Class 13: Transcriptomics and the analysis of RNA-Seq data

Solomon Kim

The data for this hands-on session comes from a published RNA-seq experiment where airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014).

Import Data

We need two things for this analysis: counts and metadata these are called “countData” and “colData” in the DESeq2 world.

```
counts <- read.csv("airway_scaledcounts.csv", row.names=1)
metadata <-  read.csv("airway_metadata.csv")

head(counts)
```

```
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
ENSG00000000003        723        486        904        445       1170
ENSG00000000005          0          0          0          0          0
ENSG00000000419        467        523        616        371        582
ENSG00000000457        347        258        364        237        318
ENSG00000000460         96         81         73         66        118
ENSG00000000938          0          0          1          0          2
                SRR1039517 SRR1039520 SRR1039521
ENSG00000000003       1097        806        604
ENSG00000000005          0          0          0
ENSG00000000419        781        417        509
ENSG00000000457        447        330        324
ENSG00000000460         94        102         74
ENSG00000000938          0          0          0
```

The counts are organized with a gene per row and experiment per column

```
head(metadata)
```

```
          id     dex celltype     geo_id
1 SRR1039508 control   N61311 GSM1275862
2 SRR1039509 treated   N61311 GSM1275863
3 SRR1039512 control  N052611 GSM1275866
4 SRR1039513 treated  N052611 GSM1275867
5 SRR1039516 control  N080611 GSM1275870
6 SRR1039517 treated  N080611 GSM1275871
```

Examine Data

> Q1. How many genes are in this dataset?

```
nrow(counts)
```

```
[1] 38694
```

> Q2. How many 'control' cell lines do we have?

```
sum(metadata$dex == "control")
```

```
[1] 4
```

```
table(metadata$dex)
```

```

control treated 
      4       4 
```

Check on match of metadata and coldata

```
colnames(counts)
```

```
[1] "SRR1039508" "SRR1039509" "SRR1039512" "SRR1039513" "SRR1039516"
[6] "SRR1039517" "SRR1039520" "SRR1039521"
```

```r
metadata$id
```

```
[1] "SRR1039508" "SRR1039509" "SRR1039512" "SRR1039513" "SRR1039516"
[6] "SRR1039517" "SRR1039520" "SRR1039521"
```

```r
colnames(counts) == metadata$id
```

```
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

If you want to know that all the elements of a vector are TRUE we ca use the `all()` function.

```r
all(c(T,T,T))
```

```
[1] TRUE
```

```r
all(c(T,T,F))
```

```
[1] FALSE
```

```r
all(colnames(counts) == metadata$id)
```

```
[1] TRUE
```

Analysis

I want to start by comparing "control" and "treated" columns, to this I will first find the average for each gene (row) in all "control" columns. Then I will find the average in the reated columns. Then I will compare them.

Let's extract all "control" columns control.

```r
control.inds <- metadata$dex == "control"
```

```r
control.counts <- counts[,control.inds]
```

Now find the mean count value per fene using the `apply()` function.

```
control.mean <- apply(control.counts, 1, mean)
```

Now do the same for the "treated" columns. i..e. find `treated.mean` values.

```
treated.inds <- metadata$dex == "treated"
```

```
treated.counts <- counts[,treated.inds]
```

```
treated.mean <- apply(treated.counts, 1, mean)
```

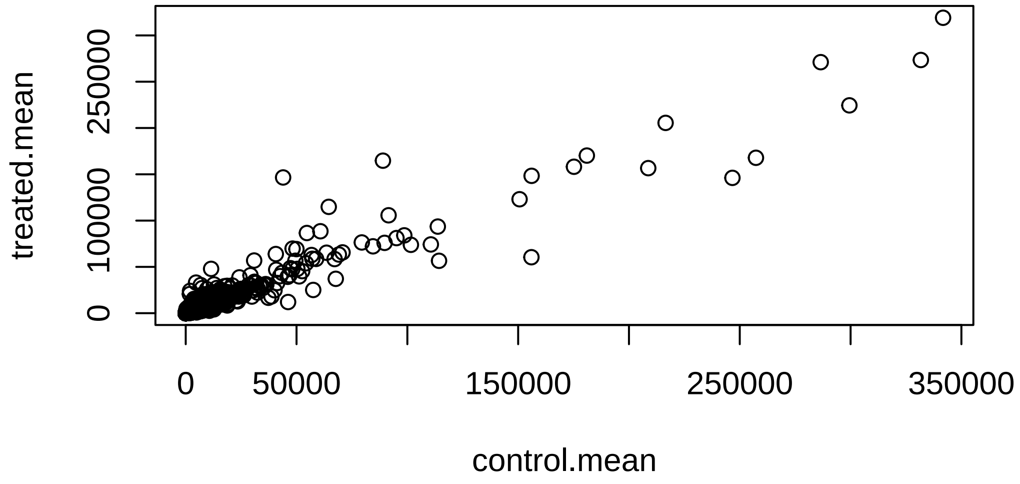
Put these two mean vectors together for ease of book-keeping

```
meancounts <- data.frame(control.mean, treated.mean)
head(meancounts)
```

```
                control.mean treated.mean
ENSG00000000003       900.75       658.00
ENSG00000000005         0.00         0.00
ENSG00000000419       520.50       546.00
ENSG00000000457       339.75       316.50
ENSG00000000460        97.25        78.75
ENSG00000000938         0.75         0.00
```

Let's have a wee look with a quick plot

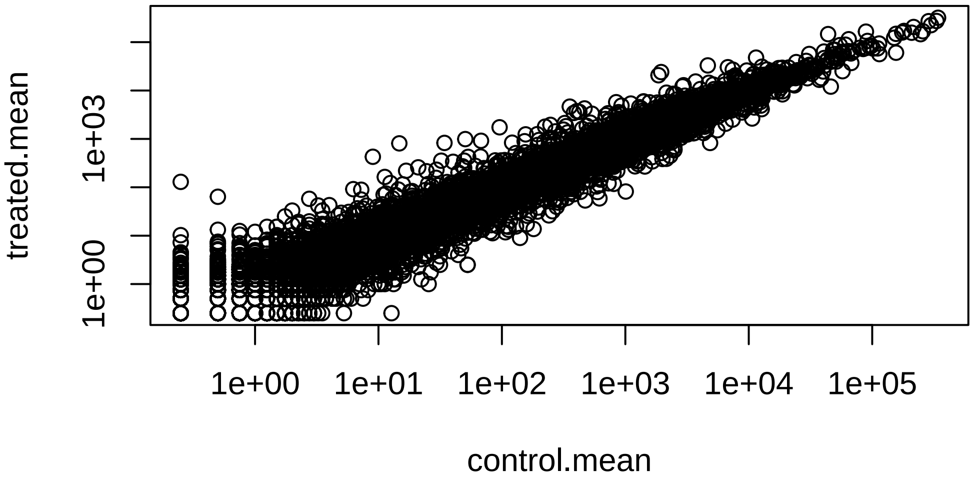
```
plot(meancounts, )
```



```
plot(meancounts,log="xy")
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15032 x values <= 0 omitted
from logarithmic plot
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted
from logarithmic plot
```



```
log(10, base=2)
```

```
[1] 3.321928
```

```
log2(10/10)
```

```
[1] 0
```

```
log2(20/10)
```

```
[1] 1
```

```
log2(10/20)
```

```
[1] -1
```

We most often work in log2 units because they have a more simple interpretation.

Here we calculate the log2 Fold-change of treated/control values and add it to our new data frame of results

```
meancounts$log2fc <- log2(meancounts$treated.mean/meancounts$control.mean)
head(meancounts)
```

```
                control.mean treated.mean      log2fc
ENSG00000000003       900.75       658.00 -0.45303916
ENSG00000000005         0.00         0.00         NaN
ENSG00000000419       520.50       546.00  0.06900279
ENSG00000000457       339.75       316.50 -0.10226805
ENSG00000000460        97.25        78.75 -0.30441833
ENSG00000000938         0.75         0.00        -Inf
```

There are some funky answers in here like NaN (not a number and -inf (minus infinity)) that all come ecause I have zero count genes in my data set.

It is common practice to filter these zero count genes out before we go too deep

```
#apply(meancounts[,1:2] == 0, 1, sum)
to.keep.inds <- (rowSums(meancounts[,1:2] == 0) == 0)

mycounts <- meancounts[to.keep.inds, ]
head(mycounts)
```

```
                control.mean treated.mean      log2fc
ENSG00000000003       900.75       658.00 -0.45303916
ENSG00000000419       520.50       546.00  0.06900279
ENSG00000000457       339.75       316.50 -0.10226805
ENSG00000000460        97.25        78.75 -0.30441833
ENSG00000000971      5219.00      6687.50  0.35769358
ENSG00000001036      2327.00      1785.75 -0.38194109
```

> Q. How many genes do we have left after zero count filtering?

```
nrow(mycounts)
```

```
[1] 21817
```

A common threshold for calling a gene “up” or “down” is a log2 fold change of +2 or -2.

> Q. How many “up” regulated gene do we have?

```
sum(mycounts$log2fc >= +2)
```

```
[1] 314
```

DESeq Analysis

We need to do this analyss properly with our inner stats person kept happy

```
library(DESeq2)
```

To use DESeq we need to get our input data in very particular format.

```
dds <- DESeqDataSetFromMatrix(countData = counts, colData = metadata, design = ~dex)
```

```
converting counts to integer mode
```

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

Run DESeq analysis

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

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

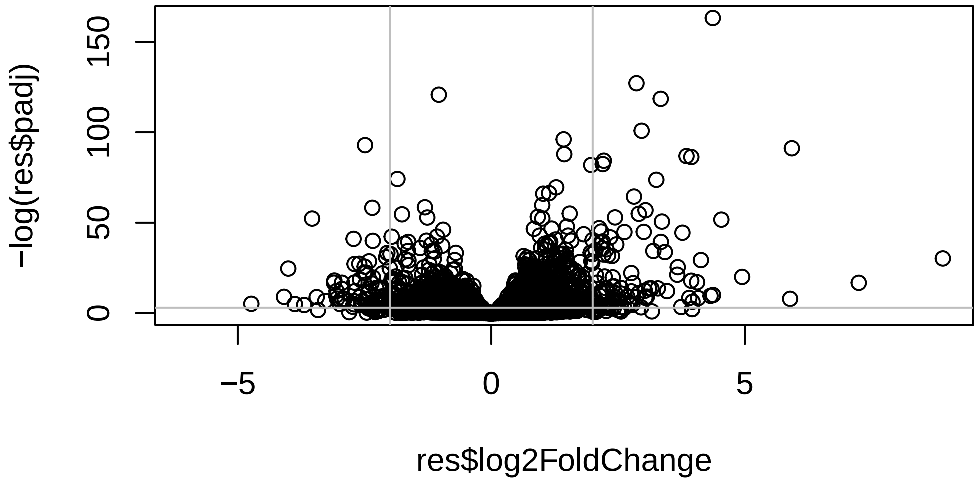
Get the resutlts

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

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue
                <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj
                <numeric>
ENSG00000000003  0.163035
ENSG00000000005        NA
ENSG00000000419  0.176032
ENSG00000000457  0.961694
ENSG00000000460  0.815849
ENSG00000000938        NA
```

I want to make a figure showing an overview of all my results to date A plot of **log2 fold change** vs the adjusted **p-value** (adjusted p-value)

```
plot(res$log2FoldChange, -log(res$padj))
abline(v=-2, col="gray")
abline(v=+2, col="gray")
abline(h=-log(0.05), col="gray")
```

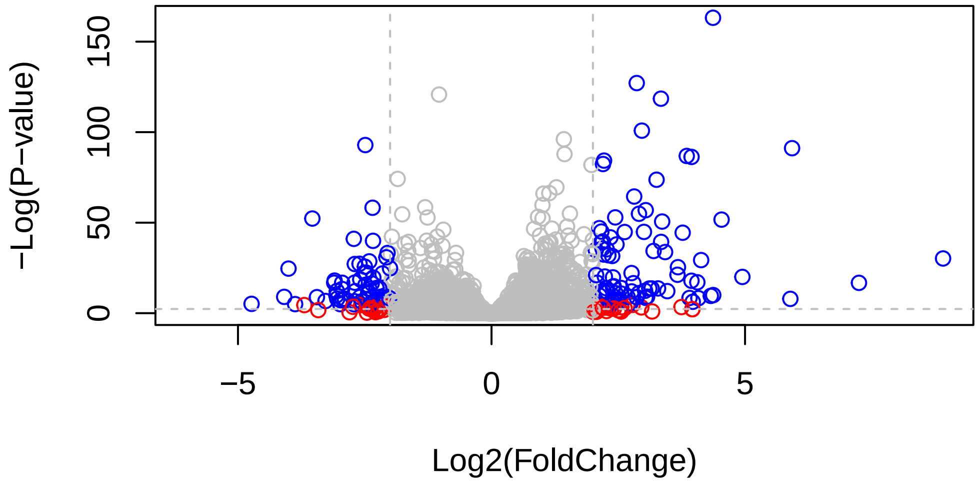


```
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ]  <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot( res$log2FoldChange,  -log(res$padj),
 col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)" )

# Cut-off lines
abline(v=c(-2,2), col="gray", lty=2)
abline(h=-log(0.1), col="gray", lty=2)
```



Add annotation data

We want to add on gene symbols (i.e. gene names) as well as other common identifiers from major databases for all our genes of interest.

```
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"
```

We can translate between the following

```
res$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="SYMBOL"
```

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

```
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 7 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue
                <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol
                <numeric> <character>
ENSG00000000003  0.163035      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.176032        DPM1
ENSG00000000457  0.961694       SCYL3
ENSG00000000460  0.815849       FIRRM
ENSG00000000938        NA         FGR
```

My IDs are in the `rownames(res)` and they are in ENSEMBL

```
#rownames(res)
```

We also want “GENENAME” and “ENTREZID”

```
res$genename <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="GENEN
```

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

```
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 8 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol               genename
                <numeric> <character>            <character>
ENSG00000000003  0.163035      TSPAN6          tetraspanin 6
ENSG00000000005        NA        TNMD            tenomodulin
ENSG00000000419  0.176032        DPM1 dolichyl-phosphate m..
ENSG00000000457  0.961694       SCYL3 SCY1 like pseudokina..
ENSG00000000460  0.815849       FIRRM FIGNL1 interacting r..
ENSG00000000938        NA         FGR FGR proto-oncogene, ..
```

```r
res$entrezid <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="ENTRE
```

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

```r
head(res)
```

```
log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 6 rows and 9 columns
                  baseMean log2FoldChange     lfcSE      stat    pvalue
                 <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG00000000003 747.194195     -0.3507030  0.168246 -2.084470 0.0371175
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160      0.2061078  0.101059  2.039475 0.0414026
ENSG00000000457 322.664844      0.0245269  0.145145  0.168982 0.8658106
ENSG00000000460  87.682625     -0.1471420  0.257007 -0.572521 0.5669691
ENSG00000000938   0.319167     -1.7322890  3.493601 -0.495846 0.6200029
                     padj      symbol               genename    entrezid
                <numeric> <character>            <character> <character>
ENSG00000000003  0.163035      TSPAN6          tetraspanin 6        7105
```

```
ENSG00000000005         NA        TNMD             tenomodulin       64102
ENSG00000000419   0.176032        DPM1 dolichyl-phosphate m..         8813
ENSG00000000457   0.961694       SCYL3 SCY1 like pseudokina..        57147
ENSG00000000460   0.815849       FIRRM FIGNL1 interacting r..        55732
ENSG00000000938         NA         FGR FGR proto-oncogene, ..         2268
```

Let's save our results to a new CSV file

```
write.csv(res, file="myresults.csv")
```

Pathway Analysis

Here we will use the "gage" package to do some pathway analysis (a.k.a geneset

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

Have a peak at KEG data

```
data(kegg.sets.hs)

#first two pathways in this kegg set for humans
data(kegg.sets.hs, 2)
```

```
Warning in data(kegg.sets.hs, 2): data set '2' not found
```

To run gage we need to provide it with a vector of fold-change values (not our big full results table)

```
foldchanges <- res$log2FoldChange
#foldchanges
```

Add the ENTREZ ids as names to this vector

```
x <- c(10, 9, 7)
names(x) <- c("chandra", "alice", "Barry")
x
```

```
chandra   alice   Barry 
     10       9       7 
```

Add Entrez ids as names to my foldchange vector

```r
names(foldchanges) <- res$entrezid
head(foldchanges)
```

```
       7105       64102        8813       57147       55732        2268 
-0.35070302          NA  0.20610777  0.02452695 -0.14714205 -1.73228897 
```

```r
keggres = gage(foldchanges, gsets=kegg.sets.hs)
attributes(keggres)
```

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

```r
head(keggres$less)
```

```
                                                         p.geomean stat.mean
hsa05332 Graft-versus-host disease                    0.0004250461 -3.473346
hsa04940 Type I diabetes mellitus                     0.0017820293 -3.002352
hsa05310 Asthma                                       0.0020045888 -3.009050
hsa04672 Intestinal immune network for IgA production 0.0060434515 -2.560547
hsa05330 Allograft rejection                          0.0073678825 -2.501419
hsa04340 Hedgehog signaling pathway                   0.0133239547 -2.248547
                                                             p.val      q.val
hsa05332 Graft-versus-host disease                    0.0004250461 0.09053483
hsa04940 Type I diabetes mellitus                     0.0017820293 0.14232581
hsa05310 Asthma                                       0.0020045888 0.14232581
hsa04672 Intestinal immune network for IgA production 0.0060434515 0.31387180
hsa05330 Allograft rejection                          0.0073678825 0.31387180
hsa04340 Hedgehog signaling pathway                   0.0133239547 0.47300039
                                                      set.size         exp1
hsa05332 Graft-versus-host disease                          40 0.0004250461
hsa04940 Type I diabetes mellitus                           42 0.0017820293
hsa05310 Asthma                                             29 0.0020045888
hsa04672 Intestinal immune network for IgA production       47 0.0060434515
hsa05330 Allograft rejection                                36 0.0073678825
hsa04340 Hedgehog signaling pathway                         56 0.0133239547
```

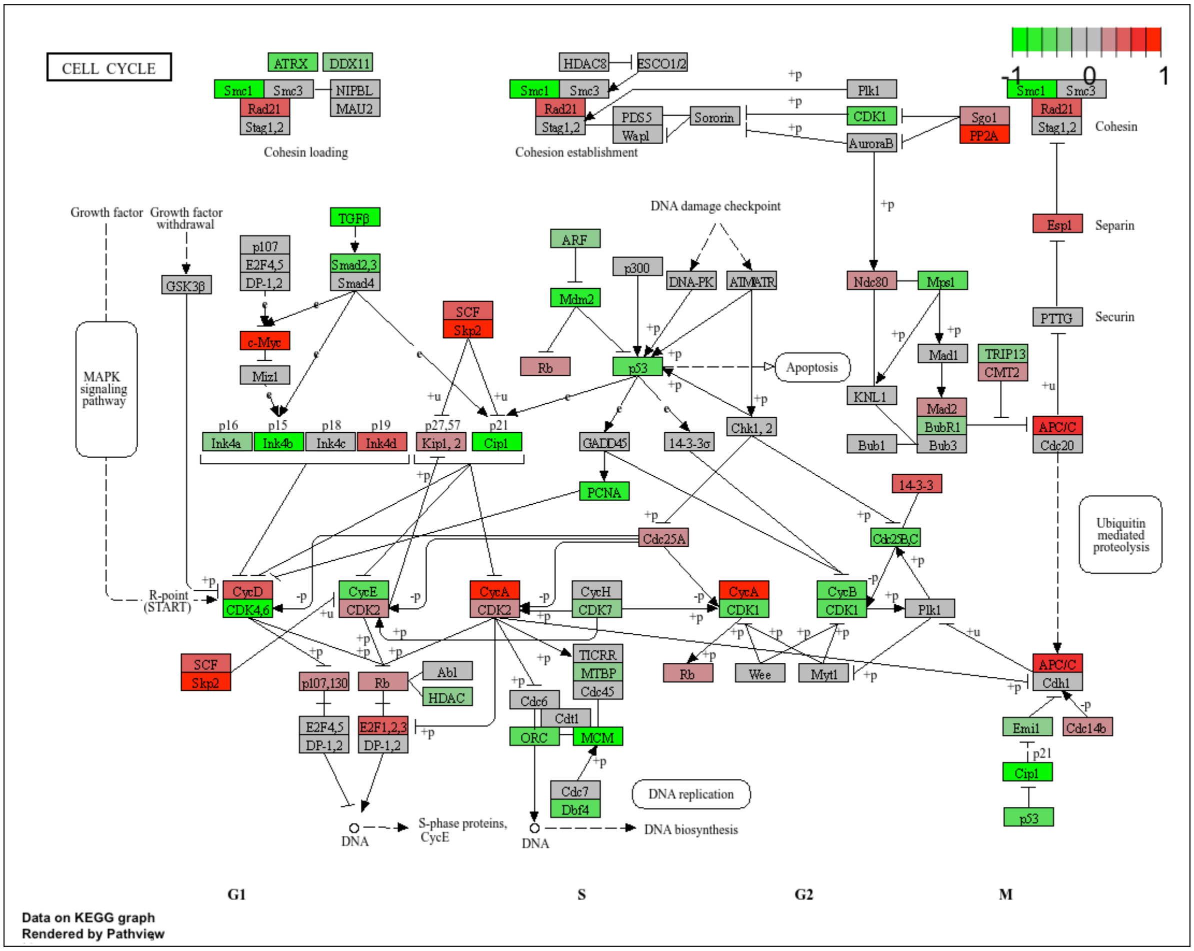
Let’s have a look at the hsa05310 Asthma pathway with our gene highlighted using the `pathview()` function:

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

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

```
Info: Working in directory /Users/solomonkim/Desktop/BIMM143/Class 13
```

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


CELL CYCLE
-1
0
1
ATRX
DDX11
Smc1
Smc3
NIPBL
MAU2
Rad21
Stag1,2
Cohesin loading
HDAC8
ESCO1/2
PDS5
Wapl
Sororin
Cohesion establishment
Plk1
CDK1
AuroraB
Sgo1
PP2A
Cohesin
Growth factor
Growth factor withdrawal
GSK3β
MAPK signaling pathway
p107
E2F4,5
DP-1,2
TGFβ
Smad2,3
Smad4
c-Myc
Miz1
SCF
Skp2
ARF
Mdm2
Rb
p300
DNA damage checkpoint
DNA-PK
ATM/ATR
p53
Apoptosis
Ndc80
Mps1
KNL1
Mad1
Mad2
BubR1
Bub3
Bub1
TRIP13
CMT2
Espl
Separin
PTTG
Securin
APC/C
Cdc20
p16
Ink4a
p15
Ink4b
p18
Ink4c
p19
Ink4d
p27,57
Kip1, 2
p21
Cip1
GADD45
14-3-3σ
Chk1, 2
PCNA
14-3-3
Cdc25B,C
Cdc25A
Ubiquitin mediated proteolysis
R-point (START)
CycD
CDK4,6
CycE
CDK2
CycA
CDK2
CycH
CDK7
CycA
CDK1
CycB
CDK1
Plk1
SCF
Skp2
p107,130
Rb
Abl
HDAC
TICRR
MTBP
Cdc45
Rb
Wee
Myt1
APC/C
Cdh1
E2F4,5
DP-1,2
E2F1,2,3
DP-1,2
Cdc6
Cdt1
ORC
MCM
Emi1
Cdc14b
p21
Cip1
p53
Cdc7
Dbf4
DNA replication
DNA
S-phase proteins, CycE
DNA
DNA biosynthesis
+p
-p
+u
e
G1
S
G2
M
Data on KEGG graph
Rendered by Pathview