Lab 15: RNAseq Pathway Analysis

Taylor Darby

11/19/2021

Background Notes:

1. Data import

2. PCA (for quality control)

3. DESeq analysis

Section 1. Differential Expression Analysis

1. Data import

```
# Load DESeq and our files
library(DESeq2)
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
```

```
# 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
```

Question 1: Complete the code below to remove the troublesome first column from countData

```
countmatrix <- as.matrix(countData[,2:7])
head(countmatrix)
```

```
##                 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
```

Question 2: 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.
countmatrix = countmatrix[rowSums(countmatrix) != 0, ]
head(countmatrix)
```

```
##                 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
```

There are 15975 genes left in the countData.

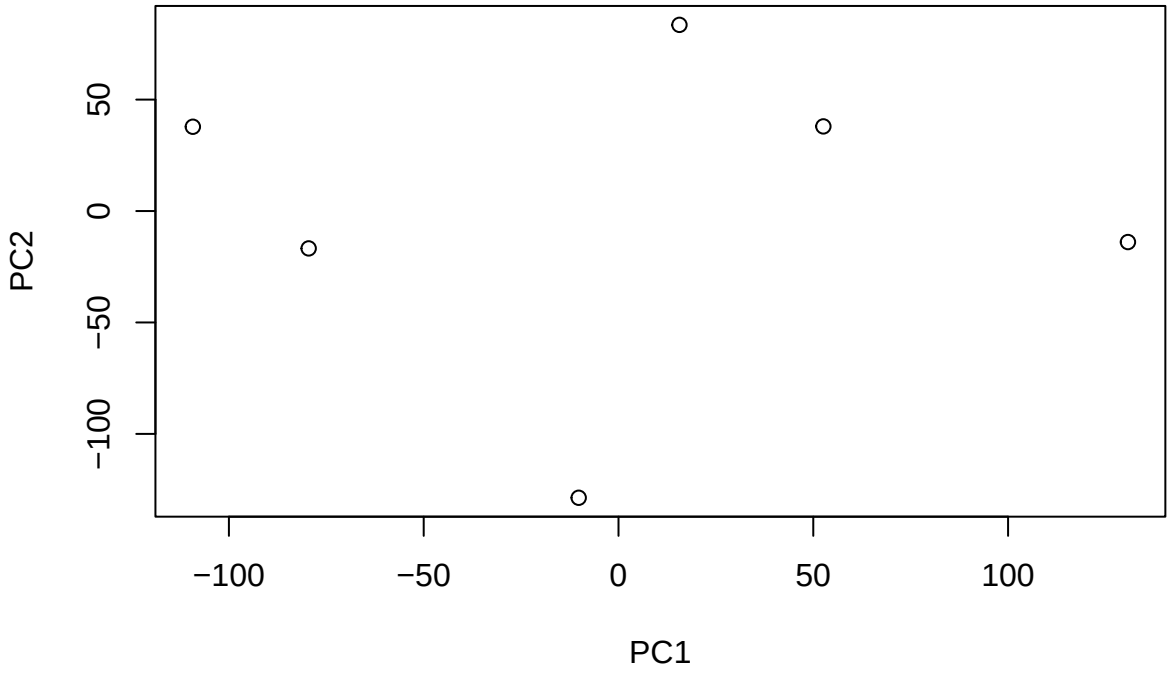
2. PCA (for quality control)

Our sample-level QC allows us to see how well our replicates cluster together, as well as, observe whether our experimental condition represents the major source of variation in the data. Performing sample-level QC can also identify any sample outliers, which may need to be explored further to determine whether they need to be removed prior to DE analysis.

```
# Perform PCA analysis on transformed data
countpca <- prcomp(t(countmatrix), scale = T)
head(countpca$x)
```

```
##                  PC1        PC2        PC3        PC4        PC5           PC6
## SRR493366 -109.25552   37.82004 -15.672855  -9.478082 -41.578831 -1.207945e-12
## SRR493367  -79.52470  -16.76884  -3.342885 -30.007959  44.778561 -1.432167e-15
## SRR493368  -10.21448 -128.65540   9.086014  30.145032  -7.226810  9.639088e-13
## SRR493369   15.64576   83.53988 -12.808025  46.849300  18.985338  3.863251e-13
## SRR493370   52.58209   37.98050  59.581067 -13.530217  -7.718522  5.743089e-13
## SRR493371  130.76685  -13.91619 -36.843317 -23.978074  -7.239737 -7.159919e-13
```

```
# plot transformed PCA
plot(countpca$x[,1:2])
```



3. DESeq analysis

```
dds = DESeqDataSetFromMatrix(countData=countmatrix,
                             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
```

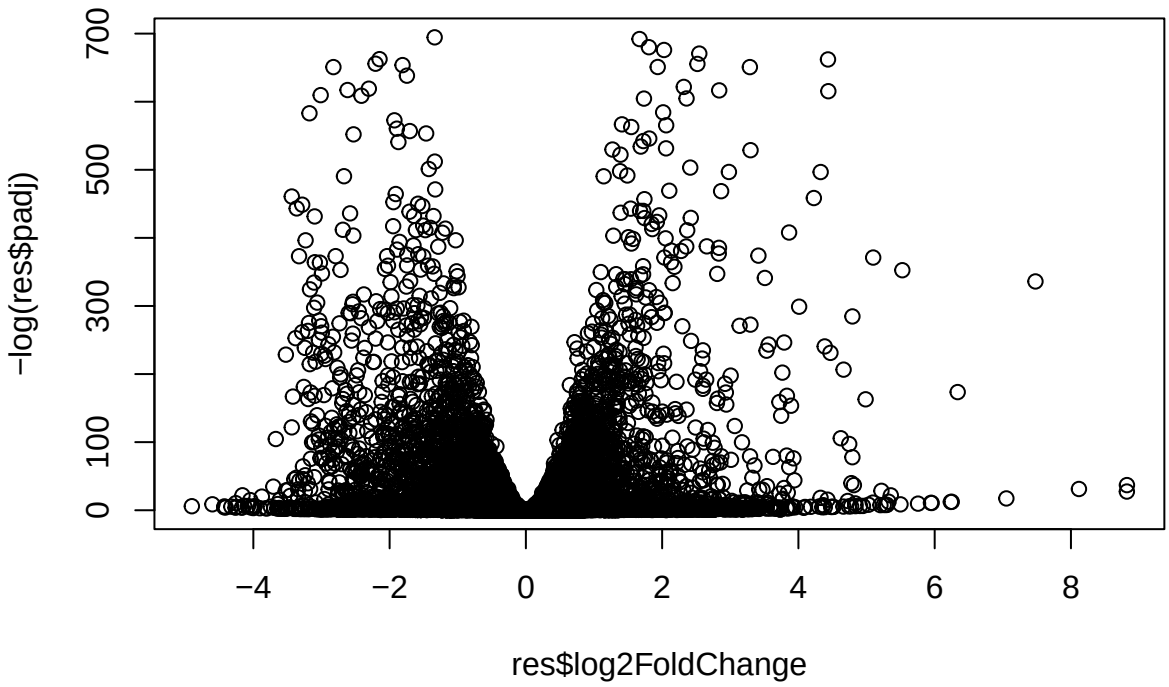
```
res <- results(dds)
```

Question 3: 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.

```
head(res)
```

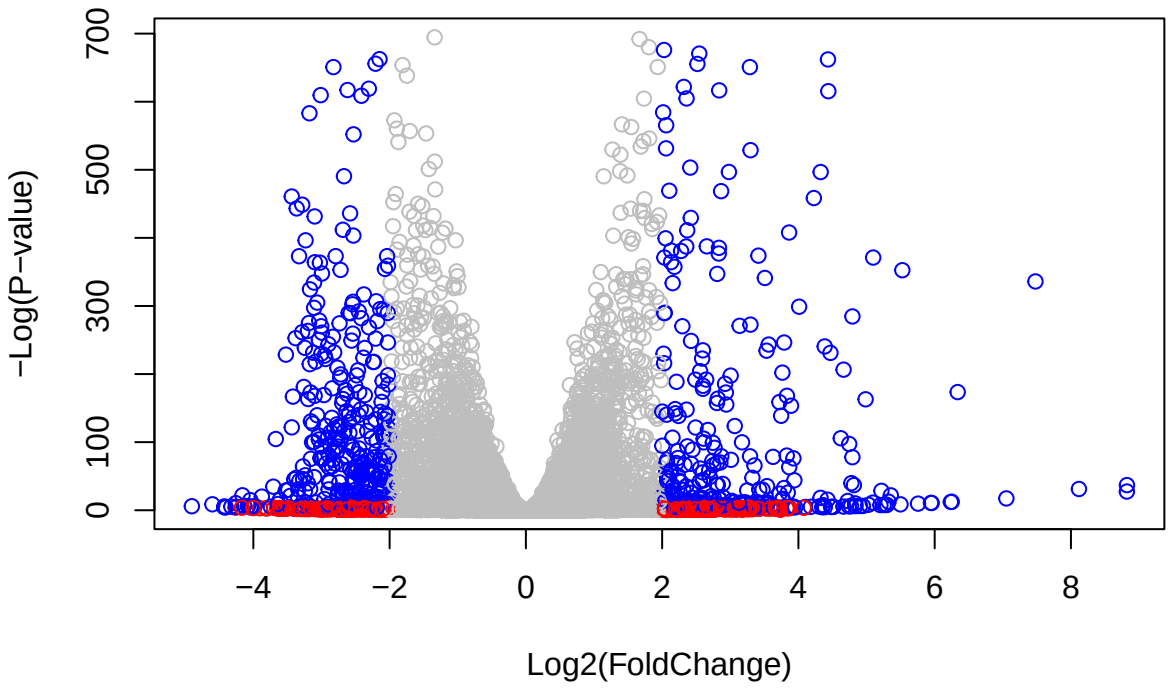
```
## log2 fold change (MLE): condition hoxa1 kd vs control sirna
## Wald test p-value: condition hoxa1 kd vs control sirna
## DataFrame with 6 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE       stat      pvalue
##                  <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000279457   29.9136      0.1792571 0.3248216   0.551863 5.81042e-01
## ENSG00000187634  183.2296      0.4264571 0.1402658   3.040350 2.36304e-03
## ENSG00000188976 1651.1881     -0.6927205 0.0548465 -12.630158 1.43990e-36
## ENSG00000187961  209.6379      0.7297556 0.1318599   5.534326 3.12428e-08
## ENSG00000187583   47.2551      0.0405765 0.2718928   0.149237 8.81366e-01
## ENSG00000187642   11.9798      0.5428105 0.5215598   1.040744 2.97994e-01
##                        padj
##                   <numeric>
## ENSG00000279457 6.86555e-01
## ENSG00000187634 5.15718e-03
## ENSG00000188976 1.76549e-35
## ENSG00000187961 1.13413e-07
## ENSG00000187583 9.19031e-01
## ENSG00000187642 4.03379e-01
```

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



Question 4: 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$pvalue < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

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



Question 5: 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=rownames(res),
                    keytype="ENSEMBL",
                    column="SYMBOL",
                    multiVals="first")
```

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

```
res$entrez = mapIds(org.Hs.eg.db,
                    keys=rownames(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      WASH9P   102723897 WAS protein family h..
## 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 ..
```

Question 6: 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

The gageData package has pre-compiled databases mapping genes to KEGG pathways and GO terms for common organisms. kegg.sets.hs is a named list of 229 elements. Each element is a character vector of member gene Entrez IDs for a single KEGG pathway. (See also go.sets.hs). The sigmet.idx.hs is an index of numbers of signaling and metabolic pathways in kegg.set.gs. In other words, KEGG pathway include other types of pathway definitions, like "Global Map" and "Human Diseases", which may be undesirable in a particular pathway analysis. Therefore, kegg.sets.hs[sigmet.idx.hs] gives you the "cleaner" gene sets of signaling and metabolic pathways only.

```
# First we need to do our one time install of these required bioconductor packages:
# 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") )
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)
```

```
# Import data:
data(kegg.sets.hs)
data(sigmet.idx.hs)

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

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

```
## $`hsa00232 Caffeine metabolism`
## [1] "10"   "1544" "1548" "1549" "1553" "7498" "9"
##
## $`hsa00983 Drug metabolism - other enzymes`
##  [1] "10"     "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
##  [9] "1553"   "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
```

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

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

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

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

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

```
# Now let's run the 'gage()' analysis:
keggres = gage(foldchanges, gsets=kegg.sets.hs)

attributes(keggres)
```

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

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

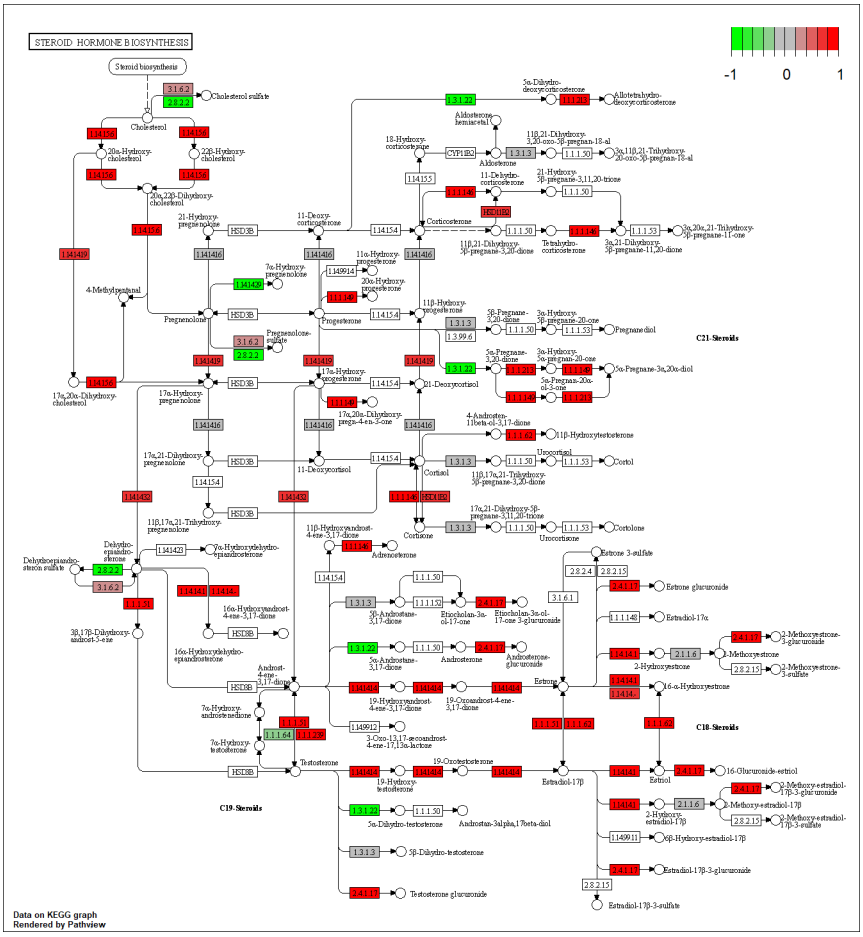
```
##                                        p.geomean stat.mean        p.val
## hsa04110 Cell cycle                   8.995727e-06 -4.378644 8.995727e-06
## hsa03030 DNA replication              9.424076e-05 -3.951803 9.424076e-05
## hsa03013 RNA transport                1.375901e-03 -3.028500 1.375901e-03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

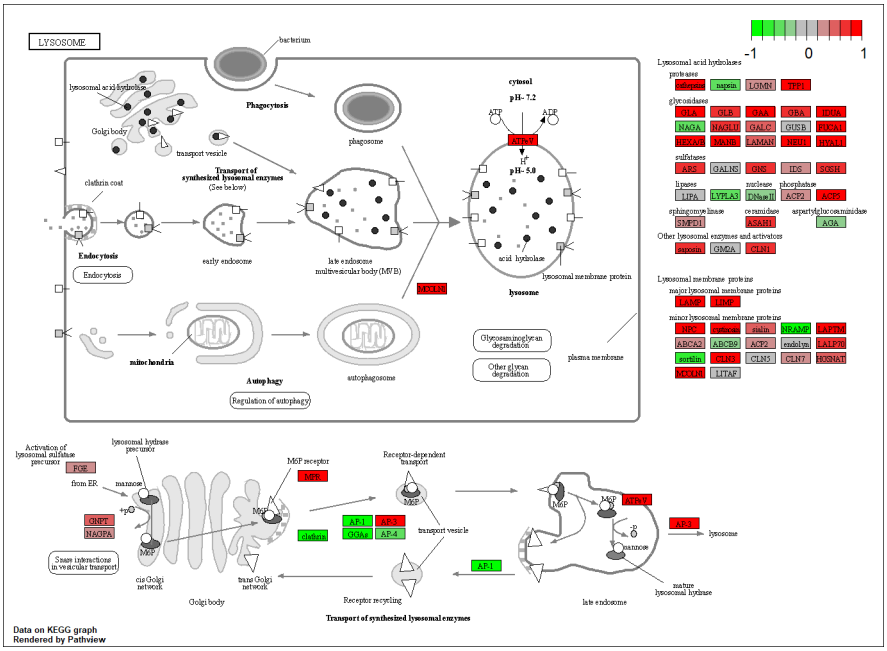
```
# Let's try out the pathview() function from the pathview package to make a pathway plot with our RNA-S

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

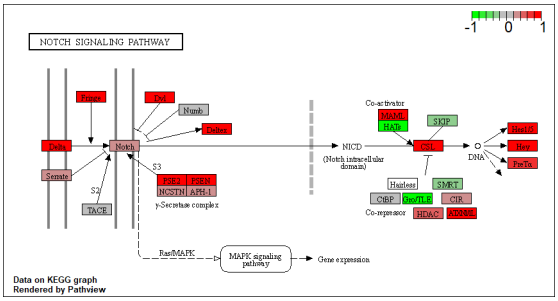
```
## Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

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

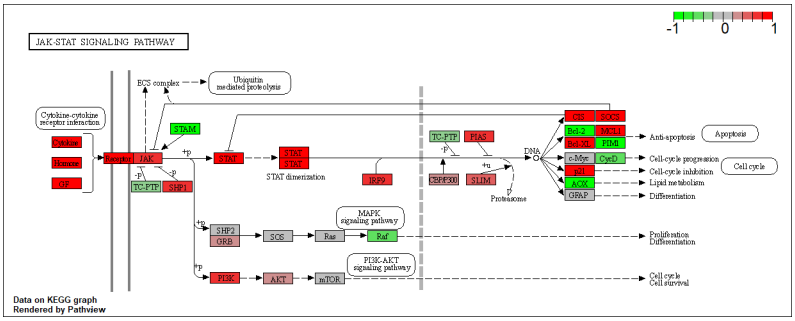
```
## [1] "hsa04640" "hsa04630" "hsa00140" "hsa04142" "hsa04330"
```

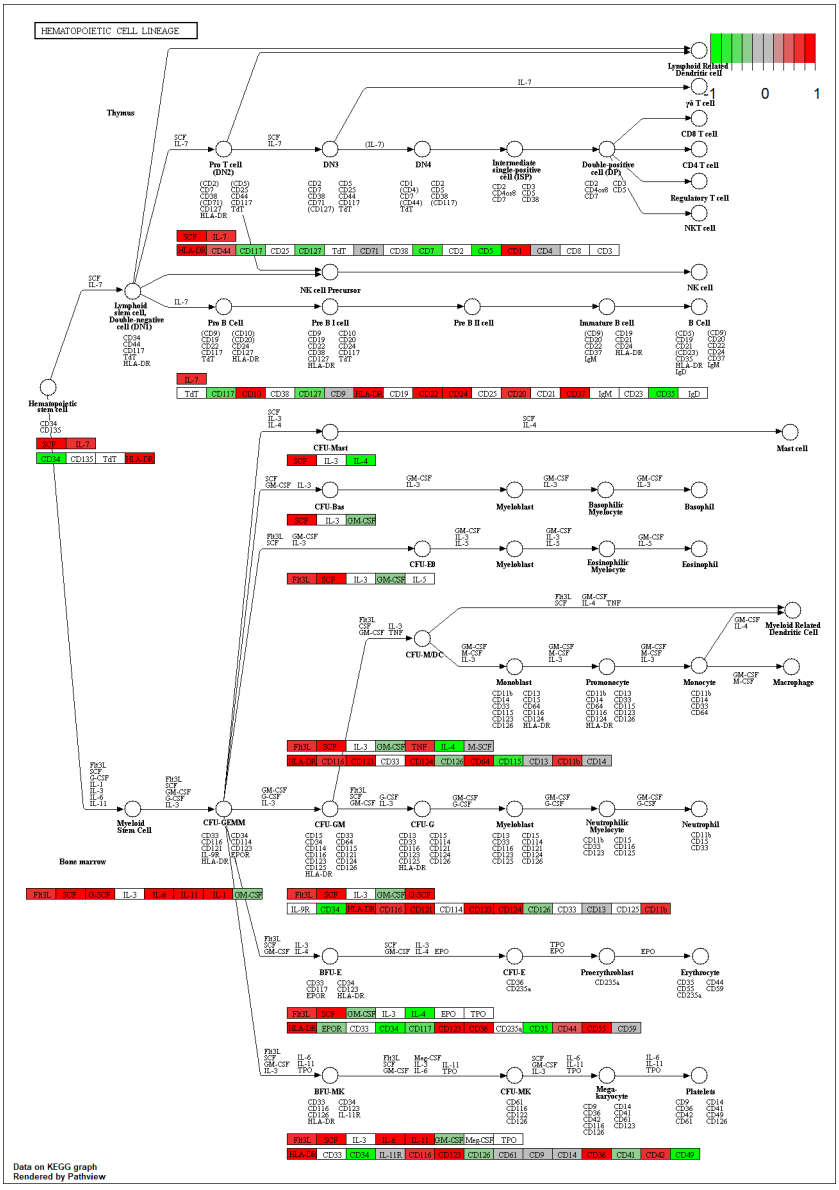
```
pathview(gene.data=foldchanges, pathway.id=keggresids, species="hsa")
```


STEROID HORMONE BIOSYNTHESIS
-1
0
1
Steroid biosynthesis
Cholesterol
Pregnenolone
Progesterone
Corticosterone
Aldosterone
Cortisol
Cortisone
Testosterone
Androstenedione
Androsterone
Estrone
Estradiol-17β
Estriol
C21-Steroids
C19-Steroids
C18-Steroids
Data on KEGG graph
Rendered by Pathview


LYSOSOME
-1
0
1
bacterium
Phagocytosis
phagosome
cytosol
pH- 7.2
pH- 5.0
ATP
ADP
ATPase
Golgi body
transport vesicle
lysosomal acid hydrolase
Transport of synthesized lysosomal enzymes
(See below)
clathrin coat
Endocytosis
early endosome
late endosome
multivesicular body (MVB)
acid hydrolase
lysosome
lysosomal membrane protein
mitochondria
Autophagy
Regulation of autophagy
autophagosome
Glycosaminoglycan degradation
Other glycan degradation
plasma membrane
Lysosomal acid hydrolases
proteases
glycosidases
sulfatases
lipases
nucleases
phosphatases
sphingomyelinase
ceramidase
aspartylglucosaminidase
Other lysosomal enzymes and activators
Lysosomal membrane proteins
major lysosomal membrane proteins
minor lysosomal membrane proteins
Activation of lysosomal sulfatase precursor
lysosomal hydrolase precursor
M6P receptor
Receptor-dependent transport
from ER
mannose
M6P
transport vesicle
clathrin
Snare interactions in vesicular transport
cis Golgi network
trans Golgi network
Golgi body
Receptor recycling
late endosome
mature lysosomal hydrolase
lysosome
Transport of synthesized lysosomal enzymes
Data on KEGG graph
Rendered by Pathview


-1 0 1
NOTCH SIGNALING PATHWAY
Fringe
Dvl
Numb
Deltex
Delta
Notch
Serrate
S3
S2
TACE
PSN2 PSEN
NCSTN APH-1
γ-Secretase complex
Co-activator
MAML
HAIR
SKIP
NICD
(Notch intracellular domain)
CSL
DNA
HesES
Hesr
FleYA
Hairless
SMRT
CtBP SunTLR CIR
Co-repressor
HDAC KMNT2C
Ras/MAPK
MAPK signaling pathway
Gene expression
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
-p
+u
SPIF30
SLIM
Proteasome
DNA
CIS SOCS
Bcl-2 MCL1
Bcl-XL PIM1
c-Myc CycD
p21
ACOX
GRAP
Anti-apoptosis
Apoptosis
Cell-cycle progression
Cell-cycle inhibition
Cell cycle
Lipid metabolism
Differentiation
+p
SHP2
GRB
SOS
Ras
Raf
MAPK signaling pathway
Proliferation
Differentiation
+p
PI3K
AKT
mTOR
PI3K-AKT signaling pathway
Cell cycle
Cell survival
Data on KEGG graph
Rendered by Pathview

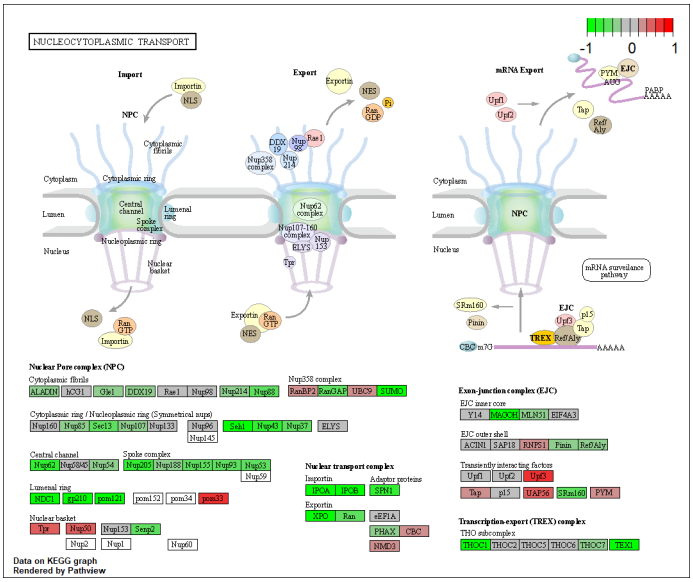


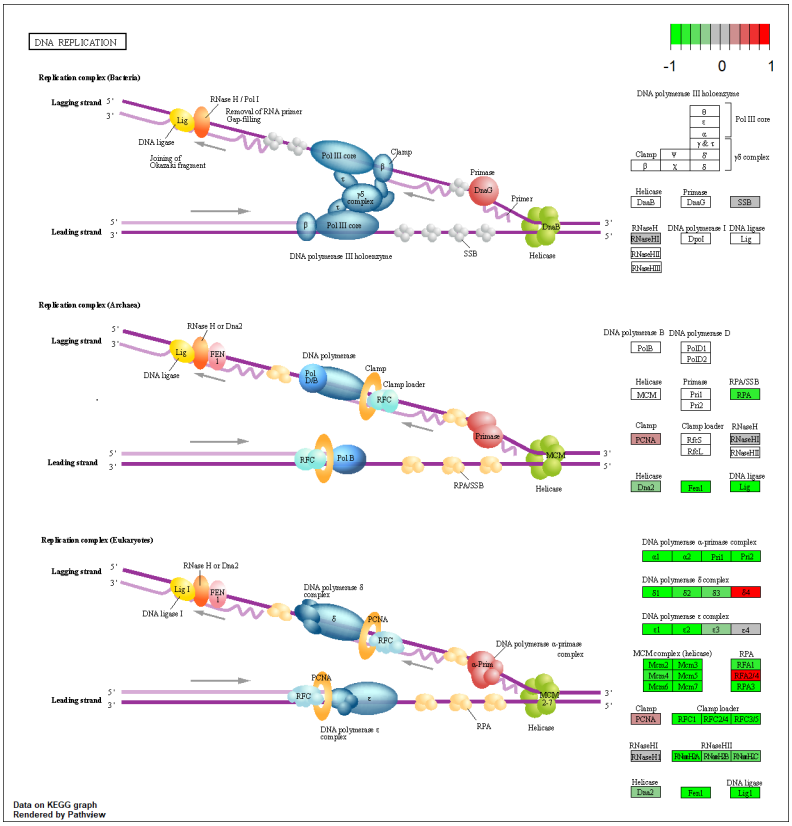
Question 7: Can you do the same procedure as above to plot the pathview figures for the top 5 down-regulated pathways?

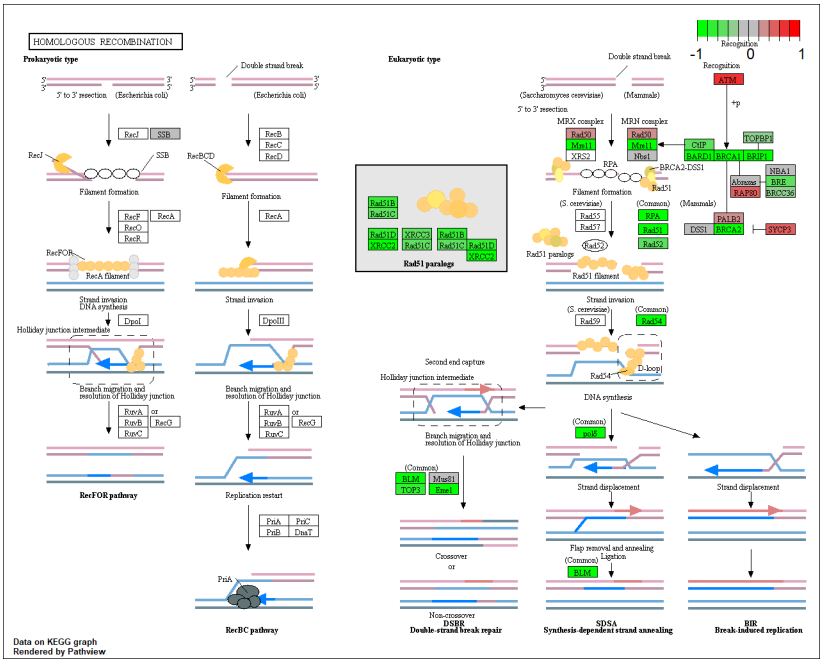
```
## Focus on top 5 downregulated pathways
keggrespathways2 <- rownames(keggres$less)[1:5]

# Extract the 8 character long IDs part of each string
keggresids2 = substr(keggrespathways2, start=1, stop=8)
keggresids2
```

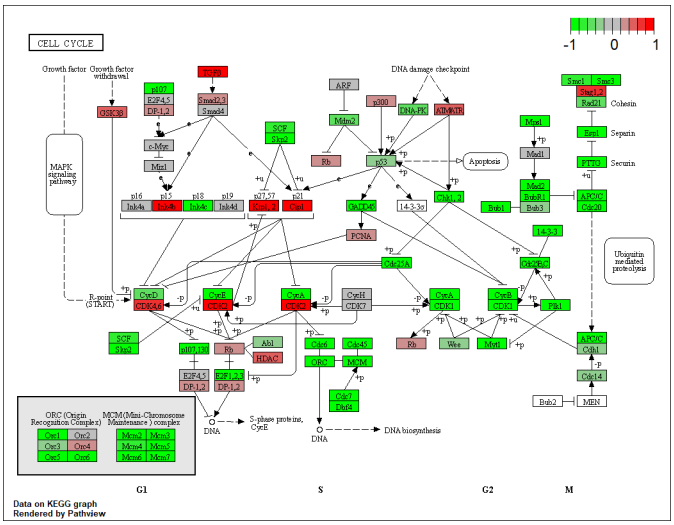
```
## [1] "hsa04110" "hsa03030" "hsa03013" "hsa03440" "hsa04114"
```

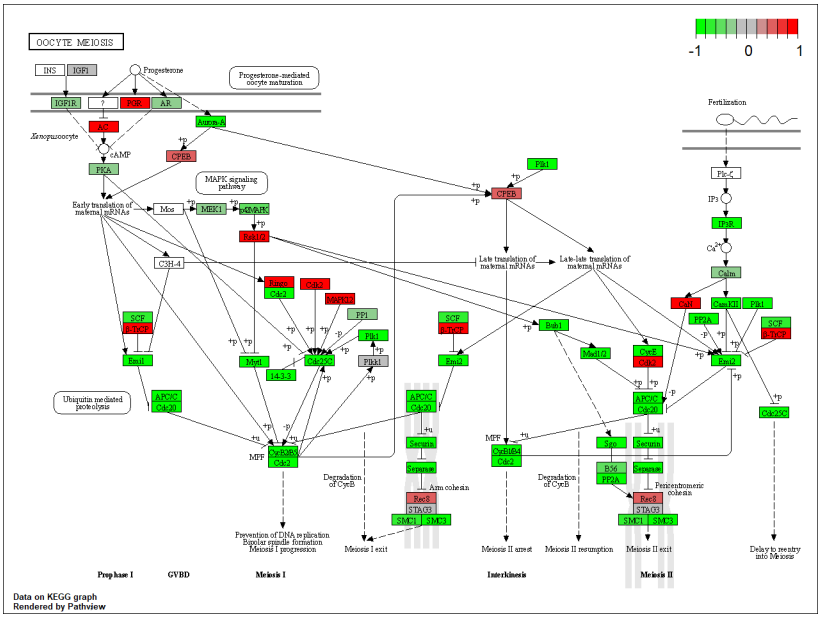
```
pathview(gene.data=foldchanges, pathway.id=keggresids2, species="hsa")
```


NUCLEOCYTOPLASMIC TRANSPORT
-1 0 1
Import
Export
mRNA Export
Importin
NLS
NPC
Cytoplasmic fibrils
Cytoplasm
Cytoplasmic ring
Lumen
Central channel
Spoke complex
Luminal ring
Nucleoplasmic ring
Nucleus
Nuclear basket
Ran GTP
Exportin
HES
Nup358 complex
Nup214
Nup62 complex
Nup107-160 complex
ELYS
Nup153
Tpr
Ras1
DDX19
Nup98
Upf1
Upf2
Tap
RefAly
PYM
EJC
PABP
AAAAA
mRNA surveillance pathway
SRm160
Pinin
Upf3
p15
TREX
CBC
m7G
Nuclear Pore complex (NPC)
Cytoplasmic fibrils
AladIN hCG1 Gle1 DDX19 Ras1 Nup98 Nup214 Nup88
Nup358 complex
RanBP2 RanGAP1 UBC9 SUMO
Cytoplasmic ring / Nucleoplasmic ring (Symmetrical unpd)
Nup160 Nup85 Seh1 Nup107 Nup133 Nup96 Nup145 Sah1 Nup43 Nup37 ELYS
Central channel
Nup62 Nup58/45 Nup54
Spoke complex
Nup205 Nup188 Nup155 Nup93 Nup53 Nup59
Luminal ring
MDT1 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
eIF1A
PHAX CBC
NMD3
Exon-junction complex (EJC)
EJC inner core
Y14 MAGOH MLN51 EIF4A3
EJC outer shell
ACIN1 SAP18 RNPS1 Pinin RefOAly
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


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
Lig
Pol III core
Clamp
β
γ complex
Primase
DnaG
Primer
DnaB
SSB
Helicase
DNA polymerase III holoenzyme
DNA polymerase III holoenzyme
θ
ε
α
γ & τ
Clamp
ψ
δ
χ
δ
Pol III core
γ complex
Helicase
DnaB
Primase
DnaG
SSB
RNaseH
RNaseHI
RNaseHII
RNaseHIII
DNA polymerase I
DpoI
DNA ligase
Lig
Replication complex (Archaea)
RNase H or Dna2
FEN
DNA polymerase
Pol B
Pol D
Clamp loader
RFC
Primase
MCM
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
DNA ligase
Fen1
Lig
Replication complex (Eukaryotes)
RNase H or Dna2
Lig I
FEN
DNA polymerase δ complex
DNA ligase I
δ
PCNA
RFC
DNA polymerase α-primase complex
α-Prim
ε
DNA polymerase ε complex
MCM2-7
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
RPA1 RPA2/4 RPA3
Clamp
PCNA
Clamp loader
RFC1 RFC2-4 RFC5
RNaseHI
RNaseH1
RNaseHII
RNaseH2A RNaseH2B RNaseH2C
Helicase
Dna2
DNA ligase
Fen1 Lig1
Data on KEGG graph
Rendered by Pathview


HOMOLOGOUS RECOMBINATION
Prokaryotic type
Eukaryotic type
Double strand break
(Escherichia coli)
(Saccharomyces cerevisiae)
(Mammals)
Recognition
-1
0
1
ATM
RecJ
SSB
RecBCD
RecA
RecF
RecO
RecR
RecFOR
Filament formation
RecA filament
Strand invasion
DNA synthesis
DpoI
DpoIII
Holliday junction intermediate
Branch migration and resolution of Holliday junction
RuvA
RuvB
RuvC
RecG
RecFOR pathway
Replication restart
PriA
PriB
PriC
DnaT
RecBC pathway
Rad51B
Rad51C
Rad51D
XRCC2
Rad51 paralogs
MRX complex
MRN complex
Mre11
Rad50
Nbs1
XRS2
RPA
BRCA2-DSS1
Rad51
(S. cerevisiae)
(Common)
(Mammals)
Rad55
Rad57
Rad52
Rad51 paralogs
Rad51 filament
Rad59
Rad54
D-loop
Second end capture
Holliday junction intermediate
Branch migration and resolution of Holliday junction
BLM
Mus81
Fmn1
Crossover
or
Non-crossover
DSBR
Double-strand break repair
DNA synthesis
pol δ
Strand displacement
Flap removal and annealing
Ligation
RLM
SDSA
Synthesis-dependent strand annealing
BIR
Break-induced replication
Data on KEGG graph
Rendered by Pathview


CELL CYCLE
-1
0
1
Growth factor
Growth factor withdrawal
MAPK signaling pathway
GSK3B
TGFB1
p107
E2F4,5
DP1,2
Smad2,3
Smad4
c-Myc
Mxi1
SCF
Skp2
ARF
Mdm2
p300
DNA damage checkpoint
DNA-PK
ATM/ATR
Rb
p53
Apoptosis
p16
Ink4a
p15
Ink4b
p18
Ink4c
p19
Ink4d
p27,57
p21
Cip1
Gfi1/36
14-3-3σ
Chk1,2
Bub1
Mad1
Mad2
Bub1B
Bub3
APC/C
Cdc20
14-3-3
PCNA
Cdc25A
Cdc25C
Ubiquitin mediated proteolysis
R-point
(STRKT)
CycD
CDK4,6
CycE
CDK2
CycA
CDK2
CycH
CDK7
CycA
CDK1
CycB
CDK1
Plk1
Abl
HDAC
Cdc45
MCM
Rb
Wee1
Myt1
Cdc7
Dbf4
Cdc14
MEN
Bub2
Smc1
Smc3
Rad21
Cohesin
Esp1
Separin
PTTG
Securin
ORC (Origin Recognition Complex)
MCM(Mini-Chromosome Maintenance) complex
Orc1
Orc2
Orc3
Orc4
Orc5
Orc6
Mcm2
Mcm3
Mcm4
Mcm5
Mcm6
Mcm7
S-phase proteins, CycE
DNA
DNA biosynthesis
G1
S
G2
M
Data on KEGG graph
Rendered by Pathview



Section 3. Gene Ontology (GO)

```
# Import data:
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       0.1951953      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                       0.2243795      427 2.195494e-04
## GO:0060562 epithelial tube morphogenesis  0.3711390      257 5.932837e-04
## GO:0035295 tube development               0.3711390      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                        3.530241 3.530241
## GO:0060562 epithelial tube morphogenesis   3.261376 3.261376
## GO:0035295 tube development                3.253665 3.253665
```

Section 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).

If you would like more information, the documentation is available here: https://reactome.org/user/guide

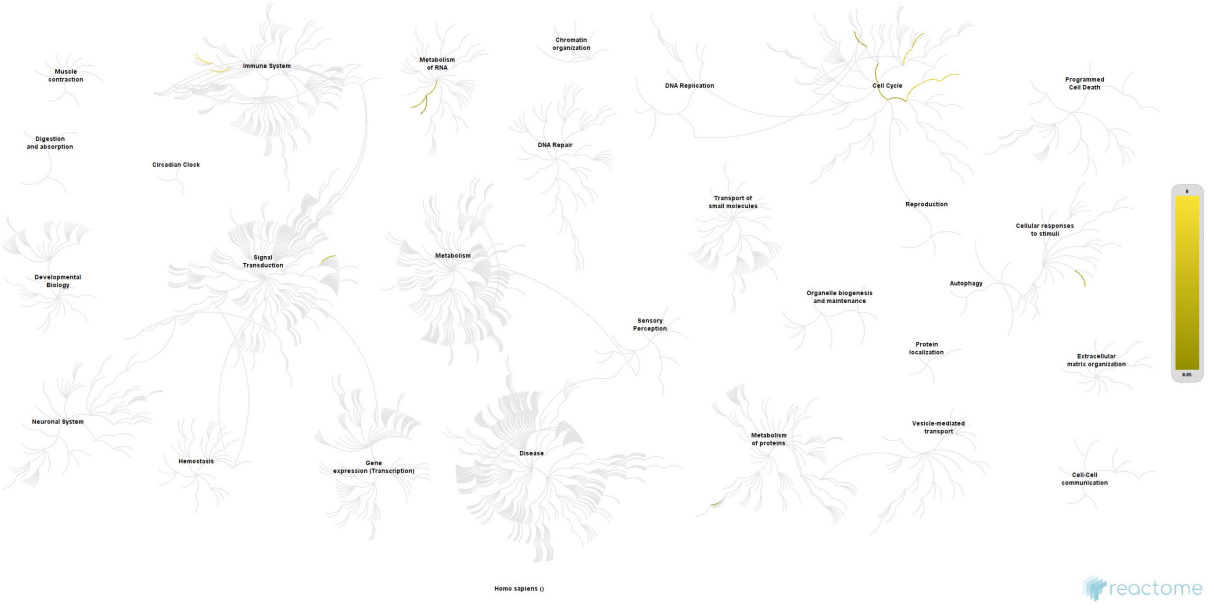
Let's now conduct over-representation enrichment analysis and pathway-topology analysis with Reactome using the previous list of significant genes generated from our differential expression results above.

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



Question 8: 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 with the most significant "Entities p-value" is the endosomal/vacuolar pathway

Section 5. GO online (OPTIONAL)

Question 9: 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?

Session Information

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
```

```
## [8] base
##
## other attached packages:
##  [1] gageData_2.32.0             gage_2.44.0
##  [3] pathview_1.34.0             org.Hs.eg.db_3.14.0
##  [5] AnnotationDbi_1.56.2        DESeq2_1.34.0
##  [7] SummarizedExperiment_1.24.0 Biobase_2.54.0
##  [9] MatrixGenerics_1.6.0        matrixStats_0.61.0
## [11] GenomicRanges_1.46.0        GenomeInfoDb_1.30.0
## [13] IRanges_2.28.0              S4Vectors_0.32.2
## [15] BiocGenerics_0.40.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2             bit64_4.0.5            splines_4.1.1
##  [4] highr_0.9              blob_1.2.2             GenomeInfoDbData_1.2.7
##  [7] yaml_2.2.1             pillar_1.6.3           RSQLite_2.2.8
## [10] lattice_0.20-45        glue_1.4.2             digest_0.6.28
## [13] RColorBrewer_1.1-2     XVector_0.34.0         colorspace_2.0-2
## [16] htmltools_0.5.2        Matrix_1.3-4           XML_3.99-0.8
## [19] pkgconfig_2.0.3        genefilter_1.76.0      zlibbioc_1.40.0
## [22] GO.db_3.14.0           purrr_0.3.4            xtable_1.8-4
## [25] scales_1.1.1           BiocParallel_1.28.0    tibble_3.1.5
## [28] annotate_1.72.0        KEGGREST_1.34.0        generics_0.1.0
## [31] ggplot2_3.3.5          ellipsis_0.3.2         cachem_1.0.6
## [34] survival_3.2-13        magrittr_2.0.1         crayon_1.4.1
## [37] KEGGgraph_1.54.0       memoise_2.0.0          evaluate_0.14
## [40] fansi_0.5.0            graph_1.72.0           tools_4.1.1
## [43] lifecycle_1.0.1        stringr_1.4.0          munsell_0.5.0
## [46] locfit_1.5-9.4         DelayedArray_0.20.0    Biostrings_2.62.0
## [49] compiler_4.1.1         rlang_0.4.11           grid_4.1.1
## [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_2.38.0
## [64] stringi_1.7.5          parallel_4.1.1         Rcpp_1.0.7
## [67] vctrs_0.3.8            geneplotter_1.72.0     png_0.1-7
## [70] tidyselect_1.1.1       xfun_0.26
```