Pathway Analysis from RNA-Seq Results

Yash Garodia

24/02/2022

In this analysis, we check for coordinated differential expression over gene sets from KEGG pathways instead of changes of individual genes. The assumption here is that consistent perturbations over a given pathway (gene set) may suggest mechanistic changes.

There are many freely available tools for pathway or over-representation analysis. As of Nov 2017 Bioconductor alone has over 80 packages categorized under gene set enrichment and over 120 packages categorized under pathways.

Here we play with just one, the GAGE package (which stands for Generally Applicable Gene set Enrichment), to do KEGG pathway enrichment analysis on RNA-seq based differential expression results.

The KEGG pathway database, unlike GO for example, provides functional annotation as well as information about gene products that interact with each other in a given pathway, how they interact (e.g., activation, inhibition, etc.), and where they interact (e.g., cytoplasm, nucleus, etc.).

Section 1: Differential Expression Analysis

First we need to load the DESeq package for differential expression analysis:

```
library(DESeq2)
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## 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, which.max, which.min

##
## Attaching package: ’S4Vectors’

## The following object is masked from ’package:base’:
##
##     expand.grid

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Warning: package ’SummarizedExperiment’ was built under R version 4.0.2

## 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")’.

## Loading required package: DelayedArray

## Warning: package ’DelayedArray’ was built under R version 4.0.2

## Loading required package: matrixStats

## Warning: package ’matrixStats’ was built under R version 4.0.2

##
## Attaching package: ’matrixStats’

## The following objects are masked from ’package:Biobase’:
##
##     anyMissing, rowMedians

##
## Attaching package: ’DelayedArray’
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

```
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
```

Great! Now we can load our data files, which we have downloaded from the following publication: Trapnell C, Hendrickson DG, Sauvageau M, Goff L et al. "Differential analysis of gene regulation at transcript resolution with RNA-seq". Nat Biotechnol 2013 Jan;31(1):46-53. PMID: 23222703

```
metaFile <- "GSE37704_metadata.csv"
countFile <- "GSE37704_featurecounts.csv"

#Import metadata and take a peak

colData = read.csv(metaFile, row.names=1)
head(colData)
```

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

```
countData = read.csv(countFile, 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
```

Hmm… remember that we need the countData and colData files to match up so we will need to remove that odd first column in countData namely contData$length.

Q1) Complete the code below to remove the troublesome first column from countData

```
# Note we need to remove the odd first $length col
countData <- as.matrix(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
```

This looks better but there are lots of zero entries in there so let's get rid of them as we have no data for these.

> Q. Complete the code below to filter countData to exclude genes (i.e. rows) where we have 0 read count across all samples (i.e. columns). Tip: What will rowSums() of countData return and how could you use it in this context?

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

Running DESeq2

Nice now lets setup the DESeqDataSet object required for the DESeq() function and then run the DESeq pipeline.

```
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
```

Great! Now lets look at the model we’ve created:

```
dds
```

```
## class: DESeqDataSet
## dim: 15975 6
## metadata(1): version
## assays(4): counts mu H cooks
## rownames(15975): ENSG00000279457 ENSG00000187634 ... ENSG00000276345
##   ENSG00000271254
## rowData names(22): baseMean baseVar ... deviance maxCooks
## colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
## colData names(2): condition sizeFactor
```

Next, get results for the HoxA1 knockdown versus control siRNA (remember that these were labeled as “hoxa1_kd” and “control_sirna” in our original colData metaFile input to DESeq, you can check this above and by running resultsNames(dds) command).

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

> Q. Call the summary() function on your results to get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff.

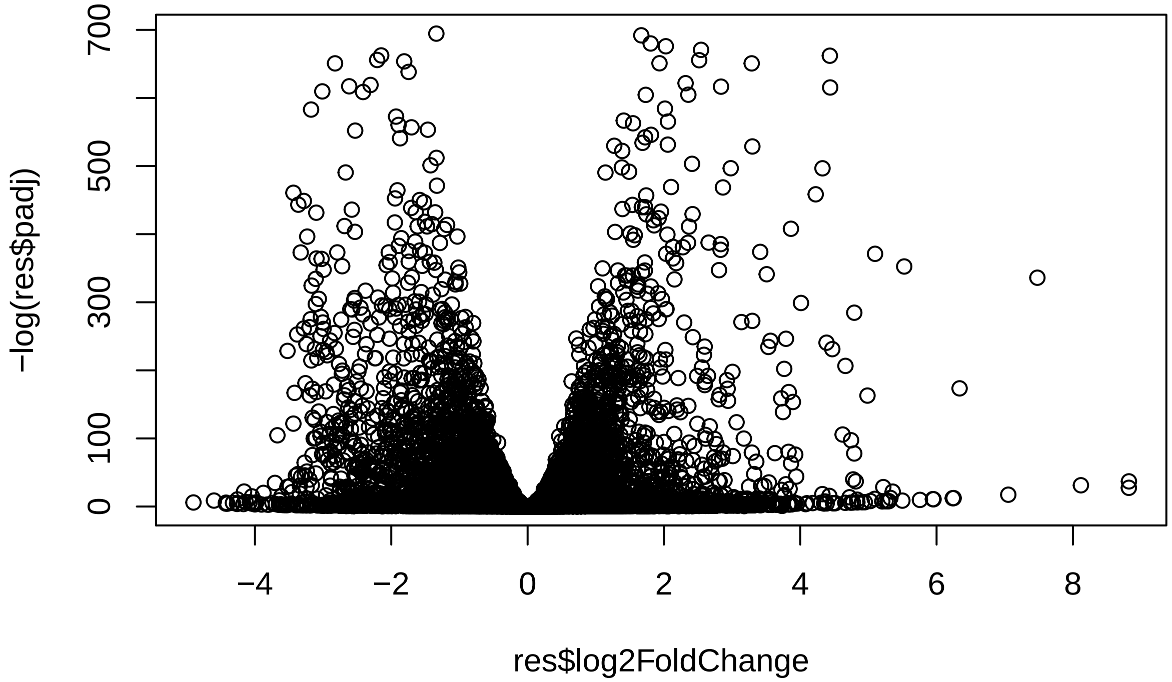
```
#summary of HoxA1 knockdown

summary(res)
```

```
##
## out of 15975 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4349, 27%
## LFC < 0 (down)     : 4396, 28%
## outliers [1]       : 0, 0%
## low counts [2]     : 1237, 7.7%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Volcano Plot

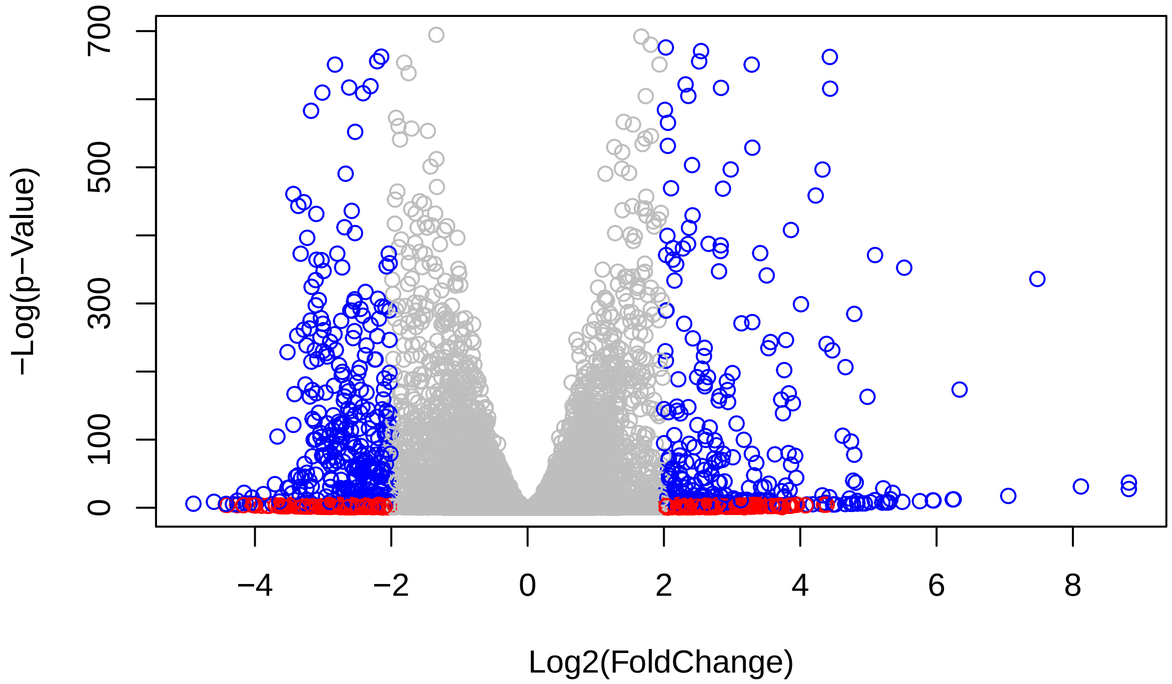
Now we will make a volcano plot, a commonly produced visualization from this type of data that we introduced last day. Basically it’s a plot of log2 fold change vs -log adjusted p-value.

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



Q) Improve this plot by completing the below code, which adds color and axis labels

```
# 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)
```



Gene Annotation

Since we mapped and counted against the Ensembl annotation, our results only have information about Ensembl gene IDs. However, our pathway analysis downstream will use KEGG pathways, and genes in KEGG pathways are annotated with Entrez gene IDs. So lets add them as we did the last day.

> Q. Use the mapIDs() function multiple times to add SYMBOL, ENTREZID and GENENAME annotation to our results by completing the code below.

In order to do this, we first need to activate the annotation and human database libraries:

```
library(AnnotationDbi)
```

```
## Warning: package 'AnnotationDbi' was built under R version 4.0.2
```

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

```
##
```

Next, we can look at all the columns included in the human database:

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

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

Now, to add symbol first:

```
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
```

Next, we add ENTREZID:

```
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
```

Finally, we add GENENAME:

```
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
```

Checking out our results:

```
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  29.913579      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634 183.229650      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.188076     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.637938      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.255123      0.0405765 0.2718928   0.149237 8.81366e-01
## ENSG00000187642   11.979750      0.5428105 0.5215598   1.040744 2.97994e-01
## ENSG00000188290  108.922128      2.0570638 0.1969053  10.446970 1.51282e-25
## ENSG00000187608  350.716868      0.2573837 0.1027266   2.505522 1.22271e-02
## ENSG00000188157 9128.439422      0.3899088 0.0467163   8.346304 7.04321e-17
## ENSG00000237330    0.158192      0.7859552 4.0804729   0.192614 8.47261e-01
##                        padj      symbol      entrez
##                   <numeric> <character> <character>
## ENSG00000279457 6.86555e-01          NA          NA
## ENSG00000187634 5.15718e-03      SAMD11      148398
## ENSG00000188976 1.76549e-35       NOC2L       26155
## ENSG00000187961 1.13413e-07      KLHL17      339451
## ENSG00000187583 9.19031e-01     PLEKHN1       84069
## ENSG00000187642 4.03379e-01       PERM1       84808
## ENSG00000188290 1.30538e-24        HES4       57801
## ENSG00000187608 2.37452e-02       ISG15        9636
## ENSG00000188157 4.21963e-16        AGRN      375790
## ENSG00000237330          NA      RNF223      401934
##                                                                   name
##                                                            <character>
## ENSG00000279457                                                     NA
## ENSG00000187634                 sterile alpha motif domain containing 11
## ENSG00000188976 NOC2 like nucleolar associated transcriptional repressor
## ENSG00000187961                              kelch like family member 17
## ENSG00000187583                 pleckstrin homology domain containing N1
## ENSG00000187642               PPARGC1 and ESRR induced regulator, muscle 1
## ENSG00000188290                  hes family bHLH transcription factor 4
## ENSG00000187608                            ISG15 ubiquitin like modifier
## ENSG00000188157                                                    agrin
## ENSG00000237330                                   ring finger protein 223
```

> Q. Finally for this section let's reorder these results by adjusted p-value and save them to a CSV file in your current project directory.

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

Great, this is looking good so far. Now lets see how pathway analysis can help us make further sense out of this ranked list of differentially expressed genes.

Section 2: Pathway Analysis

Here we are going to use the gage package for pathway analysis. Once we have a list of enriched pathways, we're going to use the pathview package to draw pathway diagrams, shading the molecules in the pathway by their degree of up/down-regulation.

First we need to do our one time install of these required bioconductor packages, which was done on R console:

```
# Run in your R console (i.e. not your Rmarkdown doc!)
#BiocManager::install( c("pathview", "gage", "gageData") )

# For old vesrsions of R only (R < 3.5.0)!
#source("http://bioconductor.org/biocLite.R")
#biocLite( c("pathview", "gage", "gageData") )
```

Now we can load packages as needed.

```
library(pathview)
```

```
## Warning: package ’pathview’ was built under R version 4.0.2
```

```
## ##############################################################################
## 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)
```

```
## Warning: package ’gage’ was built under R version 4.0.2
```

```
##
```

```
library(gageData)
```

```
data("kegg.sets.hs")
data("sigmet.idx.hs")

#Focus on signalling 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)andwehavethefoldchangeresul*

```
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, lets run the gage pathway analysis.

```
#Get the results

keggres = gage(foldchanges, gsets = kegg.sets.hs)
```

See help on the gage function with ?gage. Specifically, you might want to try changing the value of same.dir. This value determines whether to test for changes in a gene set toward a single direction (all genes up or down regulated) or changes towards both directions simultaneously (i.e. any genes in the pathway dysregulated). Here, we're using the default same.dir=TRUE, which will give us separate lists for pathways that are upregulated versus pathways that are down-regulated.

Now lets look at the object returned from gage().

```
attributes(keggres)
```

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

It is a list with three elements, “greater”, “less” and “stats”.

You can also see this in your Environmnet panel/tab window of RStudio or use the R command str(keggres).

Like any list we can use the dollar syntax to access a named element, e.g. head(keggres*greater*)*andhead*(*keggres*less).

Lets look at the first few down (less) pathway results:

```
head(keggres$less)
```

```
##                                           p.geomean stat.mean        p.val
## hsa04110 Cell cycle                    8.995727e-06 -4.378644 8.995727e-06
## hsa03030 DNA replication               9.424076e-05 -3.951803 9.424076e-05
## hsa03013 RNA transport                 1.375901e-03 -3.028500 1.375901e-03
## hsa03440 Homologous recombination      3.066756e-03 -2.852899 3.066756e-03
## hsa04114 Oocyte meiosis                3.784520e-03 -2.698128 3.784520e-03
## hsa00010 Glycolysis / Gluconeogenesis 8.961413e-03 -2.405398 8.961413e-03
##                                              q.val set.size         exp1
## hsa04110 Cell cycle                    0.001448312      121 8.995727e-06
## hsa03030 DNA replication               0.007586381       36 9.424076e-05
## hsa03013 RNA transport                 0.073840037      144 1.375901e-03
## hsa03440 Homologous recombination      0.121861535       28 3.066756e-03
## hsa04114 Oocyte meiosis                0.121861535      102 3.784520e-03
## hsa00010 Glycolysis / Gluconeogenesis 0.212222694       53 8.961413e-03
```

Each keggres*lessandkeggres*greater object is data matrix with gene sets as rows sorted by p-value.

The top “less/down” pathways is “Cell cycle” with the KEGG pathway identifier hsa04110.

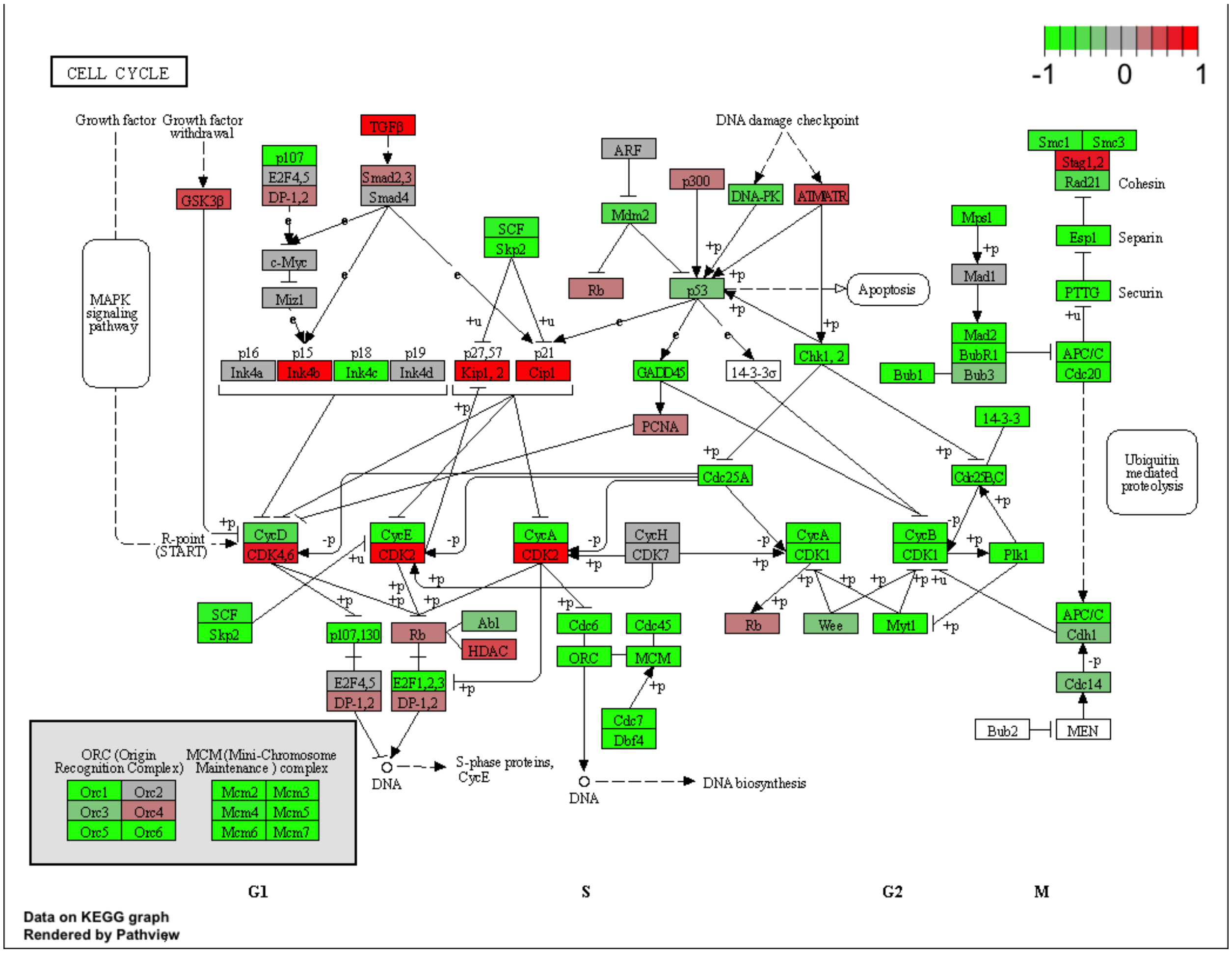
Now, let’s try out the pathview() function from the pathview package to make a pathway plot with our RNA-Seq expression results shown in color. To begin with lets manually supply a pathway.id (namely the first part of the “hsa04110 Cell cycle”) that we could see from the print out above.

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

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

```
## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09
```

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



Note how many of the genes in this pathway are perturbed (i.e. colored) in our results.

You can play with the other input arguments to pathview() to change the display in various ways including generating a PDF graph. For example:

```
# A different PDF based output of the same data
pathview(gene.data=foldchanges, pathway.id="hsa04110", kegg.native=FALSE)
```

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

```
## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09
```

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

Now, let's process our results a bit more to automagicaly pull out the top 5 upregulated pathways, then further process that just to get the pathway IDs needed by the pathview() function. We'll use these KEGG pathway IDs for pathview plotting below.

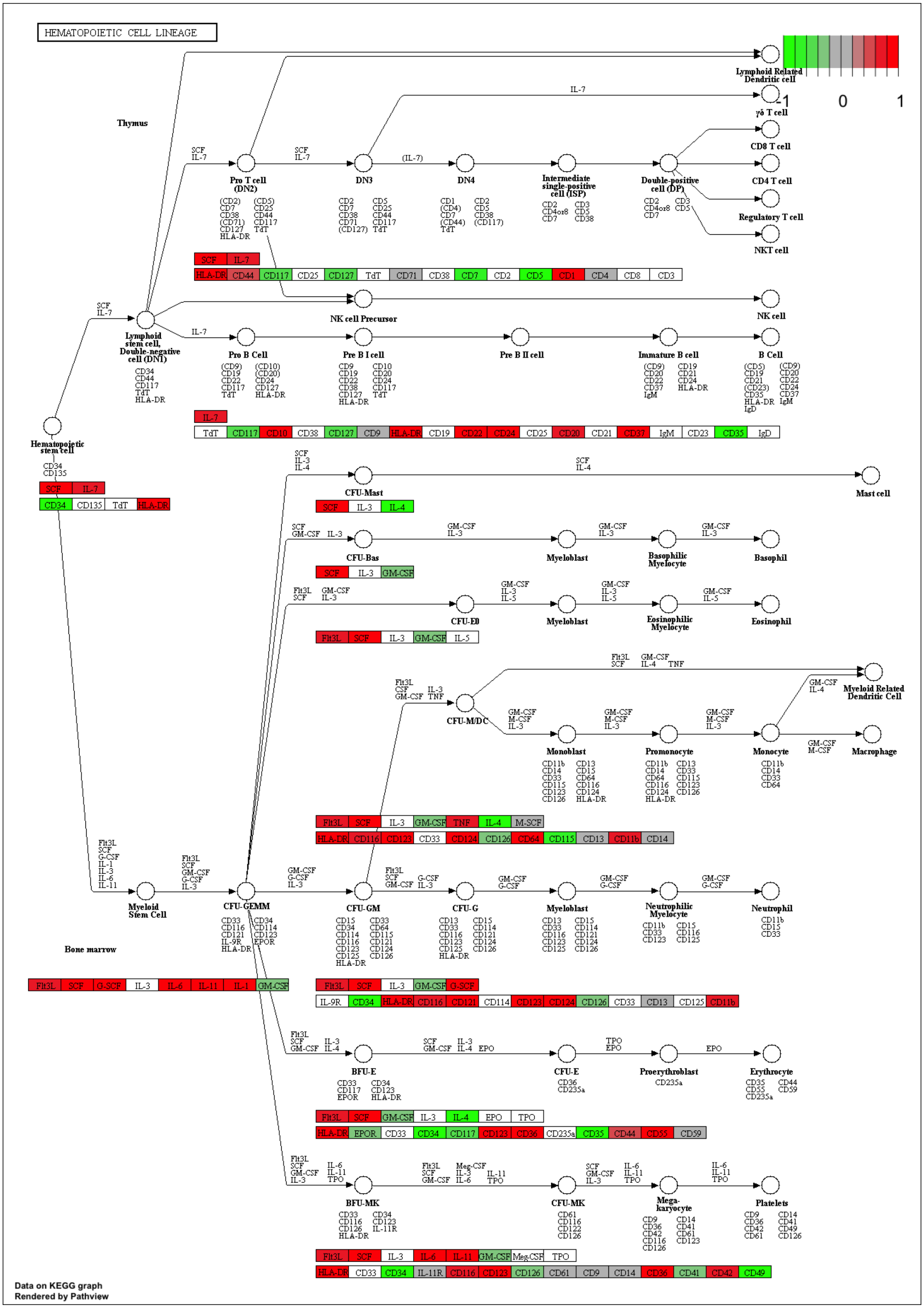
```
keggrespathways <- rownames(keggres$greater)[1:5]

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

```
## [1] "hsa04640" "hsa04630" "hsa00140" "hsa04142" "hsa04330"
```

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

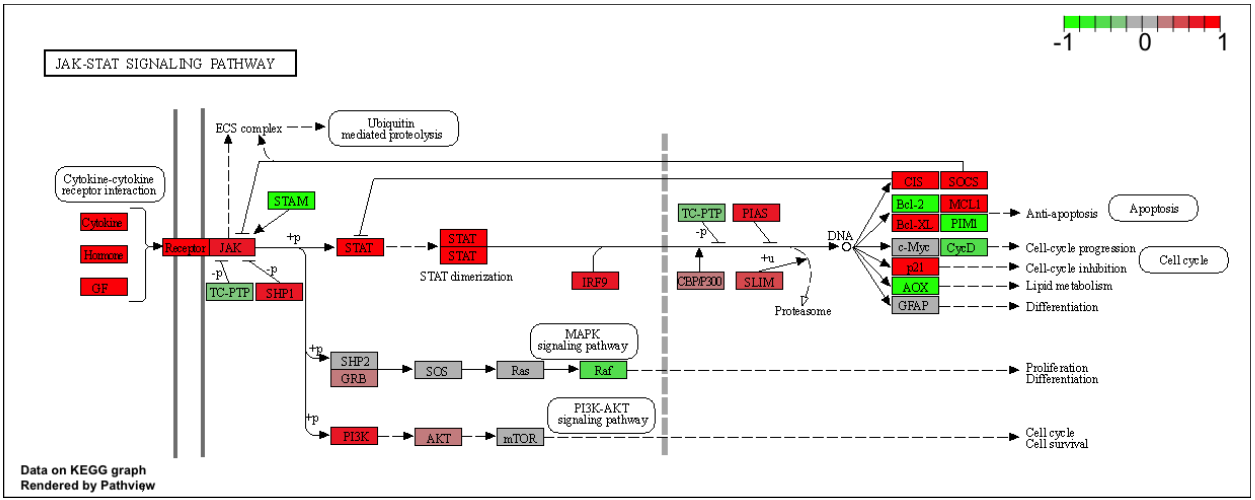
```
pathview(foldchanges, pathway.id = keggresids, species = "hsa")
```

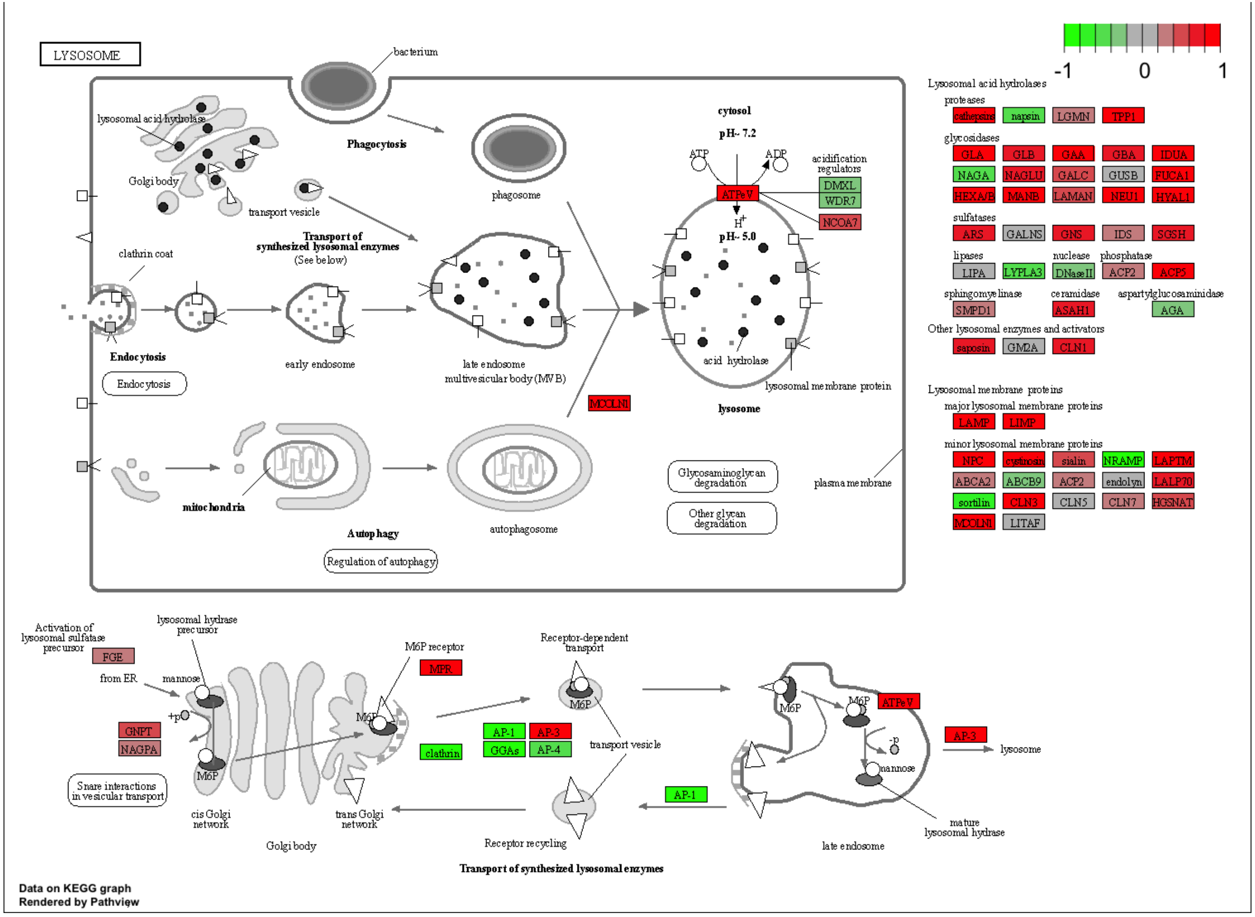
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

## 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/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

## Info: Writing image file hsa04330.pathview.png
```

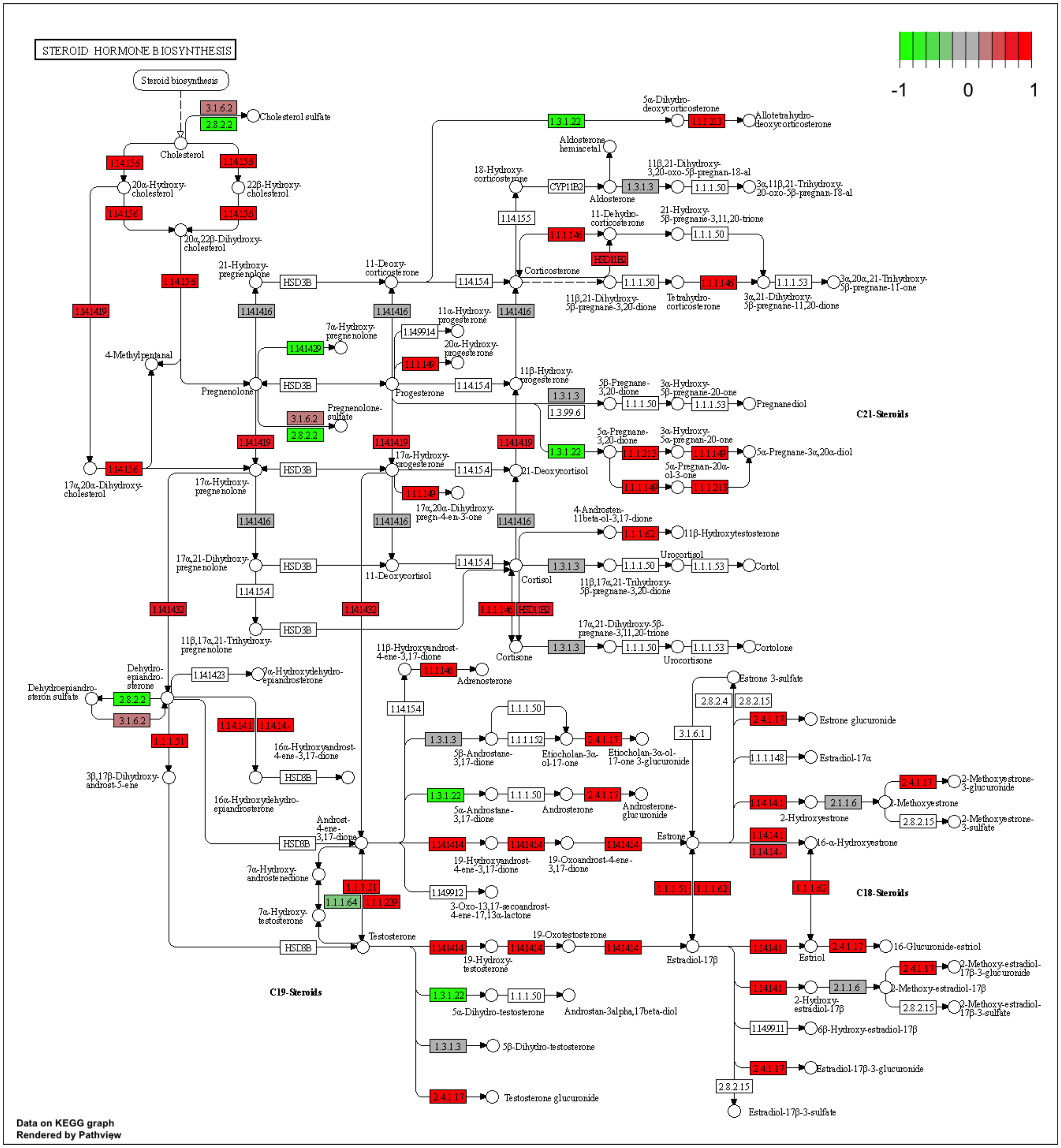


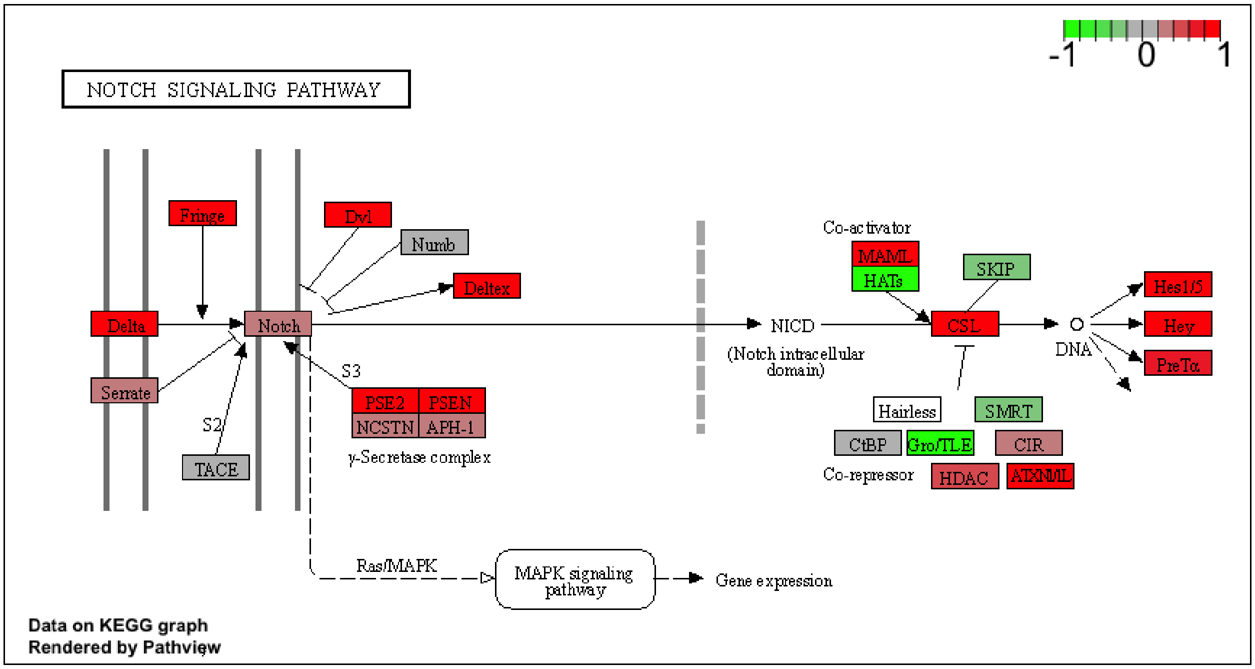
HEMATOPOIETIC CELL LINEAGE
-1
0
1
Thymus
Bone marrow
Hematopoietic stem cell
CD34 CD135
SCF IL-7
CD34 CD135 TdT HLA-DR
Lymphoid stem cell, Double-negative cell (DN1)
CD34 CD44 CD117 TdT HLA-DR
Pro T cell (DN2)
(CD2) CD7 CD38 (CD71) CD127 HLA-DR (CD5) CD25 CD44 CD117 TdT
DN3
CD2 CD7 CD38 CD71 (CD127) CD5 CD25 CD44 CD117 TdT
DN4
CD1 (CD4) CD7 (CD44) TdT CD2 CD5 CD38 (CD117)
Intermediate single-positive cell (ISP)
CD2 CD4or8 CD7 CD3 CD5 CD38
Double-positive cell (DP)
CD2 CD4or8 CD7 CD3 CD5
Lymphoid Related Dendritic cell
γδ T cell
CD8 T cell
CD4 T cell
Regulatory T cell
NKT cell
IL-7
(IL-7)
SCF IL-7
SCF IL-7 HLA-DR CD44 CD117 CD25 CD127 TdT CD71 CD38 CD7 CD2 CD5 CD1 CD4 CD8 CD3
NK cell Precursor
NK cell
Pro B Cell
(CD9) CD19 CD22 CD117 TdT (CD10) (CD20) CD24 CD127 HLA-DR
Pre B I cell
CD9 CD19 CD22 CD38 CD127 HLA-DR CD10 CD20 CD24 CD117 TdT
Pre B II cell
Immature B cell
(CD9) CD20 CD22 CD37 IgM CD19 CD21 CD24 HLA-DR
B Cell
(CD5) CD19 CD21 (CD23) CD35 HLA-DR IgD (CD9) CD20 CD22 CD24 CD37 IgM
IL-7
TdT CD117 CD10 CD38 CD127 CD9 HLA-DR CD19 CD22 CD24 CD25 CD20 CD21 CD37 IgM CD23 CD35 IgD
SCF IL-3 IL-4
CFU-Mast
Mast cell
SCF IL-3 IL-4
SCF GM-CSF IL-3
CFU-Bas
Myeloblast
Basophilic Myelocyte
Basophil
GM-CSF IL-3
SCF IL-3 GM-CSF
Flt3L SCF GM-CSF IL-3
CFU-E0
Myeloblast
Eosinophilic Myelocyte
Eosinophil
GM-CSF IL-3 IL-5
GM-CSF IL-5
Flt3L SCF IL-3 GM-CSF IL-5
Flt3L CSF GM-CSF IL-3 TNF
CFU-M/DC
Flt3L SCF GM-CSF IL-4 TNF
Myeloid Related Dendritic Cell
GM-CSF IL-4
GM-CSF M-CSF IL-3
Monoblast
CD11b CD14 CD33 CD115 CD123 CD126 CD13 CD15 CD64 CD116 CD124 HLA-DR
Promonocyte
CD11b CD14 CD64 CD116 CD124 HLA-DR CD13 CD33 CD115 CD123 CD126
Monocyte
CD11b CD14 CD33 CD64
GM-CSF M-CSF
Macrophage
Flt3L SCF IL-3 GM-CSF TNF IL-4 M-SCF
HLA-DR CD116 CD123 CD33 CD124 CD126 CD64 CD115 CD13 CD11b CD14
Flt3L SCF G-CSF IL-1 IL-3 IL-6 IL-11
Myeloid Stem Cell
Flt3L SCF GM-CSF G-CSF IL-3
CFU-GEMM
CD33 CD116 CD121 IL-9R HLA-DR CD34 CD114 CD123 EPOR
Flt3L SCF G-SCF IL-3 IL-6 IL-11 IL-1 GM-CSF
GM-CSF G-CSF IL-3
CFU-GM
CD15 CD34 CD114 CD116 CD123 CD125 HLA-DR CD33 CD64 CD115 CD121 CD124 CD126
Flt3L SCF GM-CSF G-CSF IL-3
CFU-G
CD13 CD33 CD116 CD123 CD125 HLA-DR CD15 CD114 CD121 CD124 CD126
GM-CSF G-CSF
Myeloblast
CD13 CD33 CD116 CD123 CD125 CD15 CD114 CD121 CD124 CD126
Neutrophilic Myelocyte
CD11b CD33 CD123 CD15 CD116 CD125
Neutrophil
CD11b CD15 CD33
Flt3L SCF IL-3 GM-CSF G-SCF
IL-9R CD34 HLA-DR CD116 CD121 CD114 CD123 CD124 CD126 CD33 CD13 CD125 CD11b
Flt3L SCF GM-CSF IL-3 IL-4
BFU-E
CD33 CD117 EPOR CD34 CD123 HLA-DR
SCF GM-CSF IL-3 IL-4 EPO
CFU-E
CD36 CD235a
TPO EPO
Proerythroblast
CD235a
EPO
Erythrocyte
CD35 CD55 CD235a CD44 CD59
Flt3L SCF GM-CSF IL-3 IL-4 EPO TPO
HLA-DR EPOR CD33 CD34 CD117 CD123 CD36 CD235a CD35 CD44 CD55 CD59
Flt3L SCF GM-CSF IL-3 IL-6 IL-11 TPO
BFU-MK
CD33 CD116 CD126 HLA-DR CD34 CD123 IL-11R
Flt3L SCF GM-CSF Meg-CSF IL-3 IL-6 IL-11 TPO
CFU-MK
CD61 CD116 CD122 CD126
SCF GM-CSF IL-3 IL-6 IL-11 TPO
Mega-karyocyte
CD9 CD36 CD42 CD116 CD126 CD14 CD41 CD61 CD123
IL-6 IL-11 TPO
Platelets
CD9 CD36 CD42 CD61 CD14 CD41 CD49 CD126
Flt3L SCF IL-3 IL-6 IL-11 GM-CSF Meg-CSF TPO
HLA-DR CD33 CD34 IL-11R CD116 CD123 CD126 CD61 CD9 CD14 CD36 CD41 CD42 CD49

Data on KEGG graph
Rendered by Pathview


-1
0
1
JAK-STAT SIGNALING PATHWAY
ECS complex
Ubiquitin mediated proteolysis
Cytokine-cytokine receptor interaction
Cytokine
Hormone
GF
Receptor
JAK
STAM
+p
-p
TC-PTP
SHP1
STAT
STAT
STAT
STAT dimerization
IRF9
TC-PTP
PIAS
CBP/P300
SLIM
+u
Proteasome
DNA
CIS
SOCS
Bcl-2
MCL1
Bcl-XL
PIM1
c-Myc
CycD
p21
AOX
GFAP
Anti-apoptosis
Apoptosis
Cell-cycle progression
Cell-cycle inhibition
Cell cycle
Lipid metabolism
Differentiation
MAPK signaling pathway
SHP2
GRB
SOS
Ras
Raf
Proliferation
Differentiation
PI3K-AKT signaling pathway
PI3K
AKT
mTOR
Cell cycle
Cell survival
Data on KEGG graph
Rendered by Pathview


LYSOSOME
-1
0
1
bacterium
lysosomal acid hydrolase
Golgi body
Phagocytosis
phagosome
transport vesicle
Transport of synthesized lysosomal enzymes
(See below)
clathrin coat
Endocytosis
Endocytosis
early endosome
late endosome
multivesicular body (MVB)
cytosol
pH~ 7.2
ATP
ADP
ATPeV
acidification regulators
DMXL
WDR7
NCOA7
H+
pH~ 5.0
acid hydrolase
lysosomal membrane protein
lysosome
MCOLN1
mitochondria
Autophagy
autophagosome
Regulation of autophagy
Glycosaminoglycan degradation
Other glycan degradation
plasma membrane
Lysosomal acid hydrolases
proteases
cathepsins
napsin
LGMN
TPP1
glycosidases
GLA
GLB
GAA
GBA
IDUA
NAGA
NAGLU
GALC
GUSB
FUCA1
HEXA/B
MANB
LAMAN
NEU1
HYAL1
sulfatases
ARS
GALNS
GNS
IDS
SGSH
lipases
nuclease
phosphatase
LIPA
LYPLA3
DNaseII
ACP2
ACP5
sphingomyelinase
ceramidase
aspartylglucosaminidase
SMPD1
ASAH1
AGA
Other lysosomal enzymes and activators
saposin
GM2A
CLN1
Lysosomal membrane proteins
major lysosomal membrane proteins
LAMP
LIMP
minor lysosomal membrane proteins
NPC
cystinosin
sialin
NRAMP
LAPTM
ABCA2
ABCB9
ACP2
endolyn
LALP70
sortilin
CLN3
CLN5
CLN7
HGSNAT
MCOLN1
LITAF
Activation of lysosomal sulfatase precursor
FGE
lysosomal hydrase precursor
from ER
mannose
+p
GNPT
NAGPA
M6P
Snare interactions in vesicular transport
cis Golgi network
Golgi body
M6P receptor
MPR
M6P
trans Golgi network
clathrin
AP-1
AP-3
GGAs
AP-4
Receptor-dependent transport
M6P
transport vesicle
AP-1
Receptor recycling
M6P
M6P
ATPeV
-p
mannose
AP-3
lysosome
late endosome
mature lysosomal hydrase
Transport of synthesized lysosomal enzymes
Data on KEGG graph
Rendered by Pathview


STEROID HORMONE BIOSYNTHESIS
-1
0
1
Steroid biosynthesis
3.1.6.2
2.8.2.2
Cholesterol sulfate
Cholesterol
1.14.15.6
20α-Hydroxy-cholesterol
22β-Hydroxy-cholesterol
20α,22β-Dihydroxy-cholesterol
1.14.14.19
4-Methylpentanal
17α,20α-Dihydroxy-cholesterol
21-Hydroxy-pregnenolone
HSD3B
1.14.14.16
1.14.14.29
7α-Hydroxy-pregnenolone
Pregnenolone
Pregnenolone-sulfate
17α-Hydroxy-pregnenolone
17α,21-Dihydroxy-pregnenolone
1.14.15.4
11β,17α,21-Trihydroxy-pregnenolone
1.14.14.32
Dehydroepiandro-steron sulfate
Dehydro-epiandro-sterone
1.14.14.23
7α-Hydroxydehydro-epiandrosterone
1.14.14.1
1.14.14.-
16α-Hydroxyandrost-4-ene-3,17-dione
1.1.1.51
3β,17β-Dihydroxy-androst-5-ene
16α-Hydroxydehydro-epiandrosterone
Androst-4-ene-3,17-dione
7α-Hydroxy-androstenedione
1.1.1.64
1.1.1.239
7α-Hydroxy-testosterone
Testosterone
C19-Steroids
11-Deoxy-corticosterone
1.3.1.22
5α-Dihydro-deoxycorticosterone
1.1.1.213
Allotetrahydro-deoxycorticosterone
Aldosterone hemiacetal
18-Hydroxy-corticosterone
CYP11B2
Aldosterone
3.1.3
11β,21-Dihydroxy-3,20-oxo-5β-pregnan-18-al
1.1.1.50
3α,11β,21-Trihydroxy-20-oxo-5β-pregnan-18-al
1.14.15.5
1.1.1.146
11-Dehydro-corticosterone
HSD11B2
21-Hydroxy-5β-pregnane-3,11,20-trione
Corticosterone
11β,21-Dihydroxy-5β-pregnane-3,20-dione
Tetrahydro-corticosterone
1.1.1.146
3α,21-Dihydroxy-5β-pregnane-11,20-dione
1.1.1.53
3α,20α,21-Trihydroxy-5β-pregnane-11-one
11α-Hydroxy-progesterone
1.14.99.14
20α-Hydroxy-progesterone
1.1.1.149
11β-Hydroxy-progesterone
Progesterone
5β-Pregnane-3,20-dione
1.3.1.3
1.3.99.6
3α-Hydroxy-5β-pregnane-20-one
Pregnanediol
C21-Steroids
5α-Pregnane-3,20-dione
3α-Hydroxy-5α-pregnan-20-one
5α-Pregnane-3α,20α-diol
5α-Pregnan-20α-ol-3-one
17α-Hydroxy-progesterone
21-Deoxycortisol
17α,20α-Dihydroxy-pregn-4-en-3-one
4-Androsten-11beta-ol-3,17-dione
1.1.1.62
11β-Hydroxytestosterone
11-Deoxycortisol
Cortisol
Urocortisol
Cortol
11β,17α,21-Trihydroxy-5β-pregnane-3,20-dione
17α,21-Dihydroxy-5β-pregnane-3,11,20-trione
Cortisone
Urocortisone
Cortolone
11β-Hydroxyandrost-4-ene-3,17-dione
Adrenosterone
Estrone 3-sulfate
2.8.2.4
2.8.2.15
3.1.6.1
5β-Androstane-3,17-dione
1.1.1.152
Etiocholan-3α-ol-17-one
2.4.1.17
Etiocholan-3α-ol-17-one 3-glucuronide
5α-Androstane-3,17-dione
Androsterone
Androsterone-glucuronide
Estrone
Estrone glucuronide
1.1.1.148
Estradiol-17α
2-Methoxyestrone-3-glucuronide
2.1.1.6
2-Methoxyestrone
2-Hydroxyestrone
2-Methoxyestrone-3-sulfate
19-Hydroxyandrost-4-ene-3,17-dione
19-Oxoandrost-4-ene-3,17-dione
16-α-Hydroxyestrone
1.14.99.12
3-Oxo-13,17-secoandrost-4-ene-17,13α-lactone
C18-Steroids
19-Hydroxy-testosterone
19-Oxotestosterone
Estradiol-17β
Estriol
16-Glucuronide-estriol
2-Methoxy-estradiol-17β-3-glucuronide
2-Hydroxy-estradiol-17β
2-Methoxy-estradiol-17β
2-Methoxy-estradiol-17β-3-sulfate
5α-Dihydro-testosterone
Androstan-3alpha,17beta-diol
1.14.99.11
6β-Hydroxy-estradiol-17β
5β-Dihydro-testosterone
Estradiol-17β-3-glucuronide
Testosterone glucuronide
Estradiol-17β-3-sulfate
Data on KEGG graph
Rendered by Pathview



Q) Can you do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways?

In order to do this, lets put the top 5 down regulated pathways in a different variable:

```
keggresdown <- row.names(keggres$less)[1:5]
keggresdown
```

```
## [1] "hsa04110 Cell cycle"              "hsa03030 DNA replication"
## [3] "hsa03013 RNA transport"           "hsa03440 Homologous recombination"
## [5] "hsa04114 Oocyte meiosis"
```

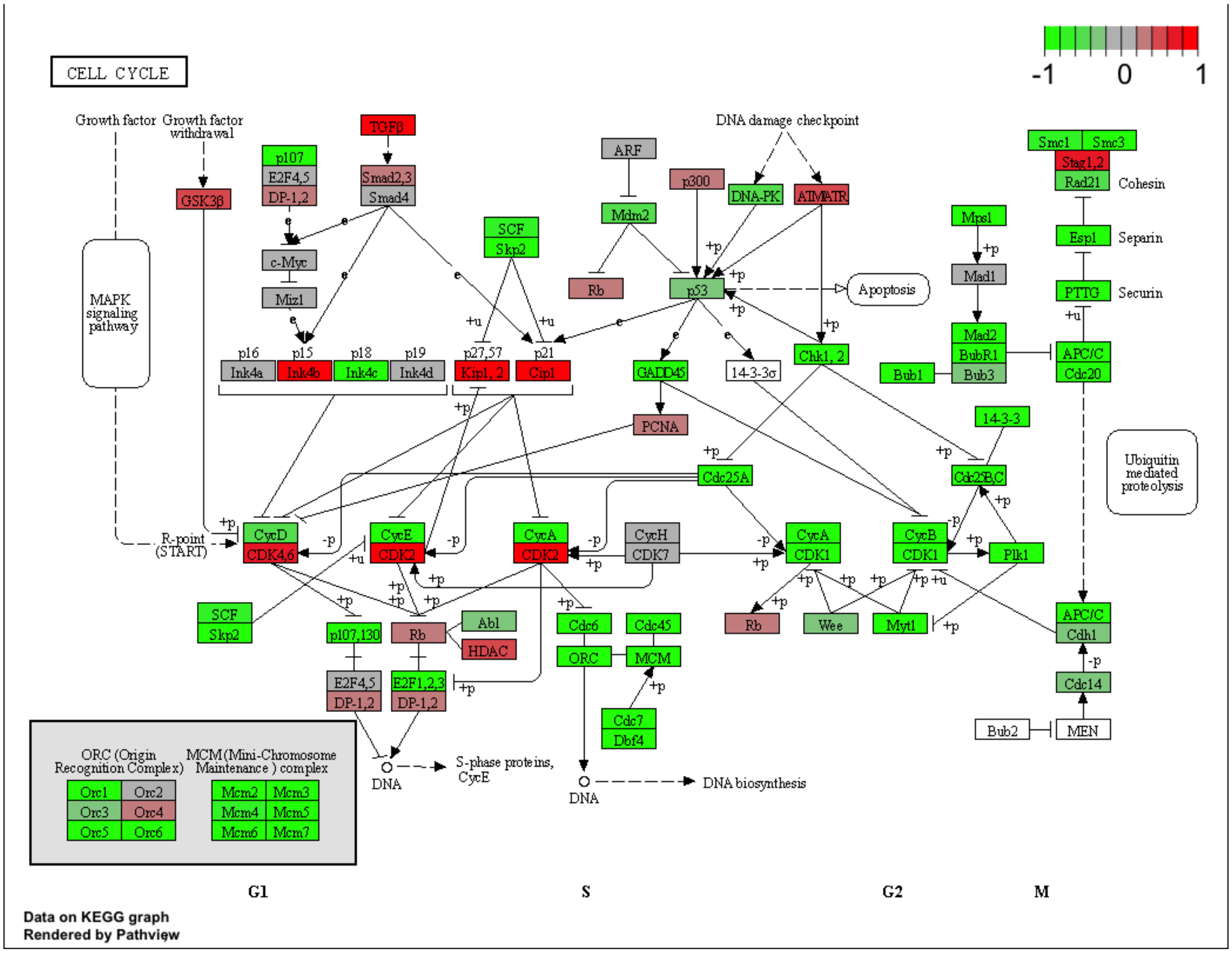
```
#Extract the 8 character long IDs part of each string
keggresid_down <- substr(keggresdown, start = 1, stop = 8)
keggresid_down
```

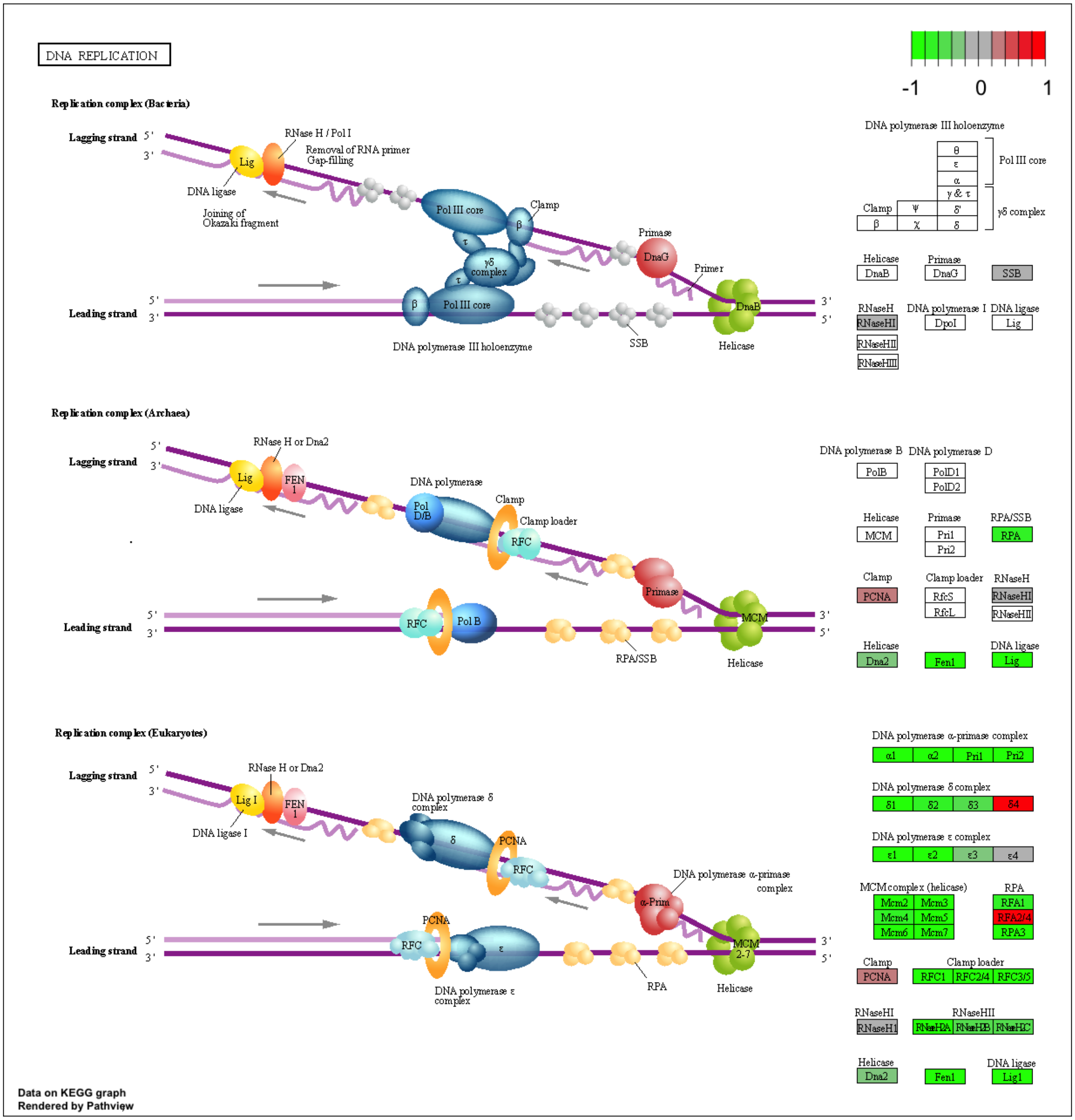
```
## [1] "hsa04110" "hsa03030" "hsa03013" "hsa03440" "hsa04114"
```

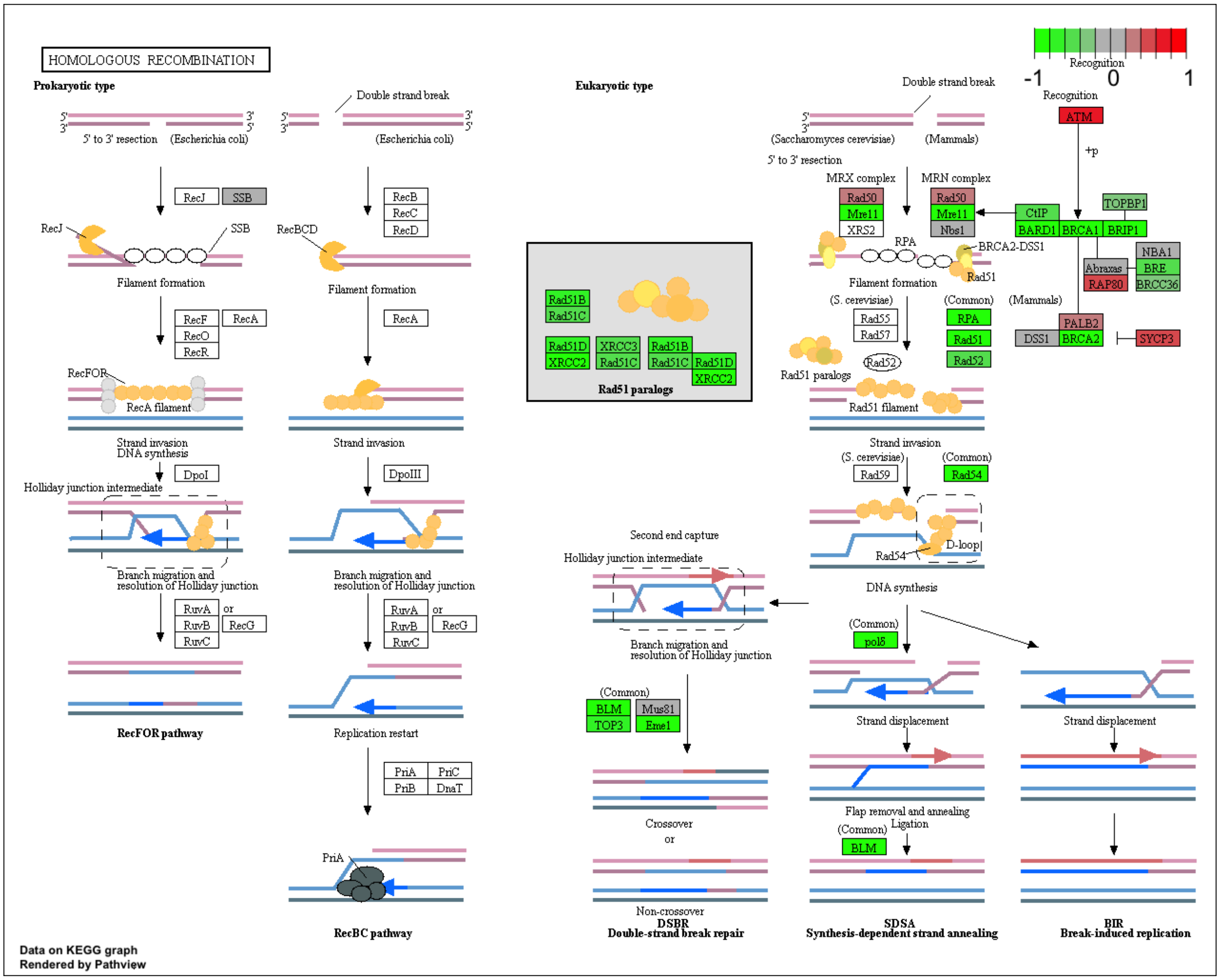
Now, to get the pathways of each of them:

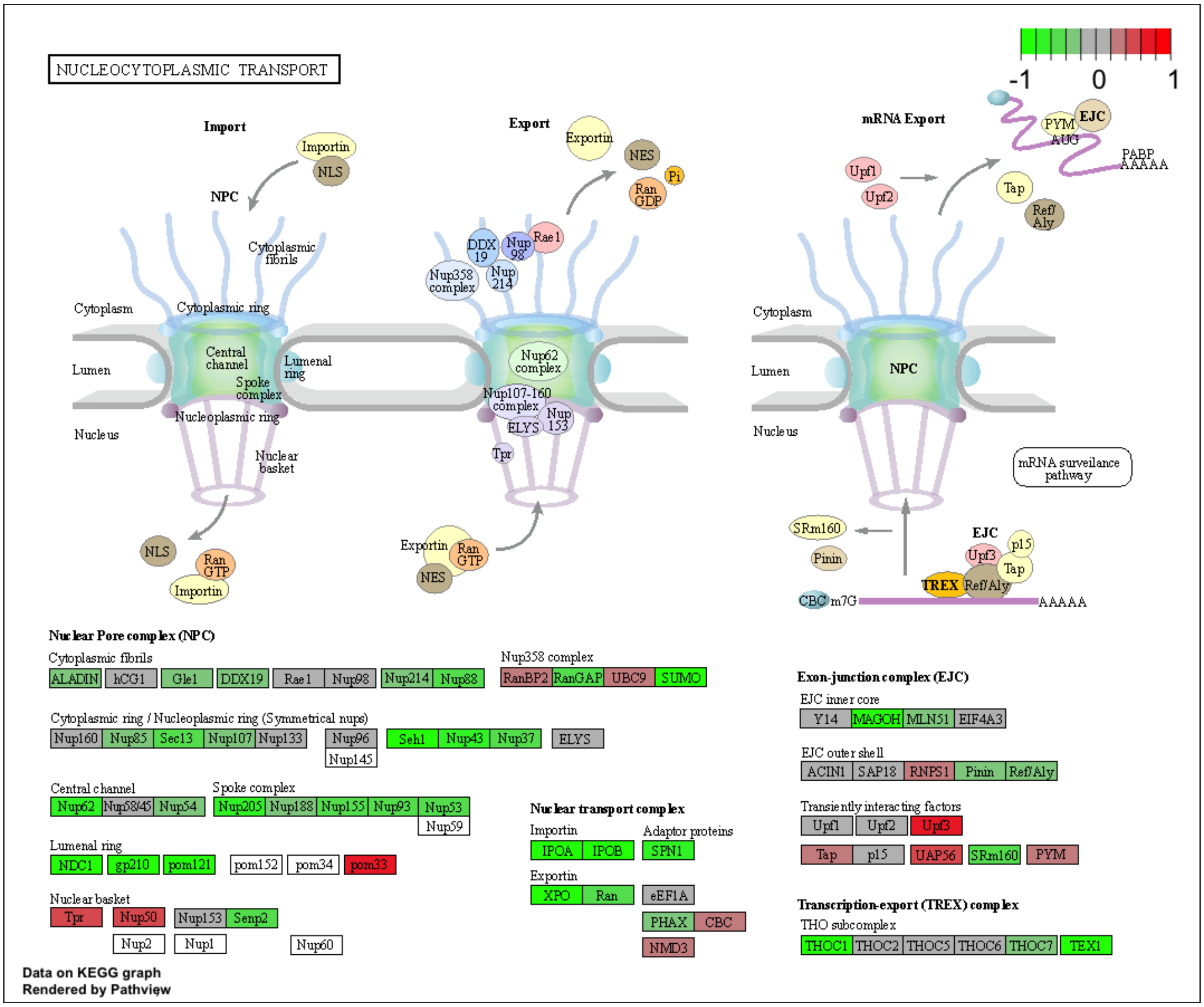
```
pathview(foldchanges, pathway.id = keggresid_down, species = "hsa")
```

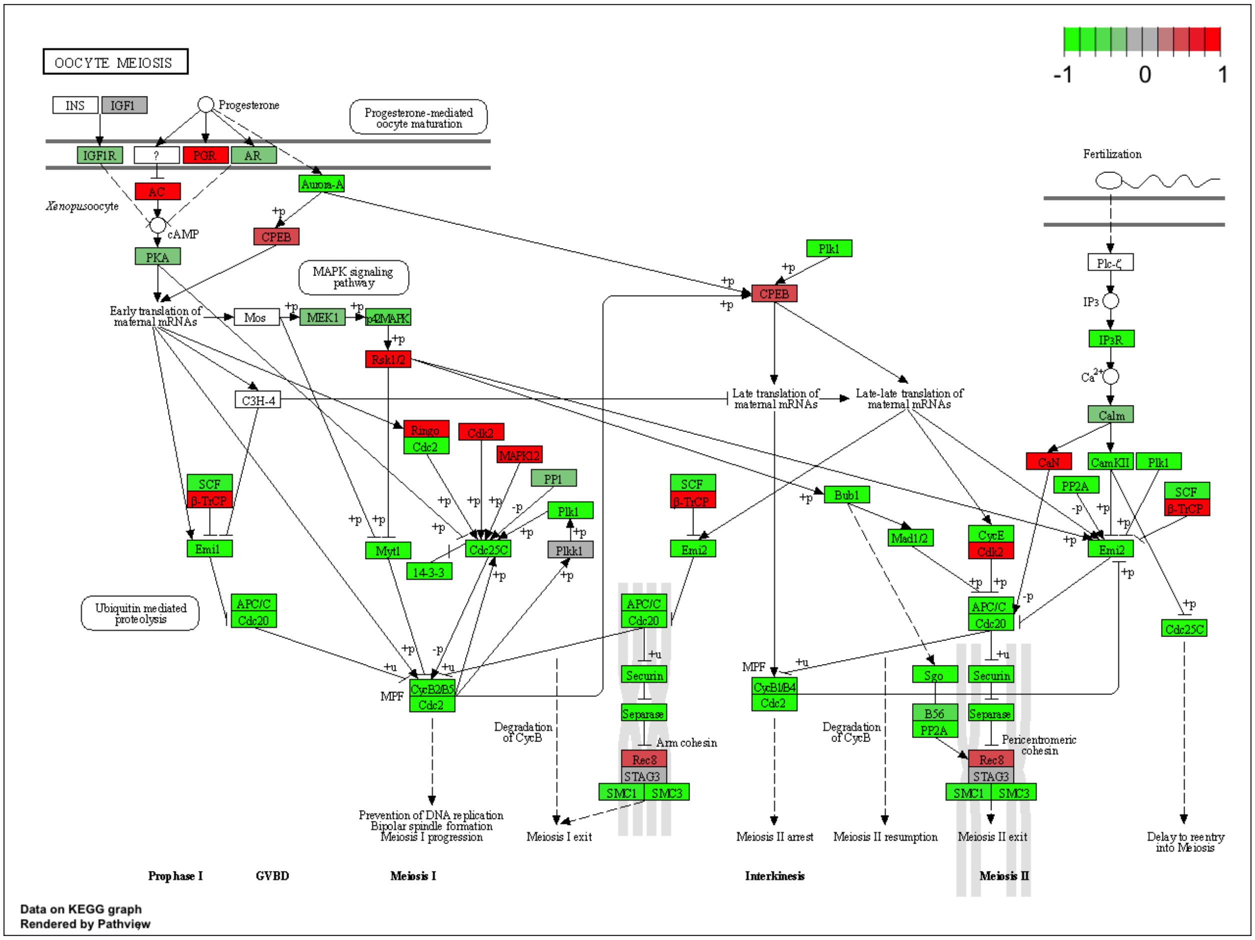
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

## 'select()' returned 1:1 mapping between keys and columns
## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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

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

## Info: Working in directory /Users/yashgarodia/Desktop/UCSD_Coursework/BIMM 143/week 09

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




DNA REPLICATION
-1
0
1
Replication complex (Bacteria)
Lagging strand
5'
3'
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
τ
Primase
DnaG
Primer
Leading strand
5'
3'
β
Pol III core
DnaB
3'
5'
DNA polymerase III holoenzyme
SSB
Helicase
DNA polymerase III holoenzyme
θ
ε
α
γ & τ
Pol III core
Clamp
ψ
δ'
β
χ
δ
γδ complex
Helicase
DnaB
Primase
DnaG
SSB
RNaseH
RNaseHI
RNaseHII
RNaseHIII
DNA polymerase I
DpoI
DNA ligase
Lig
Replication complex (Archaea)
Lagging strand
5'
3'
RNase H or Dna2
Lig
FEN 1
DNA ligase
DNA polymerase
Pol D/B
Clamp
Clamp loader
RFC
Primase
MCM
3'
5'
Leading strand
5'
3'
RFC
Pol B
RPA/SSB
Helicase
DNA polymerase B
PolB
DNA polymerase D
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)
Lagging strand
5'
3'
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
3'
5'
Leading strand
5'
3'
PCNA
RFC
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
RPA3
Clamp
PCNA
Clamp loader
RFC1
RFC2/4
RFC3/5
RNaseHI
RNaseH1
RNaseHII
RNaseH2A
RNaseH2B
RNaseH2C
Helicase
Dna2
Fen1
DNA ligase
Lig1
Data on KEGG graph
Rendered by Pathview


HOMOLOGOUS RECOMBINATION
Prokaryotic type
Eukaryotic type
Recognition
-1
0
1
5'
3'
5' to 3' resection
(Escherichia coli)
Double strand break
RecJ
SSB
RecBCD
RecB
RecC
RecD
Filament formation
RecF
RecO
RecR
RecA
RecFOR
RecA filament
Strand invasion
DNA synthesis
DpoI
DpoIII
Holliday junction intermediate
Branch migration and
resolution of Holliday junction
RuvA
RuvB
RuvC
or
RecG
RecFOR pathway
Replication restart
PriA
PriB
PriC
DnaT
RecBC pathway
Rad51B
Rad51C
Rad51D
XRCC2
XRCC3
Rad51 paralogs
(Saccharomyces cerevisiae)
(Mammals)
MRX complex
MRN complex
Rad50
Mre11
XRS2
Nbs1
CtIP
ATM
+p
TOPBP1
BARD1
BRCA1
BRIP1
NBA1
Abraxas
BRE
RAP80
BRCC36
RPA
BRCA2-DSS1
Rad51
(S. cerevisiae)
(Common)
Rad55
Rad57
Rad52
PALB2
DSS1
BRCA2
SYCP3
Rad51 filament
Rad59
Rad54
D-loop
DNA synthesis
Second end capture
polδ
BLM
TOP3
Mus81
Eme1
Crossover
Non-crossover
DSBR
Double-strand break repair
Strand displacement
Flap removal and annealing
Ligation
SDSA
Synthesis-dependent strand annealing
BIR
Break-induced replication
Data on KEGG graph
Rendered by Pathview


NUCLEOCYTOPLASMIC TRANSPORT
-1
0
1
Import
Importin
NLS
NPC
Cytoplasmic fibrils
Cytoplasm
Cytoplasmic ring
Lumen
Central channel
Lumenal ring
Spoke complex
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
Nup 153
ELYS
Tpr
Exportin
Ran GTP
NES
mRNA Export
PYM
EJC
AUG
PABP AAAAA
Upf1
Upf2
Tap
Ref/ Aly
Cytoplasm
Lumen
NPC
Nucleus
mRNA surveilance pathway
SRm160
Pinin
EJC
p15
Upf3
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
IPOA
IPOB
Adaptor proteins
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



Section 3: 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       8.519724e-05  3.824205 8.519724e-05
## GO:0002009 morphogenesis of an epithelium 1.396681e-04  3.653886 1.396681e-04
## GO:0048729 tissue morphogenesis           1.432451e-04  3.643242 1.432451e-04
## GO:0007610 behavior                       2.195494e-04  3.530241 2.195494e-04
## GO:0060562 epithelial tube morphogenesis  5.932837e-04  3.261376 5.932837e-04
## GO:0035295 tube development               5.953254e-04  3.253665 5.953254e-04
##                                                q.val set.size         exp1
## GO:0007156 homophilic cell adhesion       0.1951475      113 8.519724e-05
## GO:0002009 morphogenesis of an epithelium 0.1951475      339 1.396681e-04
## GO:0048729 tissue morphogenesis           0.1951475      424 1.432451e-04
## GO:0007610 behavior                       0.2243246      427 2.195494e-04
## GO:0060562 epithelial tube morphogenesis  0.3710482      257 5.932837e-04
## GO:0035295 tube development               0.3710482      391 5.953254e-04
##
## $less
##                                              p.geomean stat.mean        p.val
## GO:0048285 organelle fission            1.536227e-15 -8.063910 1.536227e-15
## GO:0000280 nuclear division             4.286961e-15 -7.939217 4.286961e-15
## GO:0007067 mitosis                      4.286961e-15 -7.939217 4.286961e-15
## GO:0000087 M phase of mitotic cell cycle 1.169934e-14 -7.797496 1.169934e-14
## GO:0007059 chromosome segregation       2.028624e-11 -6.878340 2.028624e-11
## GO:0000236 mitotic prometaphase         1.729553e-10 -6.695966 1.729553e-10
##                                                  q.val set.size         exp1
## GO:0048285 organelle fission            5.840269e-12      376 1.536227e-15
## GO:0000280 nuclear division             5.840269e-12      352 4.286961e-15
## GO:0007067 mitosis                      5.840269e-12      352 4.286961e-15
## GO:0000087 M phase of mitotic cell cycle 1.195380e-11      362 1.169934e-14
## GO:0007059 chromosome segregation       1.658197e-08      142 2.028624e-11
## GO:0000236 mitotic prometaphase         1.178114e-07       84 1.729553e-10
##
## $stats
##                                            stat.mean     exp1
## GO:0007156 homophilic cell adhesion        3.824205 3.824205
## GO:0002009 morphogenesis of an epithelium  3.653886 3.653886
## GO:0048729 tissue morphogenesis            3.643242 3.643242
## GO:0007610 behavior                        3.530241 3.530241
## GO:0060562 epithelial tube morphogenesis   3.261376 3.261376
## GO:0035295 tube development                3.253665 3.253665
```

4. Reactome Analysis

Reactome is database consisting of biological molecules and their relation to pathways and processes. Reactome, such as many other tools, has an online software available (https://reactome.org/) and R package available (https://bioconductor.org/packages/release/bioc/html/ReactomePA.html).

Let's now conduct over-representation enrichment analysis and pathway-topology analysis with Reactome using the previous list of significant genes generated from our differential expression results above.

First, Using R, output the list of significant genes at the 0.05 level as a plain text file:

```r
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: 8147"
```

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

Then, to perform pathway analysis online go to the Reactome website (https://reactome.org/PathwayBrowser/#TOOL=AT). Select "choose file" to upload your significant gene list. Then, select the parameters "Project to Humans", then click "Analyze".

> Q. 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 has the most significant Entities p-value. This doesn't match with my KEGG results, which suggested that the most significant pathway has to do with cell cycle. Reactome and KEGG have the same level of gene coverage, but Reactome stores much more gene data and obtains information from numerous databases, whereas KEGG is more generic and relies on the KEGG database only. This can potentially cause differences between the two pathways.

Section 5. GO online (OPTIONAL)

> Q: 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?

The pathway for the detection of chemical stimulus involved in sensory perception had the most significant p-value in GO. This doesn't match the KEGG results. The differences can possibly arise from the fact that while KEGG uses a pathway database to determine significant gene pathways, GO uses a gene ontology databases. GO is more focused on complexities of complex functional relationships between genes, and hence takes into account different factors relative to KEGG.

Session Information

The sessionInfo() prints version information about R and any attached packages. It's a good practice to always run this command at the end of your R session and record it for the sake of reproducibility in the future.

```
sessionInfo()
```

```
## R version 4.0.0 (2020-04-24)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] gageData_2.26.0             gage_2.38.3
##  [3] pathview_1.28.1             org.Hs.eg.db_3.11.4
##  [5] AnnotationDbi_1.50.3        DESeq2_1.28.1
##  [7] SummarizedExperiment_1.18.2 DelayedArray_0.14.1
##  [9] matrixStats_0.61.0          Biobase_2.48.0
## [11] GenomicRanges_1.40.0        GenomeInfoDb_1.24.2
## [13] IRanges_2.22.2              S4Vectors_0.26.1
## [15] BiocGenerics_0.34.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2             bit64_4.0.5            splines_4.0.0
##  [4] assertthat_0.2.1       highr_0.9              blob_1.2.2
##  [7] GenomeInfoDbData_1.2.3 yaml_2.2.1             pillar_1.6.3
## [10] RSQLite_2.2.8          lattice_0.20-45        glue_1.6.1
## [13] digest_0.6.28          RColorBrewer_1.1-2     XVector_0.28.0
## [16] colorspace_2.0-2       htmltools_0.5.2        Matrix_1.3-4
## [19] XML_3.99-0.8           pkgconfig_2.0.3        genefilter_1.70.0
## [22] zlibbioc_1.34.0        GO.db_3.11.4           purrr_0.3.4
## [25] xtable_1.8-4           scales_1.1.1           BiocParallel_1.22.0
## [28] tibble_3.1.5           annotate_1.66.0        KEGGREST_1.28.0
## [31] generics_0.1.0         ggplot2_3.3.5          ellipsis_0.3.2
## [34] cachem_1.0.6           survival_3.2-13        magrittr_2.0.1
## [37] crayon_1.4.1           memoise_2.0.0          evaluate_0.14
## [40] KEGGgraph_1.48.0       fansi_0.5.0            graph_1.66.0
## [43] tools_4.0.0            lifecycle_1.0.1        stringr_1.4.0
## [46] munsell_0.5.0          locfit_1.5-9.4         Biostrings_2.56.0
## [49] compiler_4.0.0         rlang_0.4.11           grid_4.0.0
## [52] RCurl_1.98-1.5         bitops_1.0-7           rmarkdown_2.11
## [55] gtable_0.3.0           DBI_1.1.1              R6_2.5.1
## [58] knitr_1.36             dplyr_1.0.7            fastmap_1.1.0
## [61] bit_4.0.4              utf8_1.2.2             Rgraphviz_2.32.0
## [64] stringi_1.7.5          Rcpp_1.0.8             png_0.1-7
## [67] vctrs_0.3.8            geneplotter_1.66.0     tidyselect_1.1.1
## [70] xfun_0.29
```