

```
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: rsud
```

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 Windows you can follow instructions at <http://brettklamer.com/diversions/statistical/faster-blas-in-r/>.

2 Vaccination Dataset

Load data

```
data(bloodCellMarkersIRISDMP)
data(svmMarkers)
data(canonicalPathways)
data(vacData)
```

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

```
allPaths=combinePaths(bloodCellMarkersIRISDMP, 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. We set the seed to get reproducible results for held-out pathway genes.

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

## [1] 26

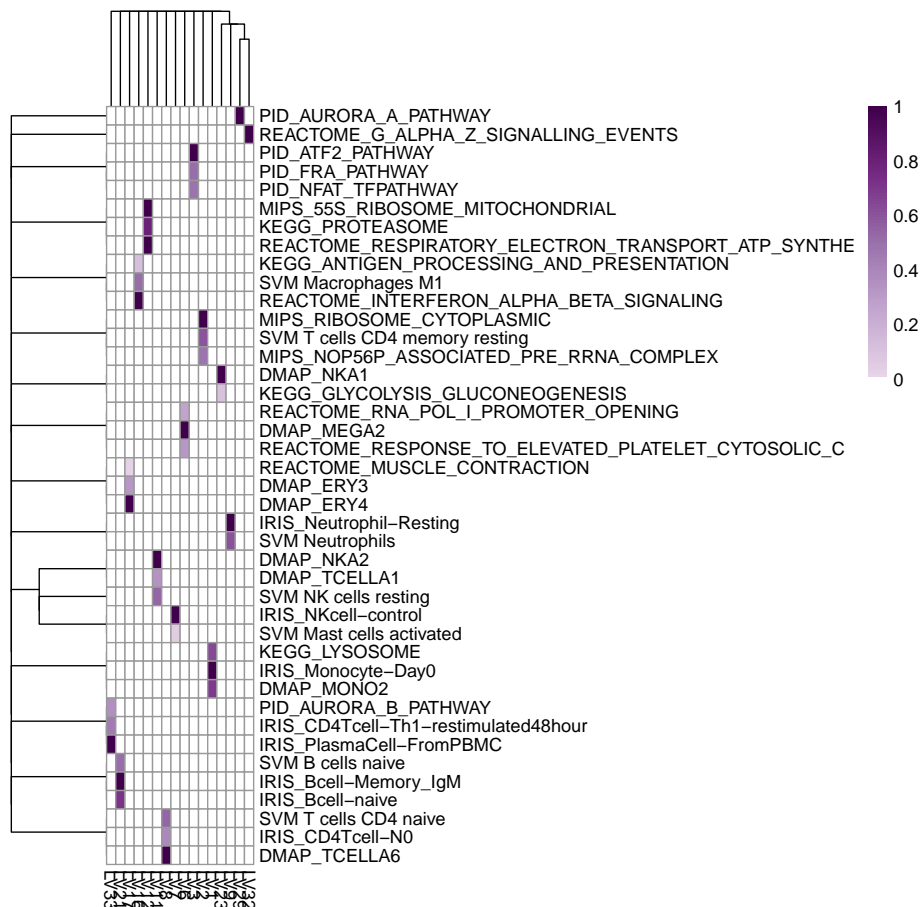
#plierResult=PLIER(vacDataN[cm.genes,], allPaths[cm.genes,],k=34, trace=T, seed = 123)
```

In the interest of speed we provide a pre-computed version though for this example it will only take less than a minute.

```
data(plierResult)
```

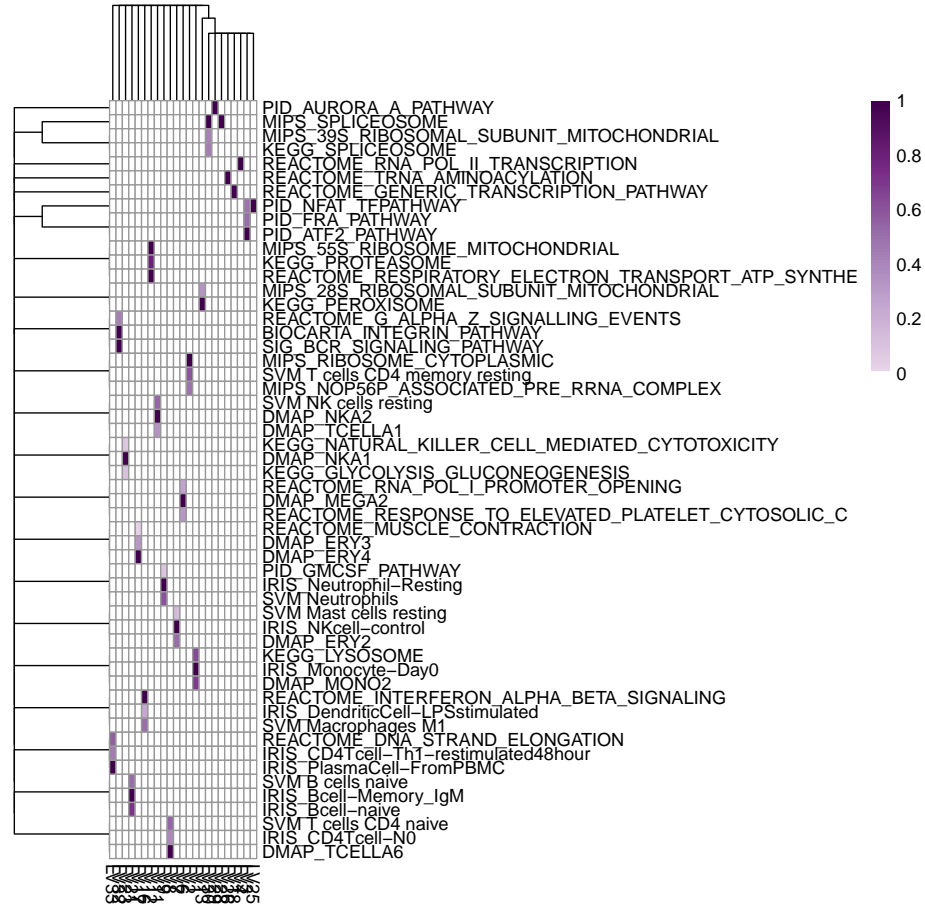
Visualize the U matrix with default cutoffs

```
plotU(plierResult, auc.cutoff = 0.70, fdr.cutoff = 0.05, top = 3)
```



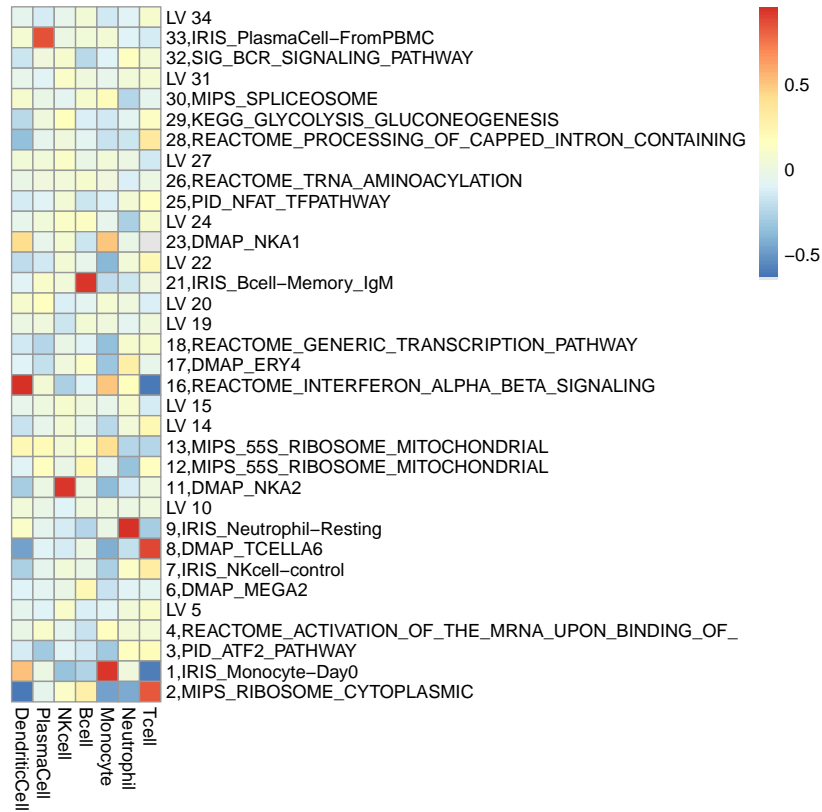
Visualize the U matrix with more permissive cutoffs

```
plotU(plierResult, auc.cutoff = 0.6, fdr.cutoff = 0.2, top = 3)
```



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

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



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: POU2AF1

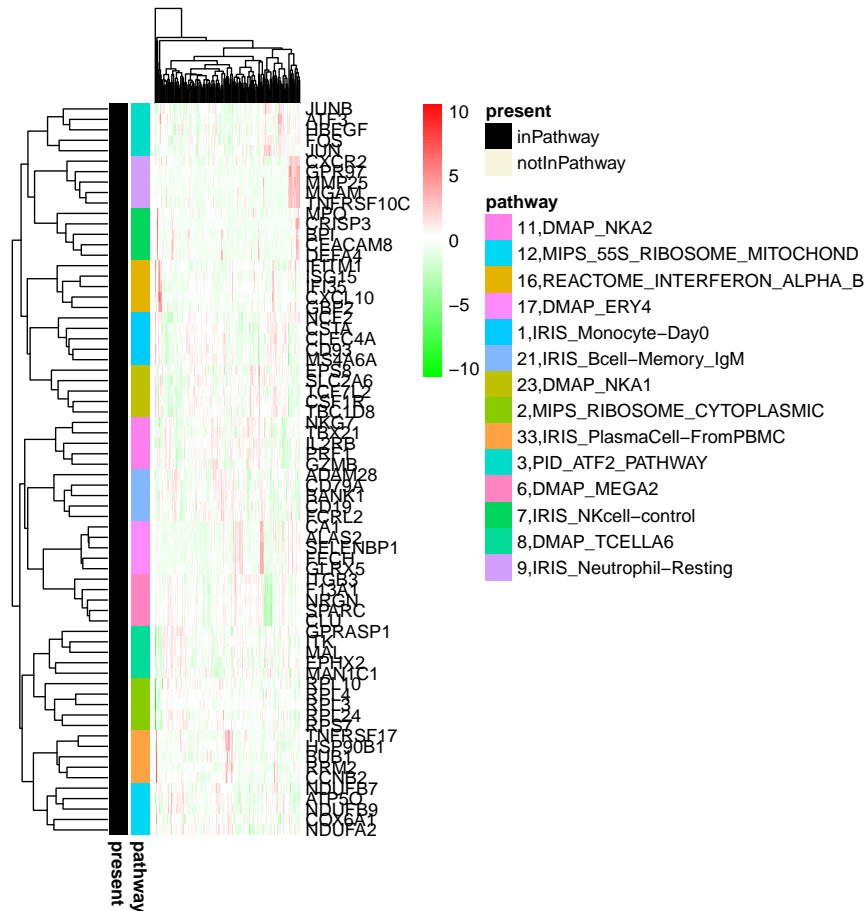
colnames(markers)

## [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"
```

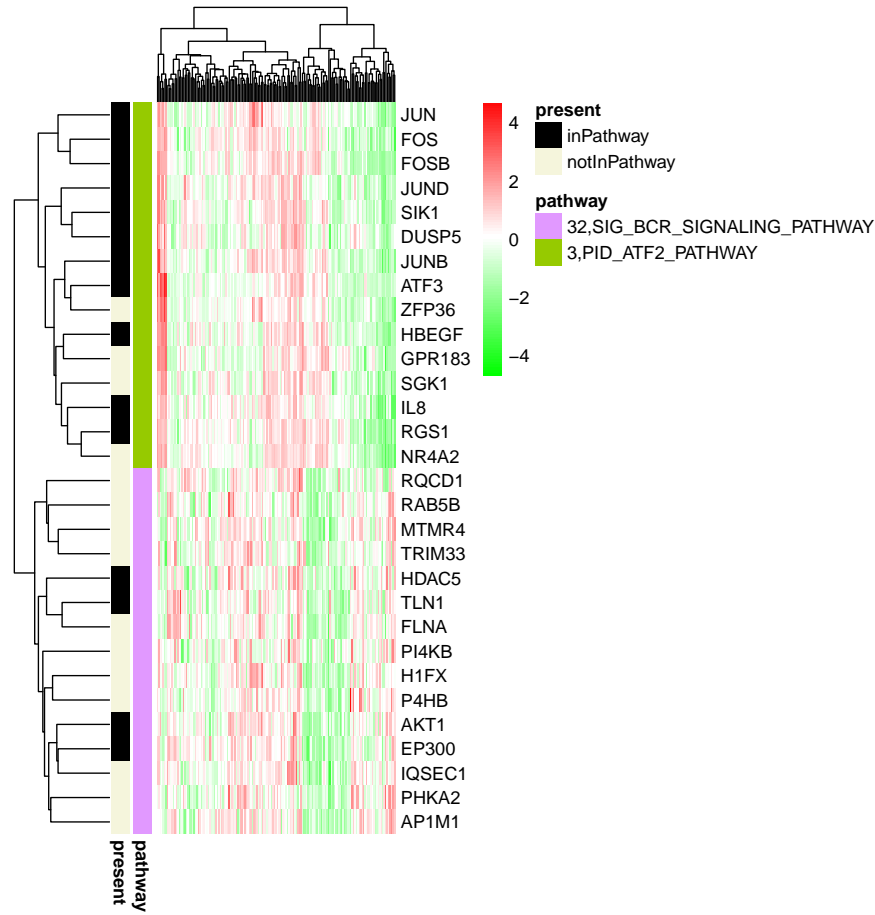
We can also visualize the top genes and their pathway associations. Note: only LVs with pathway association are plotted.

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



Let's plot some of the less reliable LVs

```
plotTopZ(plierResult, vacDataN, allPaths, top = 15, index = c(3,32))
```



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

Since each LV may (and often is) be associated with more than one pathway it can be useful to visualize all the different annotations. We can also plot more genes since for some small or weakly predicted pathways they may be quite far from the top.

```
plotTopZallPath(plierResult, vacDataN, allPaths, top = 50, index = c(32))
```

