class16

Hyeonseok Jang (PID# A59011126)

11/19/2021

Background

> 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 an differential analysis of lung fibroblasts in response to loss of the developmental transcription factor HOXA1.

1. Data Import

Read in the countdata and coldata that we need, and have a wee look.

```
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
```

```
# 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
```

```
# Filter count data where you have 0 read count across all samples.
counts <- countData[rowSums(countData)!=0,]
head(counts)
```

```
##                 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
```

2. PCA for Quality Control

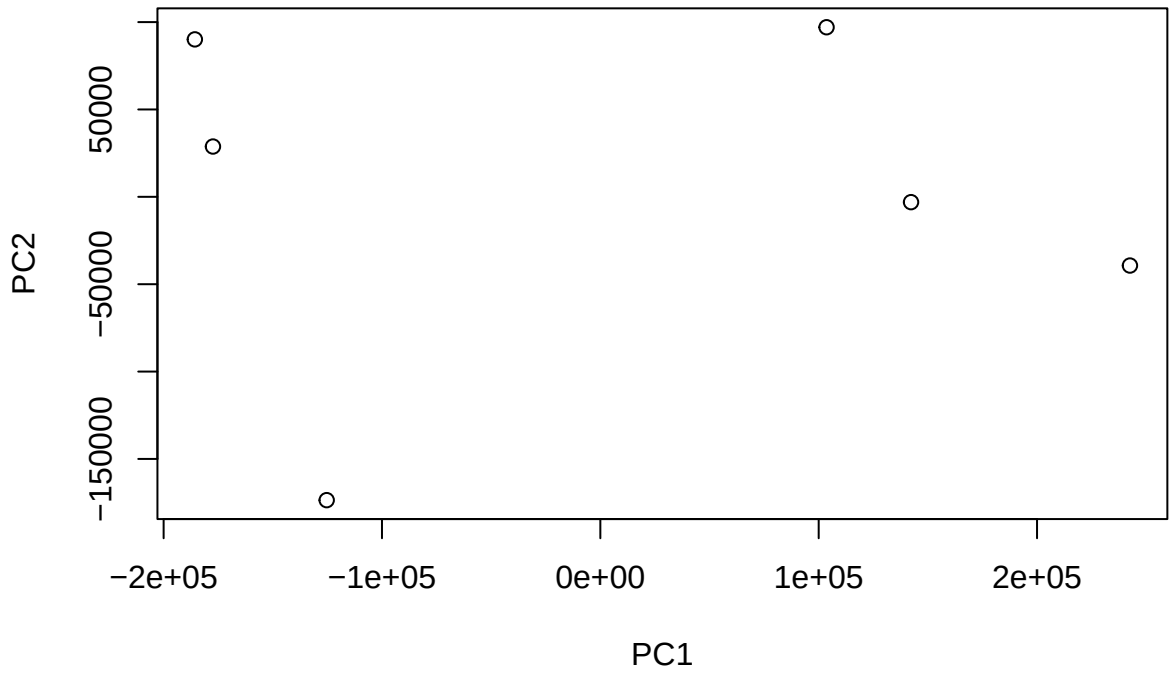
I am going to use the base R 'prcomp()' function for PCA of our counts data (form which I have removed the zero count genes).

```
pca <- prcomp(t(counts))
summary(pca)
```

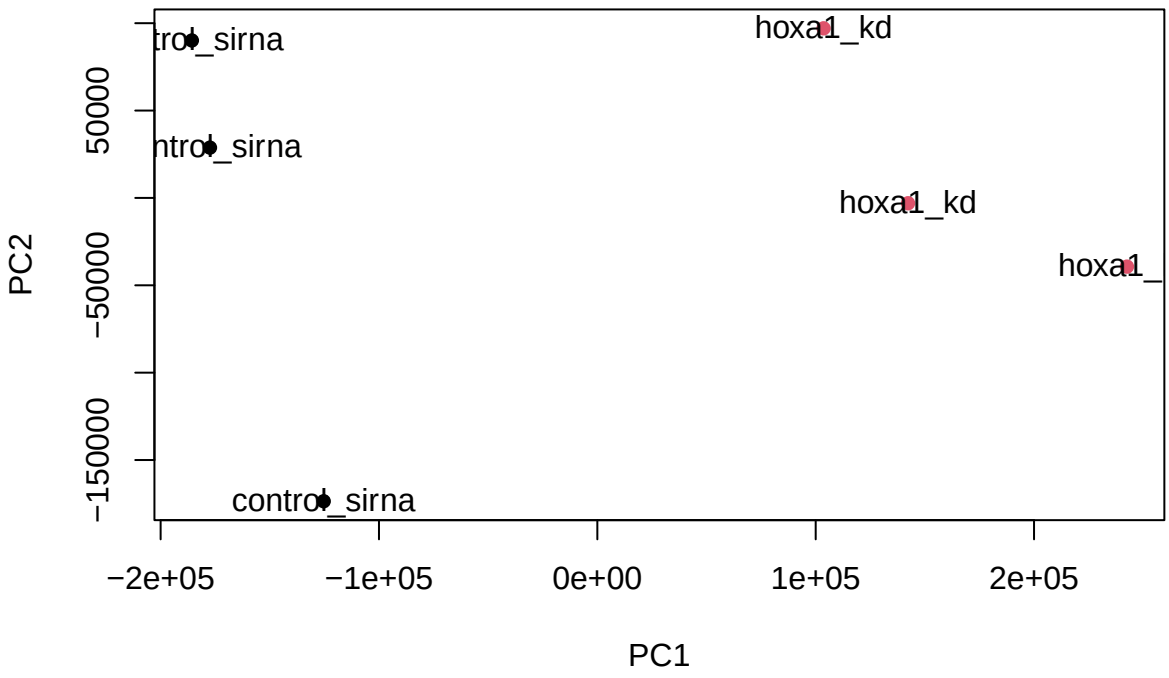
```
## Importance of components:
##                             PC1       PC2       PC3       PC4      PC5
## Standard deviation     1.852e+05 1.001e+05 1.998e+04 6.886e+03 5.15e+03
## Proportion of Variance 7.659e-01 2.235e-01 8.920e-03 1.060e-03 5.90e-04
## Cumulative Proportion  7.659e-01 9.894e-01 9.983e-01 9.994e-01 1.00e+00
##                             PC6
## Standard deviation     9.558e-10
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

Quick plot

```
plot(pca$x[,1:2])
```

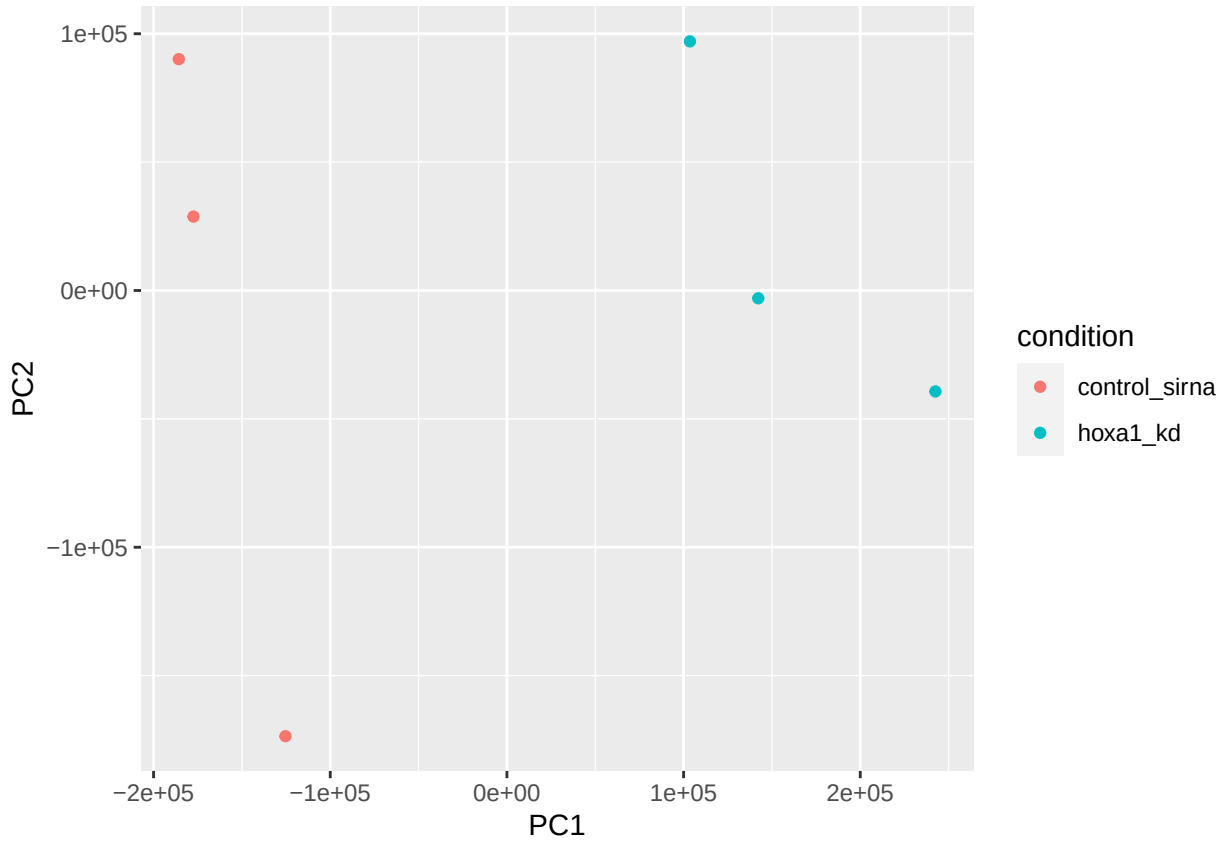



```
plot(pca$x[,1:2], pch=16, col=as.factor(colData$condition))
text(pca$x[,1:2], labels = colData$condition)
```



Or a ggplot version

```
library(ggplot2)

x <- as.data.frame(pca$x)
x$condition <- colData$condition

ggplot(x) +
  aes(PC1, PC2, col=condition) +
  geom_point()
```



This looks fine - the first PC separates the control group and experimental group well.

3. DESeq analysis

```
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, 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 objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## 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
```

```
dds = DESeqDataSetFromMatrix(countData=counts,
                             colData=colData,
                             design=~condition)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

And run the results.

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

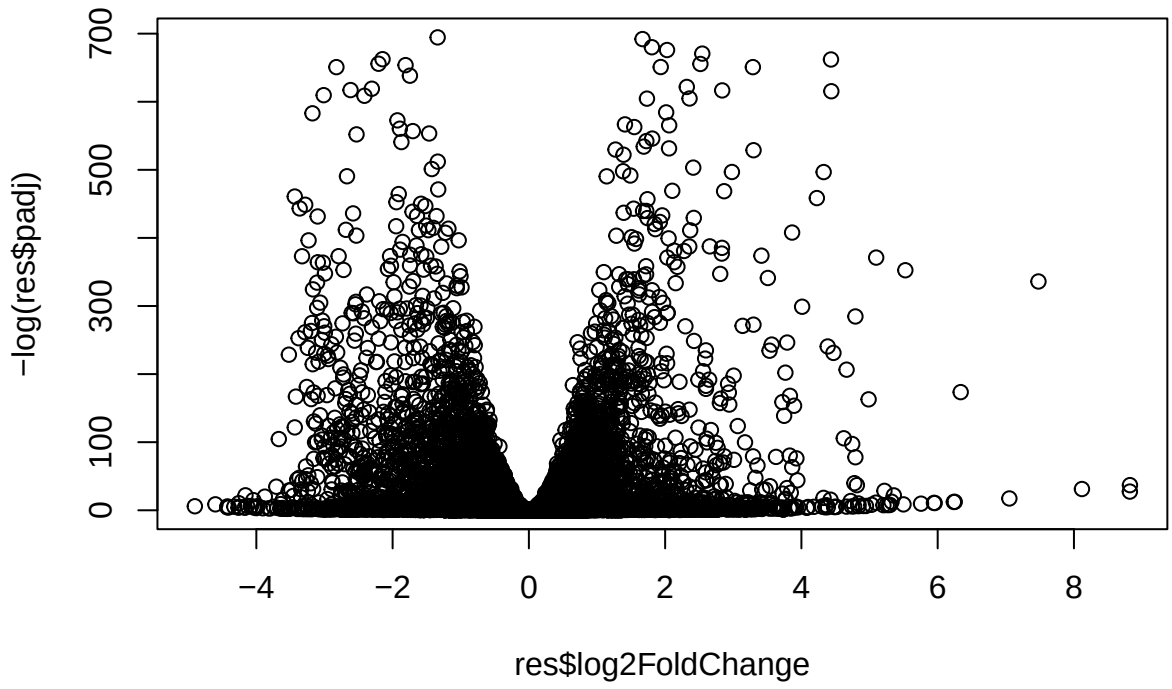
```
res <- results(dds)

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
```

4. Volcano Plot

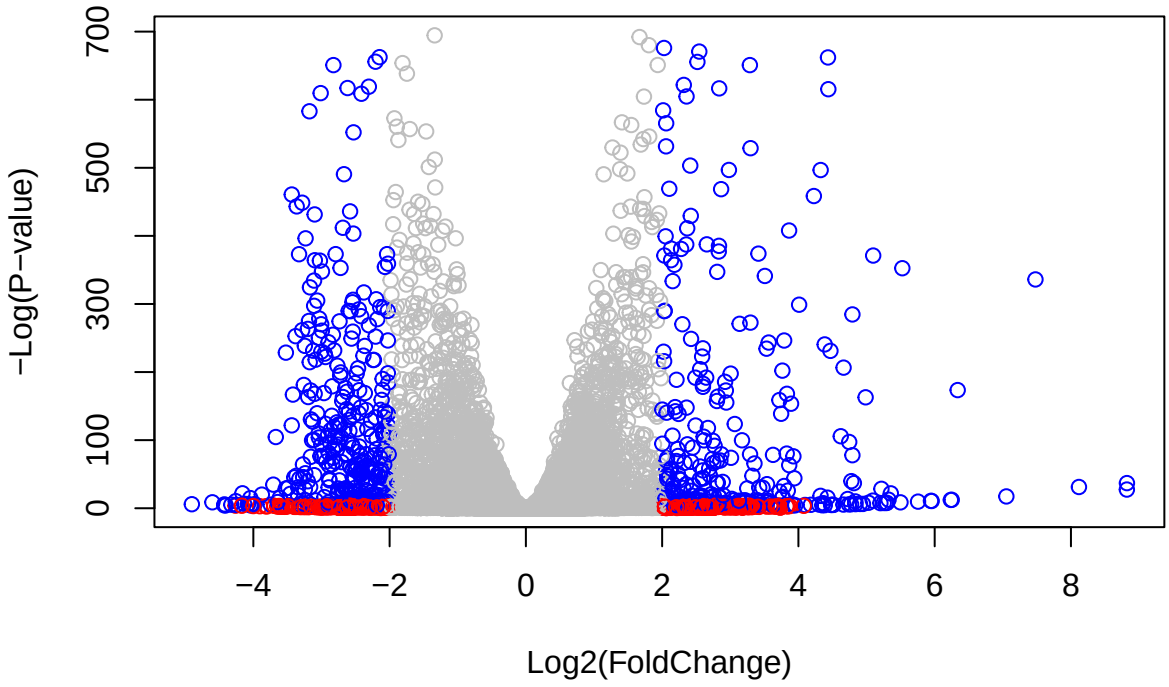
Let's make a volcano plot.

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



I can improve this plot by 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)")
```



5. Annotation

I can use the mapIDs() function multiple times to add SYMBOL, ENTREZID and GENENAME annotation to our results by 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      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 ..
```

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")
```

6. Pathway Analysis

I can load the packages and setup the KEGG data-sets we need.

```
library(pathview)
```

```
library(gage)
```

```
##
```

```
library(gageData)

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"
```

Make the input foldchange vector for KEGG and GO etc.

```
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 pathway 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
```

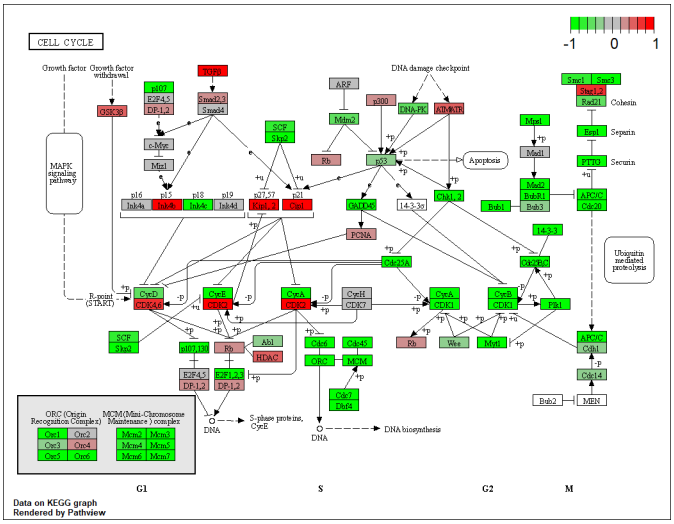
Let's try out the **pathview()** function from the pathview package to make a pathway plot with our RNA-Seq expression results shown in color.

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

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

```
## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16
```

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



A different PDF based output of the same data

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

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

```
## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16
```

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

```
## 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")
```

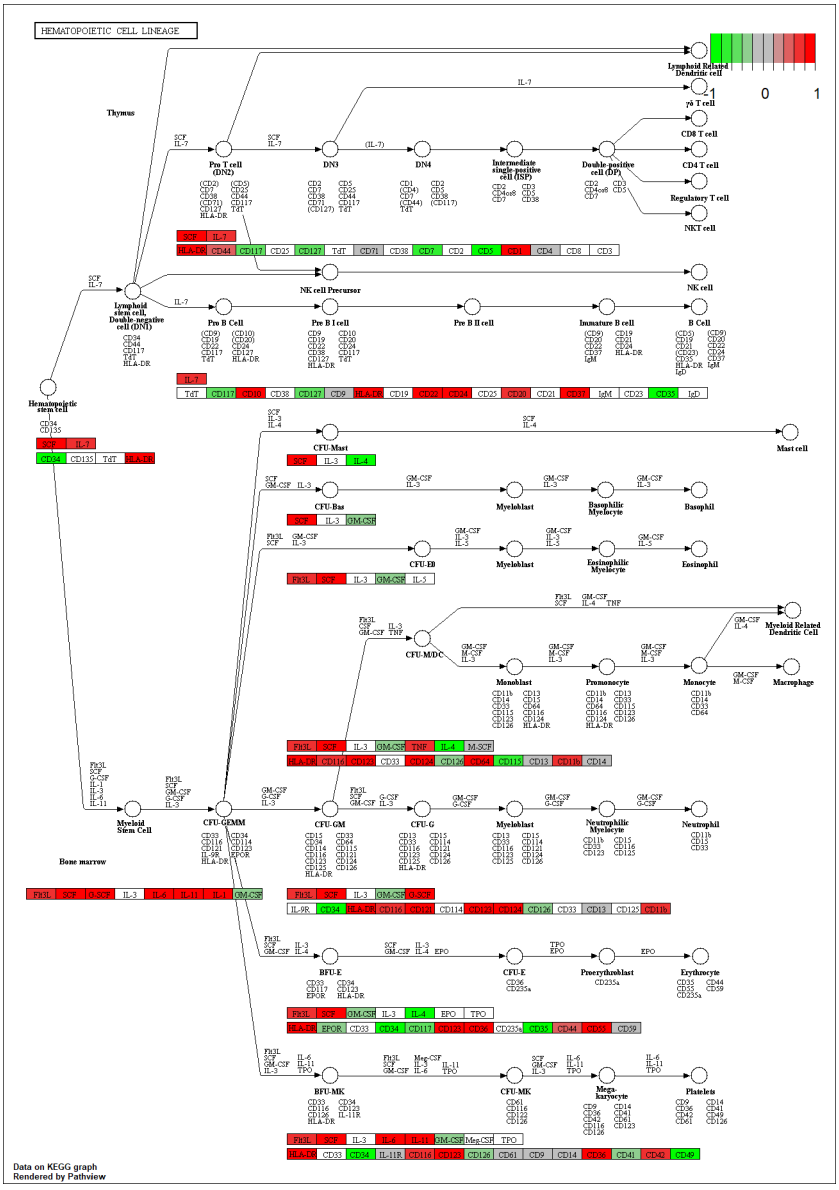
```
## 'select()' returned 1:1 mapping between keys and columns
```

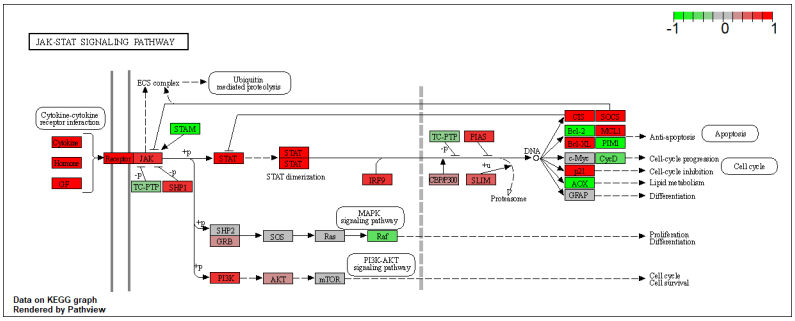
```
## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16
```

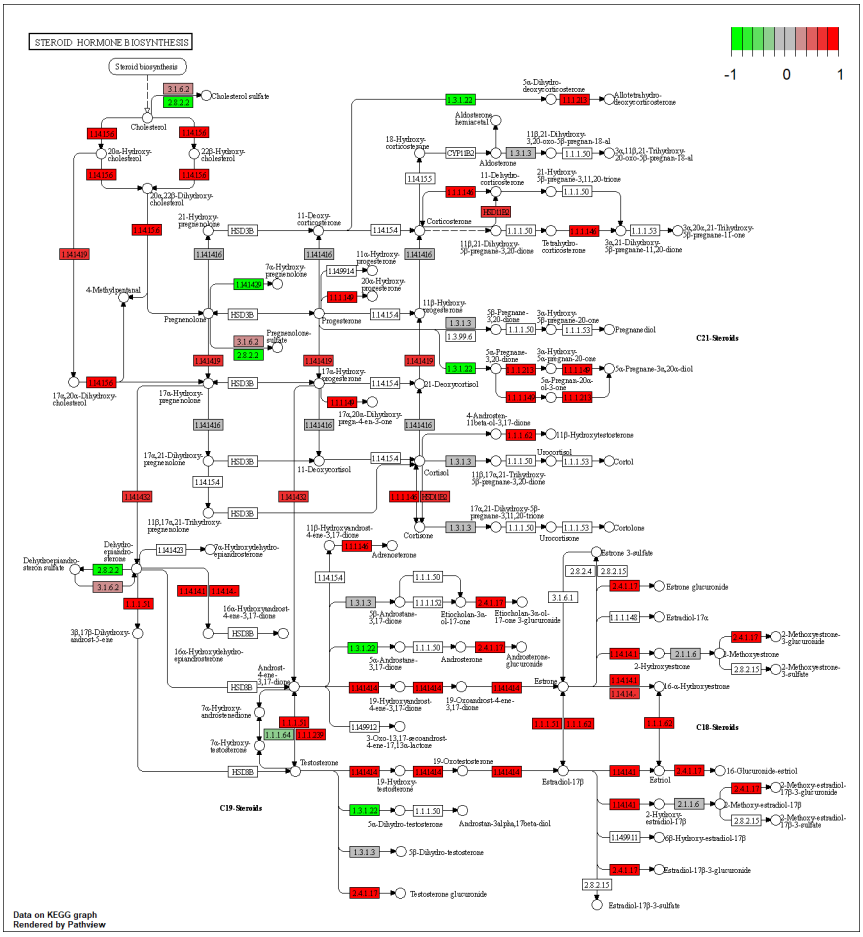
```
## Info: Writing image file hsa04640.pathview.png
```

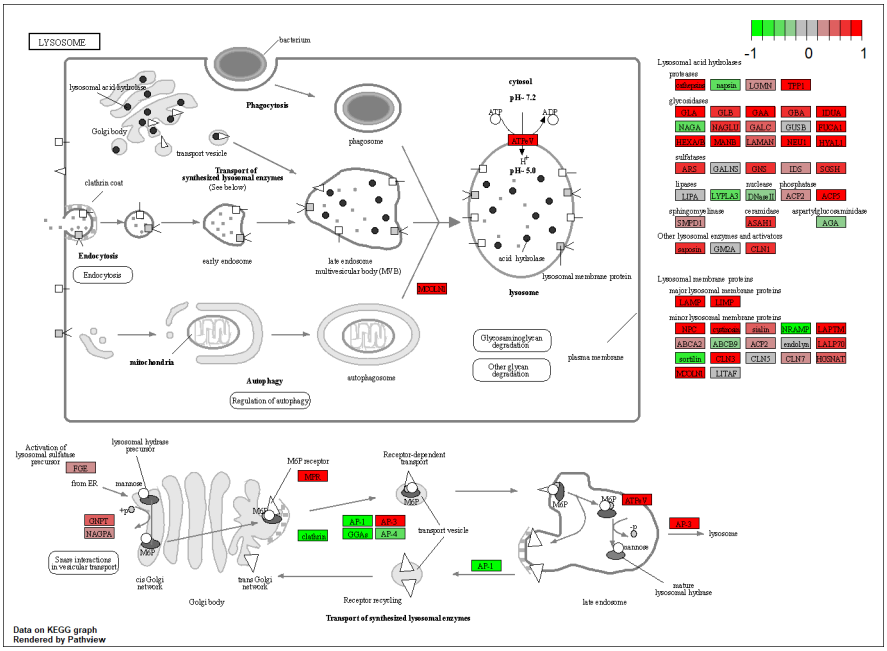
```
## 'select()' returned 1:1 mapping between keys and columns
```

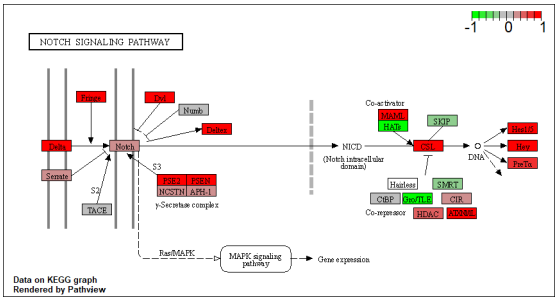
```
## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

## 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 C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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


HEMATOPOIETIC CELL LINEAGE
Lymphoid Related Dendritic cell
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
Imature B cell
B Cell
Hematopoietic stem cell
CFU-Mast
Mast cell
CFU-Ba
Myeloblast
Basophilic Myelocyte
Basophil
CFU-Eo
Eosinophilic Myelocyte
Eosinophil
CFU-M/DC
Myeloid Related Dendritic Cell
Monoblast
Promonocyte
Monocyte
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
Ubiquitin mediated proteolysis
Cytokine-cytokine receptor interaction
Cytokine
Hormone
GF
Receptor
JAK
STAT
STAT dimerization
TC-PTP
SHP1
SHP2
ORB
SOS
Ras
Raf
MAPK signaling pathway
PI3K
AKT
mTOR
PI3K-AKT signaling pathway
Proliferation Differentiation
Cell cycle Cell survival
Anti-apoptosis
Apoptosis
Cell-cycle progression
Cell cycle
Cell-cycle inhibition
Lipid metabolism
Differentiation
Data on KEGG graph
Rendered by Pathview


STEROID HORMONE BIOSYNTHESIS
-1
0
1
Steroid biosynthesis
Cholesterol
Cholesterol sulfate
Pregnenolone
Progesterone
Corticosterone
11-Deoxycorticosterone
Aldosterone
Androstenedione
Testosterone
Estradiol-17β
Estrone
Cortisol
Cortisone
Dehydroepiandrosterone sulfate
Androsterone
Androstan-3alpha,17beta-diol
Pregnanediol
Estriol
Estrone 3-sulfate
Estradiol-17β-3-sulfate
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
lysosomal acid hydrolase
Golgi body
transport vesicle
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



I can do the same procedure as above to plot the pathview figures for the top 5 down-reguled pathways.

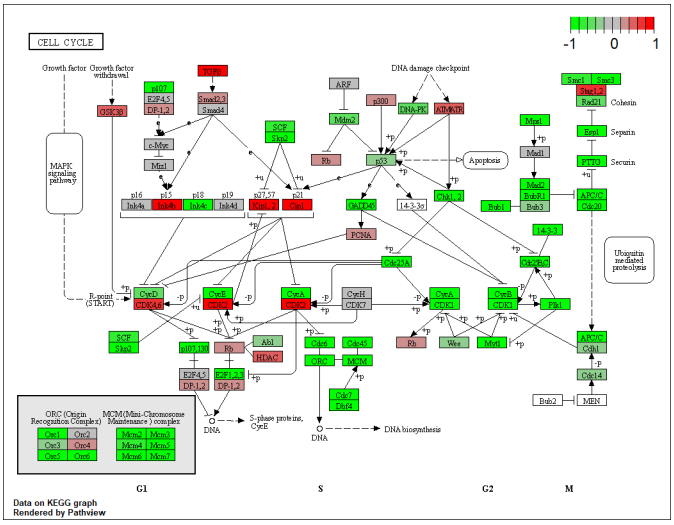
```
keggrespathways.down <- rownames(keggres$less)[1:5]

keggresids.down = substr(keggrespathways.down, start=1, stop=8)
keggresids.down
```

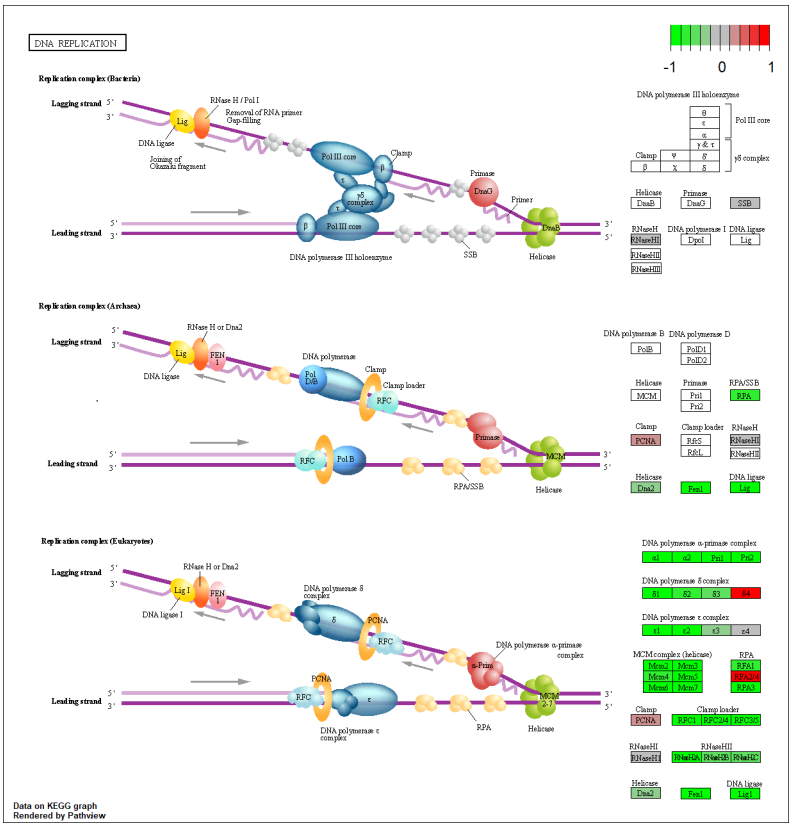
```
## [1] "hsa04110" "hsa03030" "hsa03013" "hsa03440" "hsa04114"
```

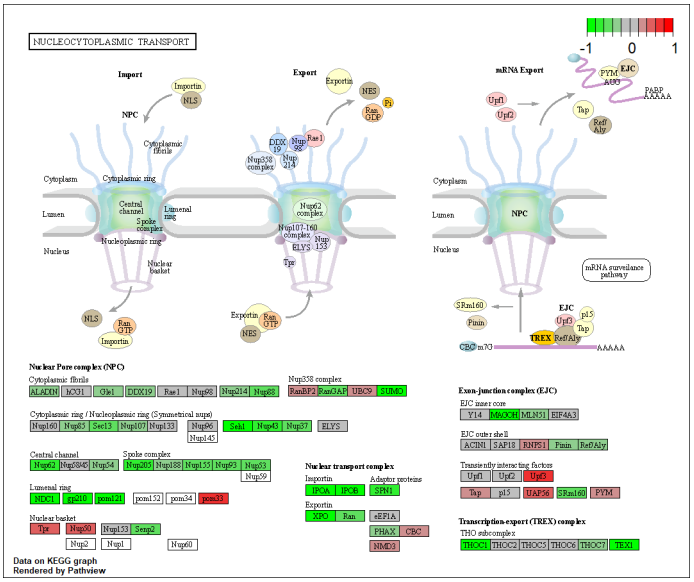
```
pathview(gene.data=foldchanges, pathway.id=keggresids.down, species="hsa")
```

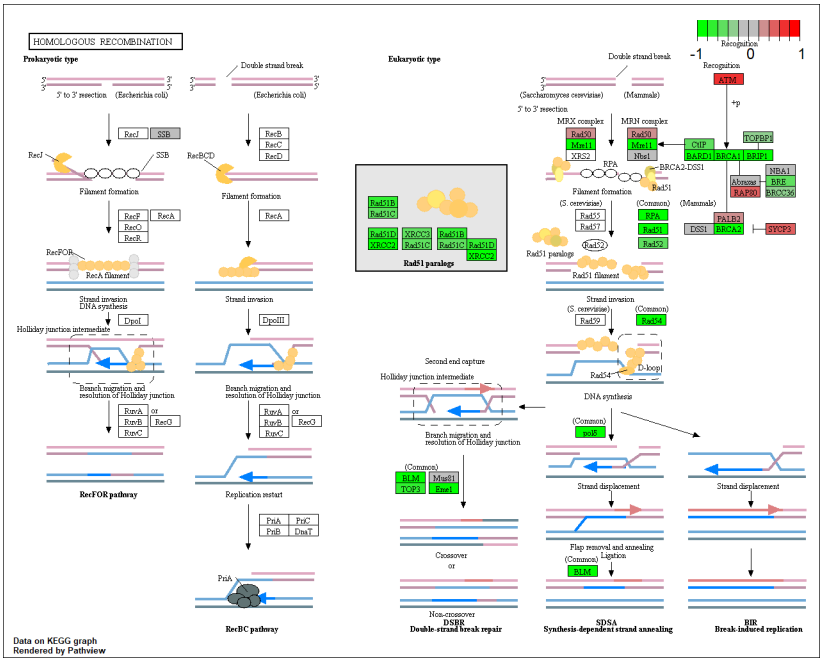
```
## 'select()' returned 1:1 mapping between keys and columns

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16

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

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

## Info: Working in directory C:/Users/Hyeonseok/Desktop/BGGN213/BGGN213_GitHub/class16
```

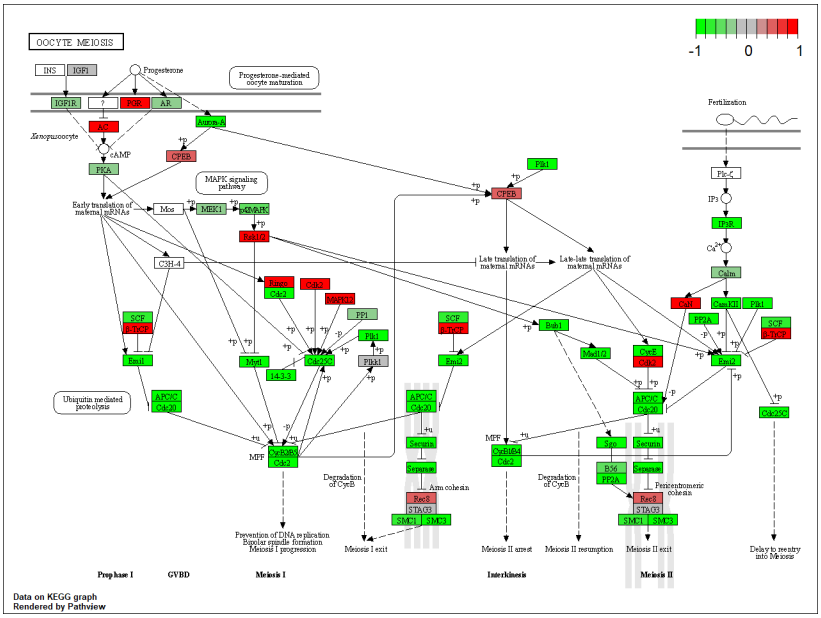
```
## Info: Writing image file hsa04114.pathview.png
```






NUCLEOCYTOPLASMIC TRANSPORT
Import
Export
mRNA Export
Importin
NLS
NPC
Cytoplasmic fibrils
Cytoplasm
Cytoplasmic ring
Lumen
Central channel
Spoke complex
Lumenal ring
Nucleoplasmic ring
Nucleus
Nuclear basket
Ran GTP
Exportin
HES
Nup358 complex
Nup62 complex
Nup107-160 complex
ELYS
Tpr
Upf1
Upf2
Tap
PABP
AAAAA
EJC
PYM
Ref/Aly
mRNA surveillance pathway
SRm160
Pinin
CBC
m7G
TREX
Nuclear Pore complex (NPC)
Cytoplasmic fibrils
Nup358 complex
Exon-junction complex (EJC)
EJC inner core
EJC outer shell
Transiently interacting factors
Cytoplasmic ring / Nucleoplasmic ring (Symmetrical unpl)
Central channel
Spoke complex
Lumenal ring
Nuclear basket
Nuclear transport complex
Importin
Adaptor proteins
Exportin
Transcription-export (TREX) complex
THO subcomplex
Data on KEGG graph
Rendered by Pathview


HOMOLOGOUS RECOMBINATION
Prokaryotic type
Eukaryotic type
Double strand break
Recognition
3' to 5' resection
(Escherichia coli)
RecF
SSB
RecA
RecBCD
RecB
RecC
RecD
Filament formation
RecFOR
RecA filament
Strand invasion
DNA synthesis
Holliday junction intermediate
DpoI
DpoIII
Branch migration and resolution of Holliday junction
RuvA
RuvB
RuvC
or
RecO
RecFOR pathway
Replication restart
PriA
PriB
PriC
DnaT
RecBC pathway
Rad51 paralogs
Rad51B
Rad51C
Rad51D
XRCC2
Rad51
(Saccharomyces cerevisiae)
(Mammals)
5' to 3' resection
MRX complex
MRN complex
Rad50
Mre11
Xrs2
Nbs1
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
(Common)
BLM
Top3
Mus81
Eme1
Crossover
or
Non-crossover
DSBR
Double-strand break repair
Strand displacement
Flap removal and annealing
Ligation
Fen1
DNA2
SDSA
Synthesis-dependent strand annealing
BIR
Break-induced replication
ATM
+p
CtIP
BARD1
BRCA1
RAP80
TOPBP1
NBA1
BRE
BRCC36
Abraxas
PALB2
BRCA2
DSS1
SYCP3
Data on KEGG graph
Rendered by Pathview


OOCYTE MEIOSIS
-1
0
1
INS
IGF1
Progesterone
Progesterone-mediated oocyte maturation
IGF1R
?
PGR
AR
Anxa6-A
AC
Xenopus oocyte
cAMP
PKA
CPEB
MAPK signaling pathway
Early translation of maternal mRNAs
Mos
MEK1
MAPK
Ringo
Cdk1
Cdk2
MAPK2
FP1
Myt1
14-3-3
Cdc25C
Plk1
Ubiquitin mediated proteolysis
APC/C
Cdc20
MPF
Cyclin B
Cdk1
Degradation of CyclB
Securin
Separase
Aim cohesin
Rec8
STAG3
SMC1
SMC3
Prevention of DNA replication
Bipolar spindle formation
Meiosis I progression
Meiosis I exit
Late translation of maternal mRNAs
Late-late translation of maternal mRNAs
Cdk2
Meiosis II arrest
Meiosis II resumption
Fertilization
Plc-Z
IP3
IP3R
Ca2+
Calm
CaMKII
Emi2
Bub1
Mad1/2
CyclE
Cdk2
Meiosis II exit
Pericentromeric cohesin
Delay to reentry into Meiosis
Prophase I
GVBD
Meiosis I
Interkinesis
Meiosis II
Data on KEGG graph
Rendered by Pathview