

# Class 13 Genomics Lab

Janie Chang-Weinberg (A69037446)

Today we will work with a dataset “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).” (from lab worksheet).

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

lets have a look!

```
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		

Not particularly helpful, will have to look at metadata

```
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
```

Q1. How many genes are in this dataset?

```
nrow(counts)
```

```
[1] 38694
```

There are 38,694 genes in this dataset

Q2. How many ‘control’ cell lines do we have?

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

```
[1] 4
```

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

```
control treated
        4         4
```

There are 4 control cell lines.

The goal here is to compare “control” vs “treated” cells.

1.) Let’s split the counts into `control.counts` and `treated.counts`

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

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

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

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

Syntax with df[ROWS, COLS]

```
#This is to pull out the control columns  
control.counts <- counts[,control inds]  
head(control.counts)
```

	SRR1039508	SRR1039512	SRR1039516	SRR1039520
ENSG000000000003	723	904	1170	806
ENSG000000000005	0	0	0	0
ENSG000000000419	467	616	582	417
ENSG000000000457	347	364	318	330
ENSG000000000460	96	73	118	102
ENSG000000000938	0	1	2	0

```
treated inds <- metadata$dex == "treated"  
treated.counts <- counts[,treated inds]  
head(treated.counts)
```

	SRR1039509	SRR1039513	SRR1039517	SRR1039521
ENSG000000000003	486	445	1097	604
ENSG000000000005	0	0	0	0
ENSG000000000419	523	371	781	509
ENSG000000000457	258	237	447	324
ENSG000000000460	81	66	94	74
ENSG000000000938	0	0	0	0

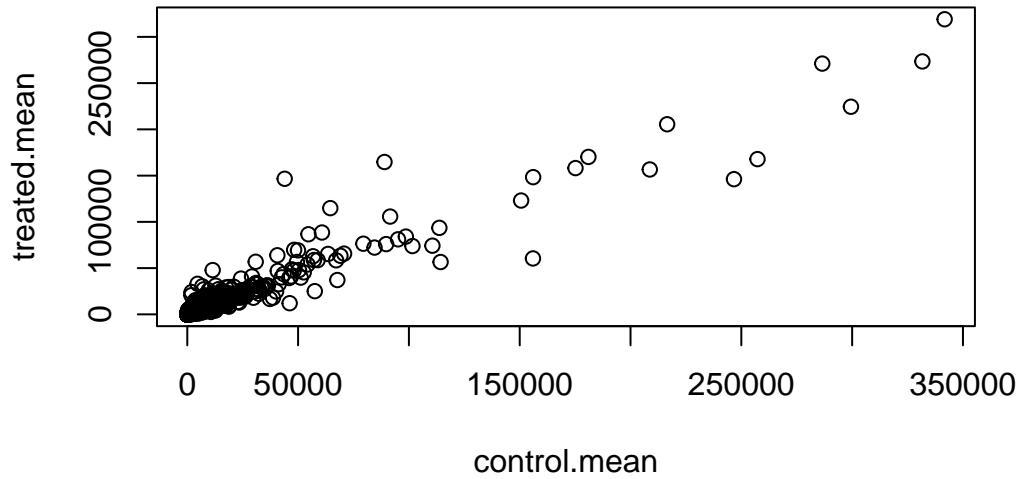
2. Let's calculate the mean counts per gene for "control" and "treated" - then we can compare these. Let's call it `control.mean` and `treated.mean`.

I can use the `apply()` function to apply `mean()` over the rows or columns of any `data.frame`.

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

Put these two together for ease of book-keeping

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

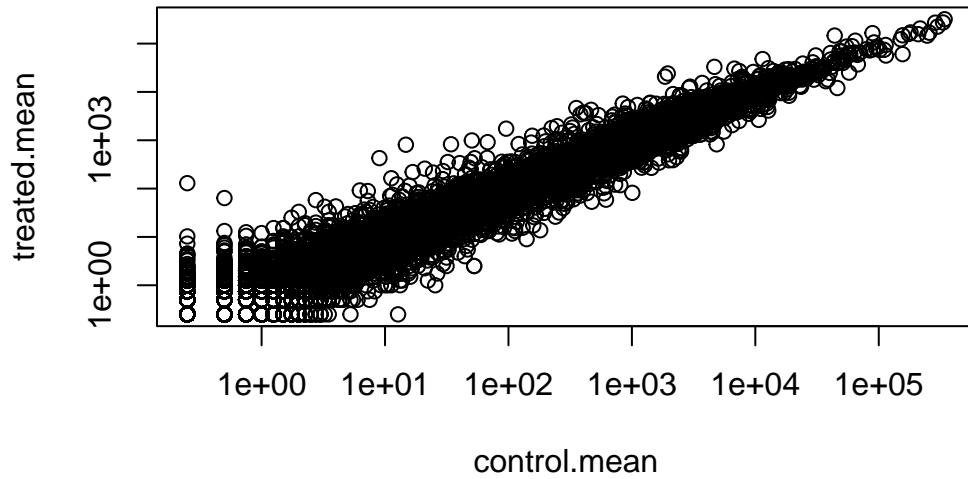


These data are heavily skewed. Let's transform it using logs.

```
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
```



We most often use `log2` transforms here because it makes the math easier

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

[1] 0

0= no change

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

[1] 1

1 = increase

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

[1] -1

-1 = decrease

```
log2(40/10)
```

```
[1] 2
```

\*log2 change of +2 is 4x increase ( $2 \times 2$ )

Let's calculate the Log2 fold-change of the drug and add it to our table, `meancounts`

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

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000005	0.00	0.00	NaN
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000938	0.75	0.00	-Inf

Filter out all genes with zero counts in either control or treated:

```
to.rm <- rowSums(meancounts[,1:2] == 0) > 0
mycounts <- meancounts[!to.rm,]

nrow(mycounts)
```

```
[1] 21817
```

```
head(mycounts)
```

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000971	5219.00	6687.50	0.35769358
ENSG00000001036	2327.00	1785.75	-0.38194109

How many “down” regulated genes do we have at the common log2 fold change values of -2?

```
sum(mycounts$log2fc < -2)
```

```
[1] 367
```

There are 367 “down” regulated genes with a log2FC of less than -2.

How many “up” at log2FC > +2?

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

```
[1] 250
```

There are 250 “up” regulated genes with a log2FC of greater than +2.

Do we trust these results? Is there anything missing? > There are no statistics! Give the people a p-value.

## DESeq analysis

```
library(DESeq2)
```

DESeq, like many BioConductor packages, wants our input data in a very specific format.

```
#have to give the countData and the colData. The ~ indicates the design info is in the dex col
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
```

The main function of DESeq2 is called `DESeq()` and this will do much of the statistical analysis for us.

```
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)
```

```
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

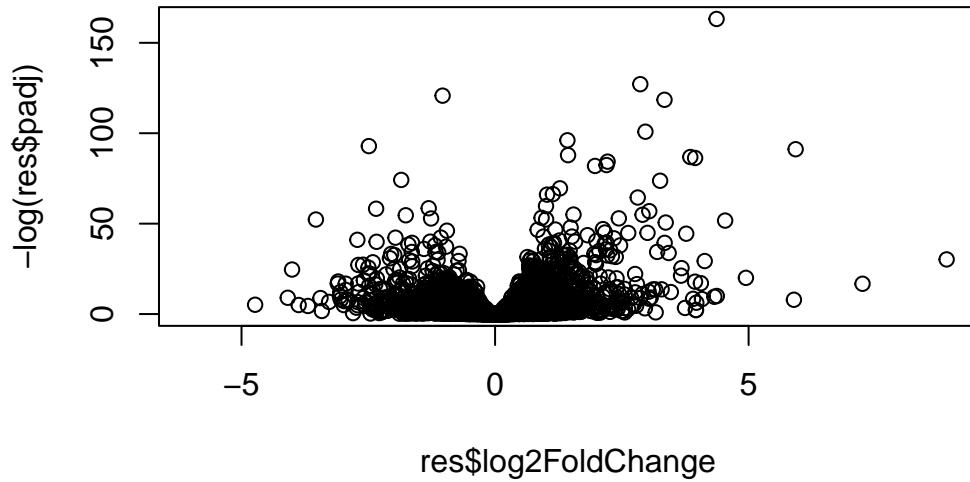
	baseMean	log2FoldChange	lfcSE	stat	pvalue
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>
ENSG00000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG00000000005	0.000000	NA	NA	NA	NA
ENSG00000000419	520.134160	0.2061078	0.101059	2.039475	0.0414026
ENSG00000000457	322.664844	0.0245269	0.145145	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

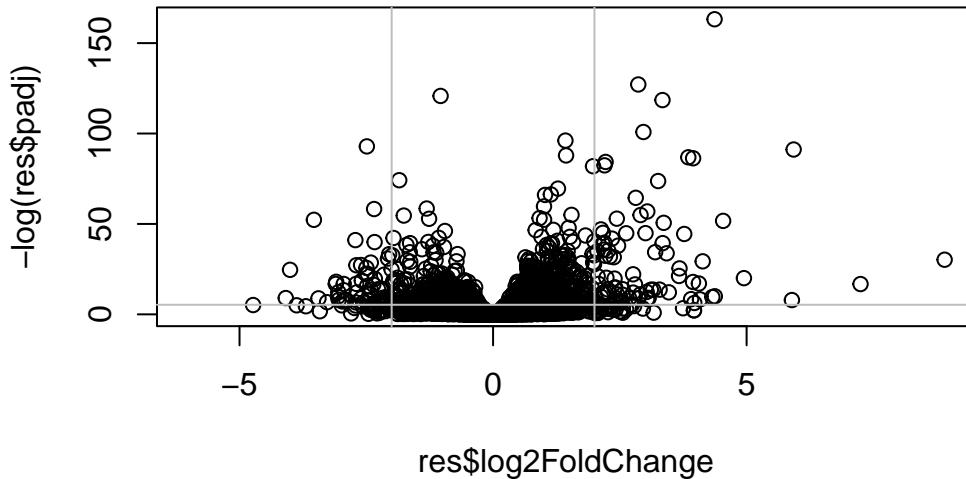
A common overview figure plots the logFC s P-value

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



(the points that are at the top have a smaller p-value— you transformed it w/ log, and then inverted it so the plot is oriented “correctly”)

```
plot(res$log2FoldChange, -log(res$padj))
abline(v=c(-2,2), col="grey")
abline(h=-log(0.005), col="grey")
```



```

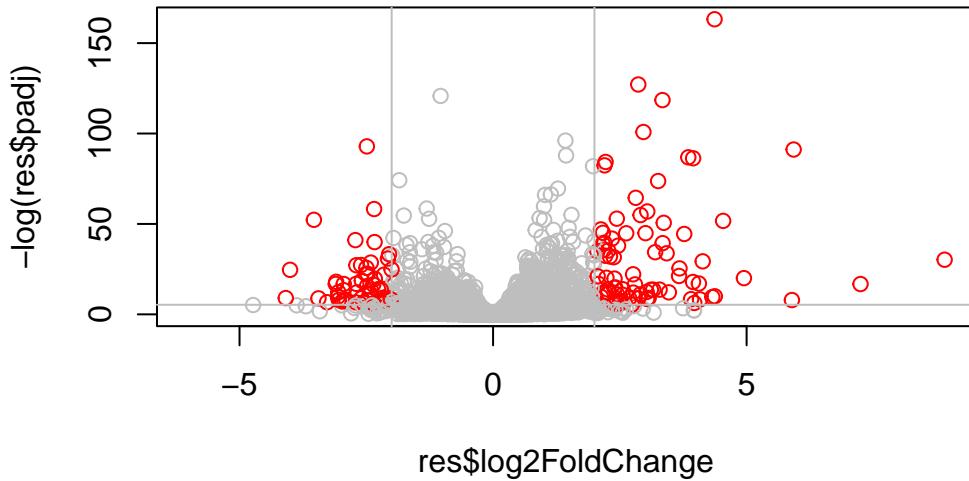
mycols <- rep("grey", nrow(res))
head(mycols)

[1] "grey" "grey" "grey" "grey" "grey" "grey"

mycols[res$log2FoldChange > 2] <- "red"
mycols[res$log2FoldChange < -2] <- "red"
mycols[res$padj > 0.005] <- "gray"

plot(res$log2FoldChange, -log(res$padj), col=mycols)
abline(v=c(-2,2), col="gray")
abline(h=-log(0.005), col="gray")

```



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

Let's annotate things!

```
library("AnnotationDbi")
library("org.Hs.eg.db")
```

```
columns(org.Hs.eg.db)
```

```
[1] "ACNUM"          "ALIAS"           "ENSEMBL"         "ENSEMLPROT"      "ENSEMLTRANS"
[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
```

```
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>
ENSG000000000003 747.194195 -0.3507030 0.168246 -2.084470 0.0371175
ENSG000000000005  0.000000      NA       NA       NA       NA
ENSG000000000419 520.134160  0.2061078 0.101059  2.039475 0.0414026
ENSG000000000457 322.664844  0.0245269 0.145145  0.168982 0.8658106
ENSG000000000460  87.682625 -0.1471420 0.257007 -0.572521 0.5669691
ENSG000000000938  0.319167 -1.7322890 3.493601 -0.495846 0.6200029
  padj      symbol
  <numeric> <character>
ENSG000000000003 0.163035   TSPAN6
ENSG000000000005  NA        TNMD
ENSG000000000419 0.176032   DPM1
ENSG000000000457 0.961694   SCYL3
ENSG000000000460 0.815849   FIRRM
ENSG000000000938  NA        FGR
```

```
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)  
  
# Examine the first 2 pathways in this kegg set for humans  
head(kegg.sets.hs, 2)
```

```
$`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"
```

I must speak with KEGG! Change column to ENTREZID to be intelligible to KEGG.

```
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  
  
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>
ENSG000000000003 747.194195 -0.3507030  0.168246 -2.084470 0.0371175
ENSG000000000005 0.000000      NA        NA        NA        NA
ENSG000000000419 520.134160  0.2061078  0.101059  2.039475 0.0414026
ENSG000000000457 322.664844  0.0245269  0.145145  0.168982 0.8658106
ENSG000000000460 87.682625 -0.1471420  0.257007 -0.572521 0.5669691
ENSG000000000938 0.319167 -1.7322890  3.493601 -0.495846 0.6200029
      padj      symbol      entrez
      <numeric> <character> <character>
ENSG000000000003 0.163035    TSPAN6      7105
ENSG000000000005  NA        TNMD       64102
ENSG000000000419 0.176032    DPM1       8813
ENSG000000000457 0.961694    SCYL3      57147
ENSG000000000460 0.815849    FIRRM      55732
ENSG000000000938  NA        FGR        2268

```

I can now use the **gage** function to check for overlap with known KEGG pathways.

```

foldchanges <- res$log2FoldChange
names(foldchanges) <- res$entrez
head(foldchanges)

```

```

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

```

Lets get some results

```

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

```

```

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

head(keggres$less, 3)

```

	p.geomean	stat.mean	p.val
hsa05332 Graft-versus-host disease	0.0004250461	-3.473346	0.0004250461
hsa04940 Type I diabetes mellitus	0.0017820293	-3.002352	0.0017820293
hsa05310 Asthma	0.0020045888	-3.009050	0.0020045888
	q.val	set.size	exp1
hsa05332 Graft-versus-host disease	0.09053483	40	0.0004250461
hsa04940 Type I diabetes mellitus	0.14232581	42	0.0017820293
hsa05310 Asthma	0.14232581	29	0.0020045888

```
pathview(gene.data=foldchanges, pathway.id="hsa05310")
```

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

Info: Working in directory /Users/janiec-w/Desktop/Classes/BGGN213 Bioinformatics/pRojects/JC...

Info: Writing image file hsa05310.pathview.png

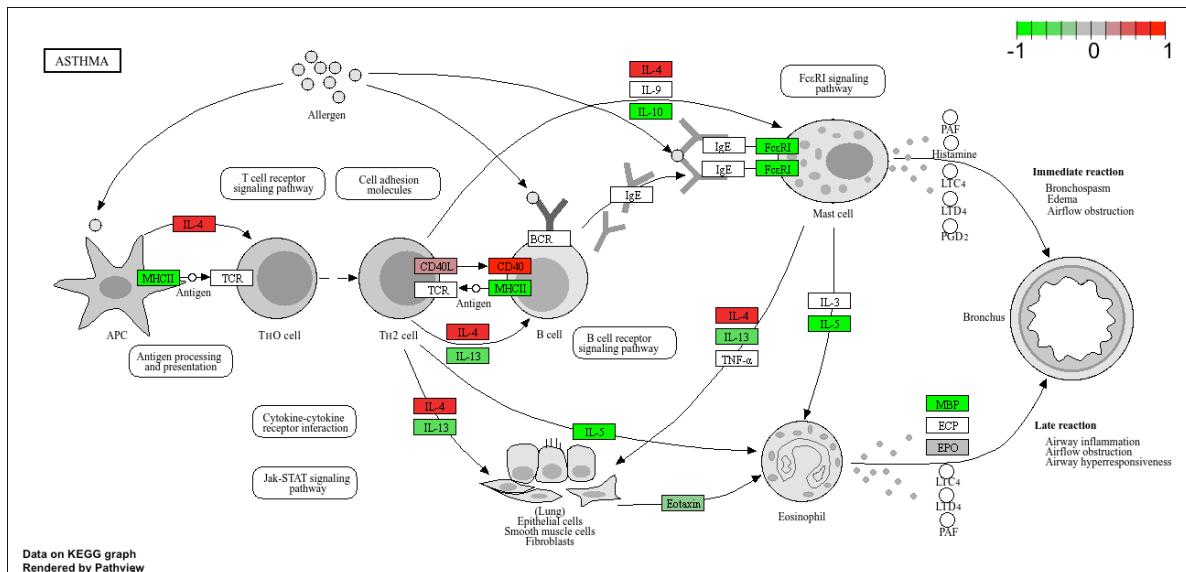


Figure 1: Genes up/down regulated in Asthma pathways