```
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
## Loading required package:  qvalue
```

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 Human whole blood

Load the data

```
data(bloodCellMarkersIRISDMAP)
data(canonicalPathways)
data(dataWholeBlood)
```

Combine markers and canonical pathways into one pathway matrix

```
allPaths=combinePaths(bloodCellMarkersIRISDMAP, canonicalPathways)
```

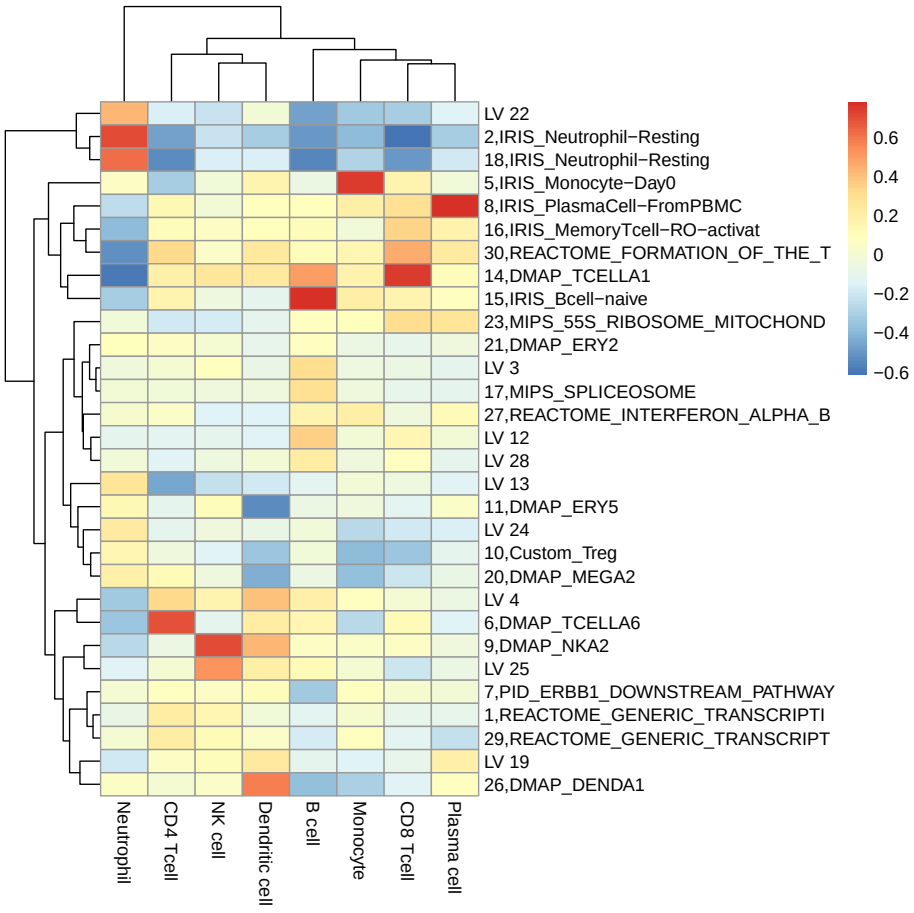
Run PLIER with all default parameters.
Note: we can also set doCrossval=F to turn off gene hold-out cross-validation. If all we care about is correlation with measured proportions it is best to use all the of the available pathway information. However, with that setting the AUCs and p-values for LV-pathway association will be anti-conservative.

```
plierResult=PLIER(dataWholeBlood, allPaths)

## Selecting common genes:  5892
## Removing 3 pathways with too few genes
## Computing SVD
## Done
## k is set to 30

## [1] 36.86117
## [1] "L2 is set to 36.8611678671894"
## [1] "L1 is set to 18.4305839335947"
## [1] 30 36

## errorY (SVD based:best possible) = 0.02237
## Updating L3, current fraction= 0, target=0.7
## 0 positive columns at L3=0.5
## 0 positive columns at L3=0.250001
## 0 positive columns at L3=0.125001
## 0 positive columns at L3=0.062501
## 2 positive columns at L3=0.031251
## 7 positive columns at L3=0.015626
## 17 positive columns at L3=0.007813
## 28 positive columns at L3=0.003907
## 25 positive columns at L3=0.00586
## 21 positive columns at L3=0.006837
## Updating L3, current fraction= 0.5667, target=0.7
## 26 positive columns at L3=0.003453
## 21 positive columns at L3=0.005145
## Updating L3, current fraction= 0.6, target=0.7
## 28 positive columns at L3=0.002598
## 24 positive columns at L3=0.003871
## 23 positive columns at L3=0.004508
## 20 positive columns at L3=0.004826
## Updating L3, current fraction= 0.6333, target=0.7
## 28 positive columns at L3=0.002437
## 24 positive columns at L3=0.003632
## 24 positive columns at L3=0.004229
## 21 positive columns at L3=0.004528
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
```

```
## Updating L3, current fraction= 0.7, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.7, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.6667, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.7, target=0.7
## L3 not changed
## Updating L3, current fraction= 0.7, target=0.7
## L3 not changed
## Bdiff is not decreasing
## Bdiff is not decreasing
## Updating L3, current fraction= 0.6333, target=0.7
## 29 positive columns at L3=0.002287
## 24 positive columns at L3=0.003407
## 23 positive columns at L3=0.003968
## 21 positive columns at L3=0.004248
## Bdiff is not decreasing
## Bdiff is not decreasing
## Bdiff is not decreasing
## Bdiff is not decreasing
## converged at  iteration 303 Bdiff is not decreasing
## There are 15  LVs with AUC>0.70
```

We can compare the results to known proportions. (Note: one of the samples does not have proportion measurements)

```
data(majorCellTypes)
corOut=cor(t(plierResult$B[, rownames(majorCellTypes)]), majorCellTypes, method="s")
rownames(corOut)=strtrim(rownames(corOut),30)
pheatmap(corOut)
```




```
apply(corOut,2,max)
```

```
##      CD8 Tcell      CD4 Tcell     Neutrophil         B cell       Monocyte
##      0.7481793      0.6938375      0.7005602      0.7759104      0.7417367
##        NK cell Dendritic cell    Plasma cell
##      0.7095238      0.5756303      0.7819203
```

3 Vaccination Dataset

Load data. Here we also include the Cibersort LM22 signature as markers (svmMarkers).

```
data(bloodCellMarkersIRISDMAP)
data(svmMarkers)
```

```
data(canonicalPathways)
data(vacData)
```

Construct a joint pathway matrix by merging canonicalPathways, bloodCellMarkersIRISDMAP 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 27. Note: num.pc() is quite conservative and we recommend to set this value about 1.5 to 2 times higher. We set k=40 and all other parameters to be default.

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

## Computing svd

## [1] 27
```

In this example we we also precompute some quantities in order to make re-running PLIER with different parameters (such as k) faster. We will need the SVD decomposition on the same set of genes (for large datasets use rsvd for speed).

```
vacData.svd=svd(vacDataN[cm.genes,])
```

Also, the pathway matrix pseudo-inverse. This is used to pre-select pathways for optimization.

```
Chat=computeChat(allPaths[cm.genes,])
```

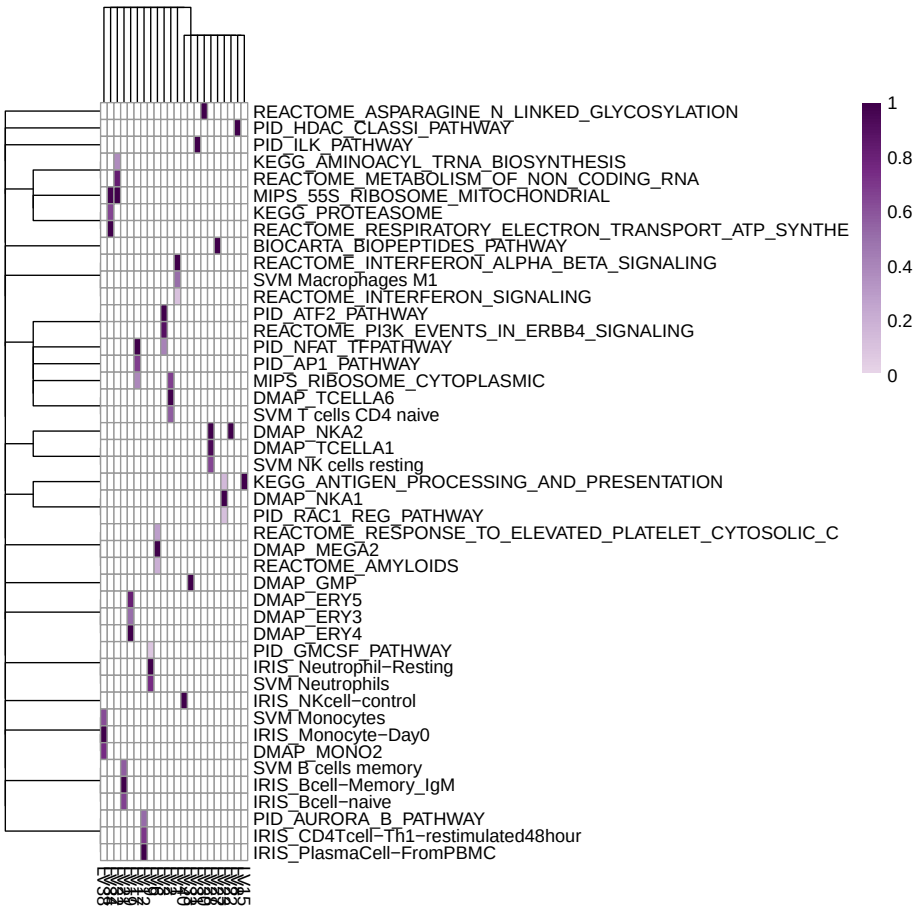
```
#plierResult=PLIER(vacDataN[cm.genes,], allPaths[cm.genes,],k=40, svdres=vacData.svd, Chat=
```

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

```
data(plierResult)
```

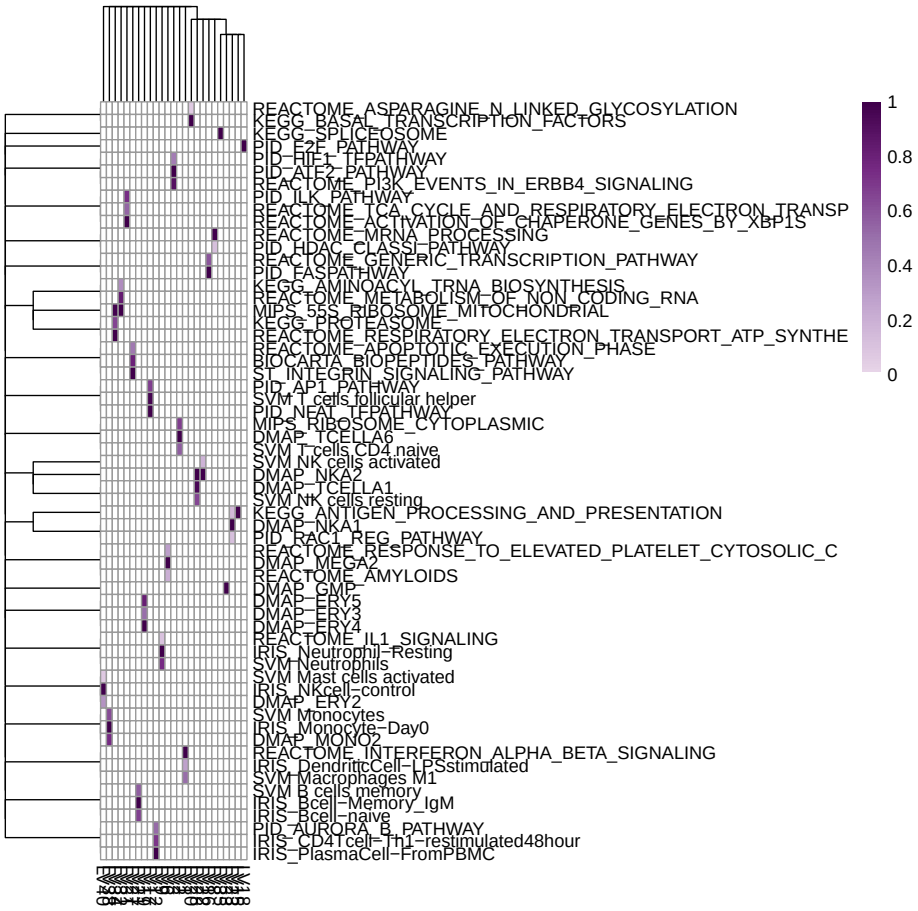
Visualize the U matrix with default cutoffs

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



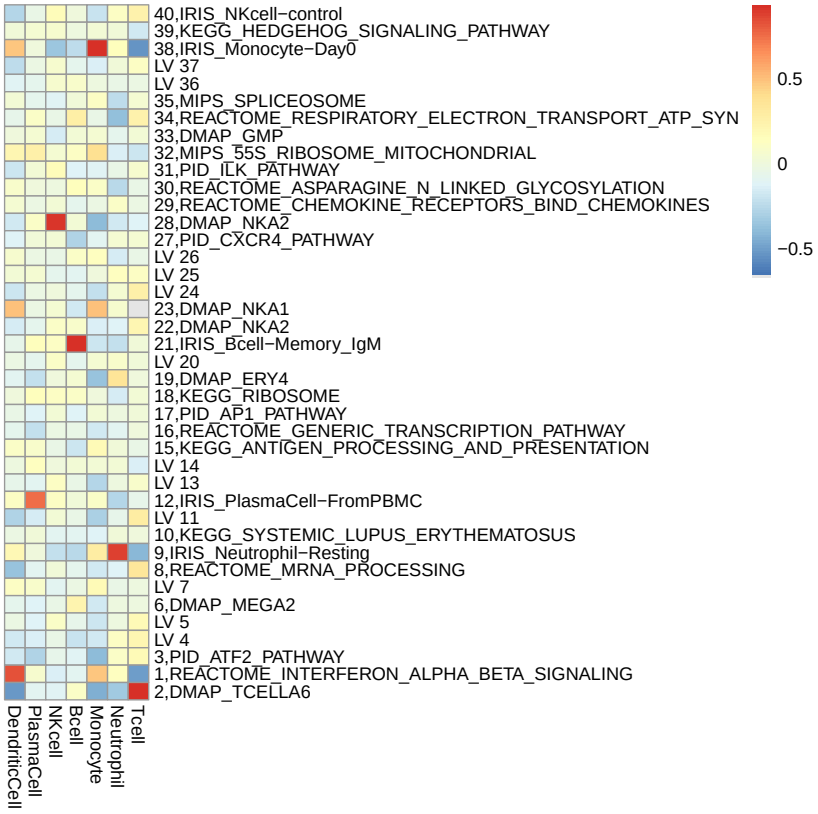
Visualize the U matrix with more permissive cutoffs

```
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(corout<-cor(t(plierResult$B), SPVs, method="s"), scale = F)
```




```
paste(sort(apply(corout,2, which.max)), collapse=", ")

## [1] "1, 2, 9, 12, 21, 28, 38"
```

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

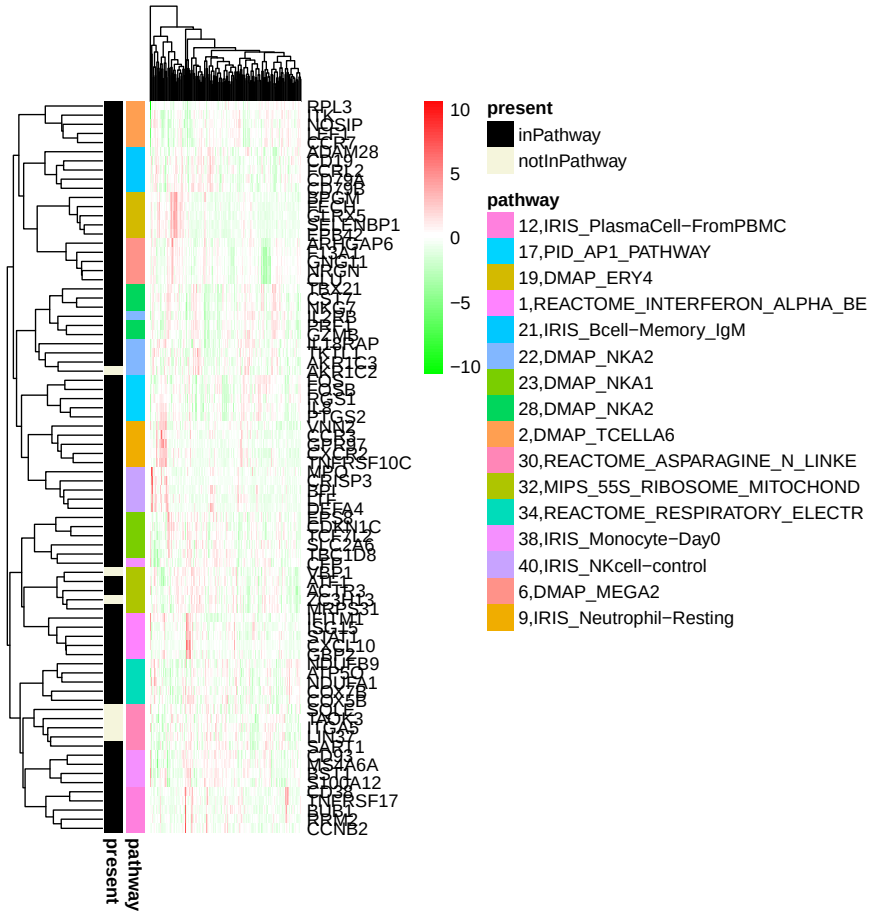
We can see that LVs 1, 2, 9, 12, 21, 28, 38 roughly correspond to major cell-types and we can use this to derive data-driven cell-type markers, which can be plugged into the CellCODE pipeline.

```
markers=plierResToMarkers(plierResult, allPaths,
                          index = c( 1, 2, 9, 12, 21, 28, 38))
colnames(markers)

## [1] "1,REACTOME_INTERFERON_ALPHA_BETA_SIGNALING"
```

```
## [2] "2,DMAP_TCELLA6"
## [3] "9,IRIS_Neutrophil-Resting"
## [4] "12,IRIS_PlasmaCell-FromPBMC"
## [5] "21,IRIS_Bcell-Memory_IgM"
## [6] "28,DMAP_NKA2"
## [7] "38,IRIS_Monocyte-Day0"
```

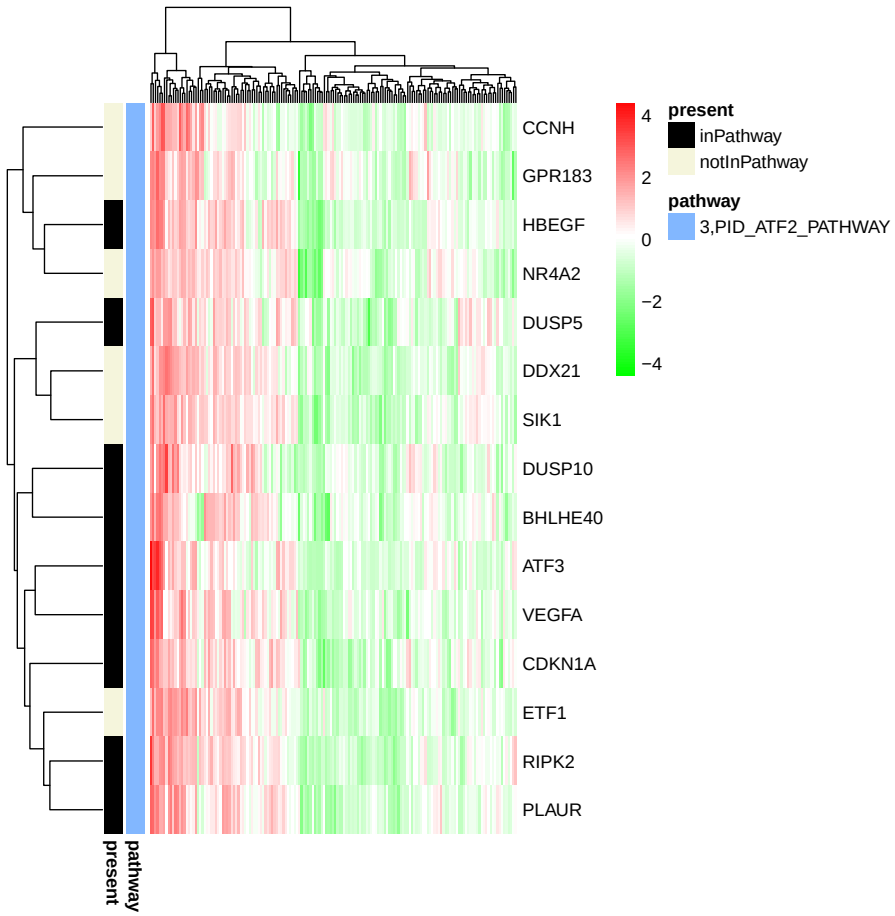
We can also visualize the top genes and their pathway associations. Note: only LVs with pathway association are plotted.

```
indexToPlot=which(apply(plierResult$Uauc*(plierResult$Up<0.001),2,max)>0.75)
plotTopZ(plierResult, vacDataN, allPaths, top = 5, index =  indexToPlot)
```



Let's plot some of the less reliable LVs

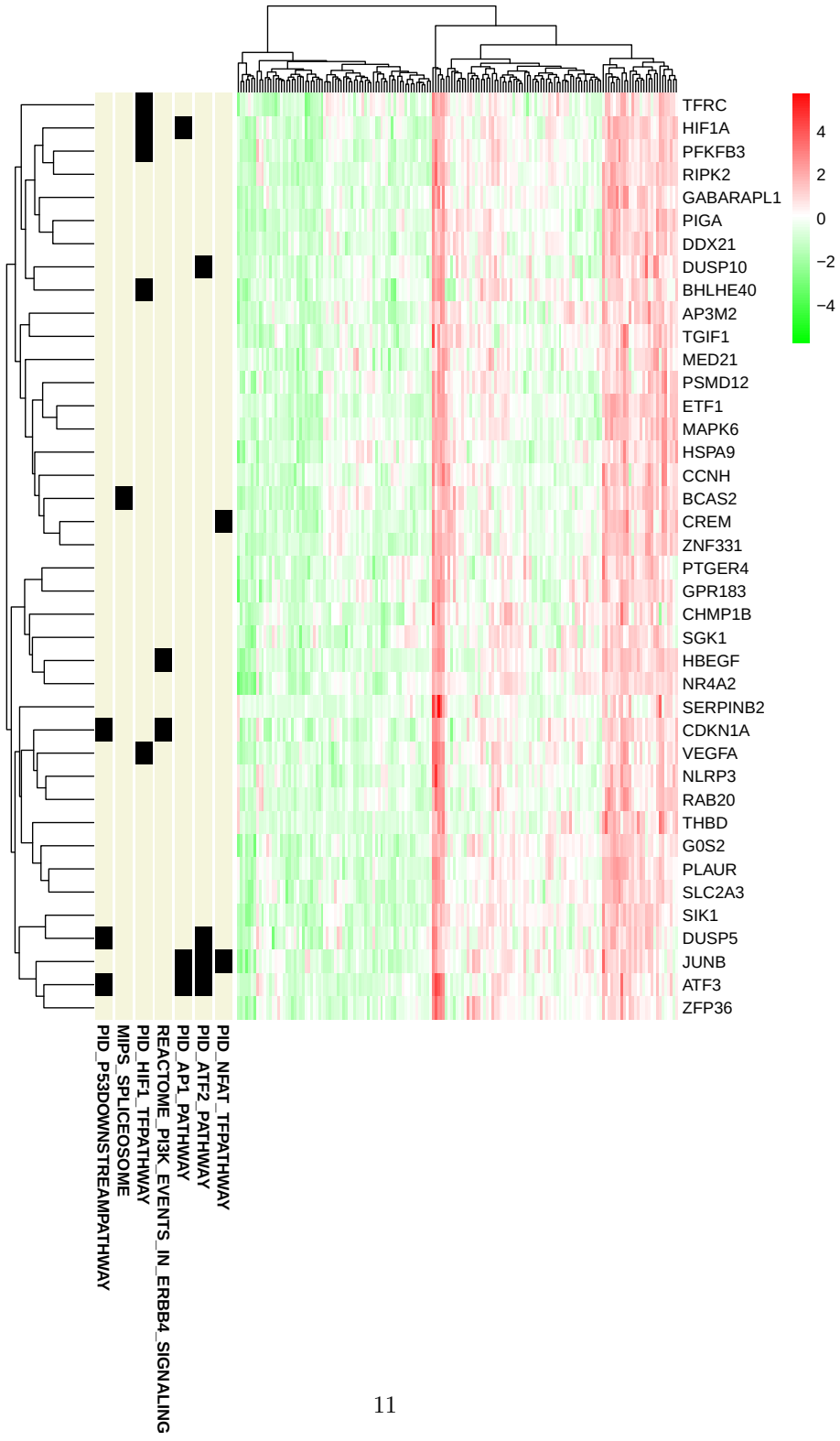
```
plotTopZ(plierResult, vacDataN, allPaths, top = 15, index = c(3))
```



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

Since each LV may 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 the annotated genes may be quite far from the top.

```
plotTopZallPath(plierResult, vacDataN, allPaths, top = 40, index = c(3))
```


TFRC
HIF1A
PFKFB3
RIPK2
GABARAPL1
PIGA
DDX21
DUSP10
BHLHE40
AP3M2
TGIF1
MED21
PSMD12
ETF1
MAPK6
HSPA9
CCNH
BCAS2
CREM
ZNF331
PTGER4
GPR183
CHMP1B
SGK1
HBEGF
NR4A2
SERPINB2
CDKN1A
VEGFA
NLRP3
RAB20
THBD
G0S2
PLAUR
SLC2A3
SIK1
DUSP5
JUNB
ATF3
ZFP36
4
2
0
−2
−4
PID_NFAT_TFPATHWAY
PID_ATF2_PATHWAY
PID_AP1_PATHWAY
REACTOME_PI3K_EVENTS_IN_ERBB4_SIGNALING
PID_HIF1_TFPATHWAY
MIPS_SPLICEOSOME
PID_P53DOWNSTREAMPATHWAY

4 (

single-cell RNAseq from Usokin et al.)

Load the relevant data

```
data("chemgenPathways")
data("canonicalPathways")
data("human2Mouse")
data("dataUsoskin")
data("cellsUsoskin")
```

Create the combined pathway matrix

```
usoskinPath = combinePaths(canonicalPathways, chemgenPathways)
```

Map the pathway matrix to mouse names

```
usoskinPath = mapPathway(usoskinPath, human2Mouse)

## [1] 16979
```

Select the subset of genes that have prior information and compute the pathway "Chat" (this saves time when the pathway matrix is large and we want to run PLIER multiple times)

```
cm = commonRows(dataUsoskin, usoskinPath)
usoskinChat = computeChat(usoskinPath[cm, ])
```

Compute the SVD and set the rownames of u and v. This makes sure names propogate when we apply data smoothing below.

```
dataUsoskin.svd = svd(rowNorm(dataUsoskin[cm, ]))
rownames(dataUsoskin.svd$u) = cm
rownames(dataUsoskin.svd$v) = colnames(dataUsoskin)
```

Calculate the number of PCs.

```
num.pc(dataUsoskin.svd)

## [1] 46
```

Smooth the data with double that number

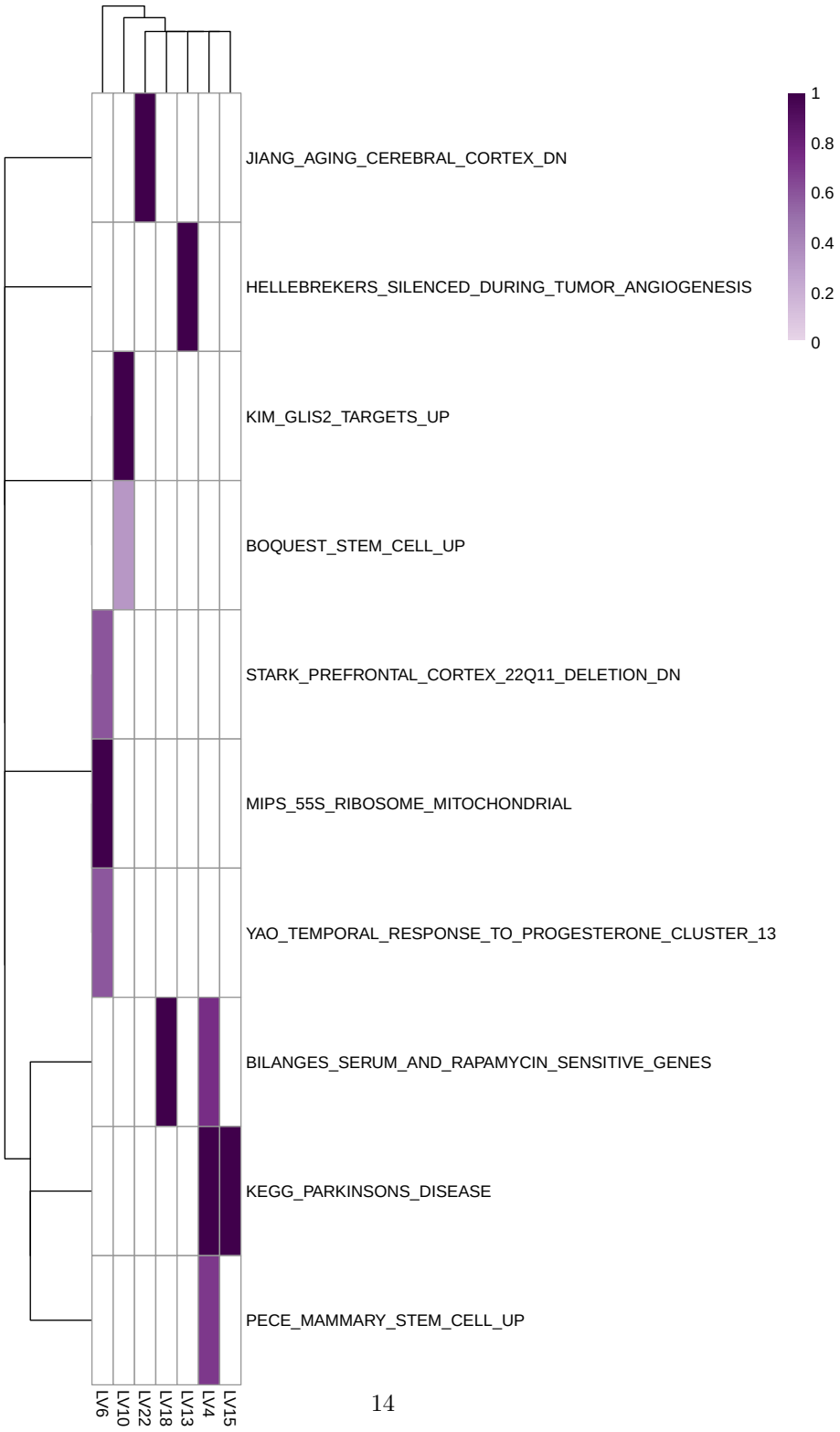
```
dataUsoskin.smooth = DataSmooth(dataUsoskin.svd, 46 * 2)
```

Run PLIER with k=num.pc. In contrast to bulk RNAseq data where increasing k seems to be benefitial and the default suggestion is k=2*num.pc for single cell data it appears that conservative k give better results. Here we just use 46 from above. As this takes some time we have can preload the results

```
# usoskinRes=PLIER(rowNorm(dataUsoskin.smooth), usoskinPath[cm,], k=46,
# svdres = dataUsoskin.svd, Chat=usoskinChat)
data(usoskinRes)
```

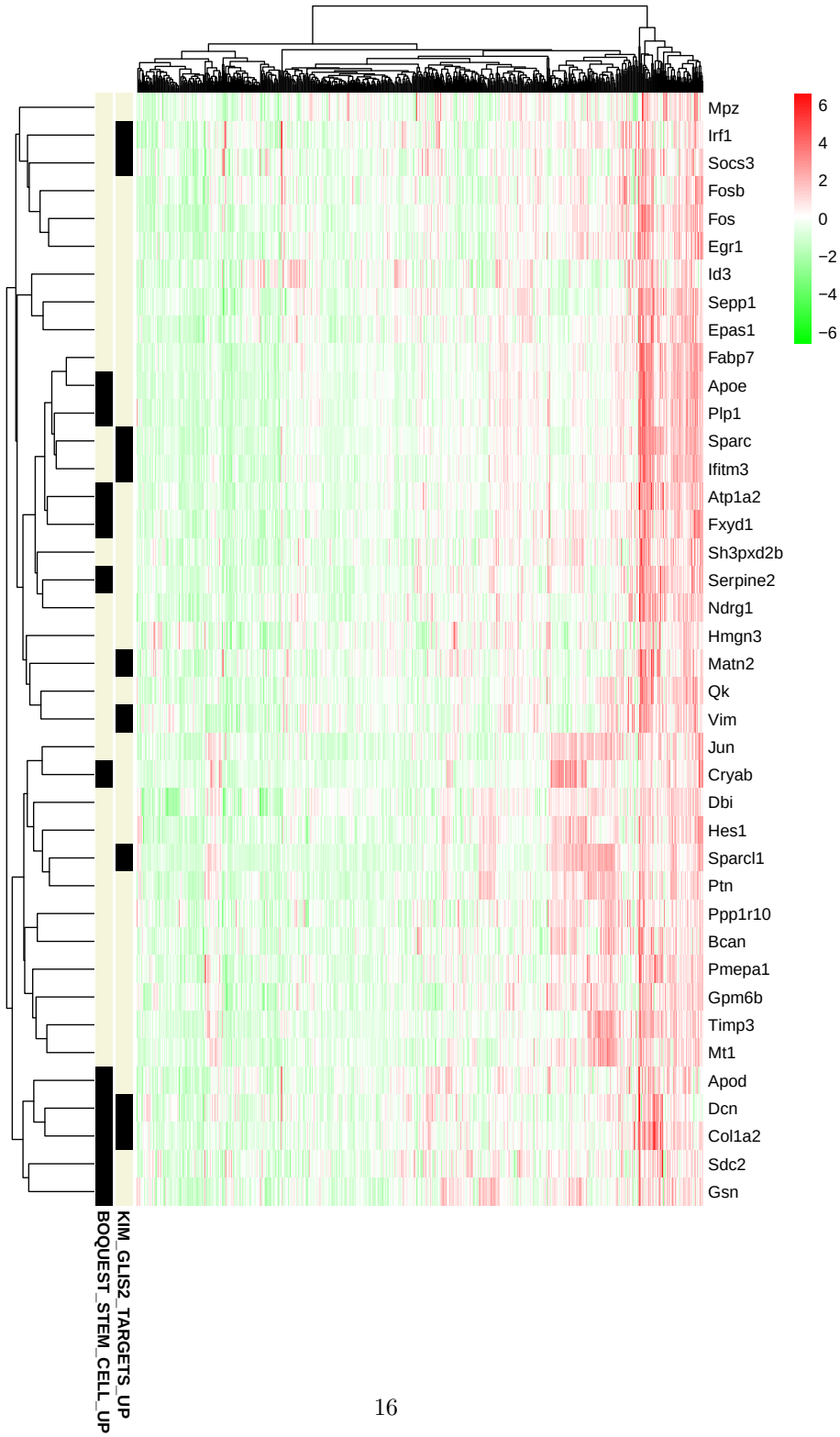
Plot the U matrix with a stringent cutoff FDR=0.01

```
plotU(usoskinRes, fdr.cutoff = 0.01)
```

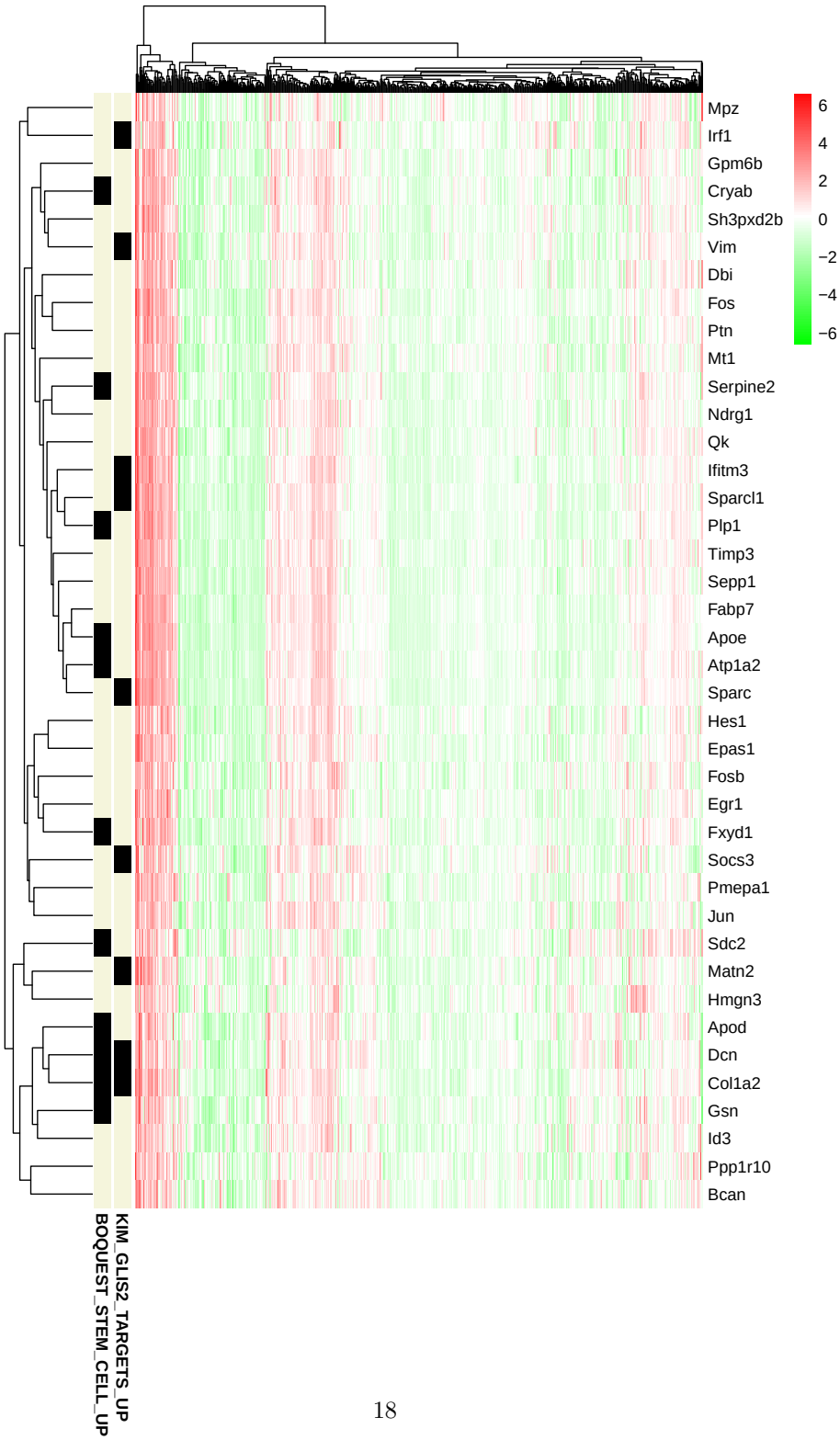

JIANG_AGING_CEREBRAL_CORTEX_DN
HELLEBREKERS_SILENCED_DURING_TUMOR_ANGIOGENESIS
KIM_GLIS2_TARGETS_UP
BOQUEST_STEM_CELL_UP
STARK_PREFRONTAL_CORTEX_22Q11_DELETION_DN
MIPS_55S_RIBOSOME_MITOCHONDRIAL
YAO_TEMPORAL_RESPONSE_TO_PROGESTERONE_CLUSTER_13
BILANGES_SERUM_AND_RAPAMYCIN_SENSITIVE_GENES
KEGG_PARKINSONS_DISEASE
PECE_MAMMARY_STEM_CELL_UP
LV6
LV10
LV22
LV18
LV13
LV4
LV15
1
0.8
0.6
0.4
0.2
0

Look at LV10, with and without regression=T

```
plotTopZallPath(usoskinRes, dataUsoskin.smooth, usoskinPath, index = 10, top = 40,
    regress = F, fdr.cutoff = 0.01)
```


Mpz
Irf1
Socs3
Fosb
Fos
Egr1
Id3
Sepp1
Epas1
Fabp7
Apoe
Plp1
Sparc
Ifitm3
Atp1a2
Fxyd1
Sh3pxd2b
Serpine2
Ndrg1
Hmgn3
Matn2
Qk
Vim
Jun
Cryab
Dbi
Hes1
Sparcl1
Ptn
Ppp1r10
Bcan
Pmepa1
Gpm6b
Timp3
Mt1
Apod
Dcn
Col1a2
Sdc2
Gsn
6
4
2
0
−2
−4
−6
KIM_GLIS2_TARGETS_UP
BOQUEST_STEM_CELL_UP

```
plotTopZallPath(usoskinRes, dataUsoskin.smooth, usoskinPath, index = 10, top = 40,
    regress = T, fdr.cutoff = 0.01)
```


Mpz
Irf1
Gpm6b
Cryab
Sh3pxd2b
Vim
Dbi
Fos
Ptn
Mt1
Serpine2
Ndrg1
Qk
Ifitm3
Sparcl1
Plp1
Timp3
Sepp1
Fabp7
Apoe
Atp1a2
Sparc
Hes1
Epas1
Fosb
Egr1
Fxyd1
Socs3
Pmepa1
Jun
Sdc2
Matn2
Hmgn3
Apod
Dcn
Col1a2
Gsn
Id3
Ppp1r10
Bcan
6
4
2
0
−2
−4
−6
KIM_GLIS2_TARGETS_UP
BOQUEST_STEM_CELL_UP