mini-project class13

Vina Nguyen

2023-05-31

Section 1. Differential Expression 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, 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 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
```

Load our data files.

```
metaFile <- read.csv("GSE37704_metadata.csv")
countFile <- read.csv("GSE37704_featurecounts.csv")
```

```
colData = read.csv("GSE37704_metadata.csv", row.names=1)
head(colData)
```

```
##               condition
## SRR493366 control_sirna
## SRR493367 control_sirna
## SRR493368 control_sirna
## SRR493369      hoxa1_kd
## SRR493370      hoxa1_kd
## SRR493371      hoxa1_kd
```

```
countData = read.csv("GSE37704_featurecounts.csv", row.names=1)
head(countData)
```

```
##                 length SRR493366 SRR493367 SRR493368 SRR493369 SRR493370
## ENSG00000186092    918         0         0         0         0         0
## ENSG00000279928    718         0         0         0         0         0
## ENSG00000279457   1982        23        28        29        29        28
## ENSG00000278566    939         0         0         0         0         0
## ENSG00000273547    939         0         0         0         0         0
## ENSG00000187634   3214       124       123       205       207       212
##                 SRR493371
## ENSG00000186092         0
## ENSG00000279928         0
## ENSG00000279457        46
## ENSG00000278566         0
## ENSG00000273547         0
## ENSG00000187634       258
```

Hmm... remember that we need the countData and colData files to match up so we will need to remove that odd first column in countData namely contData$length.

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

```
# We need to reomove the odd first $length col
# Use the negative operator to remove col
countData <- as.matrix(countData[,-1])
head(countData)
```

```
##                 SRR493366 SRR493367 SRR493368 SRR493369 SRR493370 SRR493371
## ENSG00000186092         0         0         0         0         0         0
## ENSG00000279928         0         0         0         0         0         0
```

```
## ENSG00000279457        23        28        29        29        28        46
## ENSG00000278566         0         0         0         0         0         0
## ENSG00000273547         0         0         0         0         0         0
## ENSG00000187634       124       123       205       207       212       258
```

This looks better but there are lots of zero entries in there so let's get rid of them as we have no data for these.

```
# Filter count data where you have 0 read count across all samples.
countData = countData[-row(countData)[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
```

Running DESeq2

Nice now lets setup the DESeqDataSet object required for the DESeq() function and then run the DESeq pipeline.

```
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: 13282 6
## metadata(1): version
## assays(4): counts mu H cooks
## rownames(13282): ENSG00000279457 ENSG00000187634 ... ENSG00000276345
##   ENSG00000271254
## rowData names(22): baseMean baseVar ... deviance maxCooks
## colnames(6): SRR493366 SRR493367 ... SRR493370 SRR493371
## colData names(2): condition sizeFactor
```

Next, get results for the HoxA1 knockdown versus control siRNA (remember that these were labeled as "hoxa1_kd" and "control_sirna" in our original colData metaFile input to DESeq, you can check this above and by running resultsNames(dds) command).

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

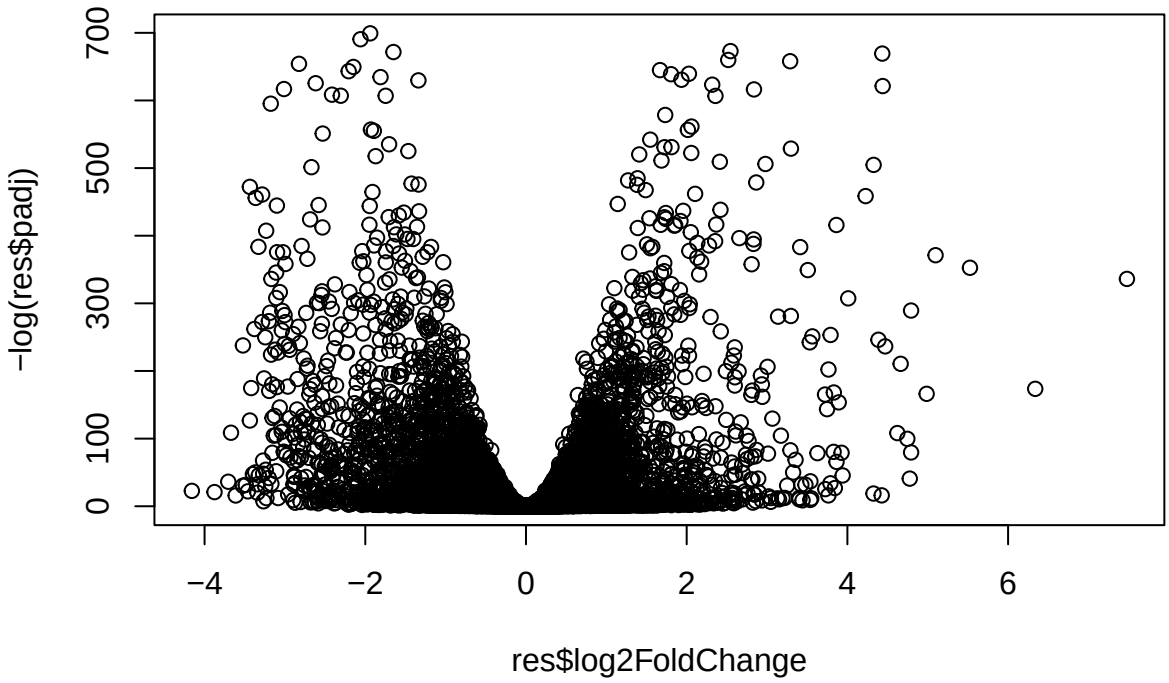
> Q. Call the summary() function on your results to get a sense of how many genes are up or down-regulated at the default 0.1 p-value cutoff.

```
summary(res)
```

```
##
## out of 13282 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4333, 33%
## LFC < 0 (down)     : 4400, 33%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Volcono plot

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



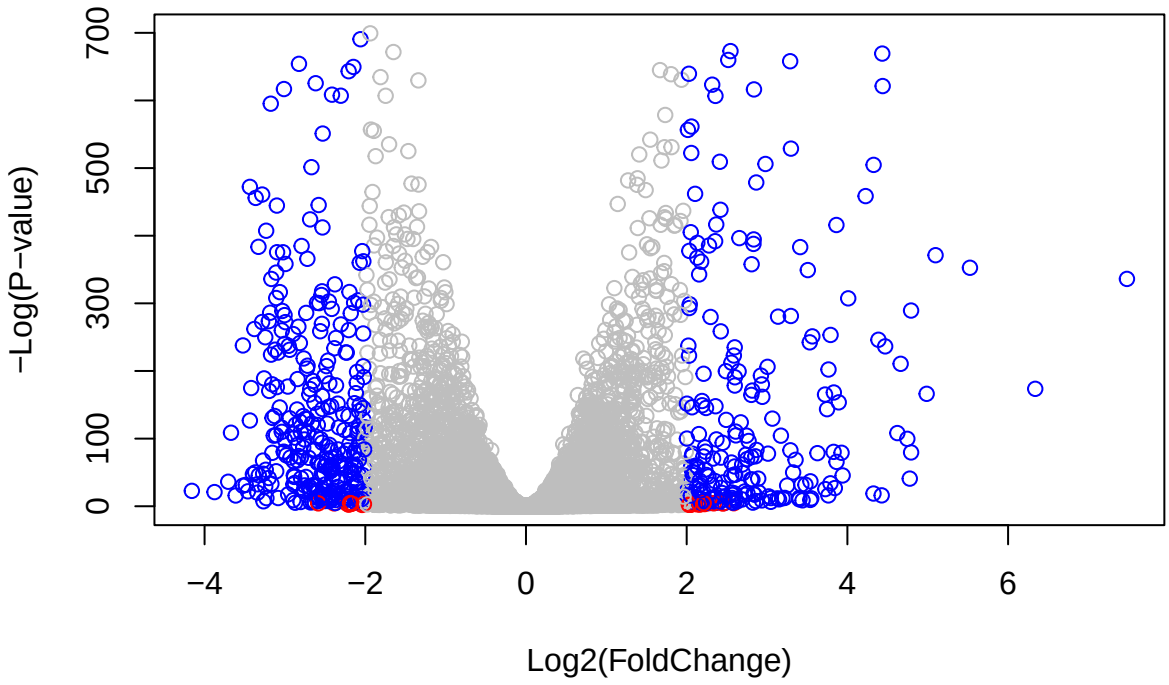
Q. Improve this plot by completing the below code, which adds color and axis labels

```
# Make a color vector for all genes
mycols <- rep("gray", nrow(res) )

# Color red the genes with absolute fold change above 2
mycols[ abs(res$log2FoldChange) > 2 ] <- "red"

# Color blue those with adjusted p-value less than 0.01
#  and absolute fold change more than 2
inds <- (res$padj < 0.01 ) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

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



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.

```
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="ENTREZID",
                    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="ENTREZID",
                    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   29.9136      0.1803039 0.3121566   0.577607 5.63529e-01
## ENSG00000187634  183.2296      0.4258966 0.1355303   3.142446 1.67543e-03
## ENSG00000188976 1651.1881     -0.6927118 0.0549876 -12.597612 2.17635e-36
## ENSG00000187961  209.6379      0.7299597 0.1277613   5.713463 1.10700e-08
## ENSG00000187583   47.2551      0.0392549 0.2606192   0.150622 8.80274e-01
## ENSG00000187642   11.9798      0.5395082 0.5001355   1.078724 2.80711e-01
## ENSG00000188290  108.9221      2.0562855 0.1910714  10.761870 5.21018e-27
## ENSG00000187608  350.7169      0.2570251 0.0999769   2.570845 1.01451e-02
## ENSG00000188157 9128.4394      0.3899096 0.0482214   8.085827 6.17439e-16
## ENSG00000131591  156.4791      0.1968918 0.1406800   1.399572 1.61641e-01
##                        padj      symbol      entrez        name
##                   <numeric> <character> <character> <character>
## ENSG00000279457 6.47026e-01          NA          NA          NA
## ENSG00000187634 3.34029e-03      148398      148398      148398
## ENSG00000188976 2.35970e-35       26155       26155       26155
## ENSG00000187961 3.69612e-08      339451      339451      339451
## ENSG00000187583 9.10931e-01       84069       84069       84069
## ENSG00000187642 3.61174e-01       84808       84808       84808
## ENSG00000188290 4.17884e-26       57801       57801       57801
## ENSG00000187608 1.79950e-02        9636        9636        9636
## ENSG00000188157 3.15902e-15      375790      375790      375790
## ENSG00000131591 2.23894e-01       54991       54991       54991
```

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")
```

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.

Run in your R console (i.e. not your Rmarkdown doc!)

BiocManager::install(c("pathview", "gage", "gageData"))

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

```
library(pathview)
```

```
## ##############################################################################
## Pathview is an open source software package distributed under GNU General
## Public License version 3 (GPLv3). Details of GPLv3 is available at
## http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
## formally cite the original Pathview paper (not just mention it) in publications
## or products. For details, do citation("pathview") within R.
##
## The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG
## license agreement (details at http://www.kegg.jp/kegg/legal.html).
## ##############################################################################
```

```
library(gage)
```

```
##
```

```
library(gageData)

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

# 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 $foldchanges$ result

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

```
##      1266     54855      1465      2034      2150      6659
## -2.422683  3.201858 -2.313713 -1.887999  3.344480  2.392257
```

Now, let's run the gage pathway analysis.

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

Now lets look at the object returned from gage().

```
attributes(keggres)
```

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

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

```
##                                       p.geomean stat.mean        p.val
## hsa04110 Cell cycle                 3.548176e-06 -4.604234 3.548176e-06
## hsa03030 DNA replication            3.992330e-05 -4.191094 3.992330e-05
## hsa04114 Oocyte meiosis             2.332810e-04 -3.564509 2.332810e-04
## hsa03440 Homologous recombination   2.248158e-03 -2.967340 2.248158e-03
## hsa03013 RNA transport              4.162613e-03 -2.662235 4.162613e-03
## hsa00670 One carbon pool by folate 8.202725e-03 -2.535331 8.202725e-03
##                                            q.val set.size         exp1
## hsa04110 Cell cycle                0.0005535155      118 3.548176e-06
## hsa03030 DNA replication           0.0031140177       36 3.992330e-05
## hsa04114 Oocyte meiosis            0.0121306145       95 2.332810e-04
## hsa03440 Homologous recombination  0.0876781678       28 2.248158e-03
## hsa03013 RNA transport             0.1298735381      140 4.162613e-03
## hsa00670 One carbon pool by folate 0.2115248982       17 8.202725e-03
```

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

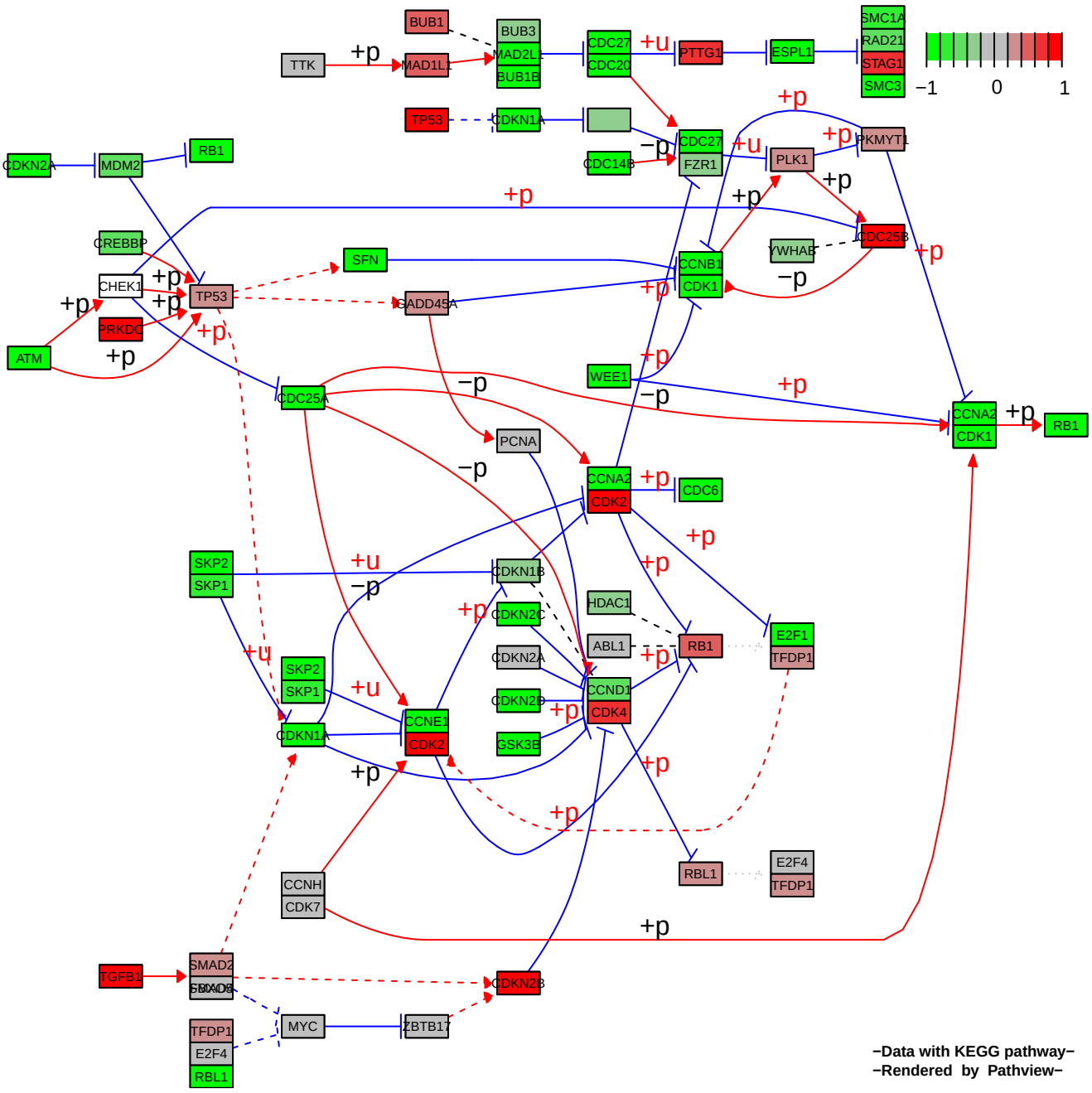
```
pathview(gene.data=foldchanges, pathway.id="hsa04110")
```

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

```
## Info: Working in directory C:/Users/myeon/OneDrive/Documents/BIMM143/week08 new file
```

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

```
# insert image
knitr::include_graphics("hsa04110.pathview.png")
```



You can play with the other input arguments to pathview() to change the display in various ways including generating a PDF graph. For example:

```
# A different PDF based output of the same data
pathview(gene.data=foldchanges, pathway.id="hsa04110", kegg.native=FALSE)
```

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

## Warning: reconcile groups sharing member nodes!

##      [,1] [,2]
```

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

```
## Info: Working in directory C:/Users/myeon/OneDrive/Documents/BIMM143/week08 new file
```

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

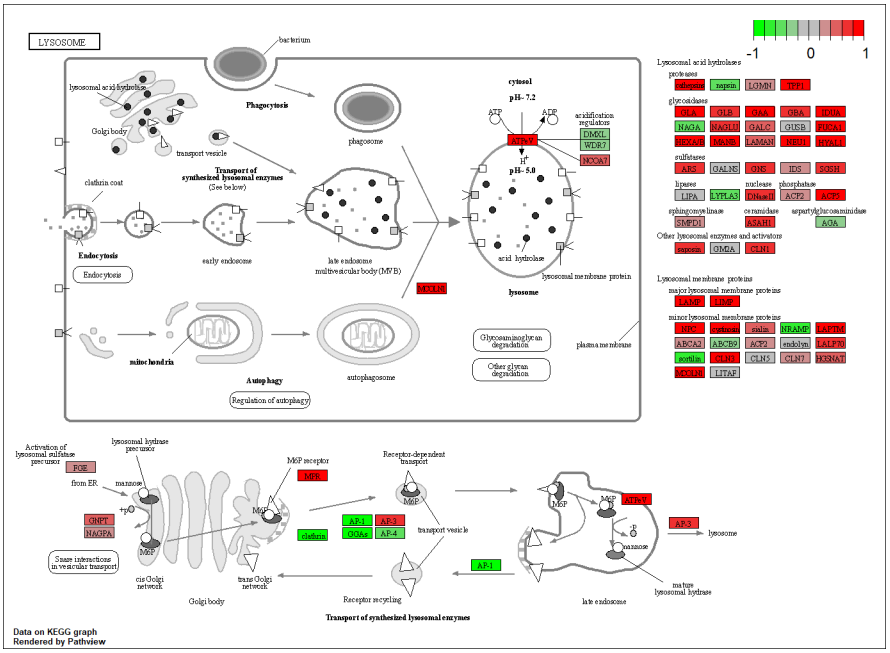
```
## Focus on top 5 upregulated pathways here for demo purposes only
keggrespathways <- rownames(keggres$greater)[1:5]

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

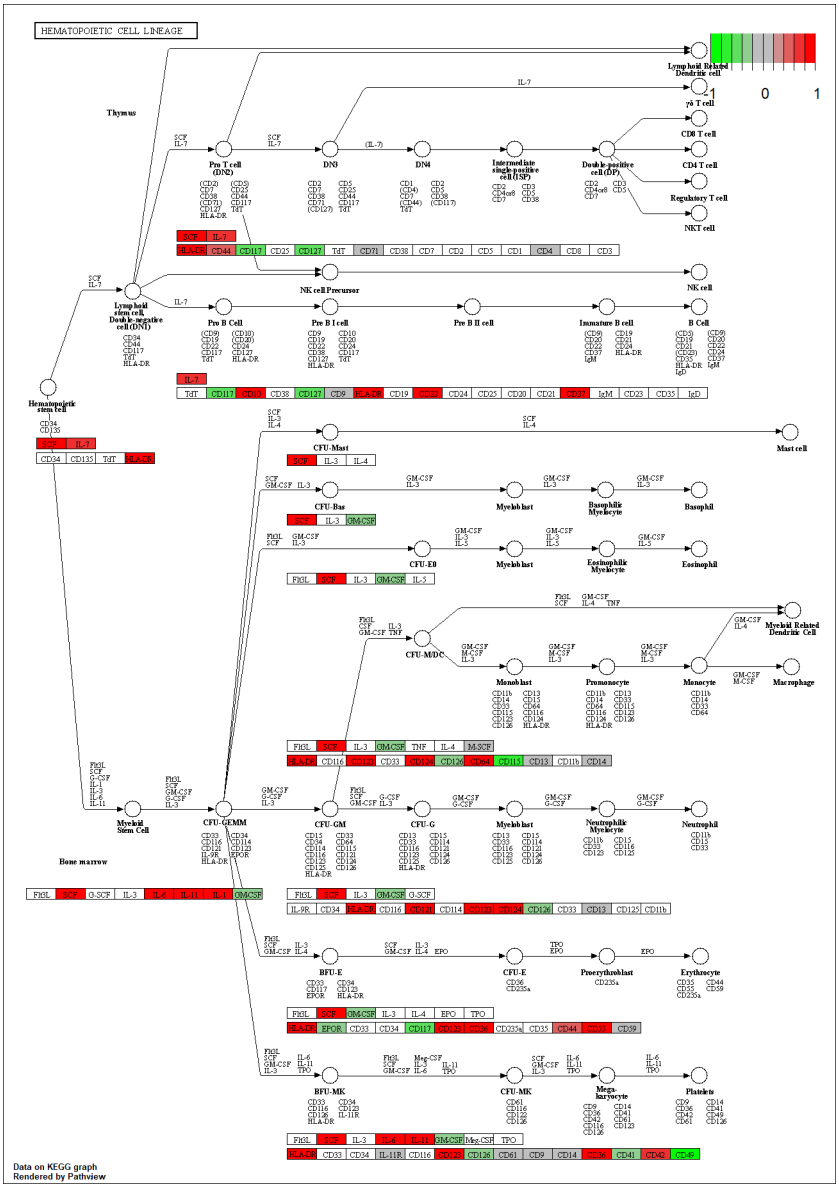
```
## [1] "hsa04142" "hsa04640" "hsa04974" "hsa00603" "hsa04380"
```

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

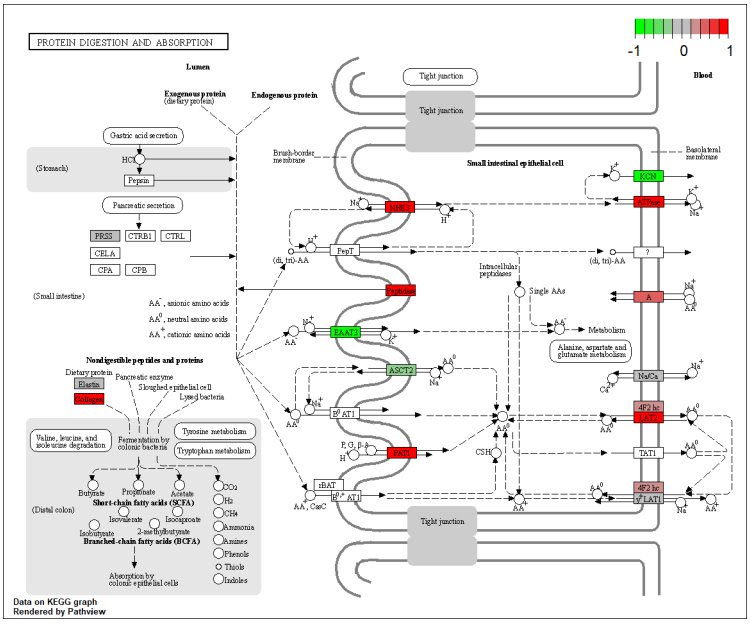
```
knitr::include_graphics("hsa04142.pathview.png")
```




```
knitr::include_graphics("hsa04640.pathview.png")
```




```
knitr::include_graphics("hsa04974.pathview.png")
```



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

We cannot plot negative length vectors.

Section 3. Gene Ontology (GO)

We can also do a similar procedure with gene ontology. Similar to above, go.sets.hs has all GO terms. go.subs.hs is a named list containing indexes for the BP, CC, and MF ontologies. Let's focus on BP (a.k.a Biological Process) here.

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

# Focus on Biological Process subset of GO
gobpsets = go.sets.hs[go.subs.hs$BP]

gobpres = gage(foldchanges, gsets=gobpsets, same.dir=TRUE)

lapply(gobpres, head)
```

```
## $greater
##                                                      p.geomean stat.mean
## GO:0007156 homophilic cell adhesion                7.523307e-05  3.873939
## GO:0016339 calcium-dependent cell-cell adhesion    8.556504e-04  3.340855
## GO:0010817 regulation of hormone levels            1.058523e-03  3.091986
## GO:0048729 tissue morphogenesis                    1.389102e-03  3.002504
## GO:0008285 negative regulation of cell proliferation 1.654959e-03  2.947174
## GO:0051047 positive regulation of secretion        1.877703e-03  2.927781
##                                                           p.val     q.val
## GO:0007156 homophilic cell adhesion                7.523307e-05 0.2796413
## GO:0016339 calcium-dependent cell-cell adhesion    8.556504e-04 0.5732589
```

```
## GO:0010817 regulation of hormone levels              1.058523e-03 0.5732589
## GO:0048729 tissue morphogenesis                      1.389102e-03 0.5732589
## GO:0008285 negative regulation of cell proliferation 1.654959e-03 0.5732589
## GO:0051047 positive regulation of secretion          1.877703e-03 0.5732589
##                                                      set.size         exp1
## GO:0007156 homophilic cell adhesion                        90 7.523307e-05
## GO:0016339 calcium-dependent cell-cell adhesion            24 8.556504e-04
## GO:0010817 regulation of hormone levels                   225 1.058523e-03
## GO:0048729 tissue morphogenesis                           347 1.389102e-03
## GO:0008285 negative regulation of cell proliferation      387 1.654959e-03
## GO:0051047 positive regulation of secretion               130 1.877703e-03
##
## $less
##                                          p.geomean stat.mean        p.val
## GO:0000279 M phase                      6.451975e-18 -8.738701 6.451975e-18
## GO:0048285 organelle fission            1.832907e-16 -8.369971 1.832907e-16
## GO:0000280 nuclear division             2.627088e-16 -8.340038 2.627088e-16
## GO:0007067 mitosis                      2.627088e-16 -8.340038 2.627088e-16
## GO:0000087 M phase of mitotic cell cycle 9.244549e-16 -8.166584 9.244549e-16
## GO:0007059 chromosome segregation       2.502912e-12 -7.264756 2.502912e-12
##                                                q.val set.size         exp1
## GO:0000279 M phase                      2.398199e-14      467 6.451975e-18
## GO:0048285 organelle fission            2.441221e-13      360 1.832907e-16
## GO:0000280 nuclear division             2.441221e-13      338 2.627088e-16
## GO:0007067 mitosis                      2.441221e-13      338 2.627088e-16
## GO:0000087 M phase of mitotic cell cycle 6.872398e-13      348 9.244549e-16
## GO:0007059 chromosome segregation       1.550554e-09      135 2.502912e-12
##
## $stats
##                                                     stat.mean     exp1
## GO:0007156 homophilic cell adhesion                  3.873939 3.873939
## GO:0016339 calcium-dependent cell-cell adhesion      3.340855 3.340855
## GO:0010817 regulation of hormone levels              3.091986 3.091986
## GO:0048729 tissue morphogenesis                      3.002504 3.002504
## GO:0008285 negative regulation of cell proliferation 2.947174 2.947174
## GO:0051047 positive regulation of secretion          2.927781 2.927781
```

Section 4. Reactome Analysis

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: 8186"
```

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

> 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?

Signaling by Rho GTPases, Miro GTPases and RHOBTB3 has the most significant entities p-value (1.68E-5). This does not match the pathways listed in the KEGG results. Database could cause differences between the two methods since Reactome seems to have a wider base.

```
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] gageData_2.36.0             gage_2.48.0
##  [3] pathview_1.38.0             org.Hs.eg.db_3.16.0
##  [5] AnnotationDbi_1.60.2        DESeq2_1.38.3
##  [7] SummarizedExperiment_1.28.0 Biobase_2.58.0
##  [9] MatrixGenerics_1.10.0       matrixStats_0.63.0
## [11] GenomicRanges_1.50.2        GenomeInfoDb_1.34.9
## [13] IRanges_2.32.0              S4Vectors_0.36.2
## [15] BiocGenerics_0.44.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.6             bit64_4.0.5            highr_0.10
##  [4] blob_1.2.4             GenomeInfoDbData_1.2.9 yaml_2.3.7
##  [7] pillar_1.9.0           RSQLite_2.3.1          lattice_0.21-8
## [10] glue_1.6.2             digest_0.6.31          RColorBrewer_1.1-3
## [13] XVector_0.38.0         colorspace_2.1-0       htmltools_0.5.5
## [16] Matrix_1.5-4.1         XML_3.99-0.14          pkgconfig_2.0.3
## [19] zlibbioc_1.44.0        GO.db_3.16.0           xtable_1.8-4
## [22] scales_1.2.1           BiocParallel_1.32.6    tibble_3.2.1
## [25] annotate_1.76.0        KEGGREST_1.38.0        generics_0.1.3
## [28] ggplot2_3.4.2          cachem_1.0.8           cli_3.6.1
## [31] magrittr_2.0.3         crayon_1.5.2           memoise_2.0.1
## [34] evaluate_0.21          KEGGgraph_1.58.3       fansi_1.0.4
## [37] graph_1.76.0           tools_4.2.3            lifecycle_1.0.3
## [40] munsell_0.5.0          locfit_1.5-9.7         DelayedArray_0.23.2
## [43] Biostrings_2.66.0      compiler_4.2.3         rlang_1.1.0
## [46] grid_4.2.3             RCurl_1.98-1.12        rstudioapi_0.14
## [49] bitops_1.0-7           rmarkdown_2.21         gtable_0.3.3
## [52] codetools_0.2-19       DBI_1.1.3              R6_2.5.1
## [55] knitr_1.43             dplyr_1.1.2            fastmap_1.1.1
## [58] bit_4.0.5              utf8_1.2.3             Rgraphviz_2.42.0
```

```
## [61] parallel_4.2.3         Rcpp_1.0.10            vctrs_0.6.2
## [64] geneplotter_1.76.0     png_0.1-8              tidyselect_1.2.0
## [67] xfun_0.39
```