Class 13: DESeq2 Mini Project

Wade Ingersoll (PID: A69038080)

Table of contents

Background/Overview

Today we will run through a complete RNAseq analysis

The data for for hands-on session comes from GEO entry: GSE37704, which is associated with 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

The authors report on differential analysis of lung fibroblasts in response to loss of the developmental transcription factor HOXA1. Their results and others indicate that HOXA1 is required for lung fibroblast and HeLa cell cycle progression.

Section 1. Differential Expression Analysis

```
library(DESeq2)
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```
Loading required package: generics
```

```

Attaching package: 'generics'
```

```
The following objects are masked from 'package:base':

    as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
    setequal, union
```

```

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, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, saveRDS, table, tapply, unique,
    unsplit, which.max, which.min
```

```

Attaching package: 'S4Vectors'
```

```
The following object is masked from 'package:utils':

    findMatches

The following objects are masked from 'package:base':

    expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomicRanges

Loading required package: Seqinfo

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
```

Read in data files:

```
metaFile <- "/Users/wadeingersoll/Desktop/BGGN213/class13/GSE37704_metadata.csv"
countFile <- "/Users/wadeingersoll/Desktop/BGGN213/class13/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
```

Import countdata:

```r
# Import countdata
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.

> **Q2:** 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).

```
tot.counts <- rowSums(countData)
head(tot.counts)
```

```
ENSG00000186092 ENSG00000279928 ENSG00000279457 ENSG00000278566 ENSG00000273547
              0               0             183               0               0
ENSG00000187634
           1129
```

Filter count data where you have 0 read count across all samples.

```
zero.inds <- tot.counts == 0
countData <- countData[!zero.inds,]
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. This is again similar to our last days hands-on session.

```
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
```

```
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)
```

> **Q3:** 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.

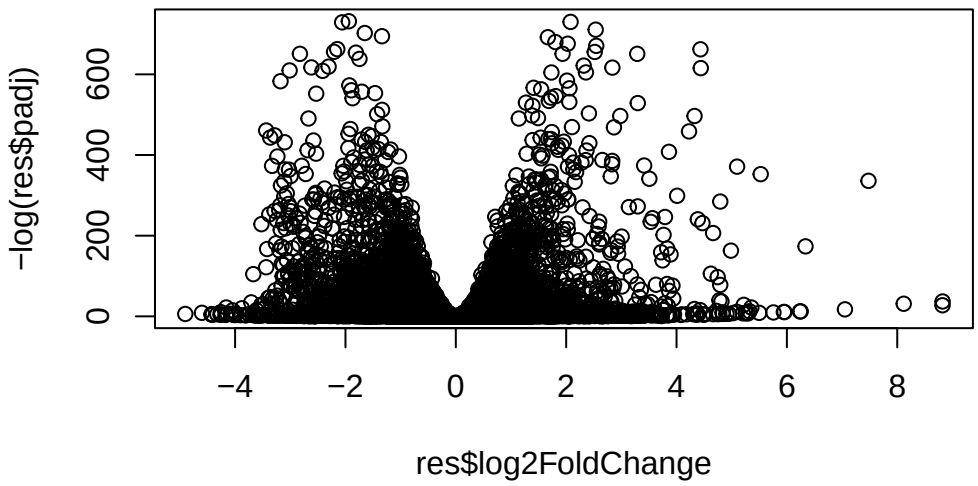
```
summary(res)
```

```
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

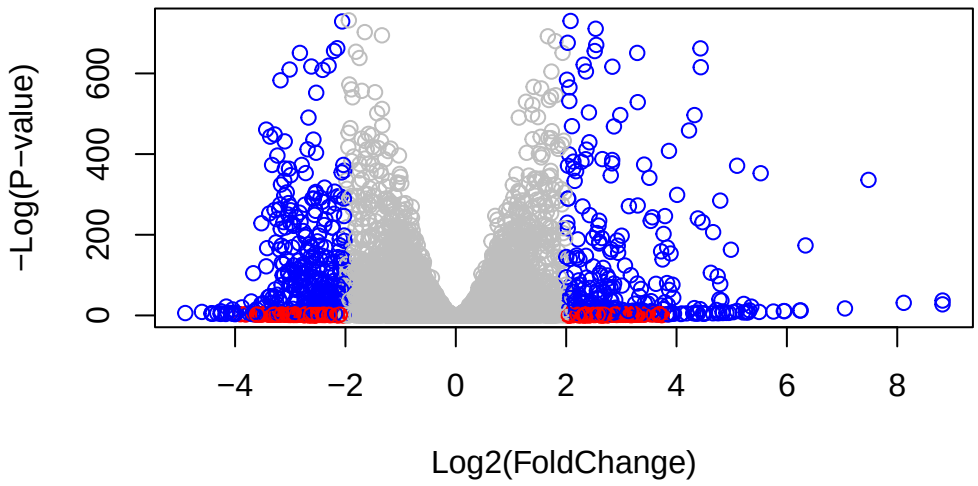
Now we will make a volcano plot, a commonly produced visualization from this type of data that we introduced last day. Basically it's a plot of log2 fold change vs -log adjusted p-value.

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


−log(res$padj)
0
200
400
600
−4
−2
0
2
4
6
8
res$log2FoldChange

Q4: 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$padj < .05) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

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



Adding 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.

> **Q5:** 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   29.913579      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                   name
                 <numeric> <character> <character>            <character>
ENSG00000279457 6.86555e-01          NA          NA                     NA
ENSG00000187634 5.15718e-03      SAMD11      148398 sterile alpha motif ..
ENSG00000188976 1.76549e-35       NOC2L       26155 NOC2 like nucleolar ..
ENSG00000187961 1.13413e-07      KLHL17      339451 kelch like family me..
ENSG00000187583 9.19031e-01     PLEKHN1       84069 pleckstrin homology ..
ENSG00000187642 4.03379e-01       PERM1       84808 PPARGC1 and ESRR ind..
ENSG00000188290 1.30538e-24        HES4       57801 hes family bHLH tran..
ENSG00000187608 2.37452e-02       ISG15        9636 ISG15 ubiquitin like..
ENSG00000188157 4.21963e-16        AGRN      375790                 agrin
ENSG00000237330          NA      RNF223      401934 ring finger protein ..
```

Q6: 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")
```

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.

Kegg Pathways

Make sure to install necessary packages in the console: "pathview", "gage", and "gageData"

```
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)

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

```r
# 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$entrez`) and we have the fold change results from DESeq2 analysis (stored in `res$log2FoldChange`).

```
foldchanges <- res$log2FoldChange
names(foldchanges) <- res$entrez
head(foldchanges)
```

```
     1266     54855      1465      2034      2150      6659
-2.422719  3.201955 -2.313738 -1.888019  3.344508  2.392288
```

Now, let's run the **gage** pathway analysis.

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

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 *Environment* 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$greater)` and `head(keggres$less)`.

Lets 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                     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$less` and `keggres$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`.

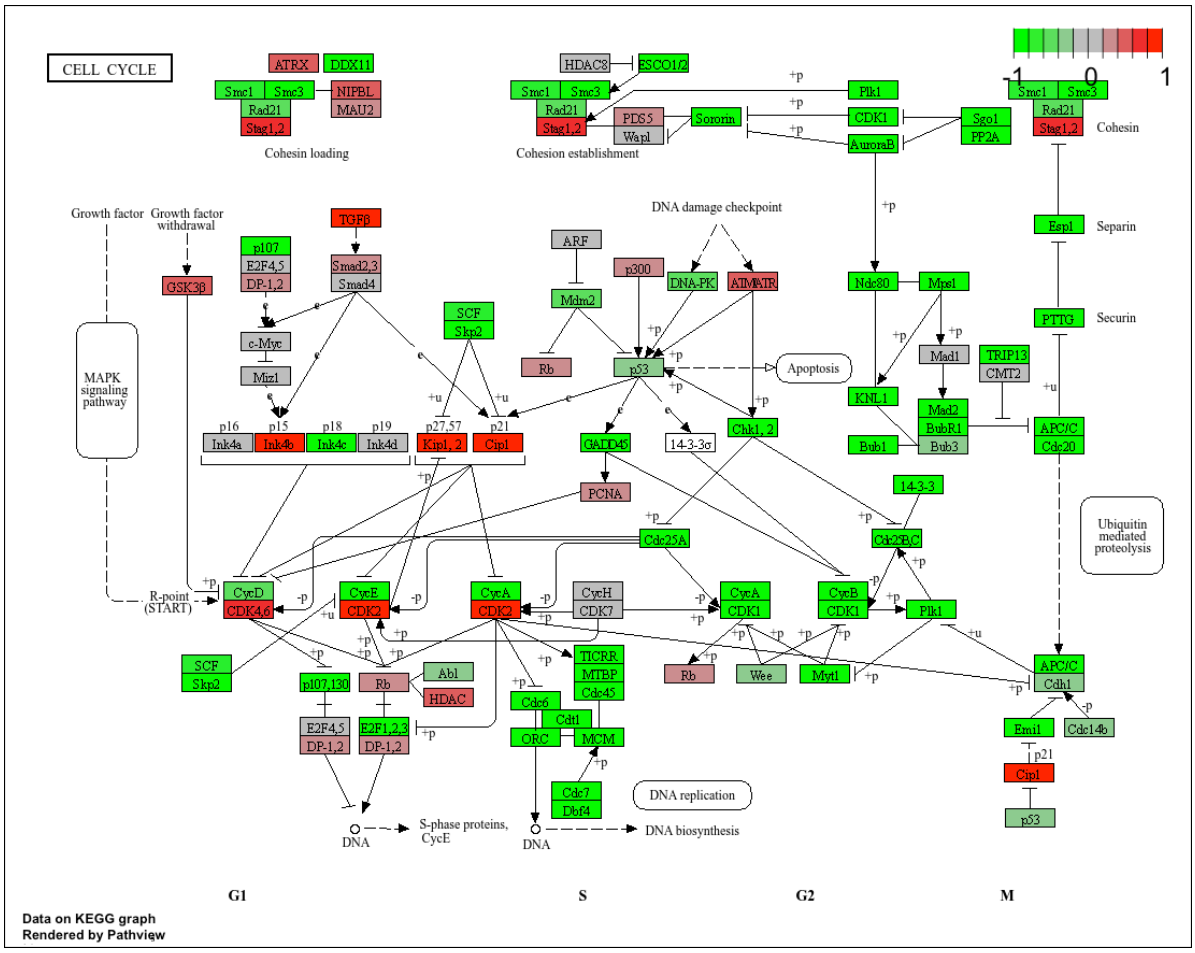
Now, let's try out the **pathview()** function from the **pathview package** to make a pathway plot with our RNA-Seq expression results shown in color. To begin with lets manually supply a `pathway.id` (namely the first part of the `"hsa04110 Cell cycle"`) that we could see from the print out above.

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

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

```
Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13
```

```
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)
```

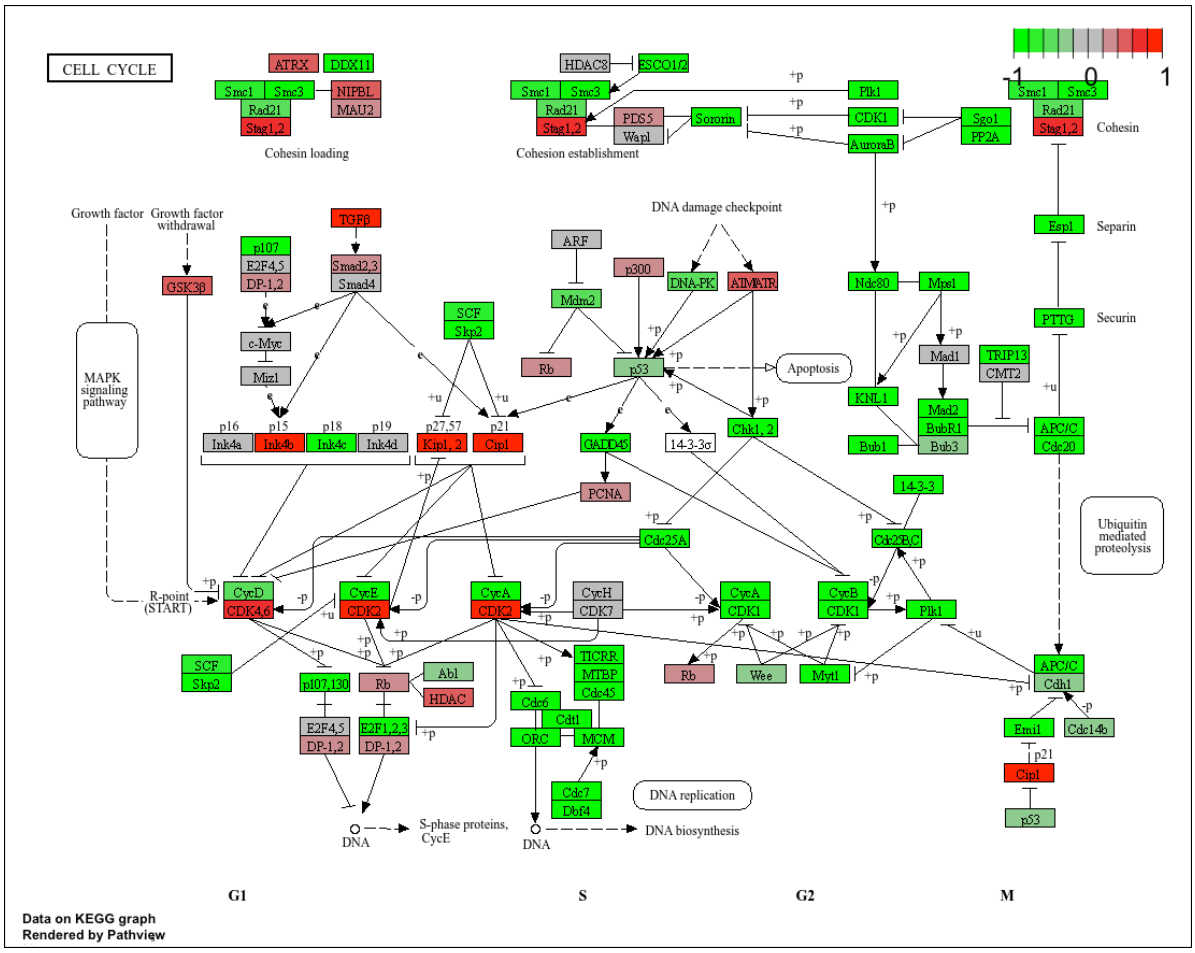
```
'select()' returned 1:1 mapping between keys and columns
```

```
Warning: reconcile groups sharing member nodes!
```

```
     [,1] [,2]
[1,] "9"  "300"
[2,] "9"  "306"
```

```
Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13
```

```
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.

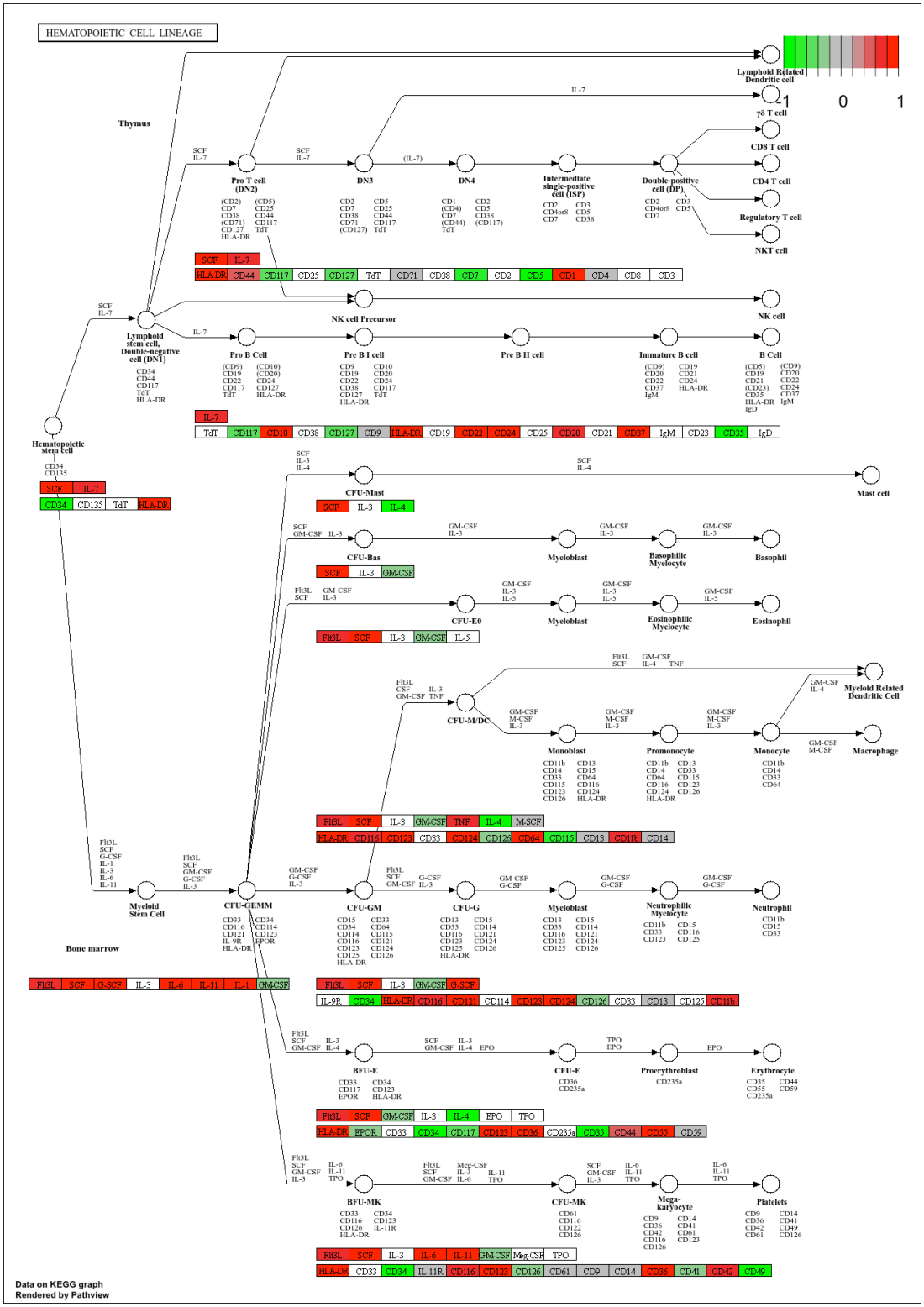
```
## Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

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

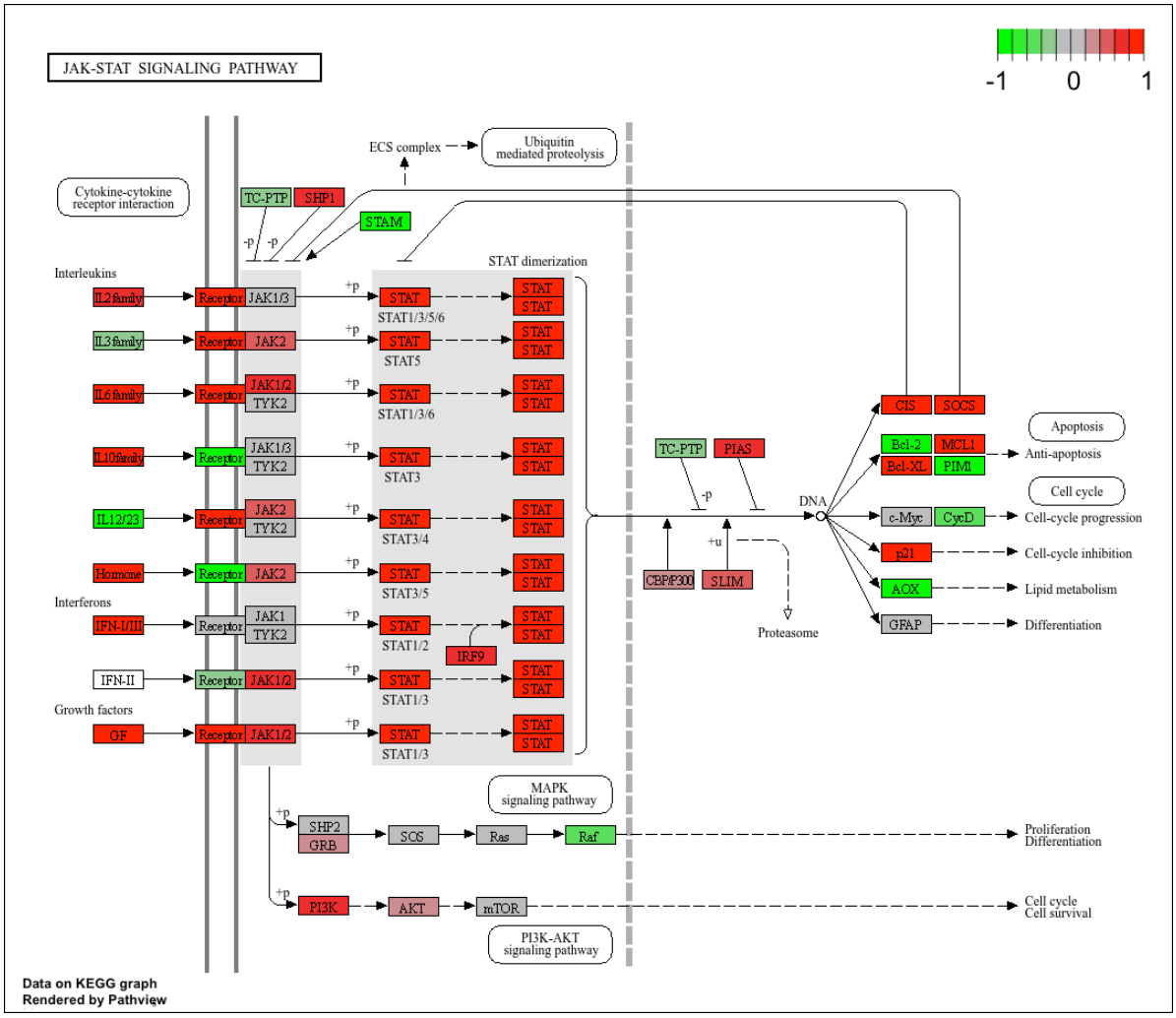
```
[1] "hsa04640" "hsa04630" "hsa00140" "hsa04142" "hsa04330"
```

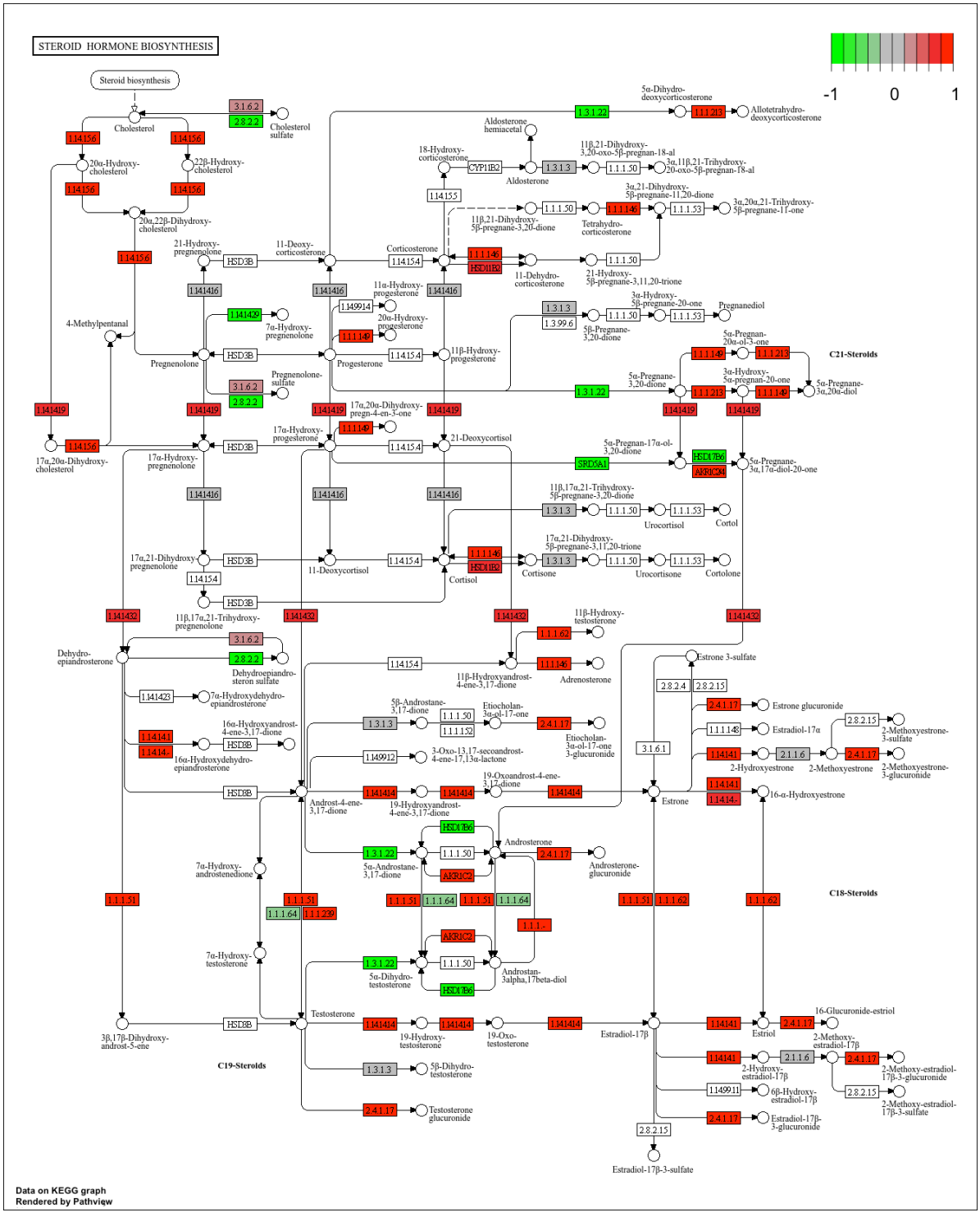
Finally, lets pass these IDs in keggresids to the **pathview()** function to draw plots for all the top 5 pathways.

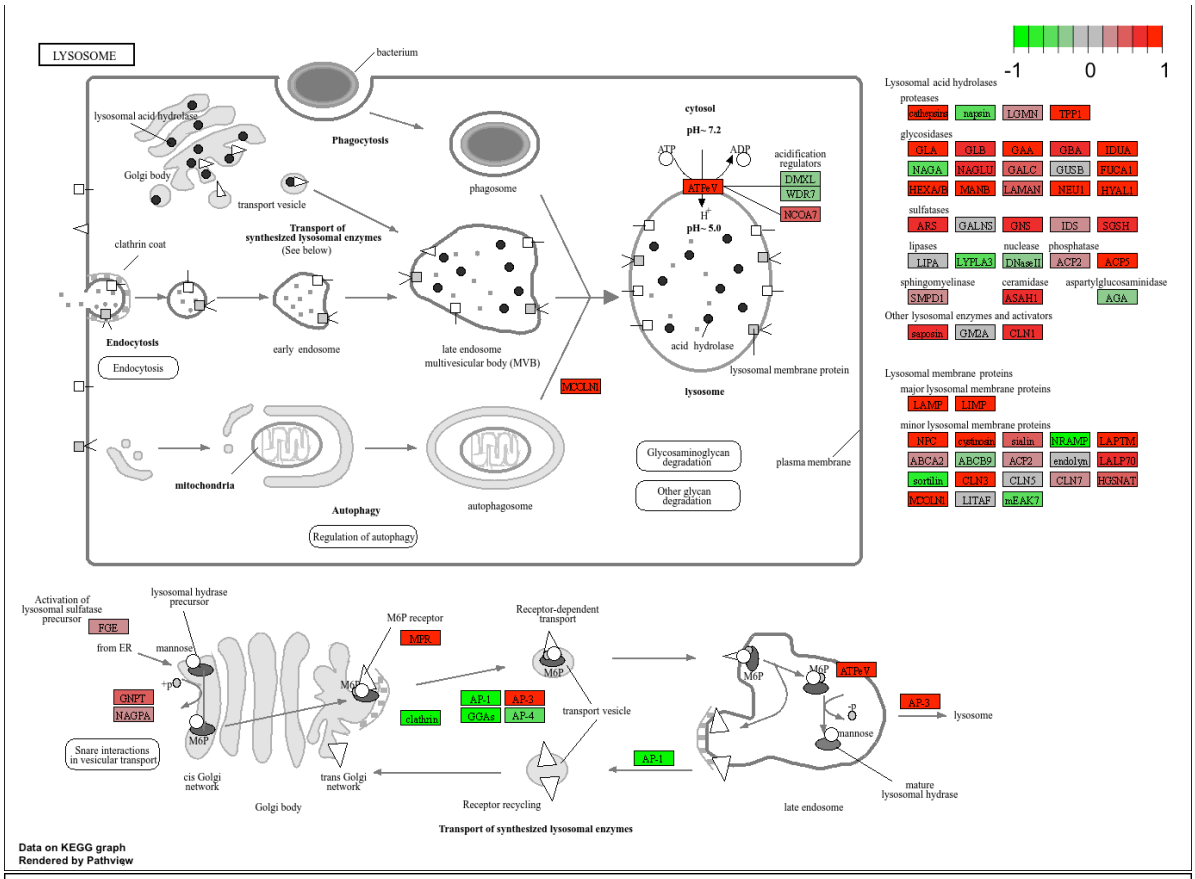
```
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```

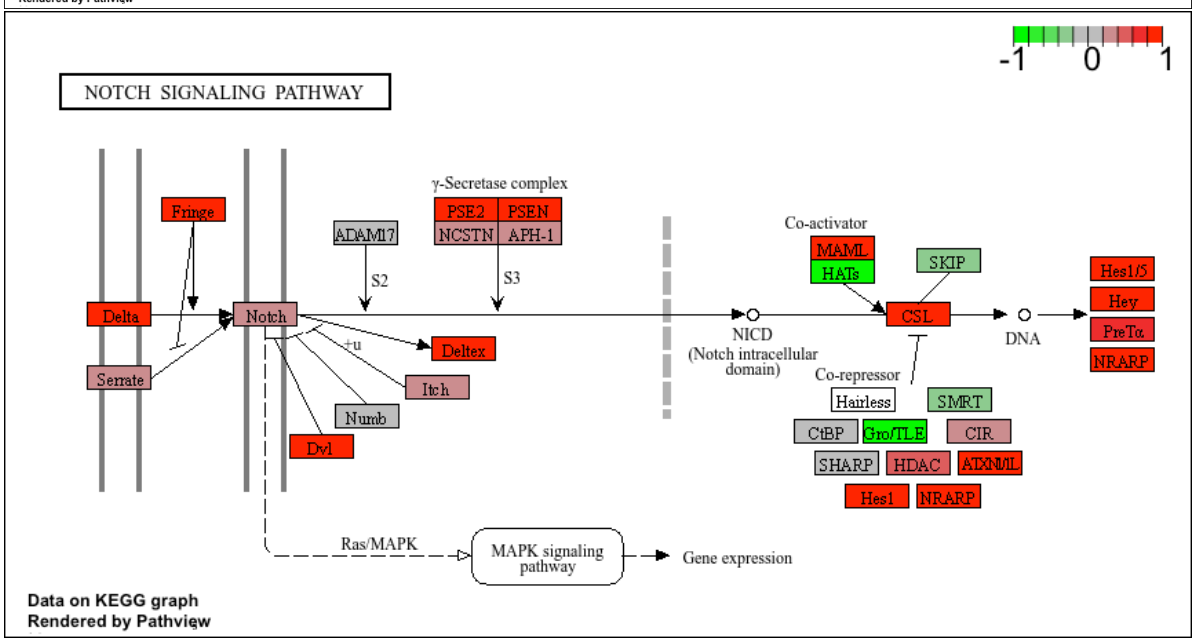
```
'select()' returned 1:1 mapping between keys and columns

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa04640.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa04630.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa00140.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa04142.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

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


HEMATOPOIETIC CELL LINEAGE
Lymphoid Related Dendritic cell
-1
0
1
Thymus
γδ T cell
CD8 T cell
CD4 T cell
Regulatory T cell
NKT cell
Pro T cell (DN2)
DN3
DN4
Intermediate single-positive cell (ISP)
Double-positive cell (DP)
NK cell Precursor
NK cell
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-E0
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
Procrythroblast
Erythrocyte
BFU-MK
CFU-MK
Mega-karyocyte
Platelets
Data on KEGG graph
Rendered by Pathview


JAK-STAT SIGNALING PATHWAY
-1
0
1
ECS complex
Ubiquitin mediated proteolysis
Cytokine-cytokine receptor interaction
TC-PTP
SHP1
STAM
STAT dimerization
Interleukins
IL2 family
IL3 family
IL6 family
IL10 family
IL12/23
Hormone
Interferons
IFN-I/III
IFN-II
Growth factors
GF
Receptor
JAK1/3
JAK2
JAK1/2
TYK2
JAK1
STAT
STAT1/3/5/6
STAT5
STAT1/3/6
STAT3
STAT3/4
STAT3/5
STAT1/2
STAT1/3
IRF9
TC-PTP
PIAS
CBP/P300
SLIM
Proteasome
DNA
CIS
SOCS
Bcl-2
MCL1
Bcl-XL
PIM1
Apoptosis
Anti-apoptosis
Cell cycle
c-Myc
CycD
Cell-cycle progression
p21
Cell-cycle inhibition
AOX
Lipid metabolism
GFAP
Differentiation
MAPK signaling pathway
SHP2
GRB
SOS
Ras
Raf
Proliferation
Differentiation
PI3K
AKT
mTOR
Cell cycle
Cell survival
PI3K-AKT signaling pathway
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
11-Dehydro-corticosterone
Aldosterone hemiacetal
Aldosterone
18-Hydroxy-corticosterone
11β,21-Dihydroxy-3,20-oxo-5β-pregnan-18-al
3α,11β,21-Trihydroxy-20-oxo-5β-pregnan-18-al
3α,21-Dihydroxy-5β-pregnane-11,20-dione
3α,20α,21-Trihydroxy-5β-pregnane-11-one
11β,21-Dihydroxy-5β-pregnane-3,20-dione
Tetrahydro-corticosterone
21-Hydroxy-5β-pregnane-3,11,20-trione
5α-Dihydro-deoxycorticosterone
Allotetrahydro-deoxycorticosterone
4-Methylpentanal
Pregnenolone
7α-Hydroxy-pregnenolone
Progesterone
11α-Hydroxy-progesterone
20α-Hydroxy-progesterone
11β-Hydroxy-progesterone
Pregnenolone-sulfate
17α,20α-Dihydroxy-pregn-4-en-3-one
17α-Hydroxy-progesterone
21-Deoxycortisol
17α,20α-Dihydroxy-cholesterol
17α-Hydroxy-pregnenolone
3α-Hydroxy-5β-pregnane-20-one
Pregnanediol
5β-Pregnane-3,20-dione
5α-Pregnan-20α-ol-3-one
5α-Pregnane-3,20-dione
3α-Hydroxy-5α-pregnan-20-one
5α-Pregnane-3α,20α-diol
C21-Steroids
5α-Pregnan-17α-ol-3,20-dione
5α-Pregnane-3α,17α-diol-20-one
11β,17α,21-Trihydroxy-5β-pregnane-3,20-dione
Urocortisol
Cortol
17α,21-Dihydroxy-pregnenolone
11-Deoxycortisol
Cortisol
17α,21-Dihydroxy-5β-pregnane-3,11,20-trione
Cortisone
Urocortisone
Cortolone
11β,17α,21-Trihydroxy-pregnenolone
Dehydro-epiandrosterone
Dehydroepiandro-steron sulfate
7α-Hydroxydehydro-epiandrosterone
16α-Hydroxyandrost-4-ene-3,17-dione
16α-Hydroxydehydro-epiandrosterone
11β-Hydroxyandrost-4-ene-3,17-dione
11β-Hydroxy-testosterone
Adrenosterone
Estrone 3-sulfate
5β-Androstane-3,17-dione
Etiocholan-3α-ol-17-one
Etiocholan-3α-ol-17-one 3-glucuronide
3-Oxo-13,17-secoandrost-4-ene-17,13α-lactone
19-Oxoandrost-4-ene-3,17-dione
Androst-4-ene-3,17-dione
19-Hydroxyandrost-4-ene-3,17-dione
Estrone
Estrone glucuronide
Estradiol-17α
2-Hydroxyestrone
2-Methoxyestrone
2-Methoxyestrone-3-sulfate
2-Methoxyestrone-3-glucuronide
16-α-Hydroxyestrone
Androsterone
Androsterone-glucuronide
5α-Androstane-3,17-dione
7α-Hydroxy-androstenedione
C18-Steroids
7α-Hydroxy-testosterone
5α-Dihydro-testosterone
Androstan-3alpha,17beta-diol
Testosterone
3β,17β-Dihydroxy-androst-5-ene
19-Hydroxy-testosterone
19-Oxo-testosterone
Estradiol-17β
Estriol
16-Glucuronide-estriol
2-Methoxy-estradiol-17β
2-Hydroxy-estradiol-17β
6β-Hydroxy-estradiol-17β
Estradiol-17β-3-glucuronide
2-Methoxy-estradiol-17β-3-glucuronide
2-Methoxy-estradiol-17β-3-sulfate
C19-Steroids
5β-Dihydro-testosterone
Testosterone glucuronide
Estradiol-17β-3-sulfate
Data on KEGG graph
Rendered by Pathview


LYSOSOME
bacterium
lysosomal acid hydrolase
Phagocytosis
Golgi body
transport vesicle
phagosome
cytosol
pH~ 7.2
ATP
ADP
acidification regulators
DMXL
WDR7
NCOA7
ATPeV
H+
pH~ 5.0
Transport of synthesized lysosomal enzymes
(See below)
clathrin coat
Endocytosis
Endocytosis
early endosome
late endosome
multivesicular body (MVB)
MCOLN1
acid hydrolase
lysosomal membrane protein
lysosome
mitochondria
Autophagy
autophagosome
Regulation of autophagy
Glycosaminoglycan degradation
Other glycan degradation
plasma membrane
-1
0
1
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
VRAMP
LAPTM
ABCA2
ABCB9
ACP2
endolyn
LALP70
sortilin
CLN3
CLN5
CLN7
HGSNAT
MCOLN1
LITAF
mEAK7
Activation of lysosomal sulfatase precursor
FGE
lysosomal hydrase precursor
from ER
mannose
GNPT
NAGPA
M6P
Snare interactions in vesicular transport
cis Golgi network
trans Golgi network
Golgi body
M6P receptor
MPR
clathrin
AP-1
GGAs
AP-3
AP-4
Receptor-dependent transport
transport vesicle
Receptor recycling
AP-1
ATPeV
AP-3
lysosome
mannose
mature lysosomal hydrase
late endosome
Transport of synthesized lysosomal enzymes
Data on KEGG graph
Rendered by Pathview


-1
0
1
NOTCH SIGNALING PATHWAY
γ-Secretase complex
PSE2
PSEN
NCSTN
APH-1
Fringe
ADAM17
S2
S3
Delta
Notch
Serrate
Deltex
Itch
Numb
Dvl
+u
NICD
(Notch intracellular domain)
Co-activator
MAML
HATs
SKIP
CSL
DNA
Hes1/5
Hey
PreTα
NRARP
Co-repressor
Hairless
SMRT
CtBP
Gro/TLE
CIR
SHARP
HDAC
ATXN1L
Hes1
NRARP
Ras/MAPK
MAPK signaling pathway
Gene expression
Data on KEGG graph
Rendered by Pathview

> **Q7:** Can you do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways? - See code below for answer

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

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

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

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

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa04110.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa03030.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa03013.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

Info: Writing image file hsa03440.pathview.png

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

Info: Working in directory /Users/wadeingersoll/Desktop/BGGN213/class13

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

Section 3. Gene Ontology (GO)

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                       1.925222e-04  3.565432 1.925222e-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       0.1951953      113 8.519724e-05
GO:0002009 morphogenesis of an epithelium 0.1951953      339 1.396681e-04
GO:0048729 tissue morphogenesis           0.1951953      424 1.432451e-04
GO:0007610 behavior                       0.1967577      426 1.925222e-04
GO:0060562 epithelial tube morphogenesis  0.3565320      257 5.932837e-04
GO:0035295 tube development               0.3565320      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              5.841698e-12      376 1.536227e-15
```

```
GO:0000280 nuclear division              5.841698e-12      352 4.286961e-15
GO:0007067 mitosis                       5.841698e-12      352 4.286961e-15
GO:0000087 M phase of mitotic cell cycle 1.195672e-11      362 1.169934e-14
GO:0007059 chromosome segregation        1.658603e-08      142 2.028624e-11
GO:0000236 mitotic prometaphase          1.178402e-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                        3.565432 3.565432
GO:0060562 epithelial tube morphogenesis   3.261376 3.261376
GO:0035295 tube development                3.253665 3.253665
```

Section 4. Reactome Analysis

Some folks really like Reactome online (i.e. their webpage viewer) rather than the R package of the same name (available from bioconductor).

To use the viewer we want to upload our set of gene symbols for the genes we want to focus on (here those with a P-value below 0.05)

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

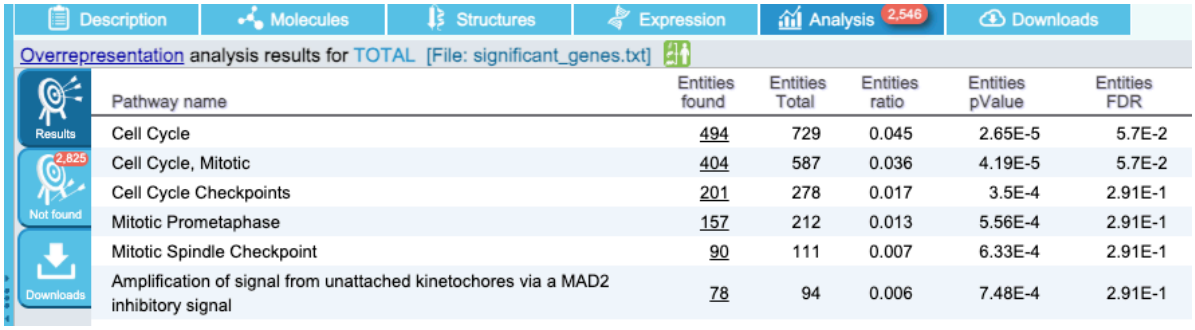
```
[1] "Total number of significant genes: 8147"
```

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

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

Q8: 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?

Answer: The pathway with the most significant Entities p-value was Cell Cyle (see screenshot below). The most significant pathways do *not* match, likely due to different statistical tests between them.



Pathway name	Entities found	Entities Total	Entities ratio	Entities pValue	Entities FDR
Cell Cycle	494	729	0.045	2.65E-5	5.7E-2
Cell Cycle, Mitotic	404	587	0.036	4.19E-5	5.7E-2
Cell Cycle Checkpoints	201	278	0.017	3.5E-4	2.91E-1
Mitotic Prometaphase	157	212	0.013	5.56E-4	2.91E-1
Mitotic Spindle Checkpoint	90	111	0.007	6.33E-4	2.91E-1
Amplification of signal from unattached kinetochores via a MAD2 inhibitory signal	78	94	0.006	7.48E-4	2.91E-1

Section 5. GO online

To perform Gene Set GO Enrichment online go to the website http://www.geneontology.org/page/go-enrichment-analysis. Paste your significant gene list from section 4. Then, select "biological process" and "homo sapiens", and click submit.

> **Q9:** 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?
>
> **Answer:** The most significant pathway was XMP metabolic process (see screenshot below). The most significant pathways do *not* match my previous KEGG resutls, again likely due to different statistical tests run between the two tools.

	Analysis Type:	PANTHER Overrepresentation Test (Released 20240807)					
	Annotation Version and Release Date:	GO Ontology database DOI: 10.5281/zenodo.16423886 Released 2025-07-22					
	Analyzed List:	upload_1 (Homo sapiens)					
	Reference List:	Homo sapiens (all genes in database)					
0	Test Type:	FISHER					
1	Correction:	FDR					
2	GO biological process complete	Homo sapiens -	upload_1 (81	upload_1 (ex	upload_	upload_1	upload_1 (raw P-value)	upload_1 (FDR)
3	XMP metabolic process (GO:0097292)	8	8	3.18	+	2.52	6.20E-04	6.36E-03
4	XMP biosynthetic process (GO:0097293)	8	8	3.18	+	2.52	6.20E-04	6.37E-03
5	wound healing, spreading of cells (GO:0044319)	30	20	11.92	+	1.68	4.23E-03	3.25E-02
6	wound healing (GO:0042060)	339	174	134.69	+	1.29	1.51E-05	2.37E-04
7	Wnt signaling pathway, planar cell polarity pathway	35	23	13.91	+	1.65	2.71E-03	2.22E-02
8	Wnt signaling pathway (GO:0016055)	288	164	114.43	+	1.43	4.12E-09	1.17E-07
9	visual system development (GO:0150063)	423	212	168.07	+	1.26	1.48E-05	2.31E-04
0	visual perception (GO:0007601)	223	54	88.6	-	0.61	1.20E-06	2.35E-05

Figure 1: go_analysis