Pathway Analysis from RNA-Seq Results

Andy Hsu

Data Import

Let's start by importing our files and converting them to R-friendly formats.

```
library(DESeq2)
```

```
Warning: package 'GenomeInfoDb' was built under R version 4.3.2
```

```
Warning: package 'matrixStats' was built under R version 4.3.2
```

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

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

countData = read.csv(countFile, row.names=1)
```

We should remove the first column from the countData set so it lines up with the colData set.

```
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

Let's also remove all entries of exclusively 0 values.

```
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

We now have an appropriate dataset prepared for DESeq.

DESeq Setup and Analysis

To perform our DESeq, remember that we need a special type of dataset. Once written, we can run the DESeq.

```
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
```

Let's also get the results for the HoxA1 knockdown versus control siRNA.

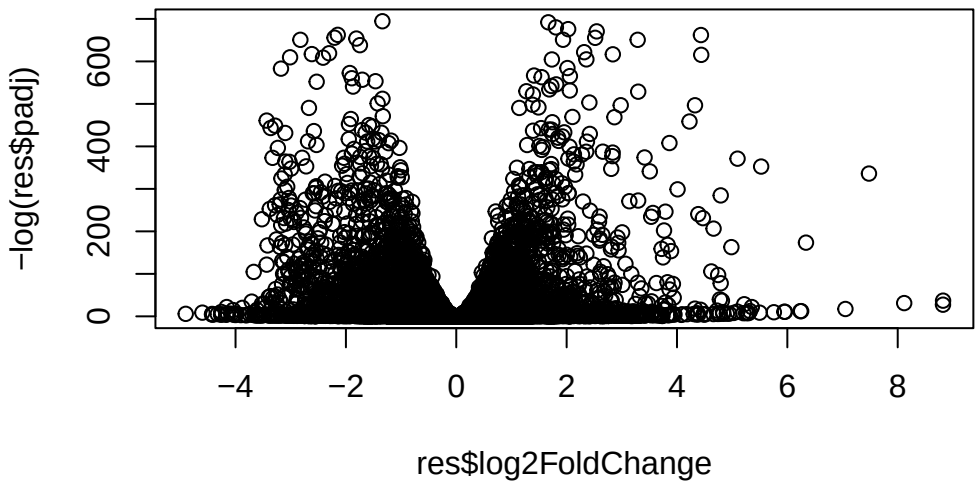
```
res = results(dds, contrast=c("condition", "hoxa1_kd", "control_sirna"))
summary(res)
```

```
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

Let's now make a volcano plot of our results.

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



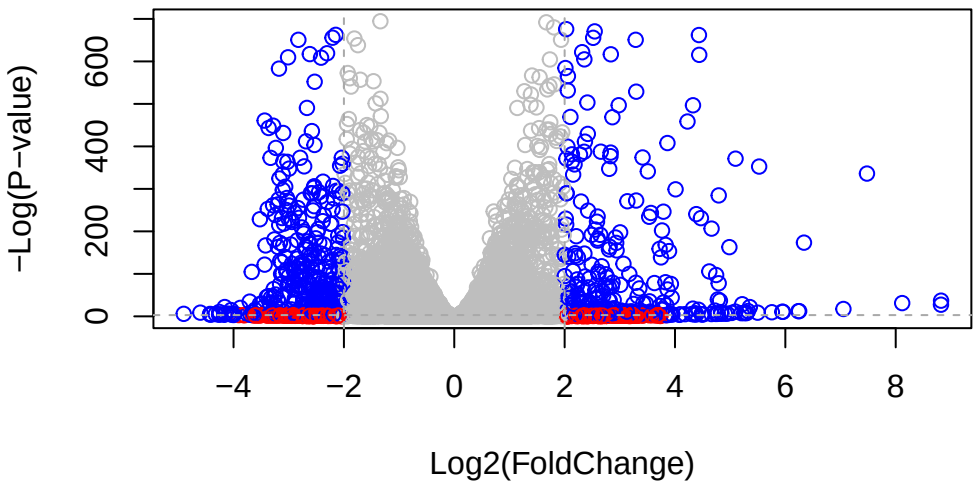
Let's not forget the color coding and ablines.

```
mycols <- rep("gray", nrow(res) )
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

inds <- (res$padj < 0.05) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

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

abline(v=c(-2,2), col="darkgray", lty=2)
abline(h=-log(0.05), col="darkgray", lty=2)
```



Gene Annotation

Next, we can annotate our results with each entry's symbol, Entrez ID, and gene name.

```
library("AnnotationDbi")
```

```
Warning: package 'AnnotationDbi' was built under R version 4.3.2
```

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

```
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, 4)
```

```
log2 fold change (MLE): condition hoxa1_kd vs control_sirna
Wald test p-value: condition hoxa1 kd vs control sirna
DataFrame with 4 rows and 9 columns
                 baseMean log2FoldChange     lfcSE       stat      pvalue
                <numeric>      <numeric> <numeric>  <numeric>   <numeric>
ENSG00000279457   29.9136       0.179257 0.3248216   0.551863 5.81042e-01
ENSG00000187634  183.2296       0.426457 0.1402658   3.040350 2.36304e-03
ENSG00000188976 1651.1881      -0.692720 0.0548465 -12.630158 1.43990e-36
ENSG00000187961  209.6379       0.729756 0.1318599   5.534326 3.12428e-08
                      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..
```

Let's also write our results to a csv file to save it.

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

Pathway Analysis

Now, let's use the **gage** package to perform pathway analysis and the **pathviewer** package to visualize our results.

```
library(pathview)
library(gage)
library(gageData)
```

Let's first load the appropriate datasets containing pathways and associated genes, filtering for just signaling and metabolic pathways.

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

kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]
```

Now, we will prepare a vector of fold changes for inputs for the `gage()` function.

```
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
```

Let's run the gage analysis now.

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

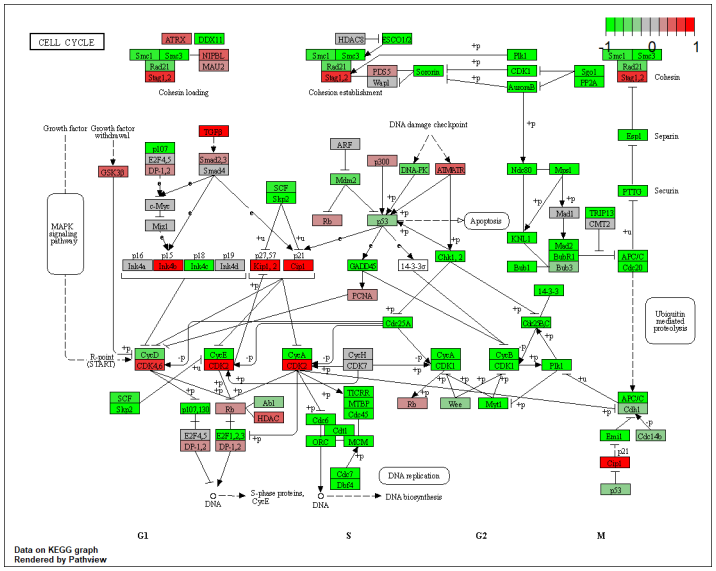
```
                                      p.geomean stat.mean        p.val
hsa04110 Cell cycle                8.995727e-06 -4.378644 8.995727e-06
hsa03030 DNA replication           9.424076e-05 -3.951803 9.424076e-05
hsa03013 RNA transport             1.375901e-03 -3.028500 1.375901e-03
hsa03440 Homologous 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
```

Finally, let's examine the first entry, the Cell Cycle pathway, using pathview.

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

The resulting image is displayed below.



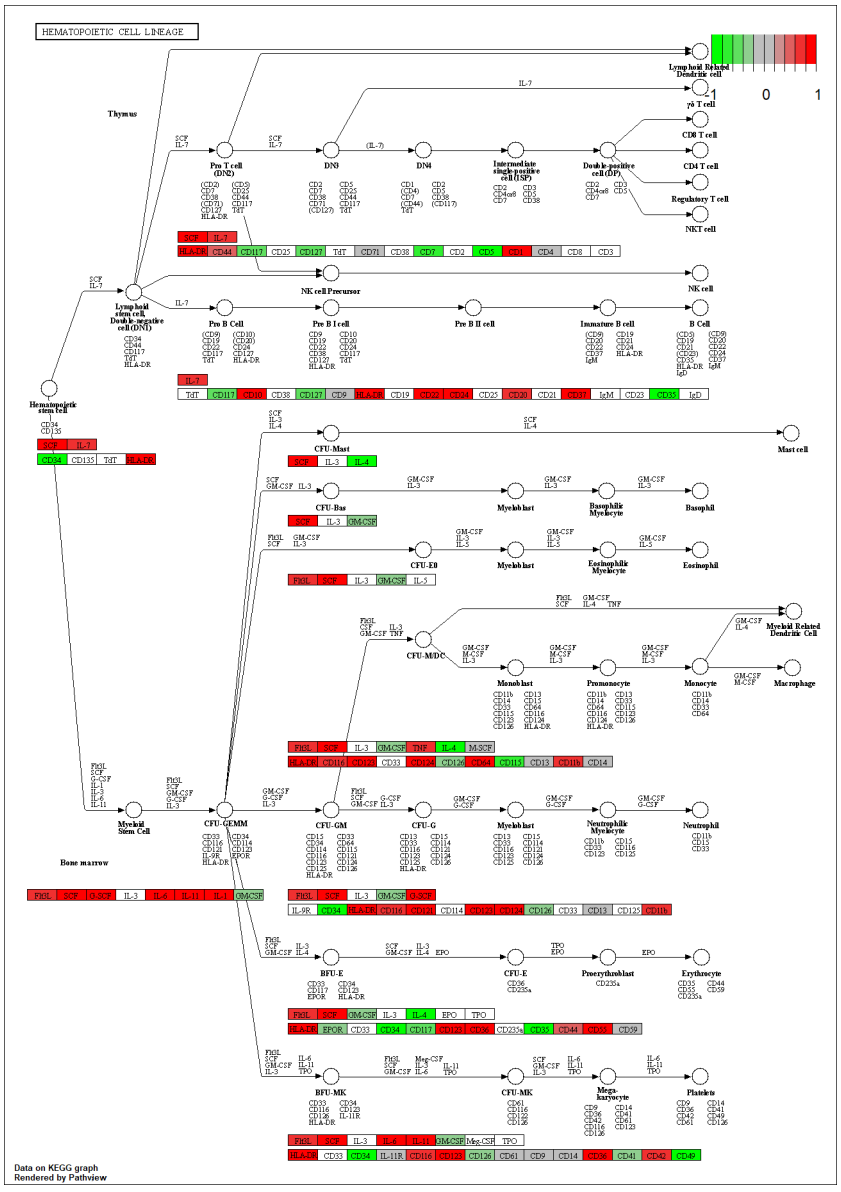
Let's repeat this process a little for the top 5 upregulated pathways.

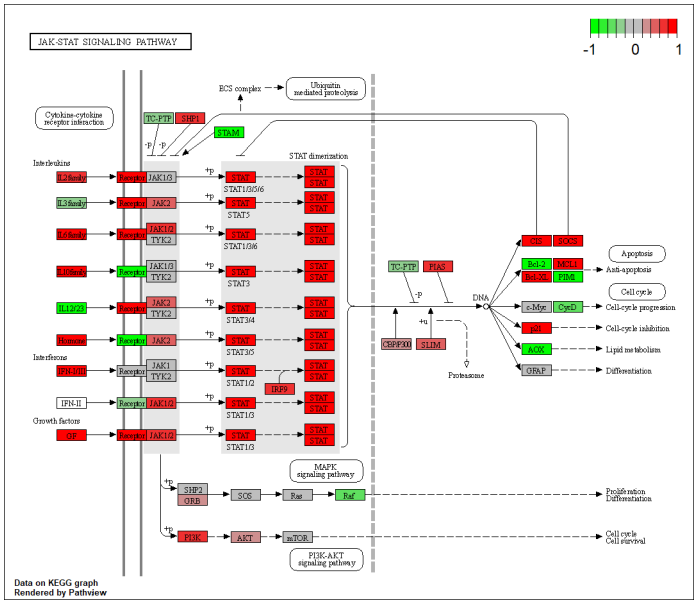
```
keggrespathways <- rownames(keggres$greater)[1:5]
keggrespathways
```

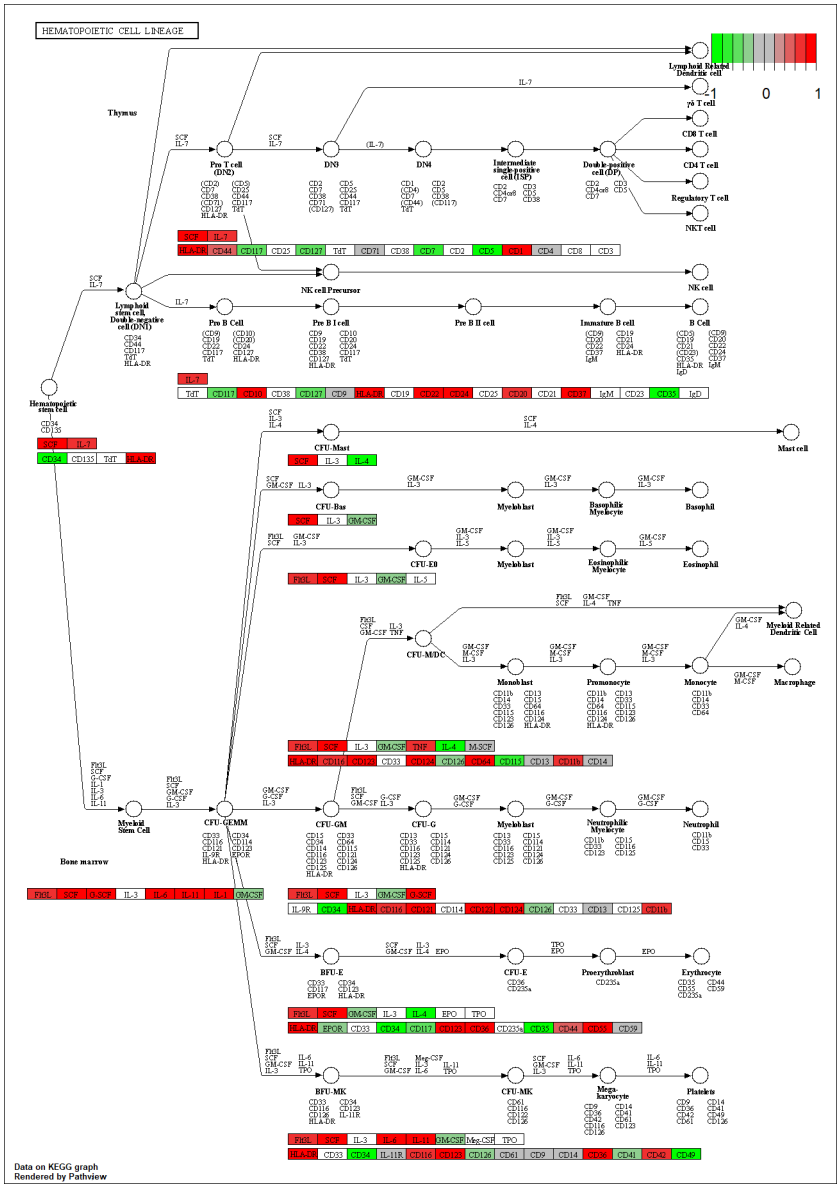
```
[1] "hsa04640 Hematopoietic cell lineage"
[2] "hsa04630 Jak-STAT signaling pathway"
[3] "hsa00140 Steroid hormone biosynthesis"
[4] "hsa04142 Lysosome"
[5] "hsa04330 Notch signaling pathway"
```

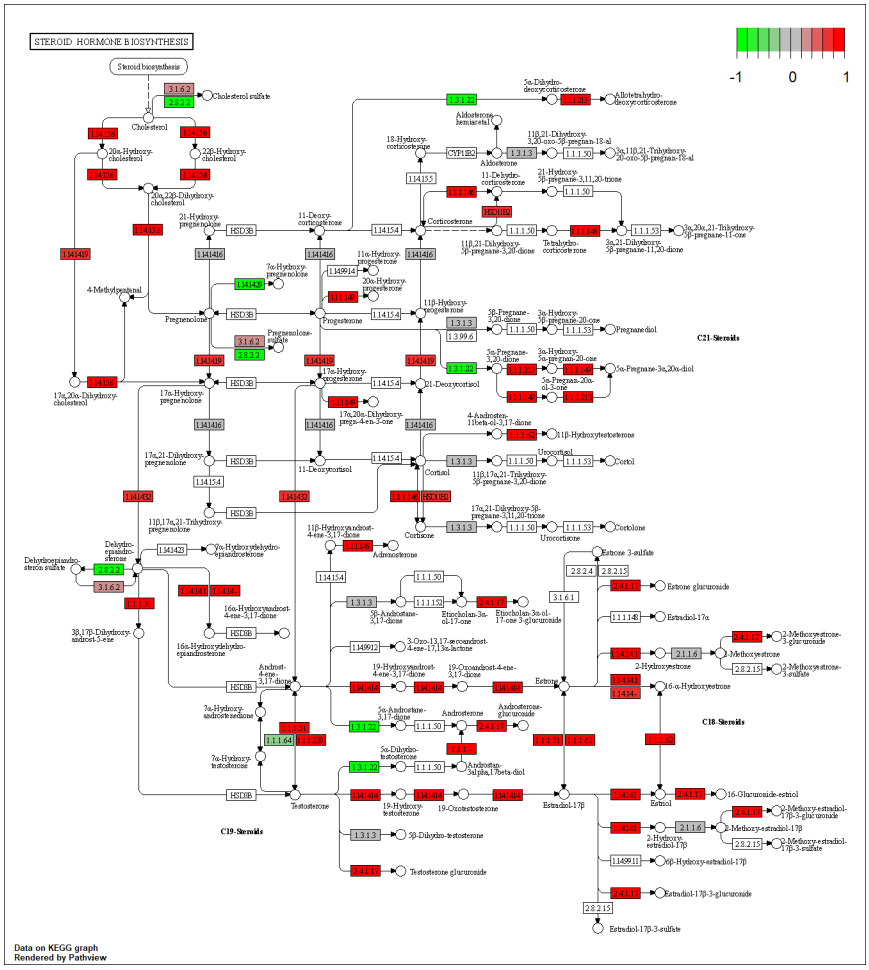
```
keggresids = substr(keggrespathways, start=1, stop=8)
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```

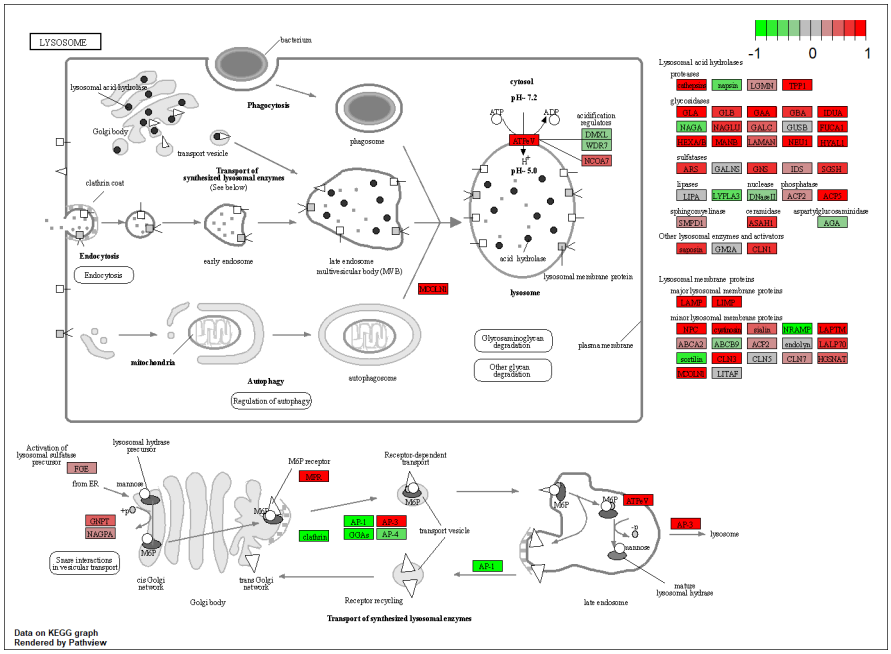
Here are the 5 resulting plots:

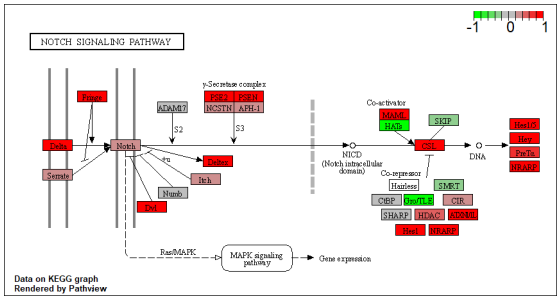



JAK-STAT SIGNALING PATHWAY
-1
0
1
Cytokine-cytokine receptor interaction
Ubiquitin mediated proteolysis
Interleukins
Interferons
Growth factors
Receptor
JAK1/2/3
TYK2
STAT1/2/3/5/6
STAT5
STAT1/2/6
STAT3
STAT3/4
STAT3/5
STAT1/2
STAT1/3
STAT dimerization
IRF9
TC-PTP
SHP1
STAM
PIAS
CBP/P300
SLIM
Proteasome
DNA
CIS
SOCS
Bcl-2
Mcl-1
Bcl-XL
FIM
c-Myc
CycD
p21
ACK
GFAP
Apoptosis
Anti-apoptosis
Cell cycle
Cell-cycle progression
Cell-cycle inhibition
Lipid metabolism
Differentiation
MAPK signaling pathway
SHP2
GRB
SOS
Ras
Raf
Proliferation Differentiation
PI3K
AKT
mTOR
PI3K-AKT signaling pathway
Cell cycle Cell survival
Data on KEGG graph
Rendered by Pathview


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
NK cell Precursor
NK cell
Pre T cell (DNC)
DN3
DN4
Intermediate single-positive cell (ISP)
Double-positive cell (DP)
Lymphoid stem cell, Double-negative cell (DNI)
Pre 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-MDC
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
Pregnenolone
Progesterone
11-Deoxycorticosterone
Corticosterone
11-Deoxycortisol
Cortisol
Cortisone
Androstenedione
Testosterone
Estrone
Estradiol-17β
Estriol
C21-Steroids
C19-Steroids
C18-Steroids
Data on KEGG graph
Rendered by Pathview





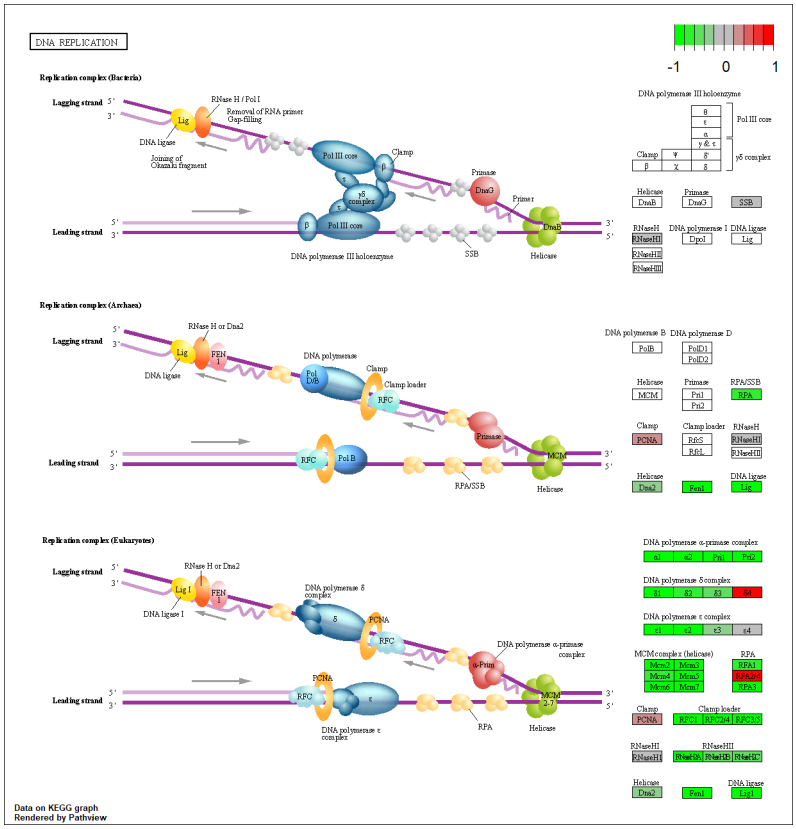
And again repeat for the top 5 downregulated pathways.

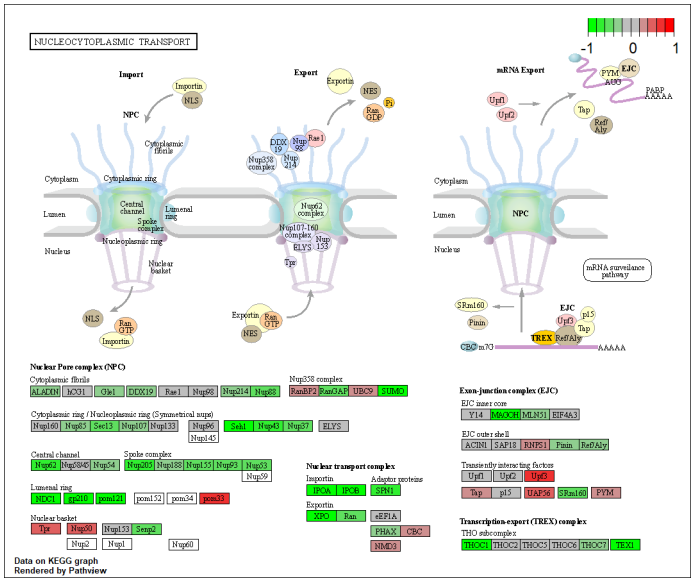
```
keggrespathways1 <- rownames(keggres$less)[2:5]
keggrespathways1
```

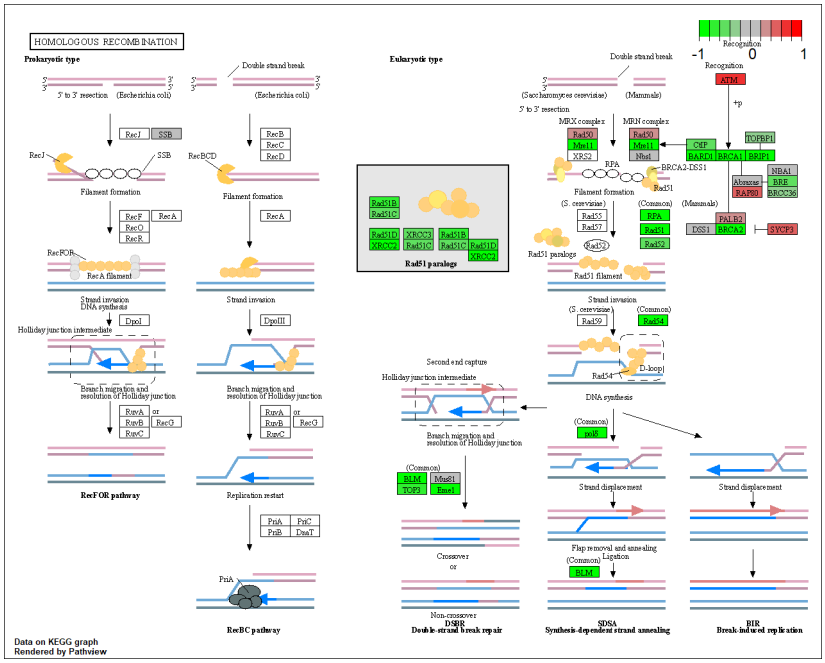
```
[1] "hsa03030 DNA replication"          "hsa03013 RNA transport"
[3] "hsa03440 Homologous recombination" "hsa04114 Oocyte meiosis"
```

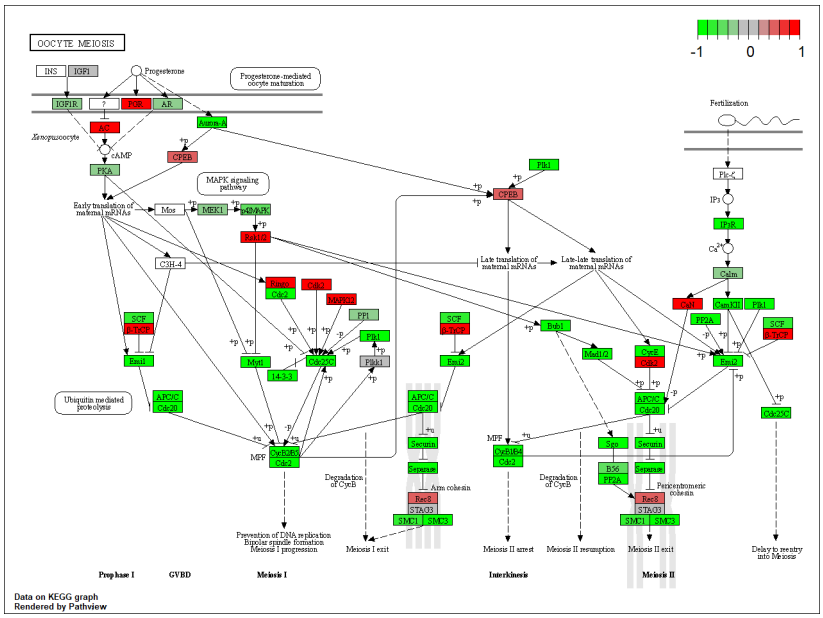
```
keggresids1 = substr(keggrespathways1, start=1, stop=8)
pathview(gene.data=foldchanges, pathway.id=keggresids1, species="hsa")
```

Here are the 5 resulting plots, with the top result (hsa04110 Cell Cycle) omitted from earlier.


DNA REPLICATION
-1
0
1
Replication complex (Bacteria)
Lagging strand
Leading strand
RNase H / Pol I
Removal of RNA primer Gap-filling
DNA ligase
Joining of Okazaki fragment
Pol III core
Clamp
γδ complex
Primase
DnaG
Primer
DnaB
SSB
Helicase
DNA polymerase III holoenzyme
Replication complex (Archaea)
RNase H or Dna2
FEN 1
DNA polymerase
Pol D/B
Clamp loader
RFC
Pol B
MCM
RPA/SSB
Replication complex (Eukaryotes)
Lig I
DNA ligase I
DNA polymerase δ complex
PCNA
DNA polymerase α-primase complex
DNA polymerase ε complex
Data on KEGG graph
Rendered by Pathview


NUCLEOCYTOPLASMIC TRANSPORT
-1
0
1
Import
Importin
NLS
NPC
Export
Exportin
NES
Ran GTP
mRNA Export
EJC
PABP
AAAAA
Cytoplasm
Lumen
Nucleus
Cytoplasmic fibrils
Cytoplasmic ring
Central channel
Spoke complex
Lumenal ring
Nucleoplasmic ring
Nuclear basket
Nup358 complex
Nup62 complex
Nup107-160 complex
Tpr
mRNA surveillance pathway
Nuclear Pore complex (NPC)
Exon-junction complex (EJC)
Nuclear transport complex
Transcription-export (TREX) complex
Data on KEGG graph
Rendered by Pathview





Gene Ontology Alternative

As an alternative to KEGG, we can use a GO (Gene Ontology) pathway set. An example is shown below.

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

gobpsets = go.sets.hs[go.subs.hs$BP]
gobpres = gage(foldchanges, gsets=gobpsets, same.dir=TRUE)

print("$greater")
```

```
[1] "$greater"
```

```
head(gobpres$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       0.1952430      113 8.519724e-05
GO:0002009 morphogenesis of an epithelium 0.1952430      339 1.396681e-04
GO:0048729 tissue morphogenesis           0.1952430      424 1.432451e-04
GO:0007610 behavior                       0.1968058      426 1.925222e-04
GO:0060562 epithelial tube morphogenesis  0.3566193      257 5.932837e-04
GO:0035295 tube development               0.3566193      391 5.953254e-04
```

```
print("$less")
```

```
[1] "$less"
```

```
head(gobpres$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             5.843127e-12      376 1.536227e-15
GO:0000280 nuclear division              5.843127e-12      352 4.286961e-15
GO:0007067 mitosis                       5.843127e-12      352 4.286961e-15
GO:0000087 M phase of mitotic cell cycle 1.195965e-11      362 1.169934e-14
GO:0007059 chromosome segregation        1.659009e-08      142 2.028624e-11
GO:0000236 mitotic prometaphase          1.178690e-07       84 1.729553e-10
```

Reactome Analysis Alternative

Similarly, the Reactome database is another alternative method of analysis that can be used. Below is an example.

```
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, quo
```

Taking this file and uploading it to the Reactome website https://reactome.org/PathwayBrowser/#TOOL=AT, we can find a list of pathways similar to the previous 2 explored methods.

Interestingly, the pathway with the lowest Entities p-value is the Cell Cycle pathway, matching the other methods, but the other pathways mentioned such as the mitotic spindle checkpoint or kinetichore signal amplification aren't seen in the others. This difference in results could be due to Reactome looking at biological molecules generally in relation to pathways rather than exclusively genes.

And with that, that's all for this pathway analysis of RNA-Seq data using multiple methods.