Class14: RNASeq mini-proj

James Woolley A16440072

```
library(DESeq2)
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```

Attaching package: 'BiocGenerics'
```

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

    IQR, mad, sd, var, xtabs
```

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

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

```

Attaching package: 'S4Vectors'
```

```
The following 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: GenomeInfoDb

Loading required package: SummarizedExperiment

Warning: package 'SummarizedExperiment' was built under R version 4.3.2

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
```

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

Let's import the data.

```
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
```

```
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
```

Q. Complete the code below to remove the troublesome first column from countData

```
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
```

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).

```
countData <- countData[rowSums(countData) != 0, ]
```

```
nrow(countData)
```

```
[1] 15975
```

##DESeq setup and analysis

```
library(DESeq2)
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
```

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

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.

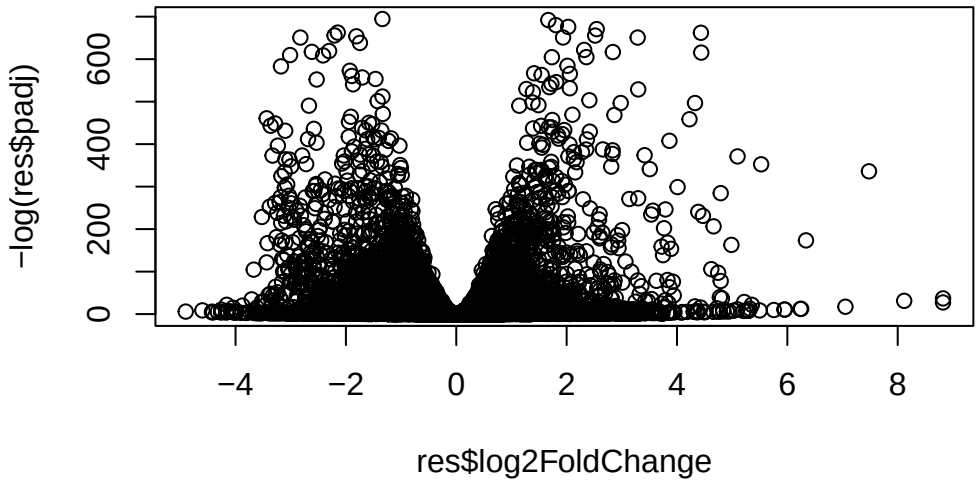
```
  summary(res)
```

```
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
```

4396 genes are down-regulated at the 0.1 p-value.

##Plotting the data

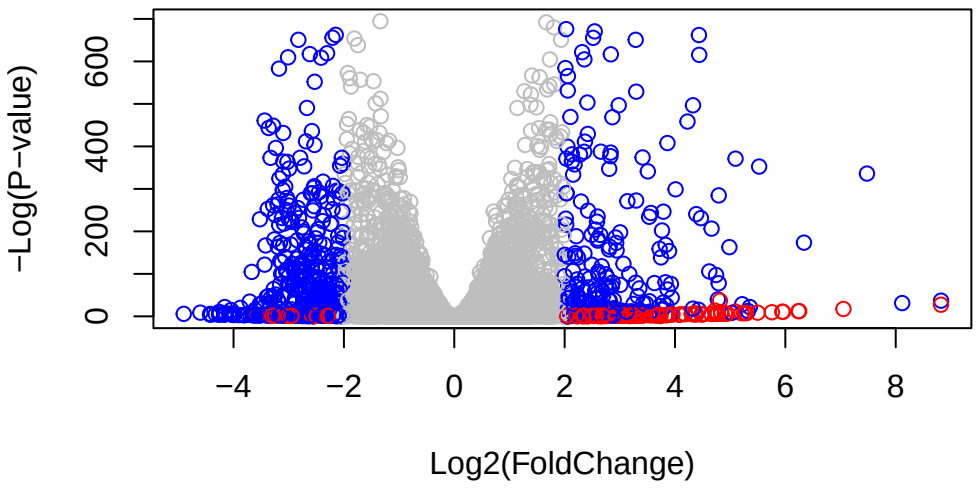
```
plot( res$log2FoldChange, -log(res$padj) )
```



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

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

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



##Adding Gene Annotation

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

Here we're adding rows to the data that are actually useful for people to read, like names, symols, and entrez numbers.

```
library("AnnotationDbi")
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, 3)
```

```
log2 fold change (MLE): condition hoxa1 kd vs control sirna
Wald test p-value: condition hoxa1 kd vs control sirna
DataFrame with 3 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 1.43989e-36
                      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 ..
```

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")
```

##Pathway Analysis

Let's load up some data we can use to generate figures.

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

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

```
kegg.sets.hs <-  kegg.sets.hs[sigmet.idx.hs] #lets focus on signaling pathways
foldchanges <-  res$log2FoldChange
names(foldchanges) <-  res$entrez
```

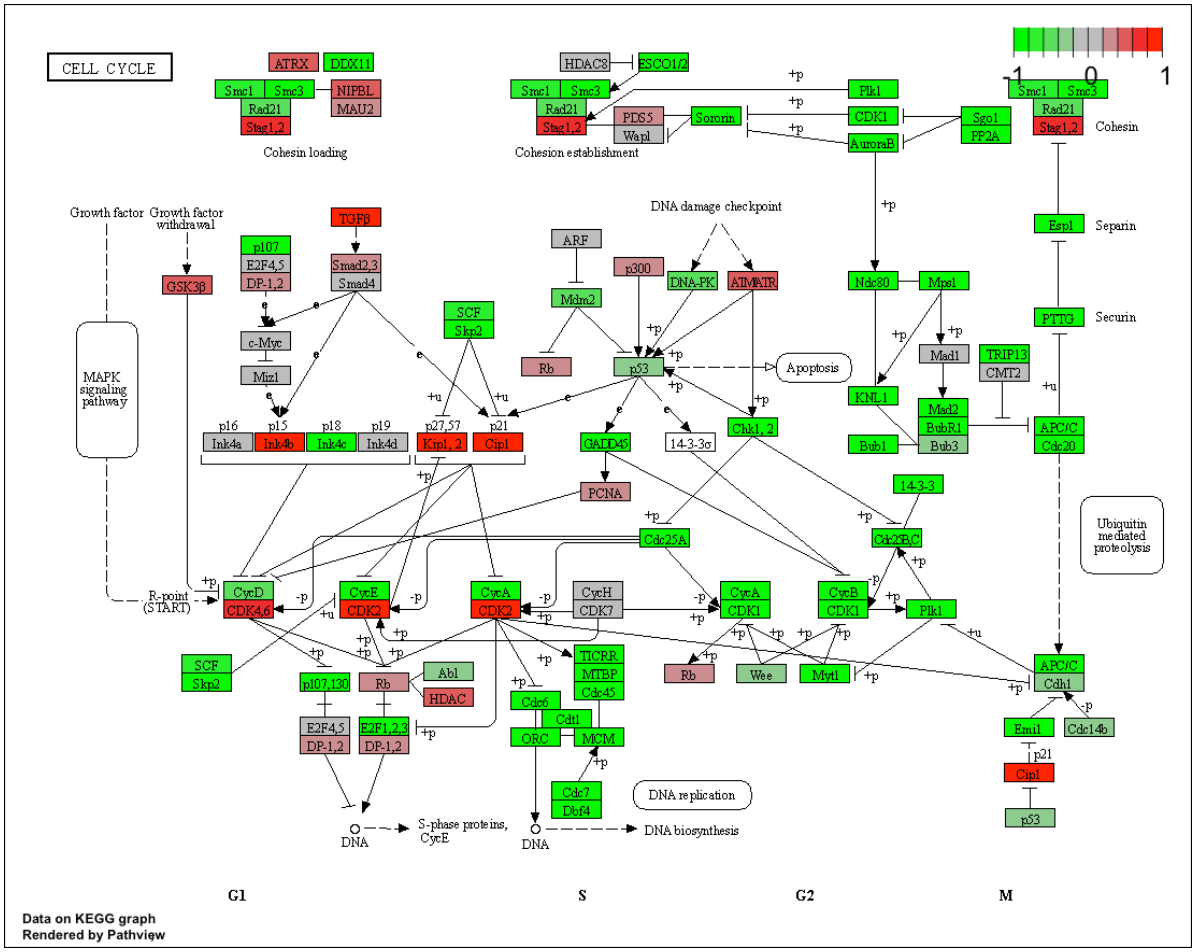
```
keggres = gage(foldchanges, gsets=kegg.sets.hs) #getting results
```

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

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

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



Here we have the entire pathway laid out for us! There are other ways to argue with it to change the way that the data is presented, but the actual information will be the same. We can also focus on the 5 highest most upregulated pathways, we just need to get their IDs first using `pathview`

```
keggrespathways <- rownames(keggres$greater)[1:5]
keggresids <-  substr(keggrespathways, start=1, stop=8)
keggresids
```

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

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

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

```
Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa04640.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa04630.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa00140.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

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/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

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

This will generate 5 plots for the IDs that we identified above.

##Question: >Q. Can you do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways?

```
keggrespathwaysLESS <- rownames(keggres$less)[1:5]
keggresidsLESS <-  substr(keggrespathwaysLESS, start=1, stop=8)
```

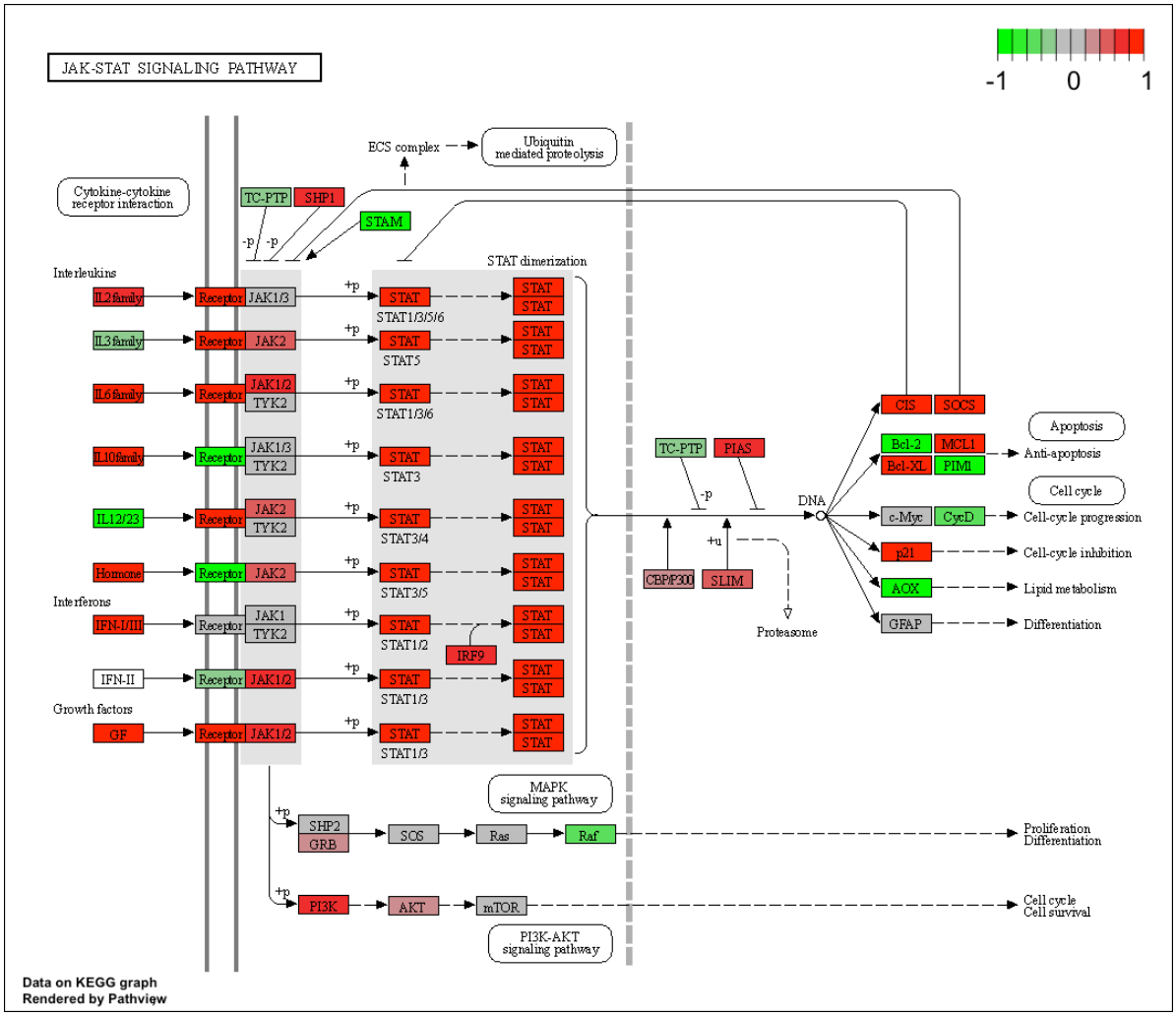


Figure 1: Gene Pathway

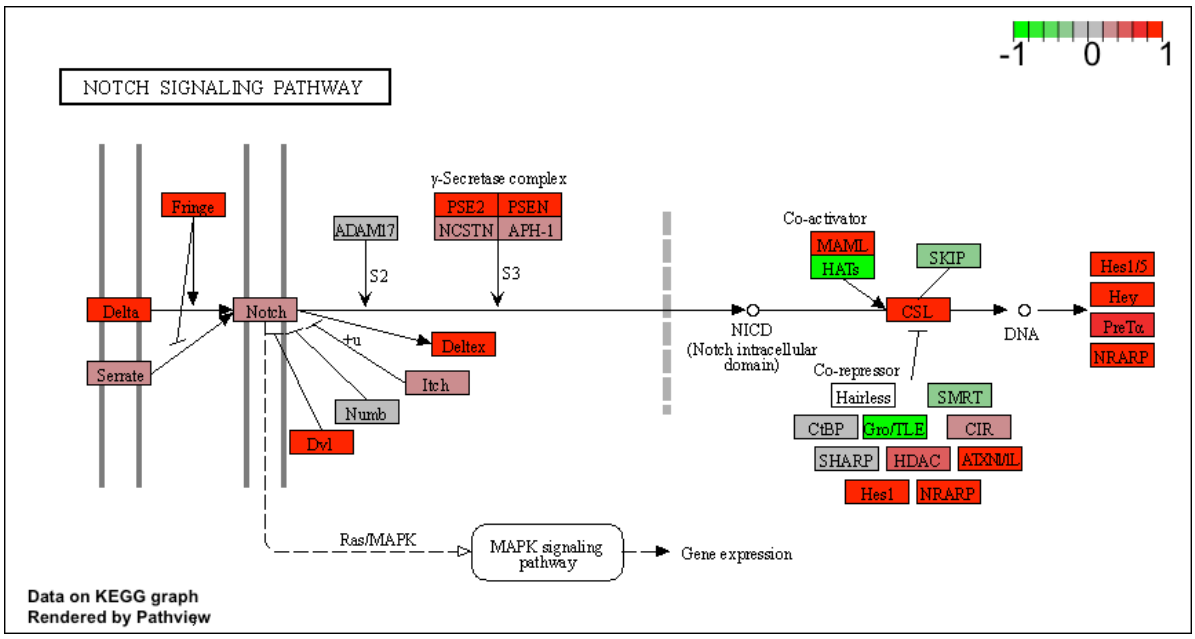


Figure 2: Gene Pathway

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

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa04110.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa03030.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

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

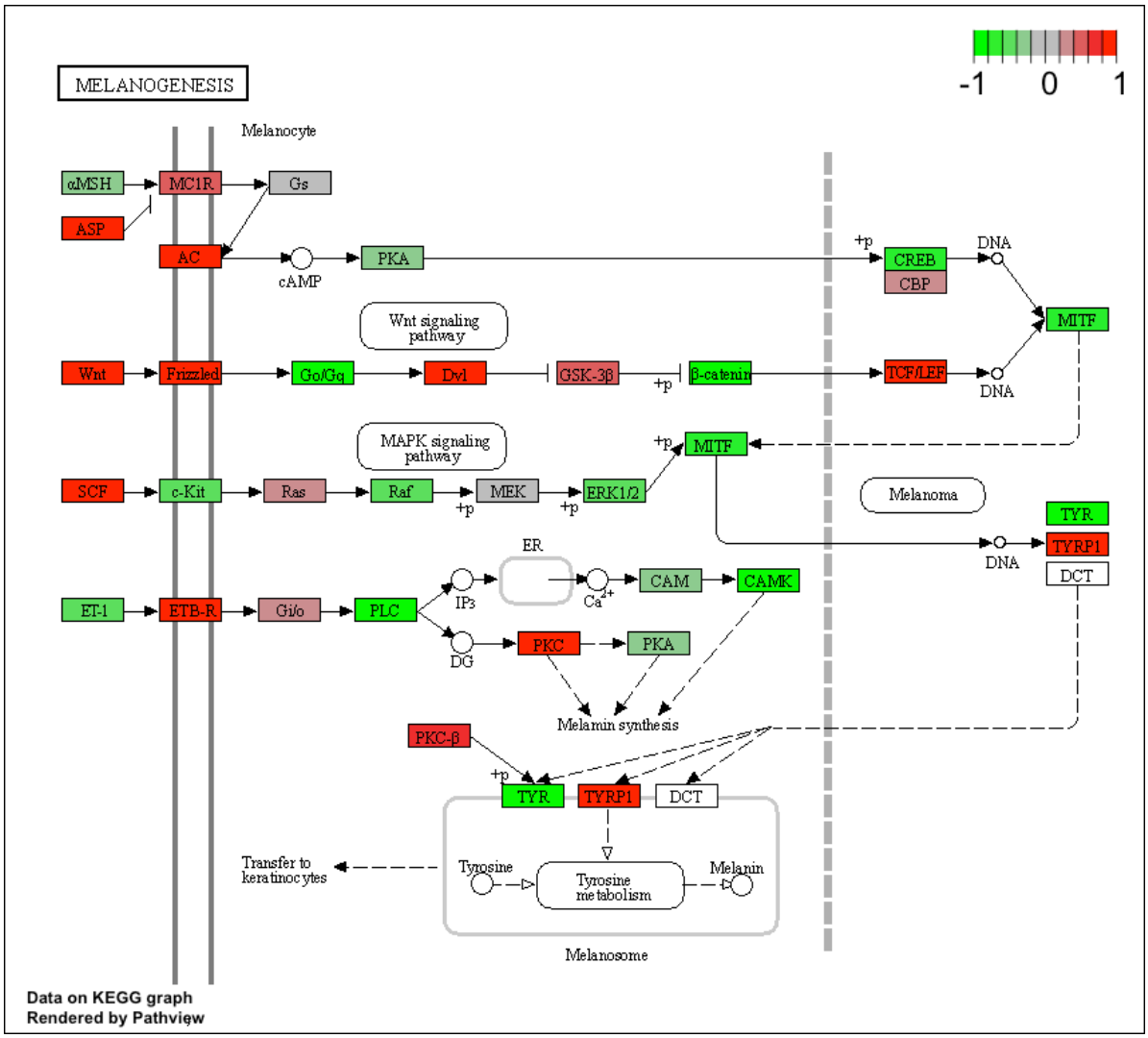


Figure 3: Gene Pathway

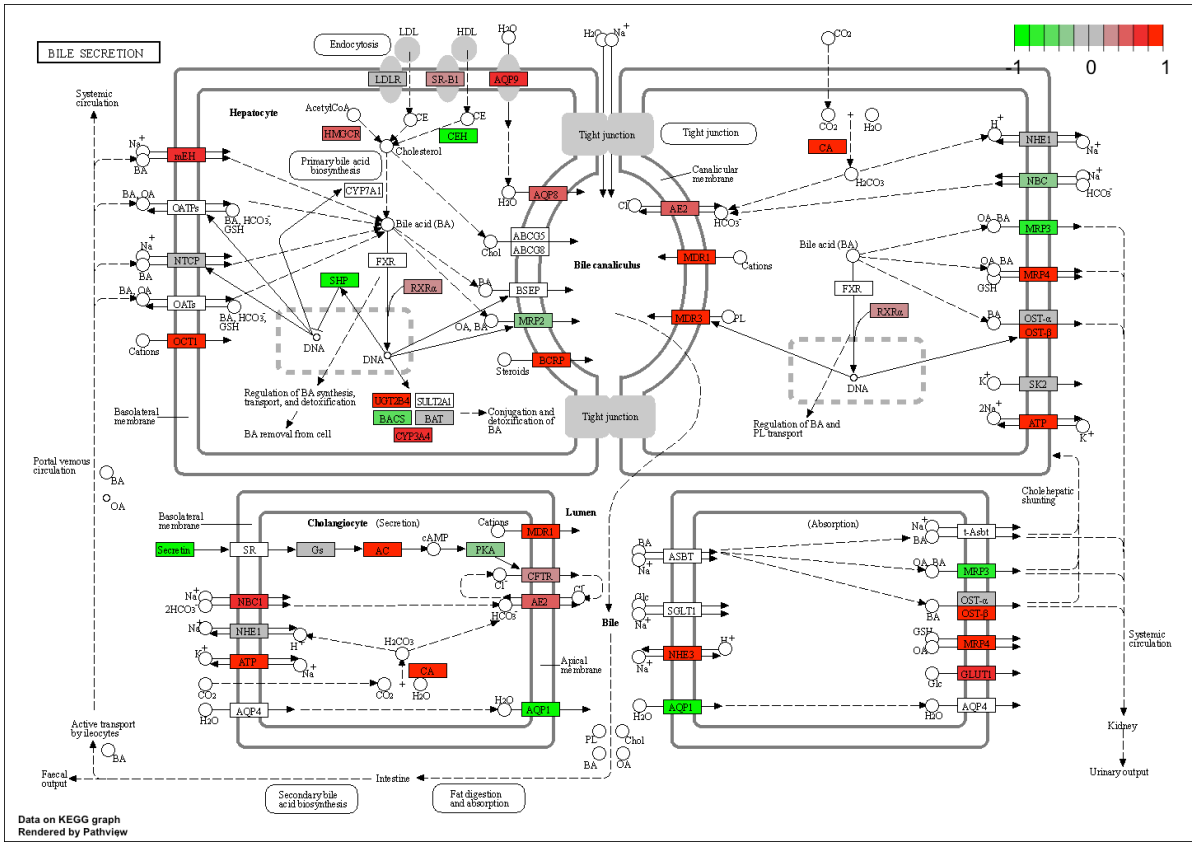


Figure 4: Gene Pathway

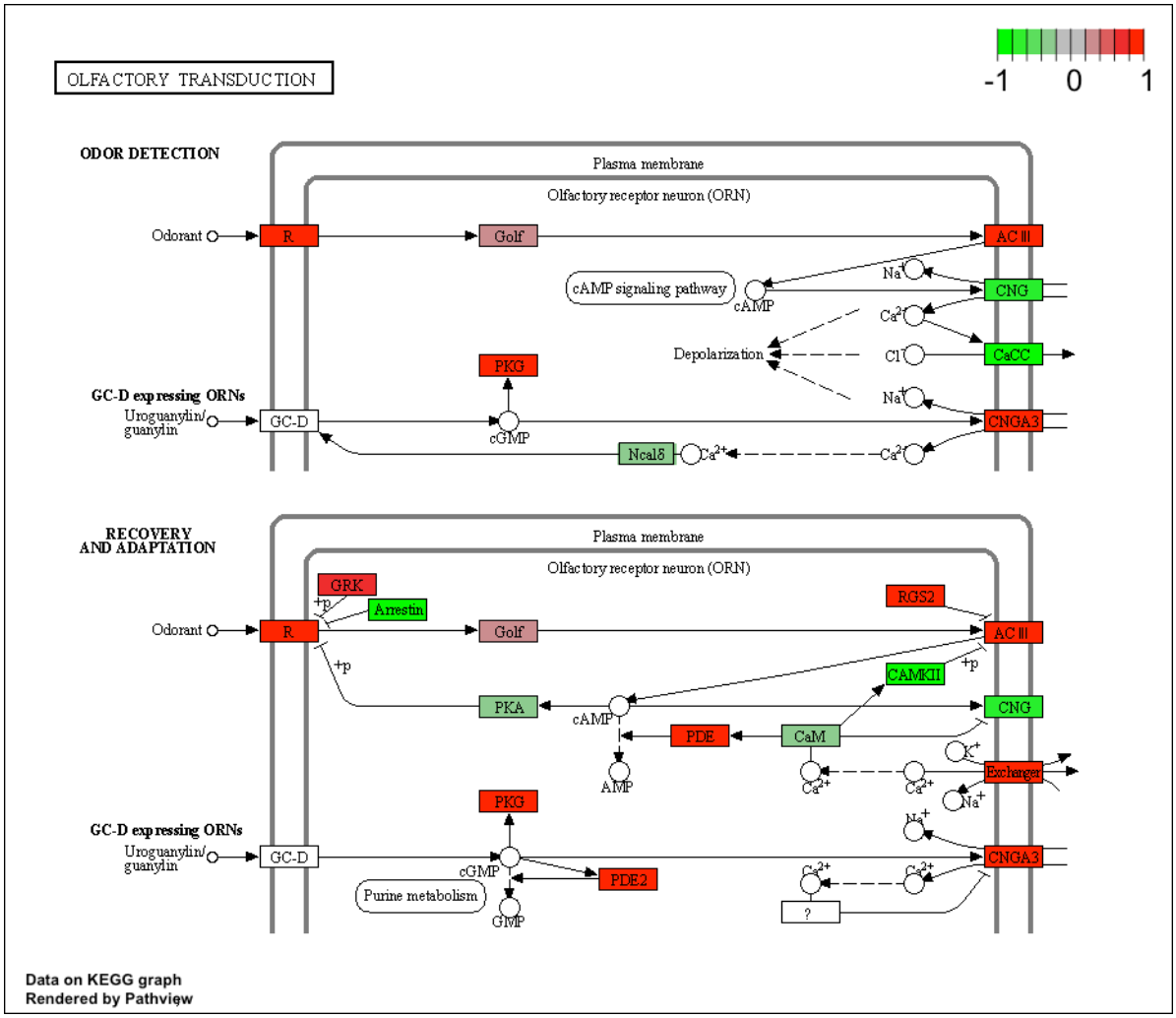


Figure 5: Gene Pathway

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

Info: Writing image file hsa03440.pathview.png

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

Info: Working in directory /Users/jameswoolley/Library/Mobile Documents/com~apple~CloudDocs/U

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

##Gene Ontology

We can do something similar with GO.

```
data(go.sets.hs)
data(go.subs.hs)
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       0.1952430      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

$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

$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
```

WE can look at hte results

```
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

Reactome is a database of biomolecules and how they work in a lot of pathways and processes. We can use Reactome to conduct overrepresentation enrichment analysis and pathway topology.

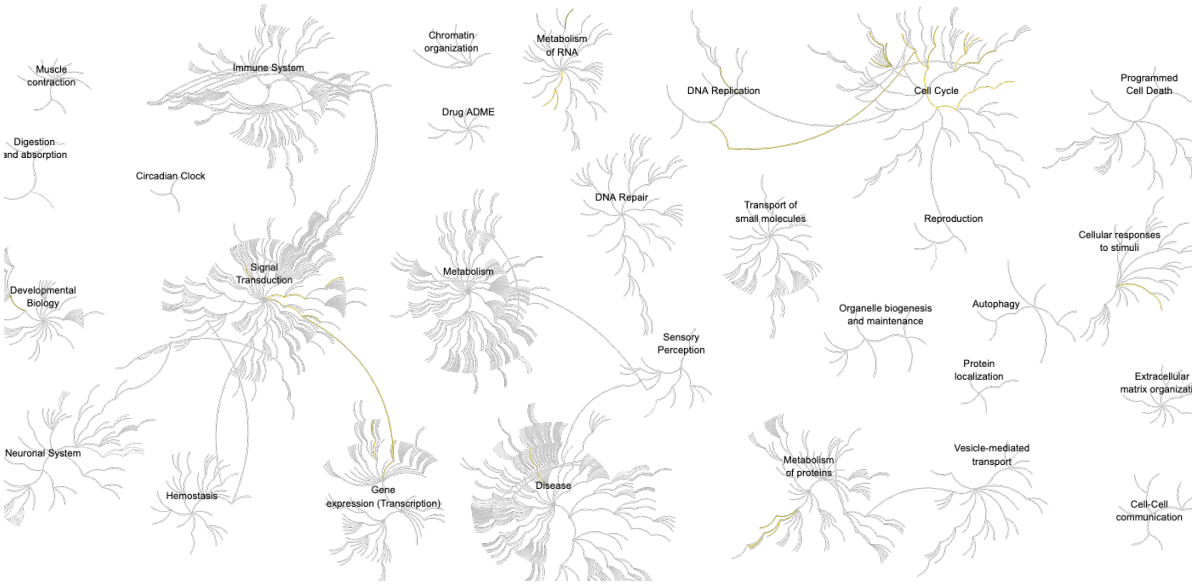
```
sig_genes <- res[res$padj <= 0.05 & !is.na(res$padj), "symbol"]
print(paste("Total number of significant genes:", length(sig_genes)))
```

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

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

This TXT file can be uploaded at the reactome website and analyzed.

> 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?



This lines up with the KEGG results. As we can see, the most significant pathways have to do with signal transduction, cell divisions (wnt, smad3,4, etc), and RNA metabolism.