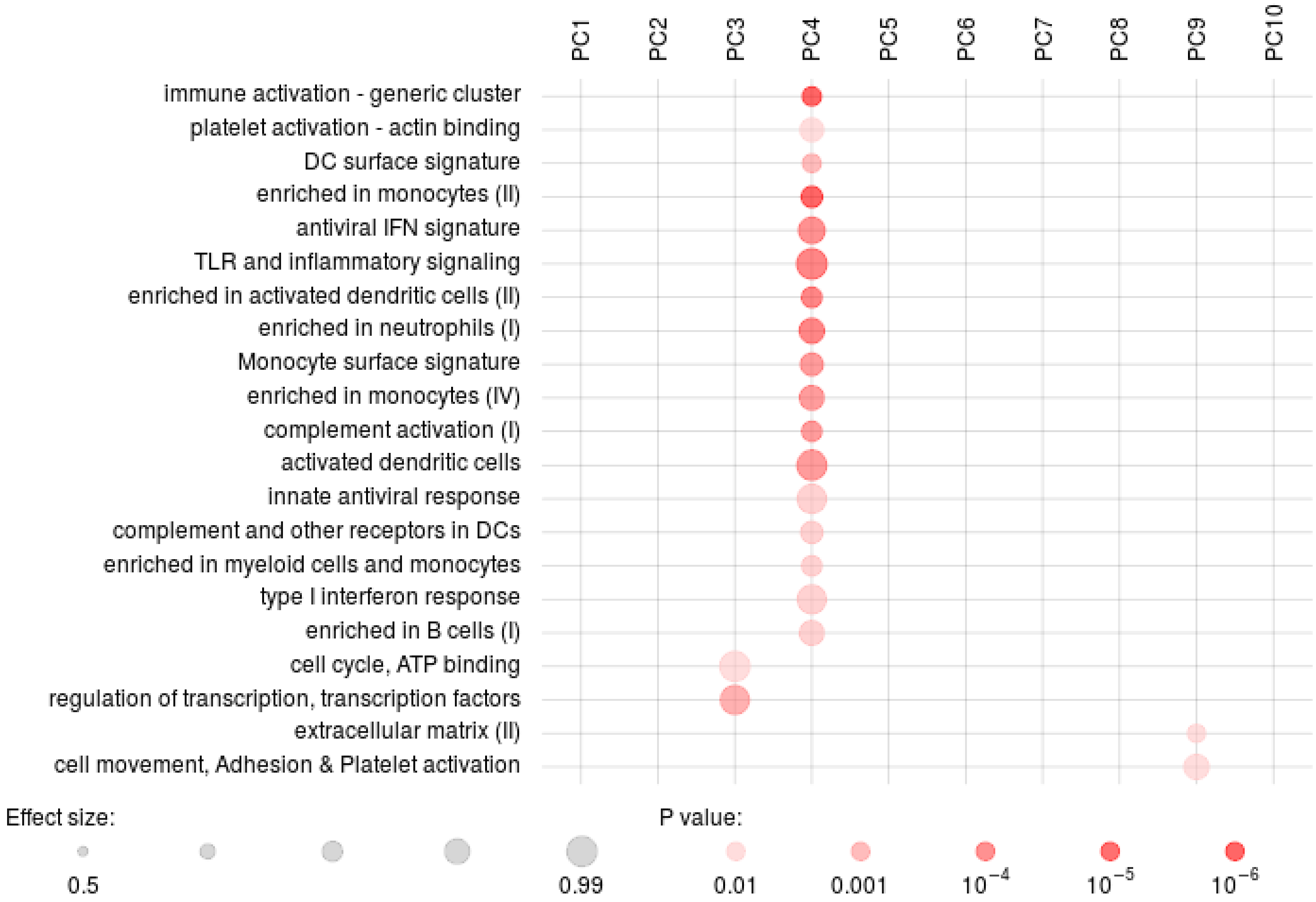
Enrichment for each component

```r
l <- Egambia$GENE_SYMBOL
encfunc <- function(r) {
  o <- order(abs(r), decreasing=TRUE)
  tmodCERNOtest(l[o])
}
res <- apply(pca$rotation[,1:10], 2, encfunc)
head(res[[4]])
```

```
##                ID                                      Title     cerno  N1
## LI.M37.0 LI.M37.0        immune activation - generic cluster 454.88172 100 0.7188
## LI.M11.0 LI.M11.0                 enriched in monocytes (II) 118.06755  20 0.7734
## LI.M165   LI.M165 enriched in activated dendritic cells (II) 101.09999  19 0.7562
## LI.M37.1 LI.M37.1                enriched in neutrophils (I)  77.04015  12 0.8671
## LI.M16     LI.M16             TLR and inflammatory signaling  50.11235   5 0.9923
## LI.M75     LI.M75                    antiviral IFN signature  67.61164  10 0.9007
```

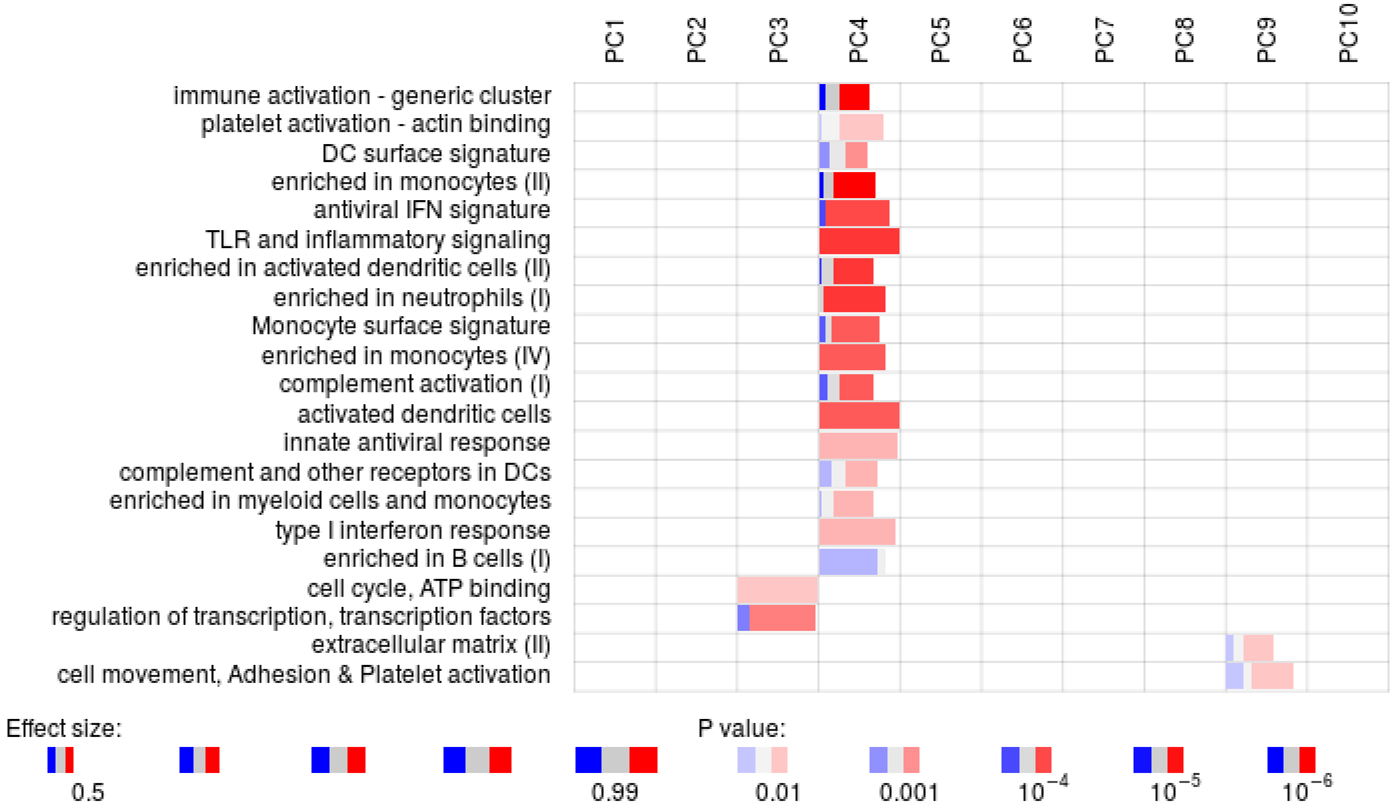
Visualization

```
tmodPanelPlot(res, filter.empty.rows=TRUE)
```



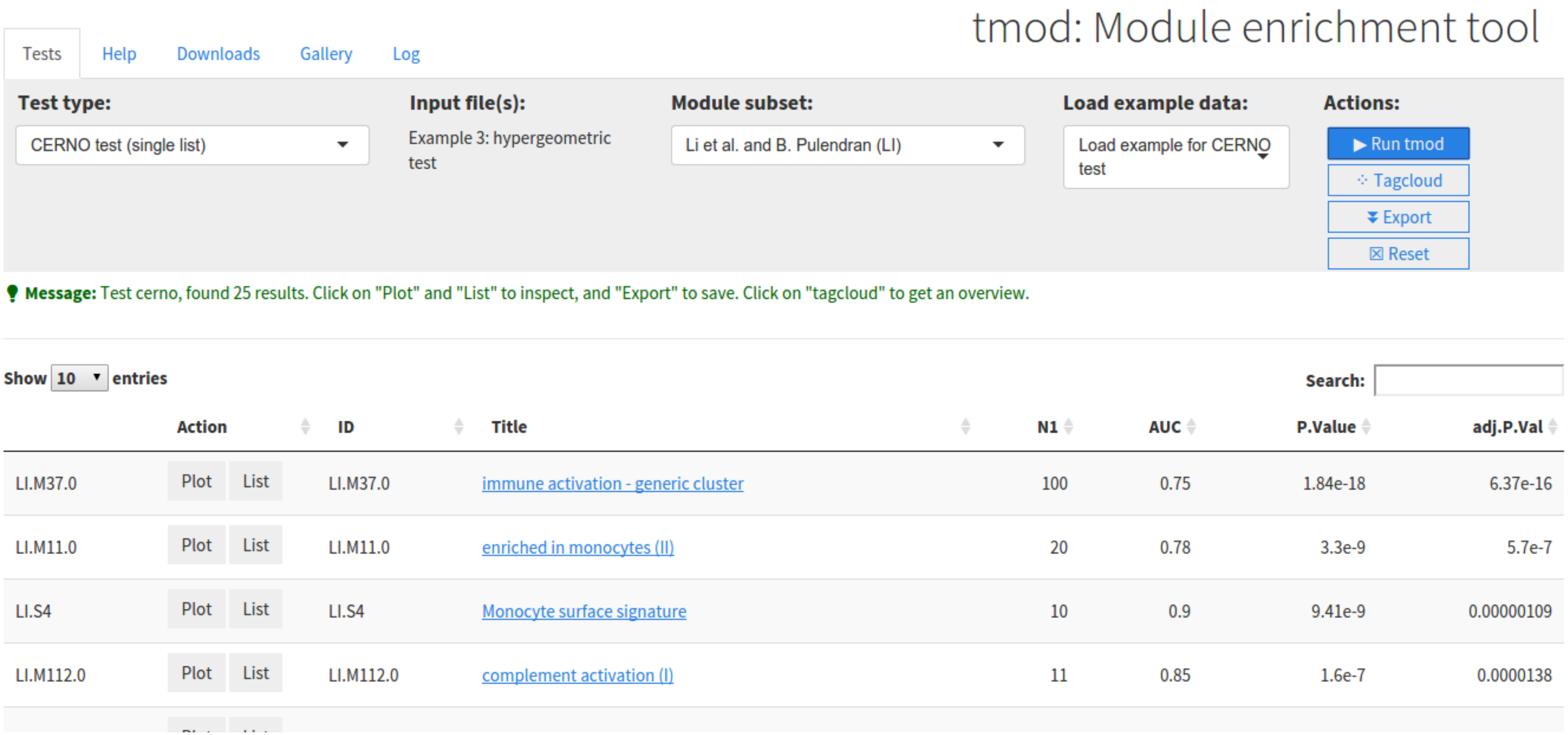
Genes with positive / negative weights?

```r
qfnc <- function(r) quantile(r, 0.75)
qqs <- apply(pca$rotation[,1:10], 2, qfnc)
pie <- tmodDecideTests(l, lfc=pca$rotation[,1:10], lfc.thr=qqs)
tmodPanelPlot(res, pie=pie, pie.style="rug", grid="between")
```


PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
immune activation - generic cluster
platelet activation - actin binding
DC surface signature
enriched in monocytes (II)
antiviral IFN signature
TLR and inflammatory signaling
enriched in activated dendritic cells (II)
enriched in neutrophils (I)
Monocyte surface signature
enriched in monocytes (IV)
complement activation (I)
activated dendritic cells
innate antiviral response
complement and other receptors in DCs
enriched in myeloid cells and monocytes
type I interferon response
enriched in B cells (I)
cell cycle, ATP binding
regulation of transcription, transcription factors
extracellular matrix (II)
cell movement, Adhesion & Platelet activation
Effect size:
0.5
0.99
P value:
0.01
0.001
10^{-4}
10^{-5}
10^{-6}

tmod Web Interface

http://bioinfo.mpiib-berlin.mpg.de/tmod/.



tmod: Module enrichment tool

Test type: CERNO test (single list)

Input file(s): Example 3: hypergeometric test

Module subset: Li et al. and B. Pulendran (LI)

Load example data: Load example for CERNO test

Actions:

Message: Test cerno, found 25 results. Click on "Plot" and "List" to inspect, and "Export" to save. Click on "tagcloud" to get an overview.

Show 10 **entries**

Search:

	Action	ID	Title	N1	AUC	P.Value	adj.P.Val
LI.M37.0	Plot List	LI.M37.0	immune activation - generic cluster	100	0.75	1.84e-18	6.37e-16
LI.M11.0	Plot List	LI.M11.0	enriched in monocytes (II)	20	0.78	3.3e-9	5.7e-7
LI.S4	Plot List	LI.S4	Monocyte surface signature	10	0.9	9.41e-9	0.00000109
LI.M112.0	Plot List	LI.M112.0	complement activation (I)	11	0.85	1.6e-7	0.0000138

Concluding remarks

- Gene set enrichment analysis is a versatile tool for functional annotation
- Functional multivariate analysis can replace differential expression analysis
- *tmod*: R package for BTM and GS enrichment analysis, available from http://bioinfo.mpiib-berlin.mpg.de/tmod/ and CRAN
- *tmod* allows functional multivariate analysis and serial enrichment analyiss
- features several visualization tools
- you know where to find me: january@mpiib-berlin.mpg.de

Conributors

- Teresa Domaszewska
- Emilio Siena

Appendix

You can download the source code of this presentation on the tmod web page, http://bioinfo.mpiib-berlin.mpg.de/tmod/.

To recreate this presentation, download the full presentation package and unzip it. Install the required packages (knitr for R and pandoc). Run the following command from inside the package archive.

Commands:

```
Rscript -e 'knitr::knit("weiner_bioinfo_2015_06_23.Rmd")'
pandoc -s -S -t revealjs weiner_bioinfo_2015_06_23.md -o weiner_bioinfo_2015_06_23.h
  --mathjax='http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTM
  --css css/mytheme.css \
  --slide-level 2 -V theme=blood
```

(Note: for an offline version, download MathJax and modify the –mathjax option)

To extract the code from this presentation, save it as “test.Rmd” and run

```
Rscript -e 'knitr::purl("tmod.Rmd")'
```

Printing:

This only works properly in Google Chromium; see reveal.js documentation

To print, follow the link below and press Ctrl-P; don’t worry if the slides appear to overlap – they will look fine on the print preview.

Print