class16.Rmd

Anu Chaparala

11/18/2021

##Differential Expression Analysis

Now let's load our data files.

```
#add variables for later use (if needed)
metaFile <- "GSE37704_metadata.csv"
countFile <- "GSE37704_featurecounts.csv"
```

```
# Import metadata and take a peak
colData = read.csv("GSE37704_metadata.csv", row.names=1)
head(colData)
```

```
##               condition
## SRR493366 control_sirna
## SRR493367 control_sirna
## SRR493368 control_sirna
## SRR493369      hoxa1_kd
## SRR493370      hoxa1_kd
## SRR493371      hoxa1_kd
```

```
# Import countdata
countData = read.csv("GSE37704_featurecounts.csv", row.names=1)
head(countData)
```

```
##                 length SRR493366 SRR493367 SRR493368 SRR493369 SRR493370
## ENSG00000186092    918         0         0         0         0         0
## ENSG00000279928    718         0         0         0         0         0
## ENSG00000279457   1982        23        28        29        29        28
## ENSG00000278566    939         0         0         0         0         0
## ENSG00000273547    939         0         0         0         0         0
## ENSG00000187634   3214       124       123       205       207       212
##                 SRR493371
## ENSG00000186092         0
## ENSG00000279928         0
## ENSG00000279457        46
## ENSG00000278566         0
## ENSG00000273547         0
## ENSG00000187634       258
```

```
# Note we need to remove the odd first $length col
countData <- countData[,-1]
head(countData)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000186092         0         0         0         0         0         0
## ENSG00000279928         0         0         0         0         0         0
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000278566         0         0         0         0         0         0
## ENSG00000273547         0         0         0         0         0         0
## ENSG00000187634       124       123       205       207       212       258
```

```
# Filter count data where you have 0 read count across all samples.
countData = countData[rowSums(countData) > 1, ]

head(countData)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000187634       124       123       205       207       212       258
## ENSG00000188976      1637      1831      2383      1226      1326      1504
## ENSG00000187961       120       153       180       236       255       357
## ENSG00000187583        24        48        65        44        48        64
## ENSG00000187642         4         9        16        14        16        16
```

DESeq Analysis

```
#load DESeq2
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
dds = DESeqDataSetFromMatrix(countData=countData,
                             colData=colData,
                             design=~condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds = DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

Get our results

```
res <- results(dds)
head(res)
```

```
## log2 fold change (MLE): condition hoxa1 kd vs control sirna
## Wald test p-value: condition hoxa1 kd vs control sirna
## DataFrame with 6 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE       stat      pvalue
##                  <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.2551      0.0405765 0.2718928   0.149237 8.81366e-01
## ENSG00000187642   11.9798      0.5428105 0.5215598   1.040744 2.97994e-01
##                        padj
##                   <numeric>
## ENSG00000279457 6.85033e-01
## ENSG00000187634 5.14039e-03
## ENSG00000188976 1.75974e-35
## ENSG00000187961 1.13044e-07
## ENSG00000187583 9.19159e-01
## ENSG00000187642 4.02066e-01
```

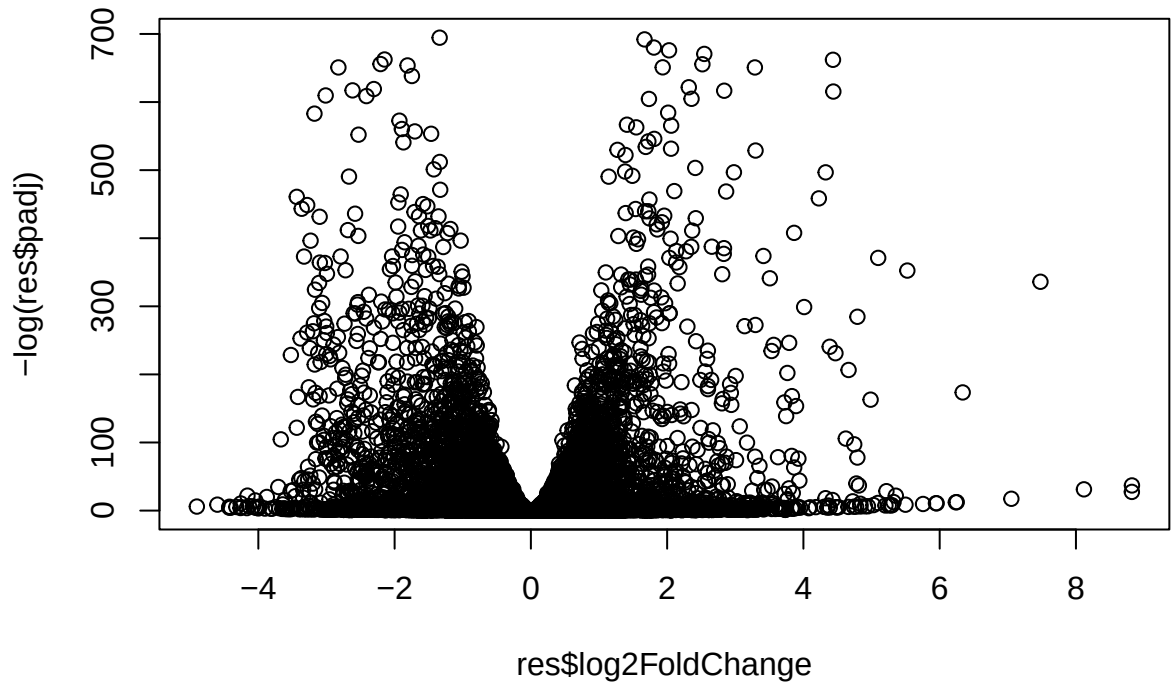
```
#Alternative results method
#res = results(dds, contrast=c("condition", "hoxa1_kd", "control_sirna"))
```

```
summary(res)
```

```
##
## out of 15280 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4351, 28%
## LFC < 0 (down)     : 4399, 29%
## outliers [1]       : 0, 0%
## low counts [2]     : 590, 3.9%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

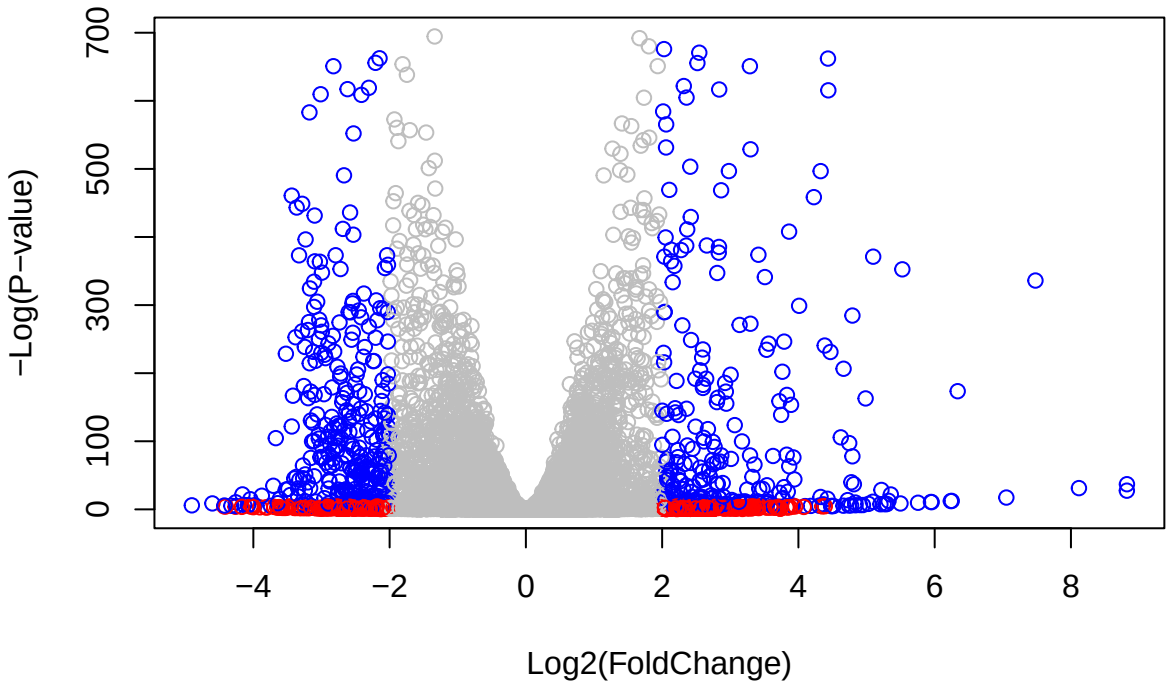
Volcano Plot

```
plot(res$log2FoldChange, -log(res$padj))
```



Let's improve our plot

```
# Make a color vector for all genes
mycols <- rep("gray", nrow(res) )

# Color red the genes with absolute fold change above 2
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

# Color blue those with adjusted p-value less than 0.01
# and absolute fold change more than 2
inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

plot( res$log2FoldChange, -log(res$padj), col= mycols, xlab="Log2(FoldChange)", ylab="-Log(P-value)" )
```



Adding Gene Annotation

Here we use the AnnotationDbi package to add gene symbols and entrez ids to our results.

```
library(AnnotationDbi)
```

```
## Warning: package 'AnnotationDbi' was built under R version 4.1.2
```

```
library(org.Hs.eg.db)
```

```
##
```

```
#to get id types in the org.Hs.eg.db data set
columns(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
## [16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
## [21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [26] "UNIPROT"
```

Let's use the mapIDs() function multiple times to add SYMBOL, ENTREZID and GENENAME annotation to our results by completing the code below.

```
library("AnnotationDbi")
library("org.Hs.eg.db")

columns(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "MAP"
## [16] "OMIM"         "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"
## [21] "PMID"         "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [26] "UNIPROT"
```

```
res$symbol = mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    keytype="ENSEMBL",
                    column="SYMBOL",
                    multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
res$entrez = mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    keytype="ENSEMBL",
                    column="ENTREZID",
                    multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
res$name =   mapIds(org.Hs.eg.db,
                    keys=row.names(res),
                    keytype="ENSEMBL",
                    column="GENENAME",
                    multiVals="first")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
head(res, 10)
```

```
## log2 fold change (MLE): condition hoxa1 kd vs control sirna
## Wald test p-value: condition hoxa1 kd vs control sirna
## DataFrame with 10 rows and 9 columns
##                  baseMean log2FoldChange     lfcSE       stat      pvalue
##                 <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.2551      0.0405765 0.2718928   0.149237 8.81366e-01
## ENSG00000187642   11.9798      0.5428105 0.5215598   1.040744 2.97994e-01
## ENSG00000188290  108.9221      2.0570638 0.1969053  10.446970 1.51282e-25
## ENSG00000187608  350.7169      0.2573837 0.1027266   2.505522 1.22271e-02
## ENSG00000188157 9128.4394      0.3899088 0.0467163   8.346304 7.04321e-17
## ENSG00000131591  156.4791      0.1965923 0.1456109   1.350121 1.76977e-01
##                       padj      symbol      entrez                  name
##                  <numeric> <character> <character>           <character>
## ENSG00000279457 6.85033e-01      WASH9P   102723897 WAS protein family h..
## ENSG00000187634 5.14039e-03      SAMD11      148398 sterile alpha motif ..
## ENSG00000188976 1.75974e-35       NOC2L       26155 NOC2 like nucleolar ..
## ENSG00000187961 1.13044e-07      KLHL17      339451 kelch like family me..
## ENSG00000187583 9.19159e-01     PLEKHN1       84069 pleckstrin homology ..
## ENSG00000187642 4.02066e-01       PERM1       84808 PPARGC1 and ESRR ind..
## ENSG00000188290 1.30113e-24        HES4       57801 hes family bHLH tran..
## ENSG00000187608 2.36679e-02       ISG15        9636 ISG15 ubiquitin like..
## ENSG00000188157 4.20589e-16        AGRN      375790                 agrin
## ENSG00000131591 2.60893e-01    C1orf159       54991 chromosome 1 open re..
```

Saving our Results

```
res = res[order(res$pvalue),]
write.csv(res, file ="deseq_results.csv")
```

##Pathway Analysis

First, we need to load our pathway, gage, and gageData packages and set up the KEGG data sets.

```
library(pathview)
```

```
## ##############################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ##############################################################################
```

```
library(gage)
```

```
##
```

```
library(gageData)
```

```
#Load data
data(kegg.sets.hs)
data(sigmet.idx.hs)

# Focus on signaling and metabolic pathways only
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]

# Examine the first 3 pathways
head(kegg.sets.hs, 3)
```

```
## $`hsa00232 Caffeine metabolism`
## [1] "10"   "1544" "1548" "1549" "1553" "7498" "9"
##
## $`hsa00983 Drug metabolism - other enzymes`
##  [1] "10"     "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
##  [9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
## [17] "3251"   "3614"   "3615"   "3704"   "51733"  "54490"  "54575"  "54576"
## [25] "54577"  "54578"  "54579"  "54600"  "54657"  "54658"  "54659"  "54963"
## [33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
## [41] "7366"   "7367"   "7371"   "7372"   "7378"   "7498"   "79799"  "83549"
## [49] "8824"   "8833"   "9"      "978"
##
## $`hsa00230 Purine metabolism`
##   [1] "100"    "10201"  "10606"  "10621"  "10622"  "10623"  "107"    "10714"
##   [9] "108"    "10846"  "109"    "111"    "11128"  "11164"  "112"    "113"
##  [17] "114"    "115"    "122481" "122622" "124583" "132"    "158"    "159"
##  [25] "1633"   "171568" "1716"   "196883" "203"    "204"    "205"    "221823"
##  [33] "2272"   "22978"  "23649"  "246721" "25885"  "2618"   "26289"  "270"
##  [41] "271"    "27115"  "272"    "2766"   "2977"   "2982"   "2983"   "2984"
##  [49] "2986"   "2987"   "29922"  "3000"   "30833"  "30834"  "318"    "3251"
##  [57] "353"    "3614"   "3615"   "3704"   "377841" "471"    "4830"   "4831"
##  [65] "4832"   "4833"   "4860"   "4881"   "4882"   "4907"   "50484"  "50940"
##  [73] "51082"  "51251"  "51292"  "5136"   "5137"   "5138"   "5139"   "5140"
##  [81] "5141"   "5142"   "5143"   "5144"   "5145"   "5146"   "5147"   "5148"
##  [89] "5149"   "5150"   "5151"   "5152"   "5153"   "5158"   "5167"   "5169"
##  [97] "51728"  "5198"   "5236"   "5313"   "5315"   "53343"  "54107"  "5422"
## [105] "5424"   "5425"   "5426"   "5427"   "5430"   "5431"   "5432"   "5433"
## [113] "5434"   "5435"   "5436"   "5437"   "5438"   "5439"   "5440"   "5441"
## [121] "5471"   "548644" "55276"  "5557"   "5558"   "55703"  "55811"  "55821"
## [129] "5631"   "5634"   "56655"  "56953"  "56985"  "57804"  "58497"  "6240"
## [137] "6241"   "64425"  "646625" "654364" "661"    "7498"   "8382"   "84172"
## [145] "84265"  "84284"  "84618"  "8622"   "8654"   "87178"  "8833"   "9060"
## [153] "9061"   "93034"  "953"    "9533"   "954"    "955"    "956"    "957"
## [161] "9583"   "9615"
```

The main gage() function requires a named vector of fold changes, where the names of the values are the Entrez gene IDs.

Note that we used the mapIDs() function above to obtain Entrez gene IDs (stored in res$entrez$) and we have the $foldchanges result

```
foldchanges = res$log2FoldChange
names(foldchanges) = res$entrez
head(foldchanges)
```

```
##      1266     54855      1465     51232      2034      2317
## -2.422719  3.201955 -2.313738 -2.059631 -1.888019 -1.649792
```

Now, let's run the gage pathway analysis.

```
# Get the results
keggres = gage(foldchanges, gsets=kegg.sets.hs)

#View the attributes of the object returned by gage.
attributes(keggres)
```

```
## $names
## [1] "greater" "less"    "stats"
```

Let's look at the first few down (less) pathway results

```
# Look at the first few down (less) pathways
head(keggres$less)
```

```
##                                       p.geomean stat.mean        p.val
## hsa04110 Cell cycle                 1.003993e-05 -4.353454 1.003993e-05
## hsa03030 DNA replication            8.909558e-05 -3.968611 8.909558e-05
## hsa03013 RNA transport              1.470985e-03 -3.007794 1.470985e-03
## hsa04114 Oocyte meiosis             1.946905e-03 -2.921710 1.946905e-03
## hsa03440 Homologous recombination   2.941989e-03 -2.868141 2.941989e-03
## hsa00010 Glycolysis / Gluconeogenesis 6.059196e-03 -2.558327 6.059196e-03
##                                            q.val set.size         exp1
## hsa04110 Cell cycle                 0.001606390      120 1.003993e-05
## hsa03030 DNA replication            0.007127646       36 8.909558e-05
## hsa03013 RNA transport              0.077876201      143 1.470985e-03
## hsa04114 Oocyte meiosis             0.077876201       99 1.946905e-03
## hsa03440 Homologous recombination   0.094143663       28 2.941989e-03
## hsa00010 Glycolysis / Gluconeogenesis 0.161578551       48 6.059196e-03
```

Note, each keggres$less$ and keggres greater object is data matrix with gene sets as rows sorted by p-value.

Now, let's try out the pathview() function from the pathview package to make a pathway plot with our RNA-Seq expression results shown in color.

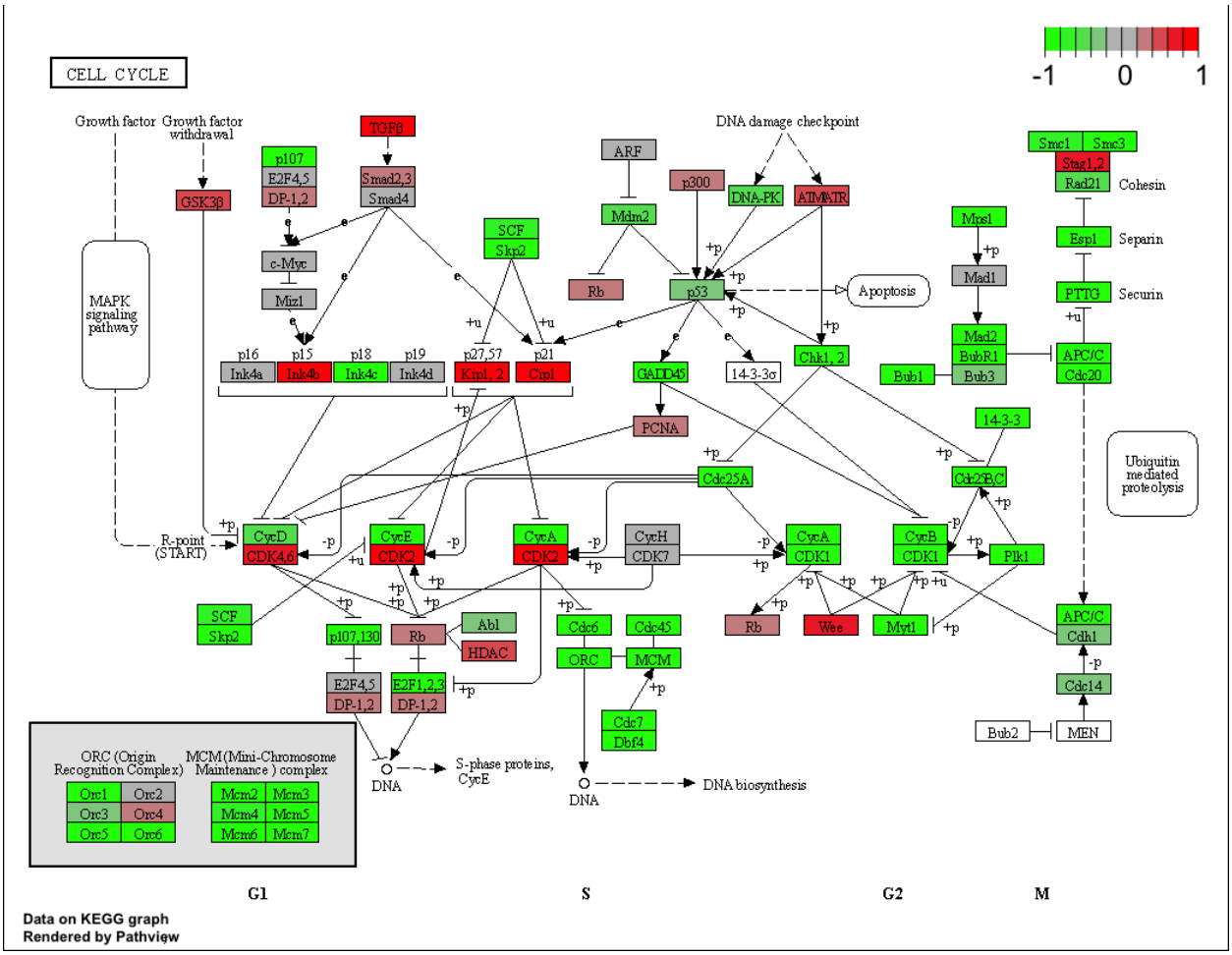
```
pathview(gene.data=foldchanges, pathway.id="hsa04110")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16
```

```
## Info: Writing image file hsa04110.pathview.png
```

Let's insert our newly generated pathway analysis results!



Let's do the same for the top 5 upregulated pathways using a more automatic method.

```
#Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

# Extract the 8 character long IDs part of each string
keggresids = substr(keggrespathways, start=1, stop=8)
keggresids
```

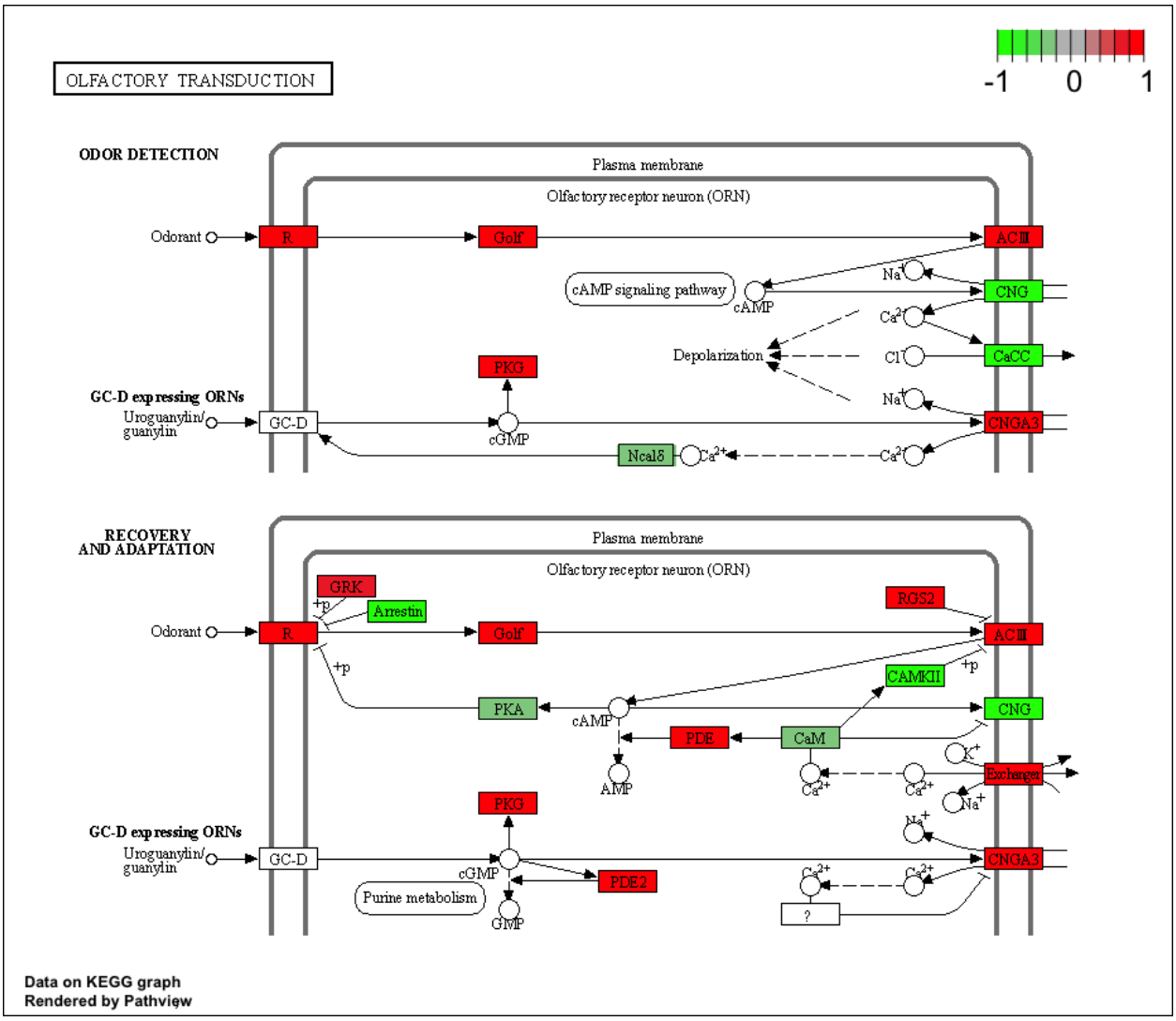
```
## [1] "hsa04640" "hsa04630" "hsa04142" "hsa00140" "hsa04740"
```

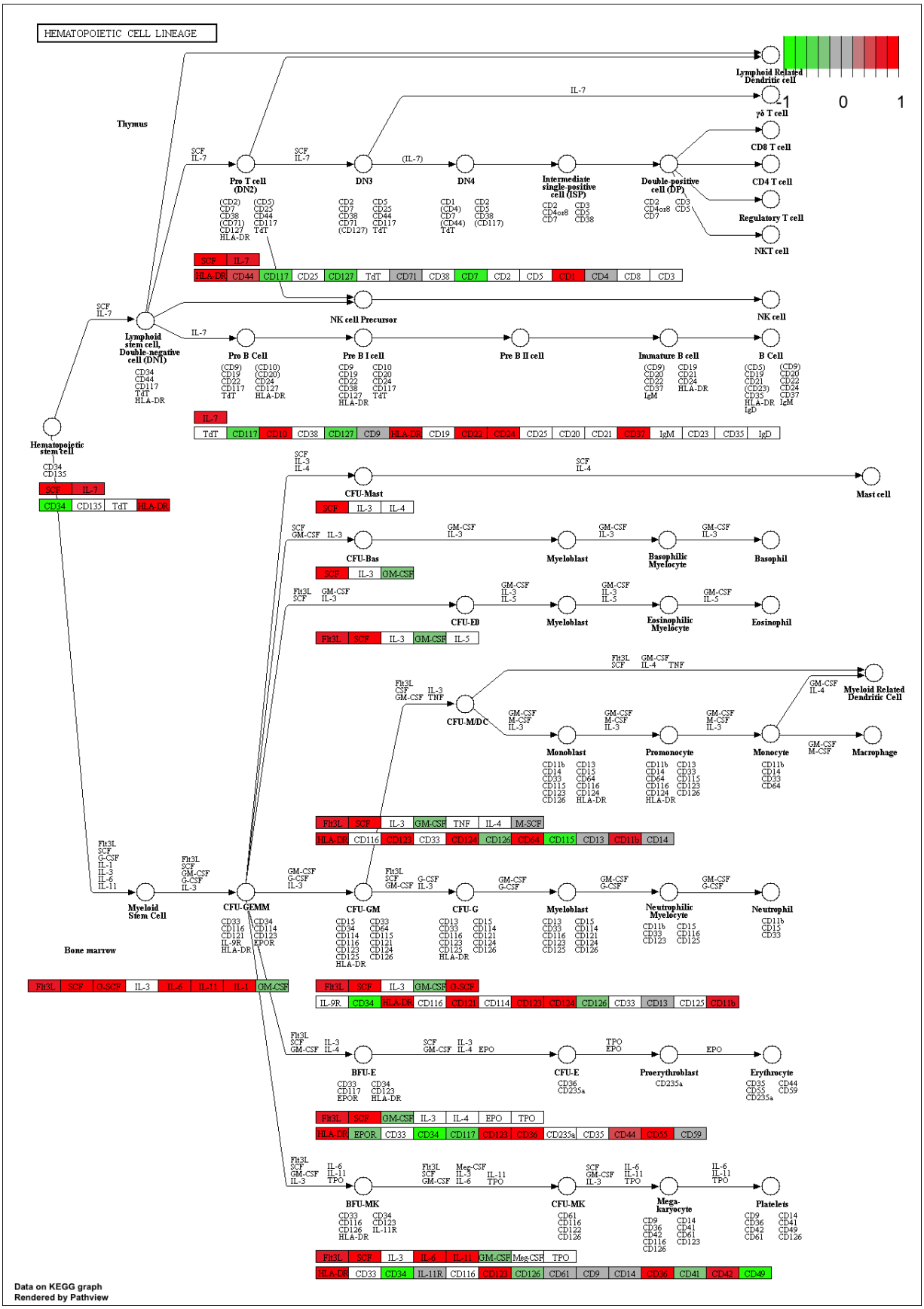
Finally, lets pass these IDs in keggresids to the pathview() function to draw plots for all the top 5 pathways.

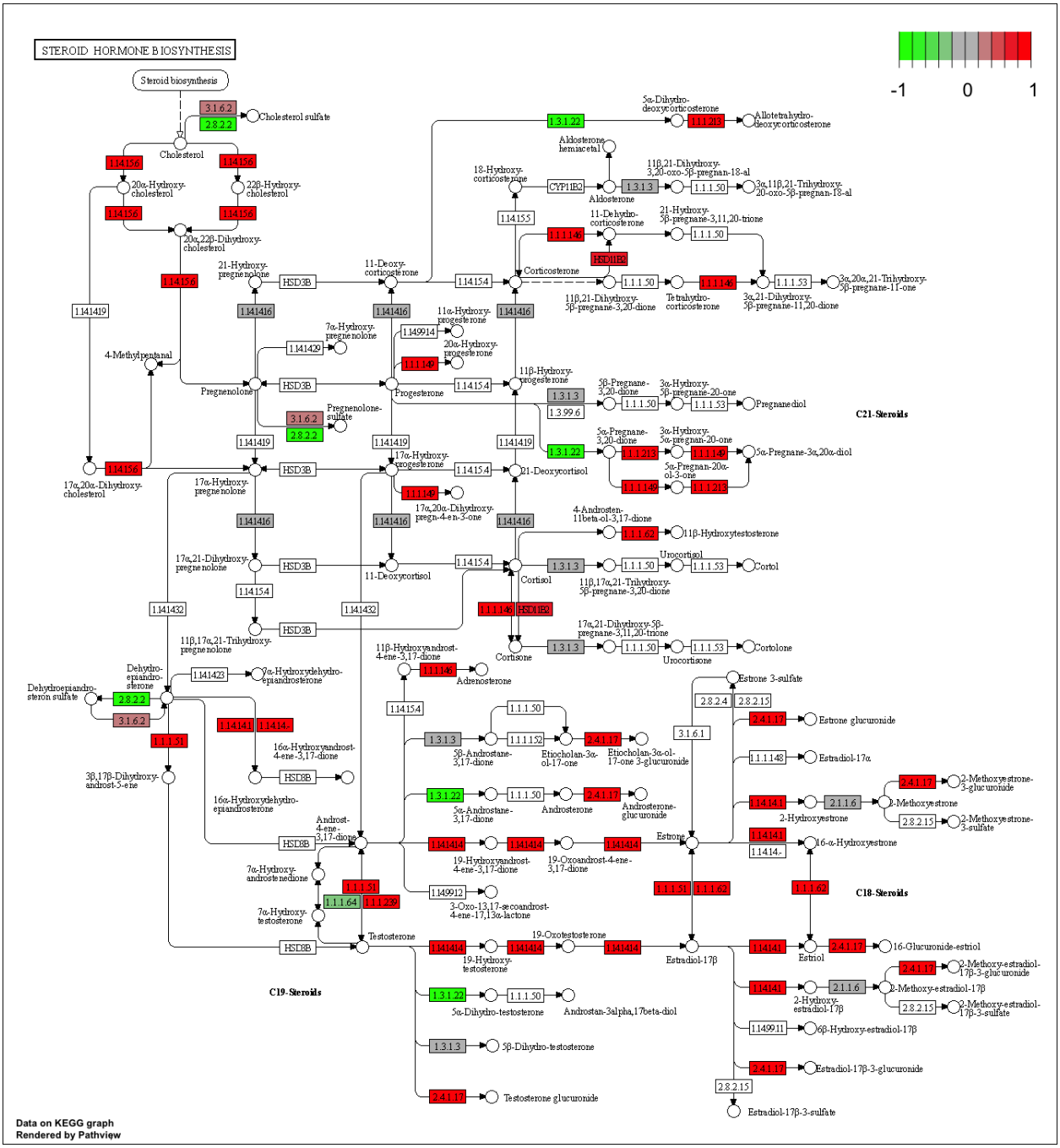
```
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```

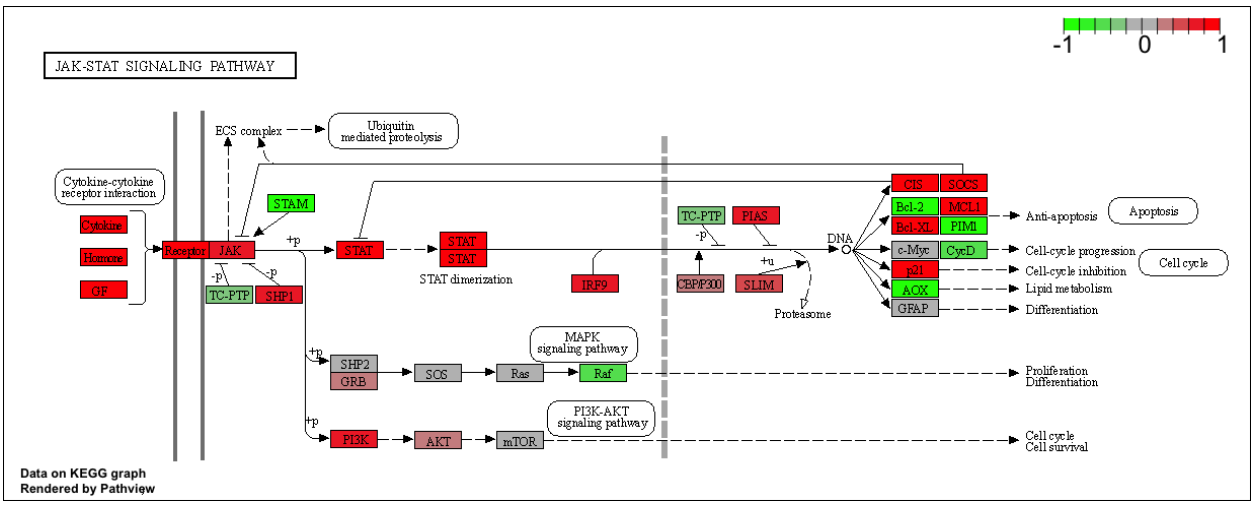
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16
```

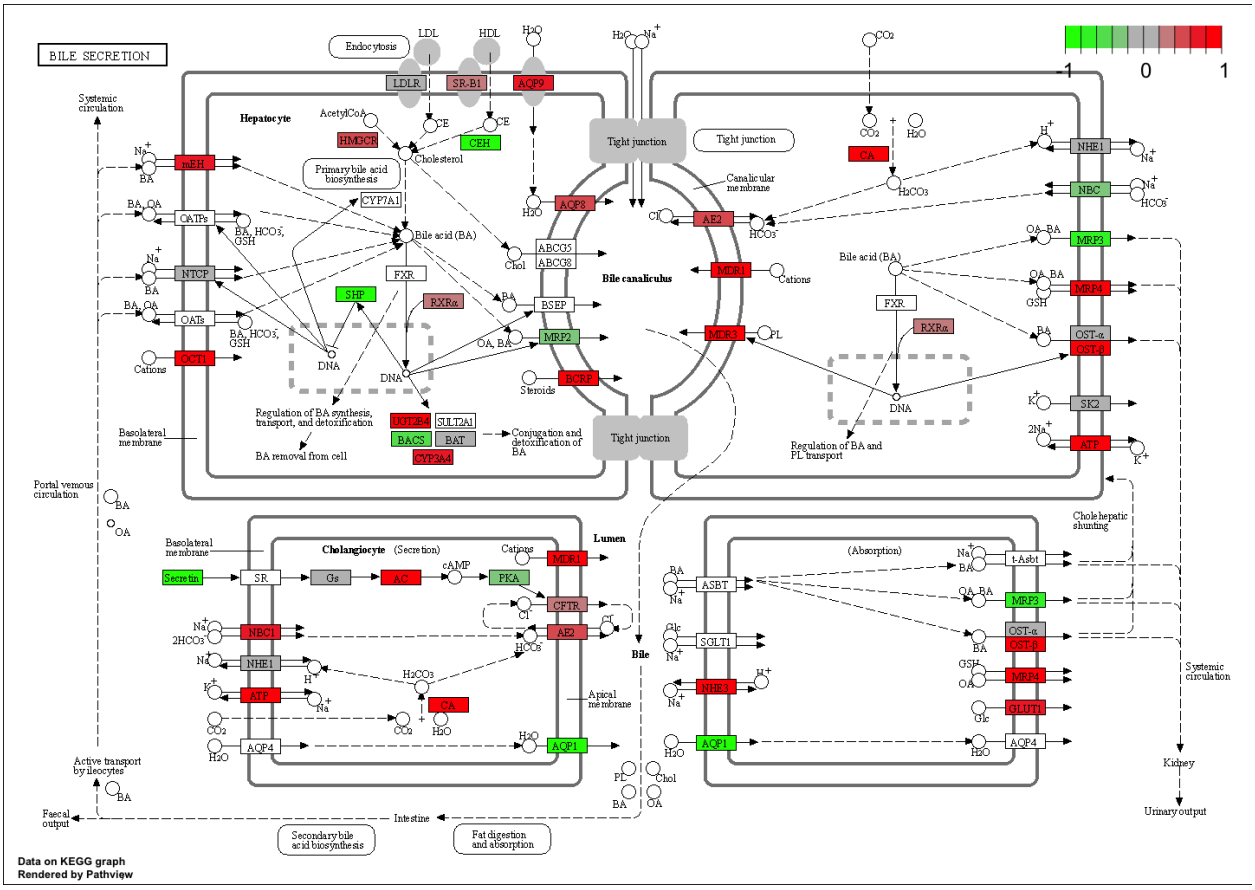
```
## Info: Writing image file hsa04640.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa04630.pathview.png

## Info: Downloading xml files for hsa04142, 1/1 pathways..

## Info: Downloading png files for hsa04142, 1/1 pathways..

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa04142.pathview.png

## Info: some node width is different from others, and hence adjusted!

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa00140.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa04740.pathview.png

## Info: some node width is different from others, and hence adjusted!
```


OLFACTORY TRANSDUCTION
-1
0
1
ODOR DETECTION
Plasma membrane
Olfactory receptor neuron (ORN)
Odorant
R
Golf
ACIII
cAMP signaling pathway
cAMP
Na+
CNG
Ca2+
Depolarization
Cl-
CaCC
PKG
GC-D expressing ORNs
Uroguanylin/
guanylin
GC-D
cGMP
Na+
CNGA3
Ncald
Ca2+
RECOVERY
AND ADAPTATION
Plasma membrane
Olfactory receptor neuron (ORN)
GRK
+p
Arrestin
RGS2
Odorant
R
Golf
ACIII
+p
CAMKII
+p
PKA
cAMP
CNG
PDE
CaM
K+
Exchanger
AMP
Ca2+
Ca2+
Na+
PKG
Na+
GC-D expressing ORNs
Uroguanylin/
guanylin
GC-D
CNGA3
cGMP
PDE2
Ca2+
Ca2+
Purine metabolism
GMP
?
Data on KEGG graph
Rendered by Pathview


HEMATOPOIETIC CELL LINEAGE
-1
0
1
Lymphoid Related Dendritic cell
γδ T cell
CD8 T cell
CD4 T cell
Regulatory T cell
NKT cell
NK cell
Thymus
Pro T cell (DN2)
DN3
DN4
Intermediate single-positive cell (ISP)
Double-positive cell (DP)
NK cell Precursor
Lymphoid stem cell, Double-negative cell (DN1)
Pro B Cell
Pre B I cell
Pre B II cell
Immature B cell
B Cell
Hematopoietic stem cell
CFU-Mast
Mast cell
CFU-Bas
Myeloblast
Basophilic Myelocyte
Basophil
CFU-Eo
Eosinophilic Myelocyte
Eosinophil
CFU-M/DC
Monoblast
Promonocyte
Monocyte
Myeloid Related Dendritic Cell
Macrophage
Myeloid Stem Cell
CFU-GEMM
CFU-GM
CFU-G
Neutrophilic Myelocyte
Neutrophil
Bone marrow
BFU-E
CFU-E
Proerythroblast
Erythrocyte
BFU-MK
CFU-MK
Mega-karyocyte
Platelets
Data on KEGG graph
Rendered by Pathview


STEROID HORMONE BIOSYNTHESIS
-1
0
1
Steroid biosynthesis
Cholesterol
Cholesterol sulfate
20α-Hydroxy-cholesterol
22β-Hydroxy-cholesterol
20α,22β-Dihydroxy-cholesterol
21-Hydroxy-pregnenolone
11-Deoxy-corticosterone
Corticosterone
11β,21-Dihydroxy-5β-pregnane-3,20-dione
Tetrahydro-corticosterone
3α,20α,21-Trihydroxy-5β-pregnane-11-one
3α,21-Dihydroxy-5β-pregnane-11,20-dione
5α-Dihydro-deoxycorticosterone
Allotetrahydro-deoxycorticosterone
Aldosterone hemiacetal
18-Hydroxy-corticosterone
Aldosterone
11β,21-Dihydroxy-3,20-oxo-5β-pregnan-18-al
3α,11β,21-Trihydroxy-20-oxo-5β-pregnan-18-al
11-Dehydro-corticosterone
21-Hydroxy-5β-pregnane-3,11,20-trione
HSD11B2
7α-Hydroxy-pregnenolone
11α-Hydroxy-progesterone
20α-Hydroxy-progesterone
4-Methylpentanal
Pregnenolone
Progesterone
11β-Hydroxy-progesterone
5β-Pregnane-3,20-dione
3α-Hydroxy-5β-pregnane-20-one
Pregnanediol
Pregnenolone-sulfate
C21-Steroids
17α-Hydroxy-progesterone
21-Deoxycortisol
5α-Pregnane-3,20-dione
3α-Hydroxy-5α-pregnan-20-one
5α-Pregnane-3α,20α-diol
5α-Pregnane-20α-ol-3-one
17α,20α-Dihydroxy-cholesterol
17α-Hydroxy-pregnenolone
17α,20α-Dihydroxy-pregn-4-en-3-one
4-Androsten-11beta-ol-3,17-dione
11β-Hydroxytestosterone
17α,21-Dihydroxy-pregnenolone
11-Deoxycortisol
Cortisol
Urocortisol
Cortol
11β,17α,21-Trihydroxy-5β-pregnane-3,20-dione
11β,17α,21-Trihydroxy-pregnenolone
17α,21-Dihydroxy-5β-pregnane-3,11,20-trione
Cortisone
Cortolone
11β-Hydroxyandrost-4-ene-3,17-dione
Adrenosterone
Dehydroepiandro-sterone sulfate
Dehydro-epiandro-sterone
7α-Hydroxydehydro-epiandrosterone
16α-Hydroxyandrost-4-ene-3,17-dione
3β,17β-Dihydroxy-androst-5-ene
16α-Hydroxydehydro-epiandrosterone
Androst-4-ene-3,17-dione
5β-Androstane-3,17-dione
Etiocholan-3α-ol-17-one
Etiocholan-3α-ol-17-one 3-glucuronide
5α-Androstane-3,17-dione
Androsterone
Androsterone-glucuronide
19-Hydroxyandrost-4-ene-3,17-dione
19-Oxoandrost-4-ene-3,17-dione
Estrone
Estrone 3-sulfate
Estrone glucuronide
Estradiol-17α
2-Methoxyestrone-3-glucuronide
2-Hydroxyestrone
2-Methoxyestrone
2-Methoxyestrone-3-sulfate
16-α-Hydroxyestrone
C18-Steroids
7α-Hydroxy-androstenedione
3-Oxo-13,17-secoandrost-4-ene-17,13α-lactone
7α-Hydroxy-testosterone
Testosterone
19-Oxotestosterone
19-Hydroxy-testosterone
Estradiol-17β
Estriol
16-Glucuronide-estriol
2-Methoxy-estradiol-17β-3-glucuronide
2-Methoxy-estradiol-17β
2-Hydroxy-estradiol-17β
2-Methoxy-estradiol-17β-3-sulfate
6β-Hydroxy-estradiol-17β
Estradiol-17β-3-glucuronide
Estradiol-17β-3-sulfate
C19-Steroids
5α-Dihydro-testosterone
Androstan-3alpha,17beta-diol
5β-Dihydro-testosterone
Testosterone glucuronide
Data on KEGG graph
Rendered by Pathview





Let's do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways?

```
#Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways2 <- rownames(keggres$less)[1:5]

# Extract the 8 character long IDs part of each string
keggresids1 = substr(keggrespathways2, start=1, stop=8)
keggresids1
```

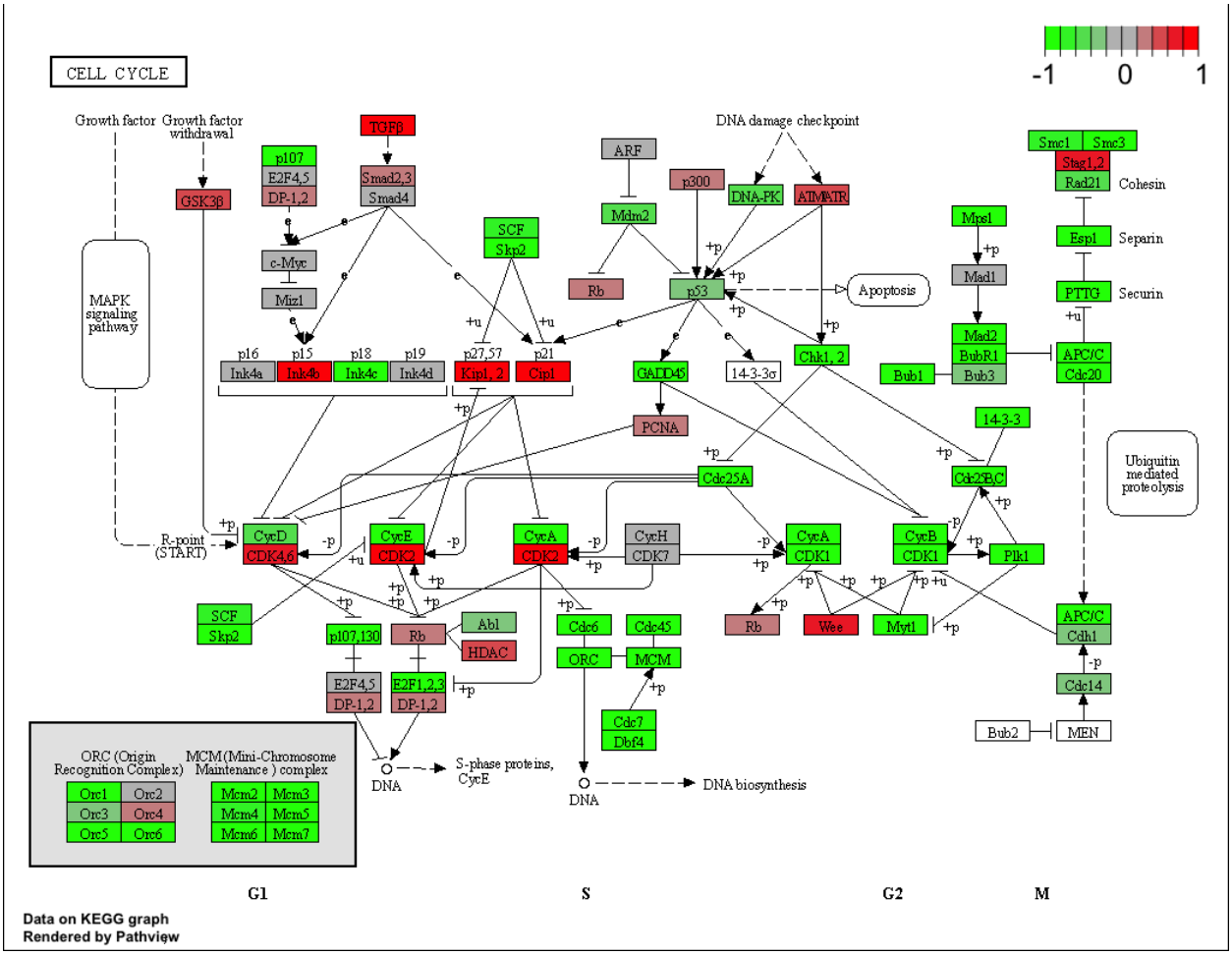
```
## [1] "hsa04110" "hsa03030" "hsa03013" "hsa04114" "hsa03440"
```

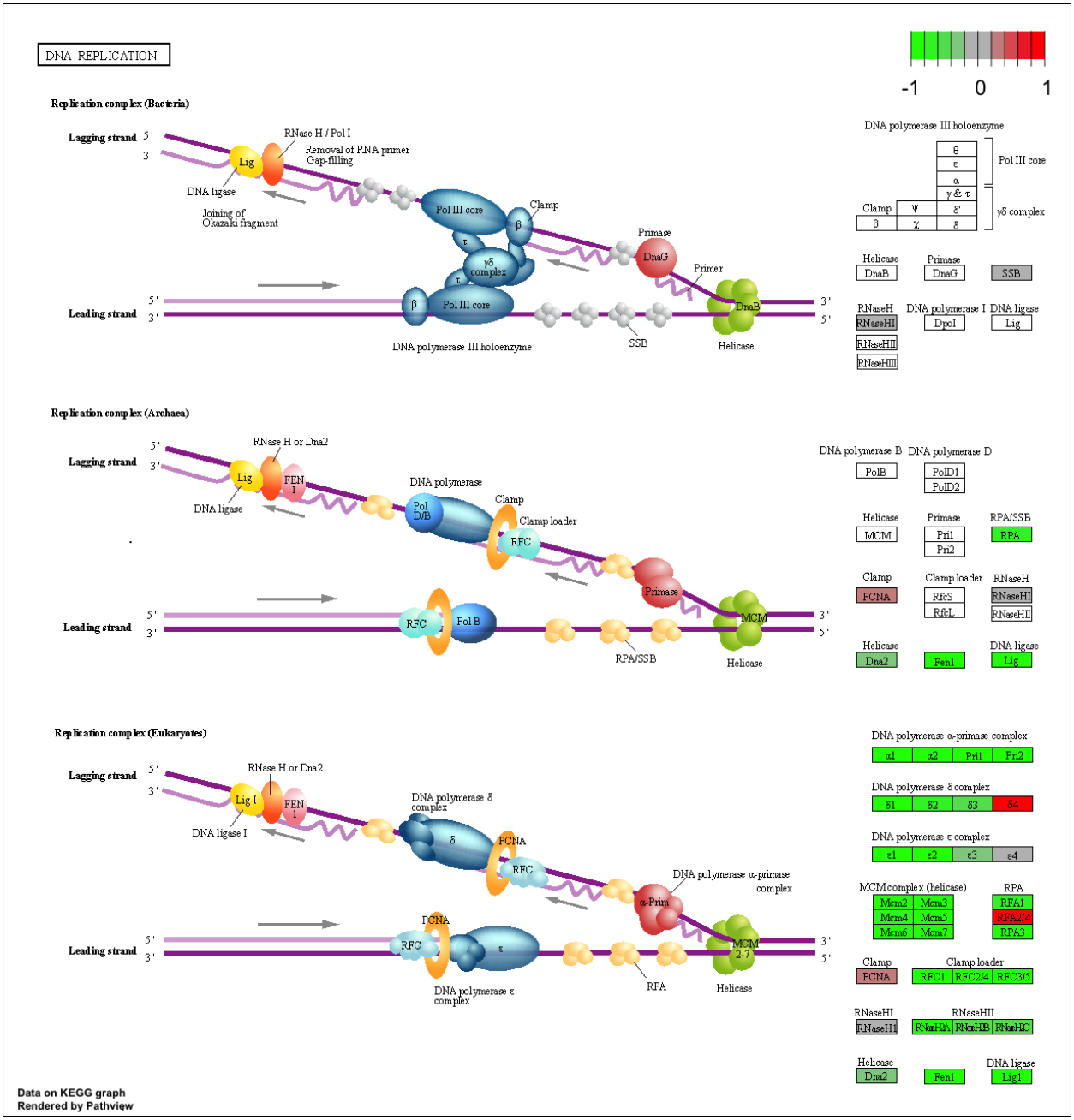
Generate the 5 pathways from above ids.

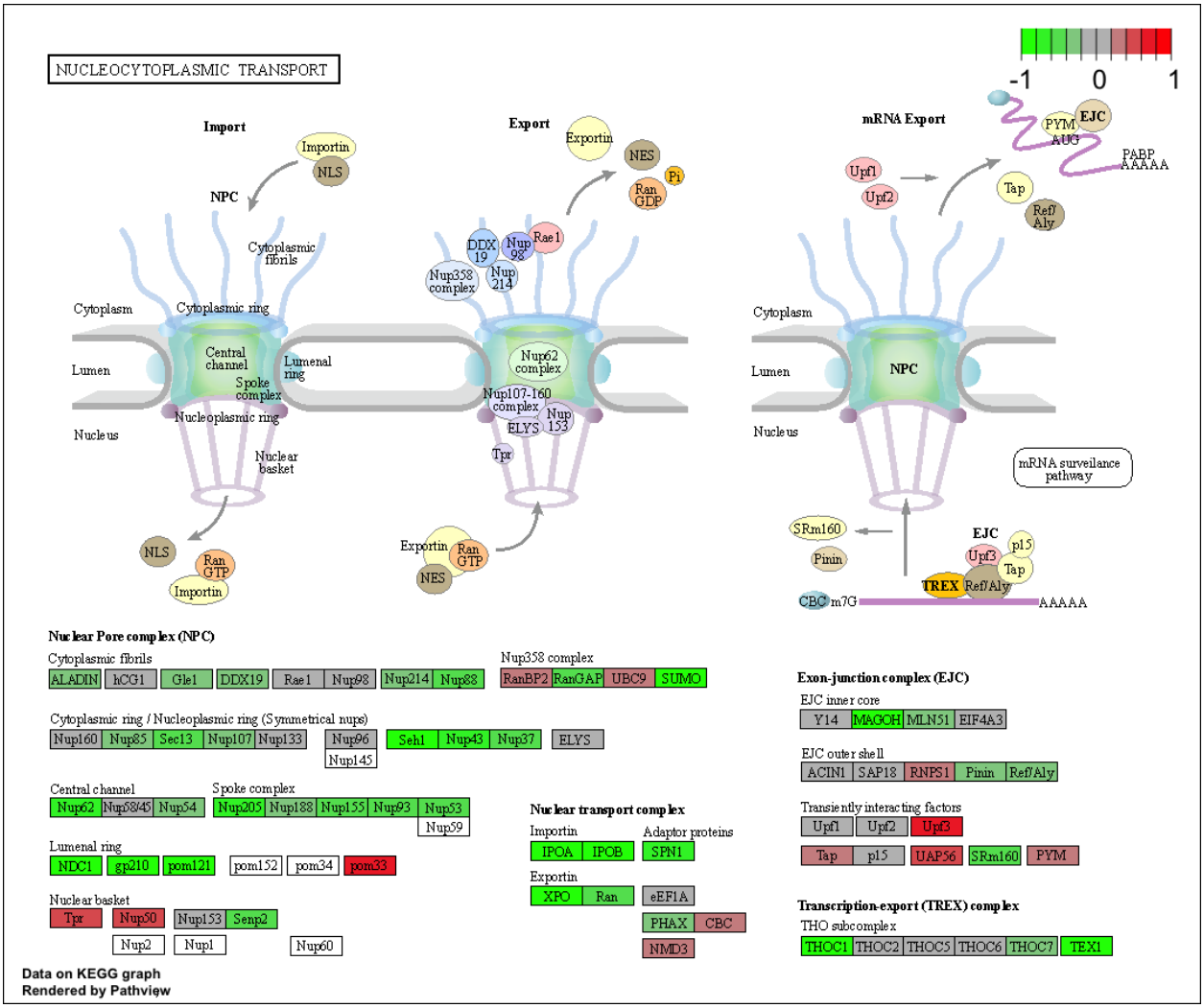
```
pathview(gene.data=foldchanges, pathway.id=keggresids1, species="hsa")
```

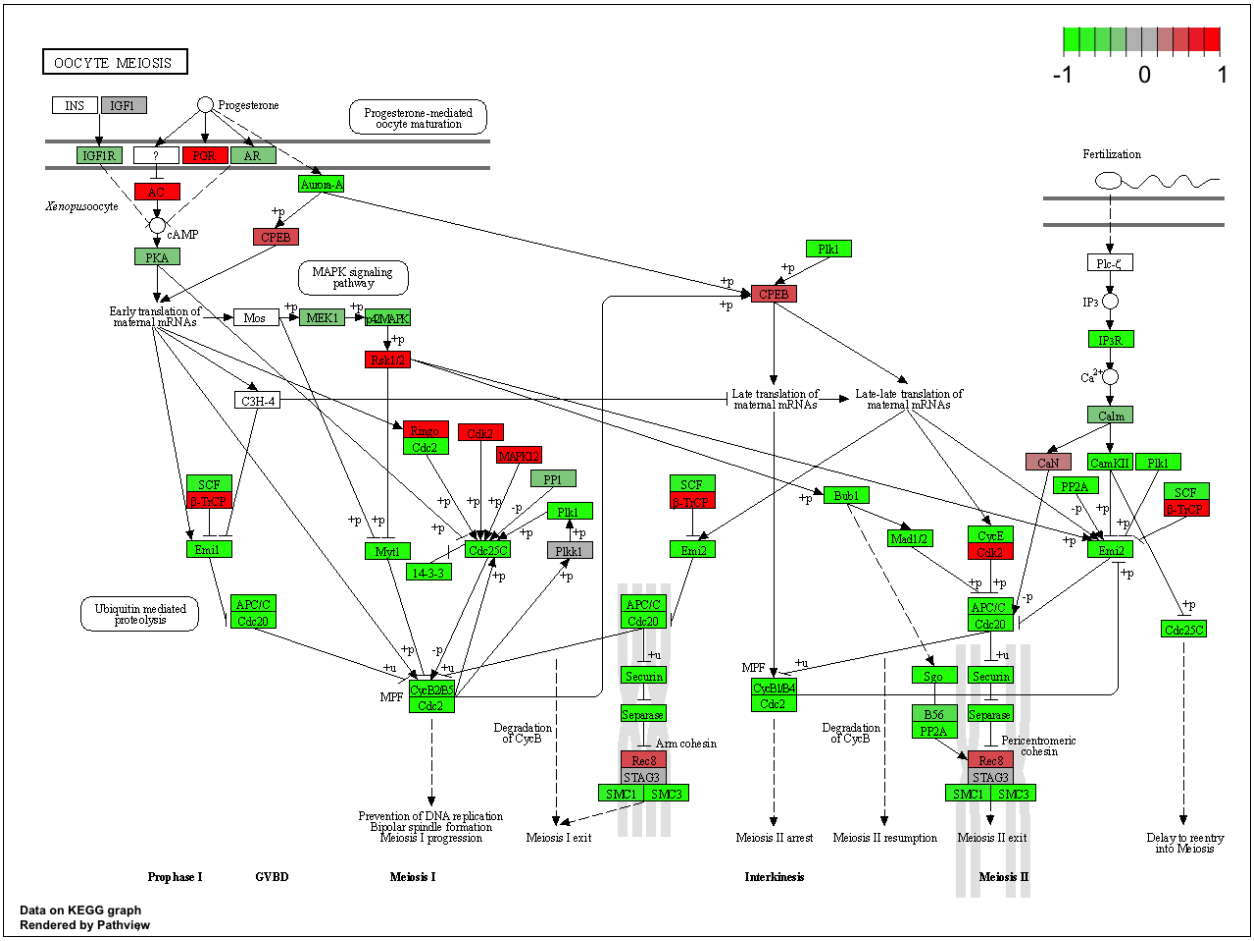
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa04110.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa03030.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa03013.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa04114.pathview.png

## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/Anu/Desktop/BIMM 143A/Bimm143_github/class16

## Info: Writing image file hsa03440.pathview.png
```

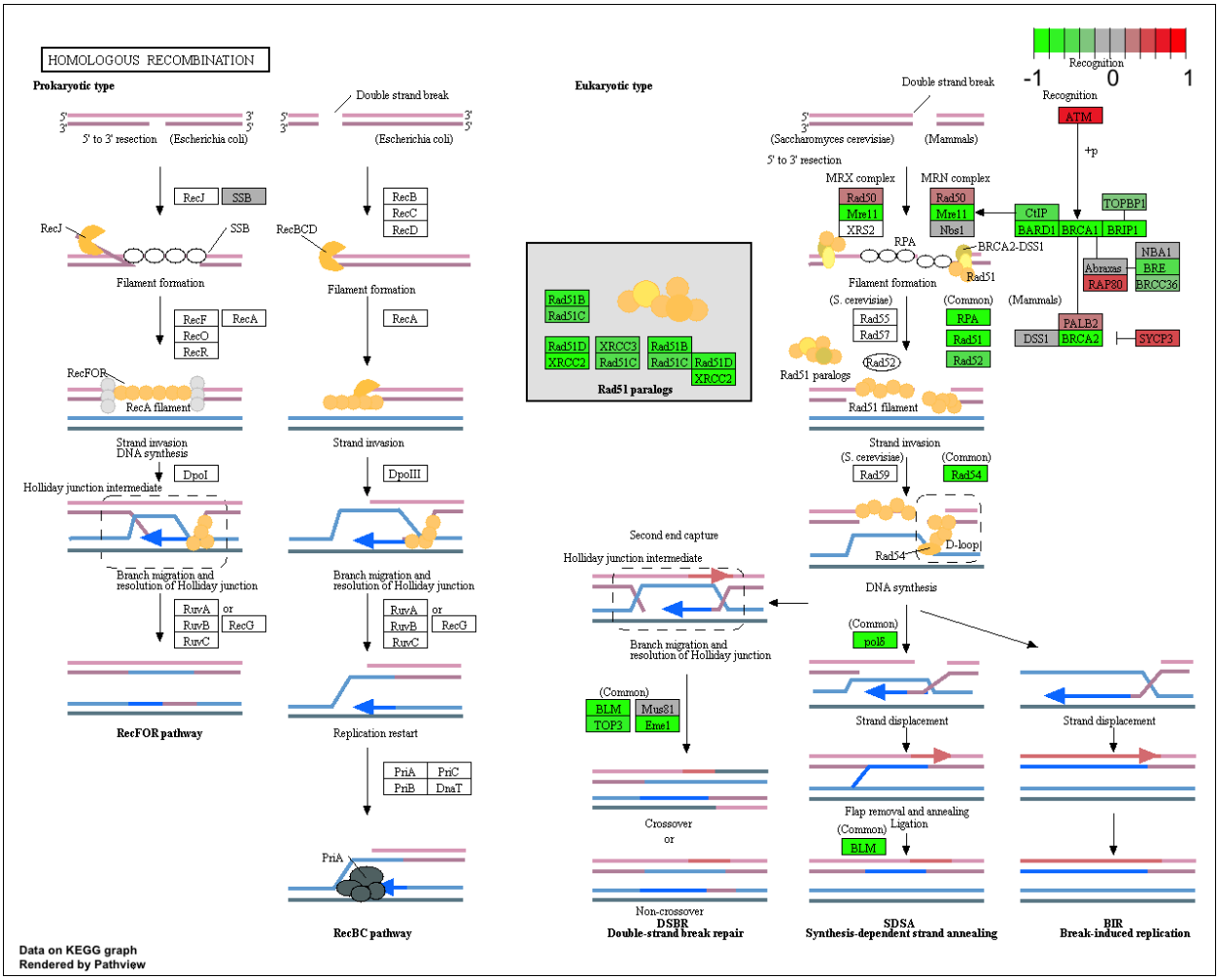
Insert the above generated pathways for the down regulated group.


CELL CYCLE
-1
0
1
Growth factor
Growth factor withdrawal
GSK3β
TGFβ
Smad2,3
Smad4
p107
E2F4,5
DP-1,2
c-Myc
Miz1
MAPK signaling pathway
SCF
Skp2
ARF
Mdm2
p300
DNA-PK
ATM/ATR
DNA damage checkpoint
Rb
p53
Apoptosis
Chk1, 2
p16
Ink4a
p15
Ink4b
p18
Ink4c
p19
Ink4d
p27,57
Kip1, 2
p21
Cip1
GADD45
14-3-3σ
PCNA
Cdc25A
R-point
(START)
CycD
CDK4,6
CycE
CDK2
CycA
CDK2
CycH
CDK7
CycA
CDK1
CycB
CDK1
Cdc25B/C
14-3-3
Plk1
Ubiquitin mediated proteolysis
Mps1
Mad1
Mad2
BubR1
Bub3
Bub1
Smc1
Smc3
Stag1,2
Rad21
Cohesin
Esp1
Separin
PTTG
Securin
APC/C
Cdc20
APC/C
Cdh1
Cdc14
Bub2
MEN
p107,130
Rb
Abl
HDAC
E2F4,5
DP-1,2
E2F1,2,3
DP-1,2
Cdc6
Cdc45
ORC
MCM
Cdc7
Dbf4
Rb
Wee
Myt1
DNA
S-phase proteins, CycE
DNA
DNA biosynthesis
ORC (Origin Recognition Complex)
Orc1
Orc2
Orc3
Orc4
Orc5
Orc6
MCM (Mini-Chromosome Maintenance) complex
Mcm2
Mcm3
Mcm4
Mcm5
Mcm6
Mcm7
G1
S
G2
M
Data on KEGG graph
Rendered by Pathview


DNA REPLICATION
-1
0
1
Replication complex (Bacteria)
Lagging strand
Leading strand
RNase H / Pol I
Removal of RNA primer
Gap-filling
Lig
DNA ligase
Joining of
Okazaki fragment
Pol III core
Clamp
β
τ
γδ
complex
Primase
DnaG
Primer
DnaB
SSB
Helicase
DNA polymerase III holoenzyme
DNA polymerase III holoenzyme
θ
ε
α
γ & τ
Pol III core
Clamp
ψ
δ'
γδ complex
β
χ
δ
Helicase
DnaB
Primase
DnaG
SSB
RNaseH
RNaseHII
RNaseHII
RNaseHIII
DNA polymerase I
DpoI
DNA ligase
Lig
Replication complex (Archaea)
RNase H or Dna2
Lig
FEN
1
DNA ligase
DNA polymerase
Pol
D/B
Clamp
Clamp loader
RFC
Primase
MCM
RFC
Pol B
RPA/SSB
Helicase
DNA polymerase B
DNA polymerase D
PolB
PolD1
PolD2
Helicase
MCM
Primase
Pri1
Pri2
RPA/SSB
RPA
Clamp
PCNA
Clamp loader
RfcS
RfcL
RNaseH
RNaseHI
RNaseHII
Helicase
Dna2
Fen1
DNA ligase
Lig
Replication complex (Eukaryotes)
RNase H or Dna2
Lig I
FEN
1
DNA ligase I
DNA polymerase δ
complex
δ
PCNA
RFC
DNA polymerase α-primase
complex
α-Prim
MCM
2-7
RFC
PCNA
ε
DNA polymerase ε
complex
RPA
Helicase
DNA polymerase α-primase complex
α1
α2
Pri1
Pri2
DNA polymerase δ complex
δ1
δ2
δ3
δ4
DNA polymerase ε complex
ε1
ε2
ε3
ε4
MCM complex (helicase)
Mcm2
Mcm3
Mcm4
Mcm5
Mcm6
Mcm7
RPA
RFA1
RFA2/4
RFA3
Clamp
PCNA
Clamp loader
RFC1
RFC2/4
RFC3/5
RNaseHI
RNaseHI
RNaseHII
RNaseHA
RNaseHB
RNaseHC
Helicase
Dna2
Fen1
DNA ligase
Lig1
Data on KEGG graph
Rendered by Pathview


NUCLEOCYTOPLASMIC TRANSPORT
-1
0
1
Import
Importin
NLS
NPC
Cytoplasmic fibrils
Cytoplasm
Cytoplasmic ring
Central channel
Spoke complex
Lumenal ring
Lumen
Nucleoplasmic ring
Nucleus
Nuclear basket
NLS
Ran GTP
Importin
Export
Exportin
NES
Pi
Ran GDP
DDX 19
Nup 98
Rae1
Nup358 complex
Nup 214
Nup62 complex
Nup107-160 complex
ELYS
Nup 153
Tpr
Exportin
Ran GTP
NES
mRNA Export
EJC
PYM
AUG
PABP
AAAAA
Upf1
Upf2
Tap
Ref/ Aly
Cytoplasm
Lumen
NPC
Nucleus
mRNA surveilance pathway
SRm160
Pinin
EJC
Upf3
p15
Tap
TREX
Ref/Aly
CBC
m7G
AAAAA
Nuclear Pore complex (NPC)
Cytoplasmic fibrils
ALADIN
hCG1
Gle1
DDX19
Rae1
Nup98
Nup214
Nup88
Nup358 complex
RanBP2
RanGAP
UBC9
SUMO
Cytoplasmic ring / Nucleoplasmic ring (Symmetrical nups)
Nup160
Nup85
Sec13
Nup107
Nup133
Nup96
Nup145
Seh1
Nup43
Nup37
ELYS
Central channel
Nup62
Nup58/45
Nup54
Spoke complex
Nup205
Nup188
Nup155
Nup93
Nup53
Nup59
Lumenal ring
NDC1
gp210
pom121
pom152
pom34
pom33
Nuclear basket
Tpr
Nup50
Nup153
Senp2
Nup2
Nup1
Nup60
Nuclear transport complex
Importin
Adaptor proteins
IPOA
IPOB
SPN1
Exportin
XPO
Ran
eEF1A
PHAX
CBC
NMD3
Exon-junction complex (EJC)
EJC inner core
Y14
MAGOH
MLN51
EIF4A3
EJC outer shell
ACIN1
SAP18
RNPS1
Pinin
Ref/Aly
Transiently interacting factors
Upf1
Upf2
Upf3
Tap
p15
UAP56
SRm160
PYM
Transcription-export (TREX) complex
THO subcomplex
THOC1
THOC2
THOC5
THOC6
THOC7
TEX1
Data on KEGG graph
Rendered by Pathview


OOCYTE MEIOSIS
-1
0
1
INS
IGF1
Progesterone
Progesterone-mediated oocyte maturation
IGF1R
?
PGR
AR
AC
Xenopus oocyte
cAMP
PKA
Aurora-A
CPEB
MAPK signaling pathway
Early translation of maternal mRNAs
Mos
MEK1
MAPK
Rsk1/2
C3H-4
Ringo
Cdc2
Cdk2
MAPK12
PP1
Plk1
Myt1
14-3-3
Cdc25C
Plkk1
SCF
β-TrCP
Emi1
Ubiquitin mediated proteolysis
APC/C
Cdc20
MPF
CycB2/B5
Prevention of DNA replication
Bipolar spindle formation
Meiosis I progression
Degradation of CycB
Securin
Separase
Arm cohesin
Rec8
STAG3
SMC1
SMC3
Meiosis I exit
Late translation of maternal mRNAs
Late-late translation of maternal mRNAs
Emi2
Bub1
CycB1/B4
Meiosis II arrest
Mad1/2
Sgo
B56
PP2A
Meiosis II resumption
CycE
Pericentromeric cohesin
Meiosis II exit
Fertilization
Plc-ζ
IP3
IP3R
Ca2+
Calm
CaN
CamKII
Delay to reentry into Meiosis
Prophase I
GVBD
Meiosis I
Interkinesis
Meiosis II
Data on KEGG graph
Rendered by Pathview



##Gene Ontology

We can also do a similar procedure with gene ontology. Similar to above, go.sets.hs has all GO terms. go.subs.hs is a named list containing indexes for the BP, CC, and MF ontologies. Let's focus on BP (a.k.a Biological Process) here.

```
data(go.sets.hs)
data(go.subs.hs)

# Focus on Biological Process subset of GO
gobpsets = go.sets.hs[go.subs.hs$BP]

gobpres = gage(foldchanges, gsets=gobpsets, same.dir=TRUE)

lapply(gobpres, head)
```

```
## $greater
##                                             p.geomean stat.mean        p.val
## GO:0007156 homophilic cell adhesion      4.892477e-05  3.971899 4.892477e-05
## GO:0060429 epithelium development        6.727546e-05  3.834595 6.727546e-05
## GO:0007610 behavior                      1.988039e-04  3.557821 1.988039e-04
## GO:0048729 tissue morphogenesis          2.470962e-04  3.498983 2.470962e-04
```

```
## GO:0002009 morphogenesis of an epithelium 3.227439e-04  3.429317 3.227439e-04
## GO:0016337 cell-cell adhesion             8.195506e-04  3.163057 8.195506e-04
##                                               q.val set.size         exp1
## GO:0007156 homophilic cell adhesion       0.1337436      107 4.892477e-05
## GO:0060429 epithelium development         0.1337436      478 6.727546e-05
## GO:0007610 behavior                       0.2456136      403 1.988039e-04
## GO:0048729 tissue morphogenesis           0.2456136      403 2.470962e-04
## GO:0002009 morphogenesis of an epithelium 0.2566460      326 3.227439e-04
## GO:0016337 cell-cell adhesion             0.3782658      318 8.195506e-04
##
## $less
##                                             p.geomean stat.mean        p.val
## GO:0000279 M phase                        1.475361e-16 -8.323749 1.475361e-16
## GO:0048285 organelle fission              7.498413e-16 -8.160305 7.498413e-16
## GO:0000280 nuclear division               2.135098e-15 -8.034814 2.135098e-15
## GO:0007067 mitosis                        2.135098e-15 -8.034814 2.135098e-15
## GO:0000087 M phase of mitotic cell cycle 5.927567e-15 -7.891758 5.927567e-15
## GO:0007059 chromosome segregation         1.055918e-11 -6.988373 1.055918e-11
##                                                q.val set.size         exp1
## GO:0000279 M phase                        5.866036e-13      492 1.475361e-16
## GO:0048285 organelle fission              1.490684e-12      373 7.498413e-16
## GO:0000280 nuclear division               2.122288e-12      349 2.135098e-15
## GO:0007067 mitosis                        2.122288e-12      349 2.135098e-15
## GO:0000087 M phase of mitotic cell cycle 4.713601e-12      359 5.927567e-15
## GO:0007059 chromosome segregation         6.997217e-09      141 1.055918e-11
##
## $stats
##                                           stat.mean     exp1
## GO:0007156 homophilic cell adhesion        3.971899 3.971899
## GO:0060429 epithelium development          3.834595 3.834595
## GO:0007610 behavior                        3.557821 3.557821
## GO:0048729 tissue morphogenesis            3.498983 3.498983
## GO:0002009 morphogenesis of an epithelium  3.429317 3.429317
## GO:0016337 cell-cell adhesion              3.163057 3.163057
```

##Reactome Analysis Reactome is database consisting of biological molecules and their relation to pathways and processes.

```
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]
print(paste("Total number of significant genes:", length(sig_genes)))
```

```
## [1] "Total number of significant genes: 8149"
```

```
write.table(sig_genes, file="significant_genes.txt", row.names=FALSE, col.names=FALSE, quote=FALSE)
```

Online Analyisis: https://reactome.org/PathwayBrowser/#TOOL=AT

> What pathway has the most significant "Entities p-value"? Do the most significant pathways listed match your previous KEGG results? What factors could cause differences between the two methods?

Endosomal/Vacuolar pathway. No they do not match entirely. Some factors that may have differentiated our results is possibly how we set our p-value limit, how we initially cut down our data/exclusion criteria,

and the differences in updated information between the online source and our manually uploaded data (not intersecting with multiple updating data sets as in the case of the online resource).

For good practice:

```
#sessionInfo()
```