```
library(PLIER)

## Loading required package:  RColorBrewer
## Loading required package:  gplots
##
## Attaching package:  'gplots'
## The following object is masked from 'package:stats':
##
##      lowess
## Loading required package:  pheatmap
## Loading required package:  glmnet
## Loading required package:  Matrix
## Loading required package:  foreach
## Loaded glmnet 2.0-10
## Loading required package:  rsvd
```

1 Some Notes

PLIER runs reasonably fast but to get the best performance we recommend that
you use linear algebra libraries optimized for your system. For Ubuntu and Win-
dows you can follow instructions at http://brettklamer.com/diversions/statistical/faster-
blas-in-r/.

2 Vaccination Dataset

Load data

```
data(bloodCellMarkersIRISDMAP)
data(svmMarkers)
data(canonicalPathways)
data(vacData)
```

Construct a joint pathway matrix by merging canonicalPathways, bloodCell-
MarkersIRISDMAP and svmMarkers and select genes appearing in both gene
expression profile and the joint pathway matrix.

```
allPaths=combinePaths(bloodCellMarkersIRISDMAP, svmMarkers,canonicalPathways)
cm.genes=commonRows(allPaths, vacData)
```

Normalize the data and approximate the number of latent variables in the
data by num.pc(). The result is 26. Note: num.pc() is quite conservative and
we recommend to set this value about 30 to 50% higher. We set k=34 and all
other parameters to be default.

```
vacDataN=rowNorm(vacData)
num.pc(vacDataN)

## [1] 26

#plierResult=PLIER(vacDataN[cm.genes,], allPaths[cm.genes,],k=34, trace=T)
```
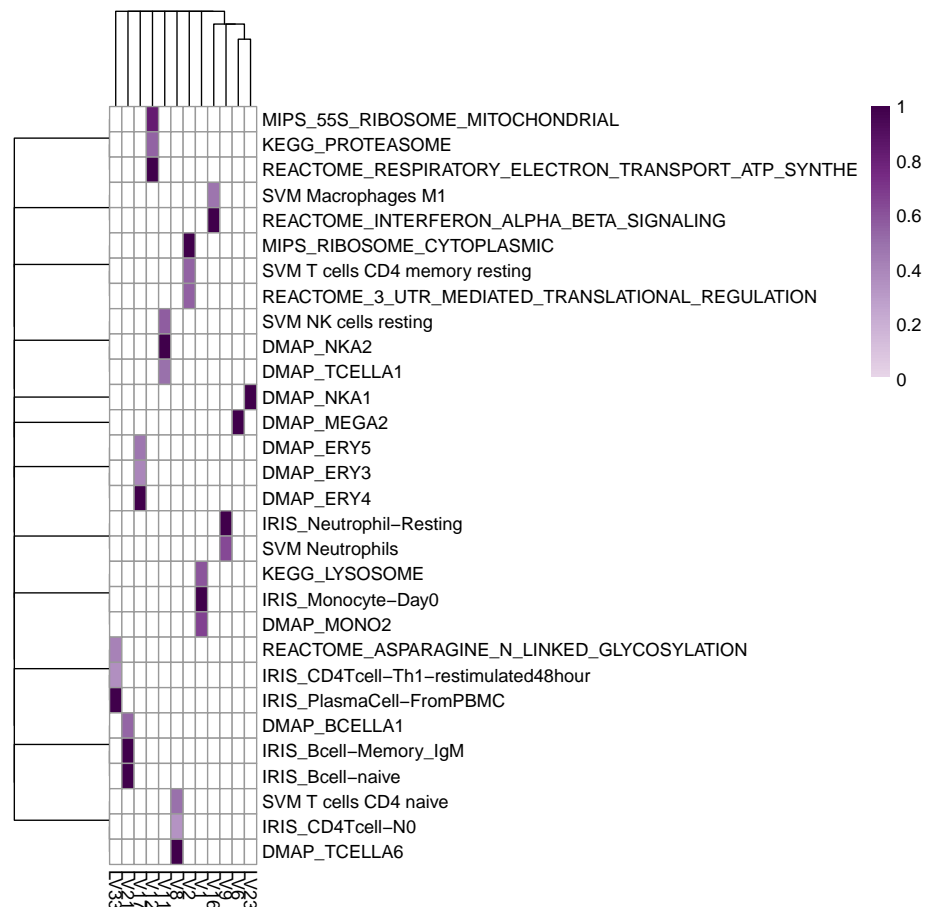
In the interest of speed we provide a pre-computed version though for this example it will only take a around a minute.

```
data(plierResult)
```
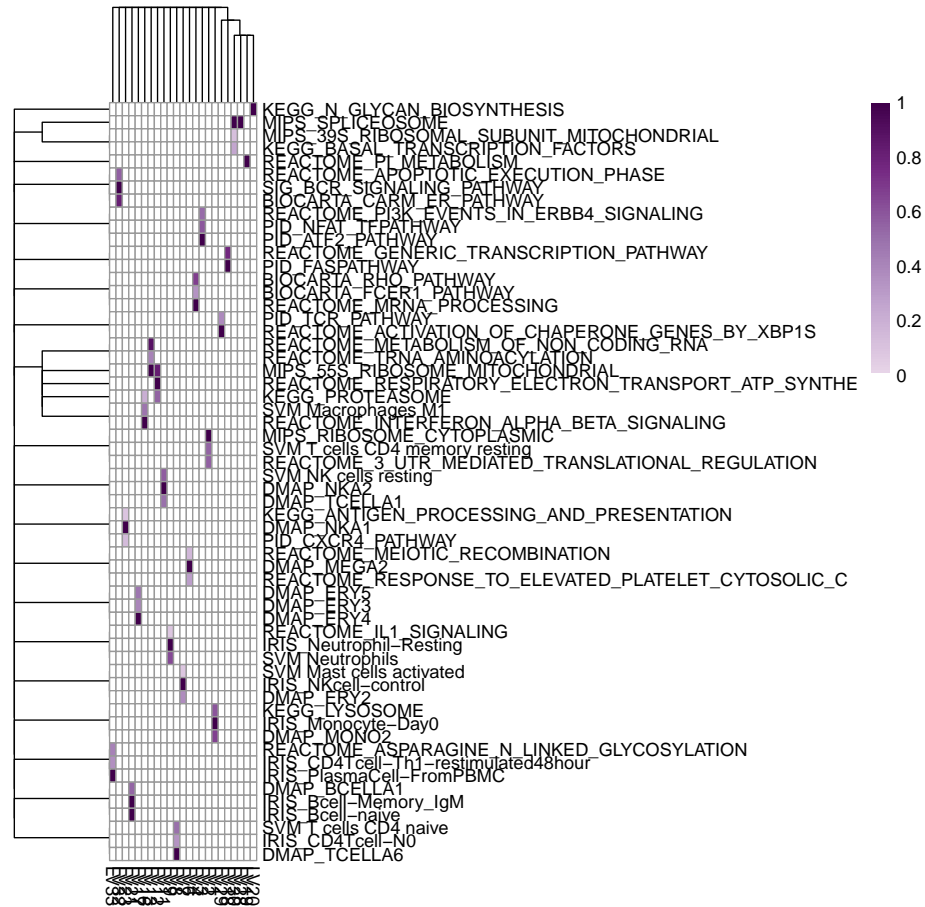
Visualize the U matrix with default cutoffs

```
plotU(plierResult, auc.cutoff = 0.75, pval.cutoff = 0.01, top = 3)
```
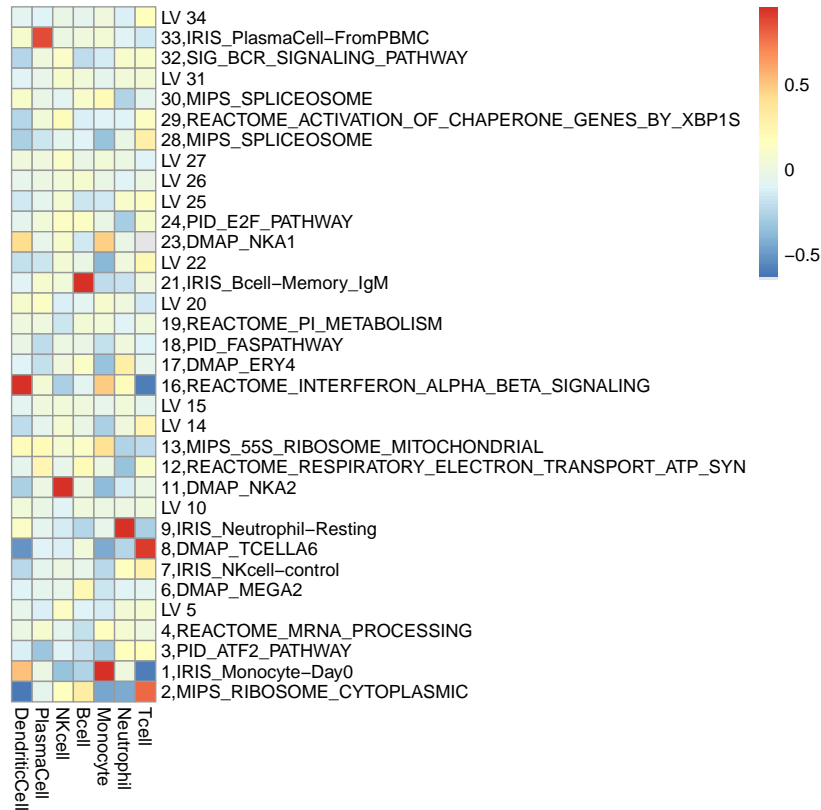

Visualize the U matrix with more permissive cutoffs

```
plotU(plierResult, auc.cutoff = 0.6, pval.cutoff = 0.05, top = 3)
```


We can correlate the decomposition result with SPVs from CellCODE like this.

```
data(SPVs)
plotMat(cor(t(plierResult$B), SPVs), scale = F)
```

LV 34
33,IRIS_PlasmaCell–FromPBMC
32,SIG_BCR_SIGNALING_PATHWAY
LV 31
30,MIPS_SPLICEOSOME
29,REACTOME_ACTIVATION_OF_CHAPERONE_GENES_BY_XBP1S
28,MIPS_SPLICEOSOME
LV 27
LV 26
LV 25
24,PID_E2F_PATHWAY
23,DMAP_NKA1
LV 22
21,IRIS_Bcell–Memory_IgM
LV 20
19,REACTOME_PI_METABOLISM
18,PID_FASPATHWAY
17,DMAP_ERY4
16,REACTOME_INTERFERON_ALPHA_BETA_SIGNALING
LV 15
LV 14
13,MIPS_55S_RIBOSOME_MITOCHONDRIAL
12,REACTOME_RESPIRATORY_ELECTRON_TRANSPORT_ATP_SYN
11,DMAP_NKA2
LV 10
9,IRIS_Neutrophil–Resting
8,DMAP_TCELLA6
7,IRIS_NKcell–control
6,DMAP_MEGA2
LV 5
4,REACTOME_MRNA_PROCESSING
3,PID_ATF2_PATHWAY
1,IRIS_Monocyte–Day0
2,MIPS_RIBOSOME_CYTOPLASMIC

DendriticCell
PlasmaCell
NKcell
Bcell
Monocyte
Neutrophil
Tcell

0.5
0
−0.5

We have nice one-to-one correspondence, though the "DendriticCell"ignature from CellCODE is more closely related to the Type-I interferon transcriptional response so it is probably not cell-type induced variation.

We can see that LVs 1, 8, 9, 11, 16, 21, 33 roughly correspond to major cell-types and we can use this to derive data-driven cell-type markers, which can be plugged into the CellCODE pipeline.

```
markers=plierResToMarkers(plierResult, allPaths, index = c( 1, 8, 9, 11, 16, 21, 33))
```

```
## Genes not matched uniquely:  ZAP70, CD2
```

```
colnames(markers)
```

```
## [1] "1,IRIS_Monocyte-Day0"
## [2] "8,DMAP_TCELLA6"
## [3] "9,IRIS_Neutrophil-Resting"
## [4] "11,DMAP_NKA2"
## [5] "16,REACTOME_INTERFERON_ALPHA_BETA_SIGNALING"
```

```
## [6] "21,IRIS_Bcell-Memory_IgM"
## [7] "33,IRIS_PlasmaCell-FromPBMC"
```
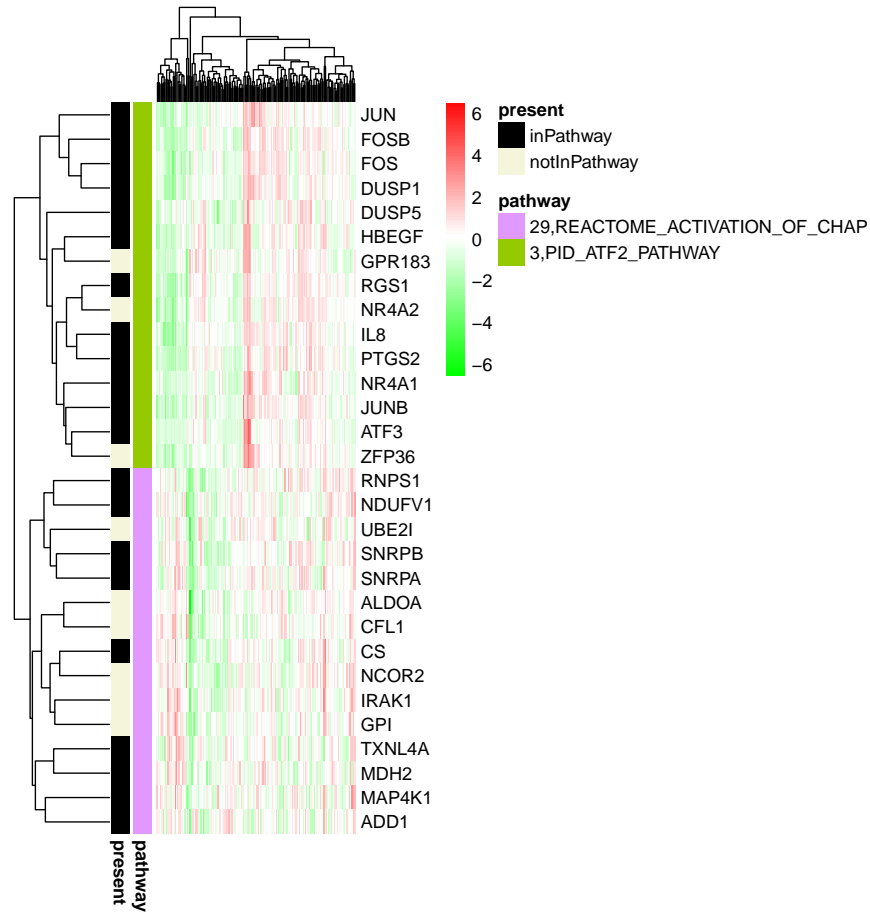
We can also visualize the top genes and their pathway associations. Note: only LVs with pathway association are plotted.

```
indexToPlot=which(apply(plierResult$Uauc,2,max)>0.75)
plotTopZ(plierResult, vacDataN, allPaths, top = 5, index =  indexToPlot)
```
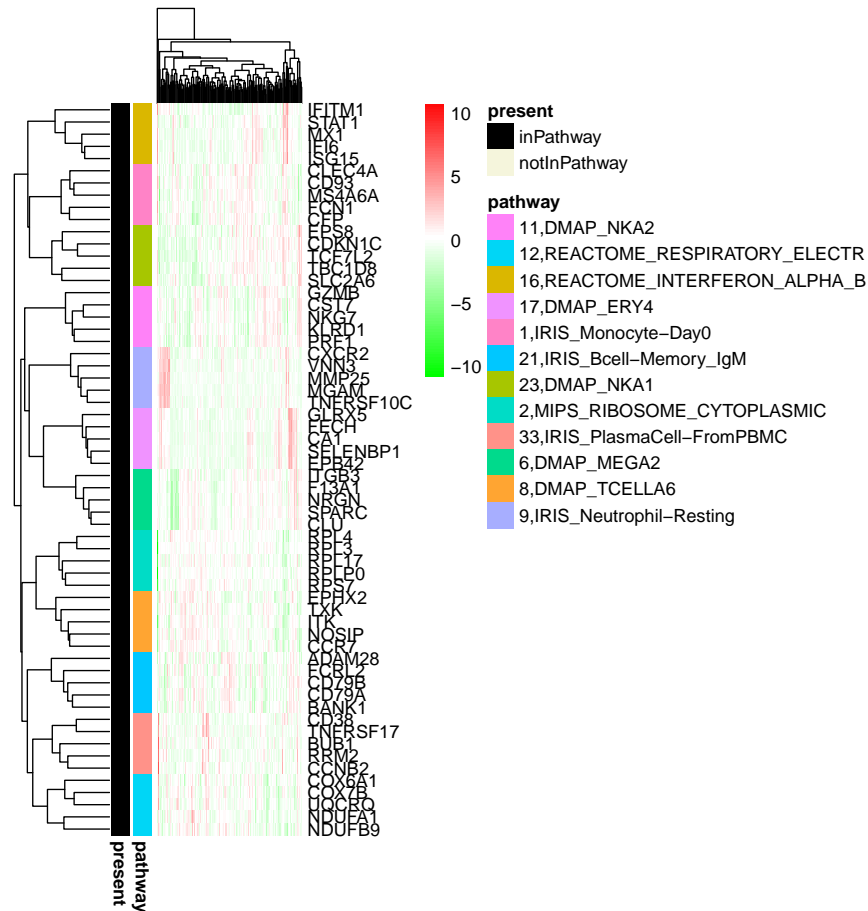

Let's plot some of the less reliable LVs

```
plotTopZ(plierResult, vacDataN, allPaths, top = 15, index =  c(3,29))
```


We can see that in this case not all genes have the pathway(s) annotation like this.

Since each LV may (and often is) be associated with more than one pathway it can be useful to visualize all the different annotations.

```
plotTopZallPath(plierResult, vacDataN, allPaths, top = 20, index =  c(3,29), regress = T)
```