

class 13

Yiyu

Table of contents

Data Import	1
Mean counts per condition	2
Log fold change	6
DESeq analysis	7
Volcano Plot	9
Adding annotation data	19
Pathway Analysis	21

In today's class we will analyze some published RNA-seq experiments where airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects (Himes et al. 2014)

We will use the **DESeq2** package for the heavy lifting in a little but first let's read the data in and get to know how things work.

Data Import

There are two dat sets that I need for this type of analysis:

-contData: the transcript abundances (e.g. read counts per gene)
-colData: metadata about the columns in countData (i.e. experimental setup)

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

Q1. How many genes are in this dataset?

```
nrow(counts)
```

```
[1] 38694
```

Q2. How many “control” experiments are there in the dataset?

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

```
control treated
        4       4
```

Mean counts per condition

Let's find the average gene counts (i.e. rows) for control and treated conditions (i.e. columns)
-extract all “control” columns/experiments -then find the row wise average for these columns

```
control inds <- metadata$dex == "control"
control counts <- counts[,control inds]
control mean <- rowMeans(control counts)
```

Do the same to produce `treated.mean`

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

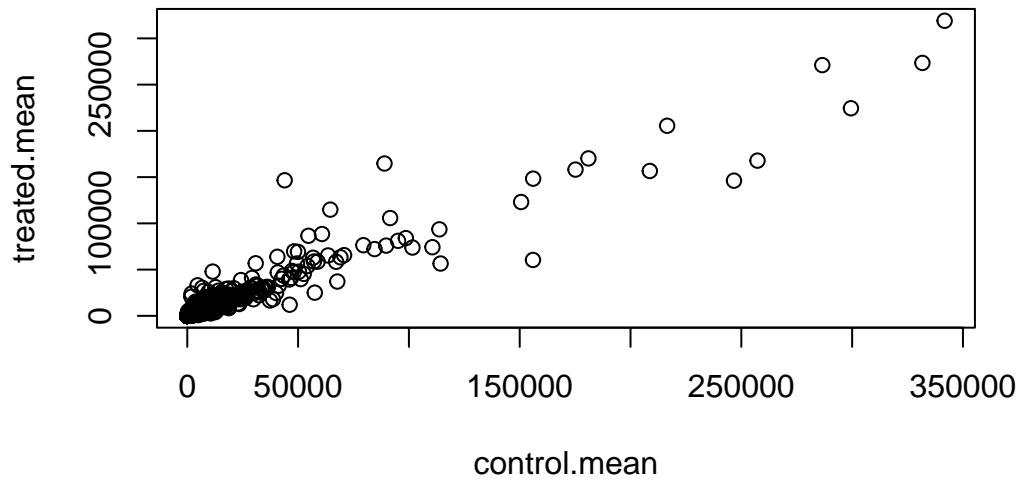
Let's store these mean values all in one data.frame called `meancounts`.

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

	control mean	treated mean
ENSG000000000003	900.75	658.00
ENSG000000000005	0.00	0.00
ENSG000000000419	520.50	546.00
ENSG000000000457	339.75	316.50
ENSG000000000460	97.25	78.75
ENSG000000000938	0.75	0.00

Make of plot of control vs. treated

```
plot(meancounts)
```

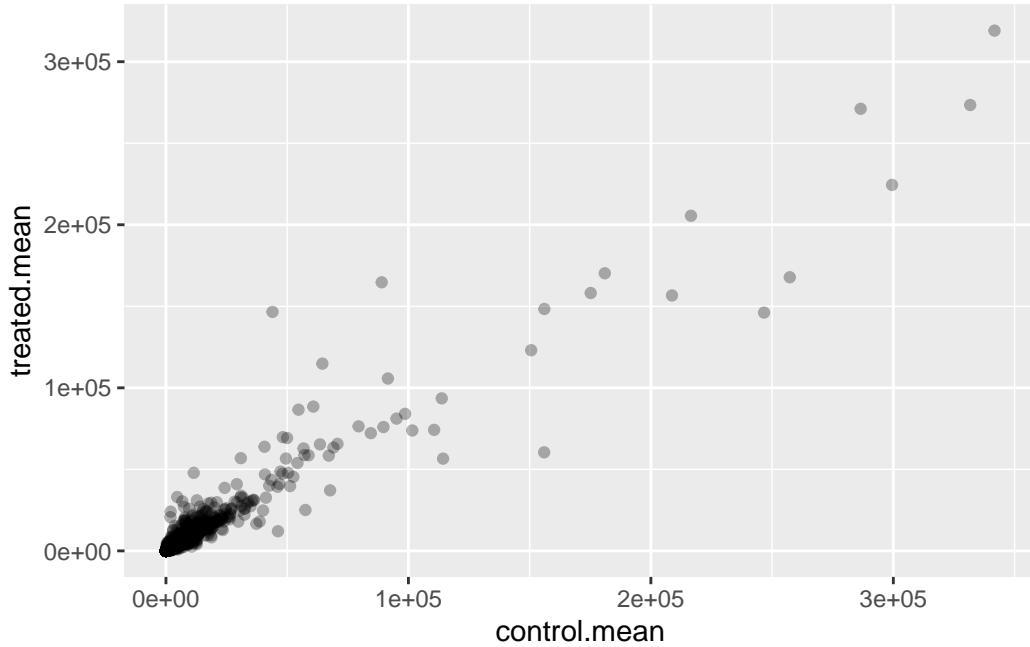


Where are all of the genes??

In ggplot, can better see density of plot with low alpha.

```
library(ggplot2)

ggplot(meancounts) +
  aes (control.mean, treated.mean) +
  geom_point(alpha = 0.3)
```

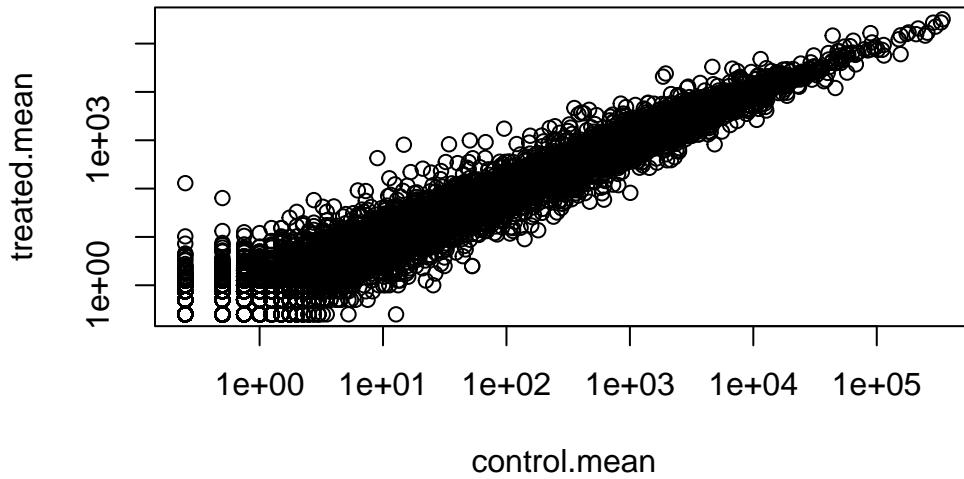


Data is heavily skewed, we can transform the plot to better see.

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

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

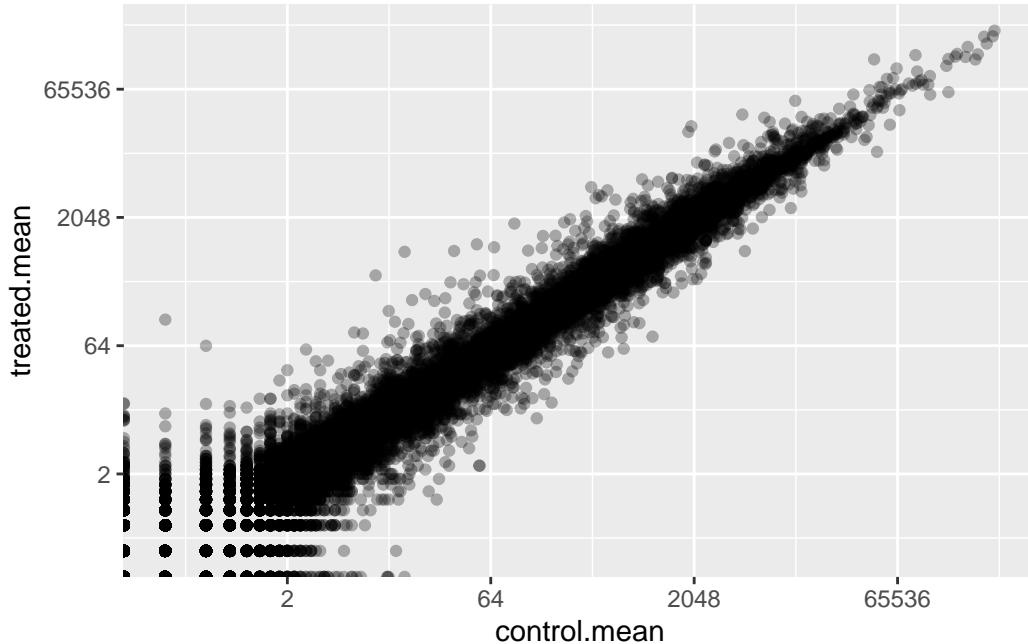
```
Warning in xy.coords(x, y, xlabel, ylabel, log): 15281 y values <= 0 omitted  
from logarithmic plot
```



```
ggplot(meancounts) +  
  aes (control.mean, treated.mean) +  
  geom_point(alpha = 0.3) +  
  scale_x_continuous(trans = "log2") +  
  scale_y_continuous(trans = "log2")
```

Warning in scale_x_continuous(trans = "log2"): log-2 transformation introduced infinite values.

Warning in scale_y_continuous(trans = "log2"): log-2 transformation introduced infinite values.



Log fold change

We most often work with log2 units - why? Because interpretation is much more straightforward.

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

```
[1] 0
```

```
log2(20/40)
```

```
[1] -1
```

```
log2(40/20)
```

```
[1] 1
```

Let's calculate the log2 fold change (`log2fc`) of treated/control for our data.

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

There are some weird values in the log2fc column like -Inf and NaN all because I have zero count genes. I need to filter these out (i.e. remove them) before going any further

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

Q. How many non-zero genes do we have left?

```
nrow(mycounts)
```

```
[1] 21817
```

Q. How many genes are “up” regulated at a log2fc > 2?

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

```
[1] 250
```

Q. How many genes are “down” regulated with a log2fc < 2?

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

```
[1] 367
```

Q. Do you trust these results? Why or why not?

Not quite, we need to account for significance still.

To do this analysis properly we can use the bioConductor package **DESeq2**:

DESeq analysis

```
library(DESeq2)
```

Like most BioConductor packages DESeq wants it's input in a 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

```
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>
ENSG000000000003	747.194195	-0.3507030	0.168246	-2.084470	0.0371175
ENSG000000000005	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

```

<numeric>
ENSG000000000003 0.163035
ENSG000000000005 NA
ENSG000000000419 0.176032
ENSG000000000457 0.961694
ENSG000000000460 0.815849
ENSG000000000938 NA

```

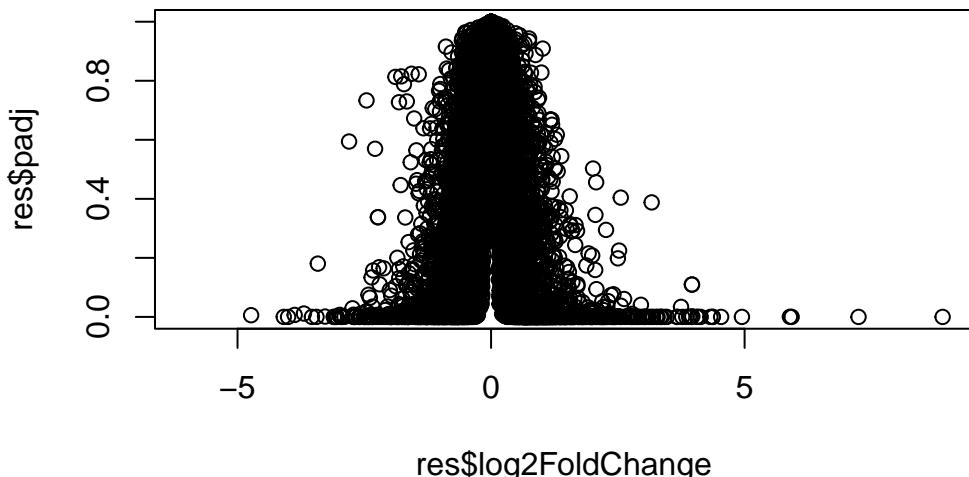
Save out results to CSV file:

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

Volcano Plot

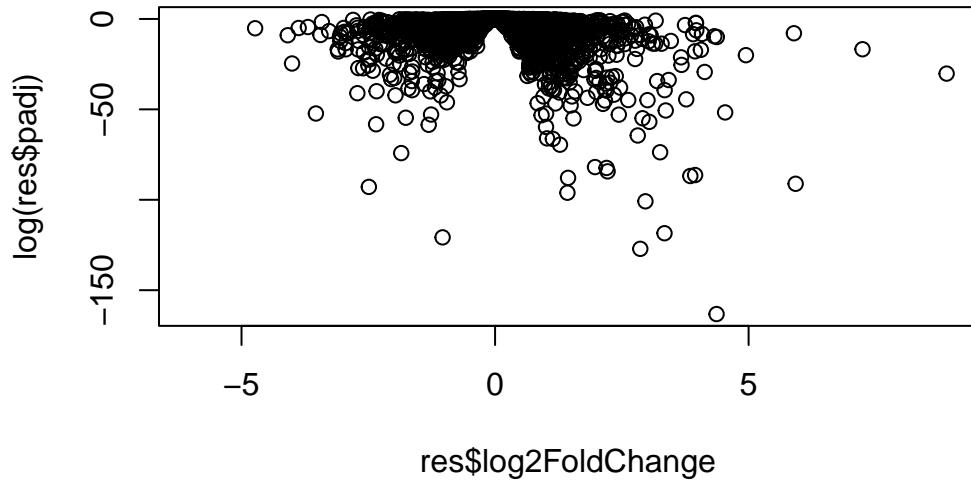
Let's make a common summary plot of our results. Our main results are the log2fc and adjusted p-value.

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



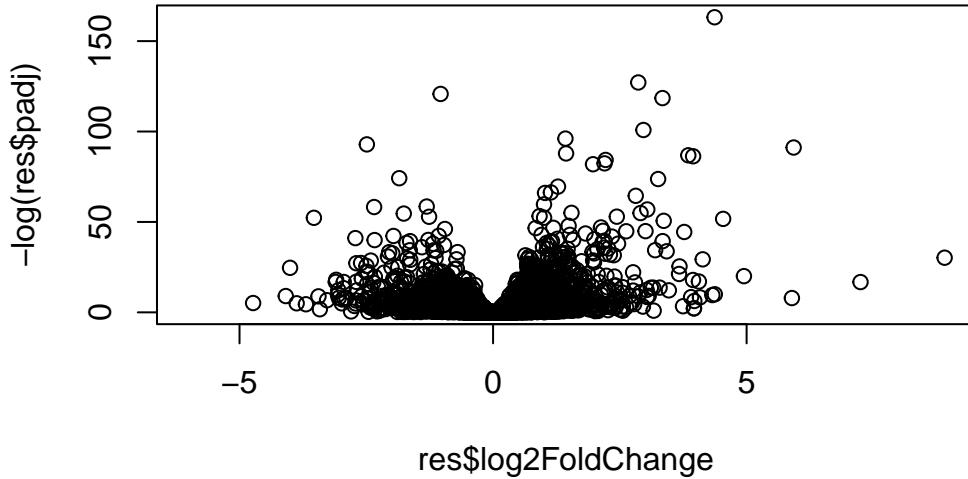
We need to transform p-value axis because we're interested in the very low p-values, not any values with large p-values.

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



We are interested in the larger values (larger negative in the above plot). We tend to flip the y axis so that it's $-\log(p\text{-value})$.

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



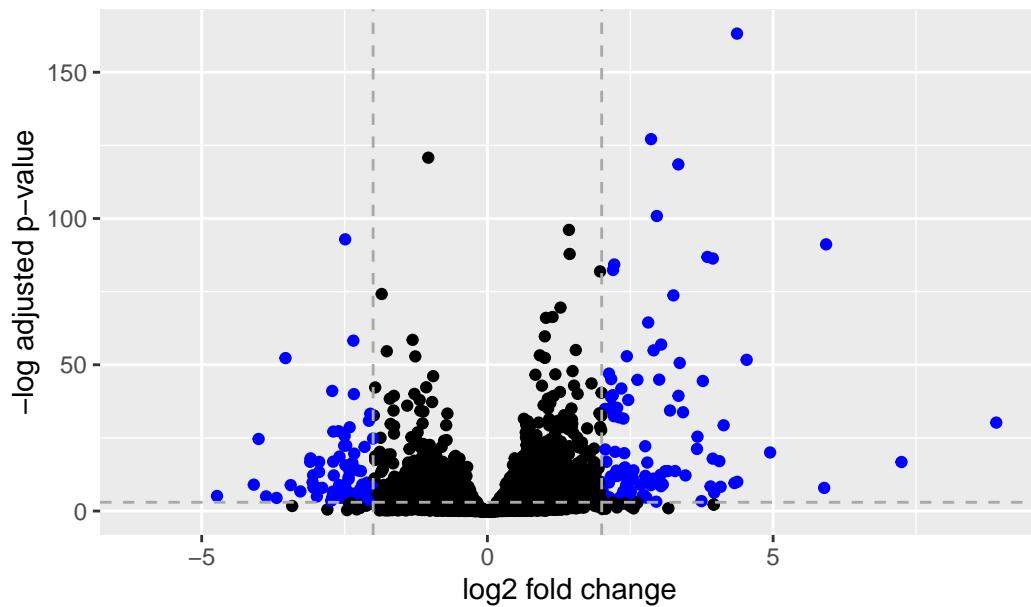
This is our standard volcano plot. Let's plot it in ggplot to make it more interesting. We can use color to highlight the most important subset of transcripts with a $\log_{2}\text{FC} > +2$ and <-2 that have a $p\text{-value} < 0.05$. We will need a custom color vector for this `mycols`

```
mycols <- rep("black", nrow(res))
mycols[res$log2FoldChange >= 2] <- "blue"
mycols[res$log2FoldChange <= -2] <- "blue"
mycols[ res$padj >= 0.05] <- "black"
```

```
ggplot(res) +
  aes(log2FoldChange, -log(padj)) +
  geom_point(col=mycols) +
  labs(title = "Summary Volcano Plot") +
  xlab( "log2 fold change") +
  ylab (" -log adjusted p-value") +
  geom_vline(xintercept = c(-2,2), col = "darkgray", lty = 2) +
  geom_hline (yintercept = -log(0.05), col = "darkgray", lty =2)
```

Warning: Removed 23549 rows containing missing values or values outside the scale range
(`geom_point()`).

Summary Volcano Plot



```
theme_bw()
```

```
List of 136
$ line                               :List of 6
..$ colour      : chr "black"
..$ linewidth   : num 0.5
..$ linetype    : num 1
..$ lineend     : chr "butt"
..$ arrow       : logi FALSE
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_line" "element"
$ rect                               :List of 5
..$ fill       : chr "white"
..$ colour     : chr "black"
..$ linewidth  : num 0.5
..$ linetype   : num 1
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_rect" "element"
$ text                               :List of 11
..$ family     : chr ""
..$ face       : chr "plain"
..$ colour     : chr "black"
```

```

..$ size          : num 11
..$ hjust         : num 0.5
..$ vjust         : num 0.5
..$ angle         : num 0
..$ lineheight    : num 0.9
..$ margin        : 'margin' num [1:4] 0points 0points 0points 0points
... .-. attr(*, "unit")= int 8
..$ debug         : logi FALSE
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ title          : NULL
$ aspect.ratio   : NULL
$ axis.title     : NULL
$ axis.title.x   :List of 11
..$ family        : NULL
..$ face          : NULL
..$ colour        : NULL
..$ size          : NULL
..$ hjust         : NULL
..$ vjust         : num 1
..$ angle         : NULL
..$ lineheight    : NULL
..$ margin        : 'margin' num [1:4] 2.75points 0points 0points 0points
... .-. attr(*, "unit")= int 8
..$ debug         : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.top :List of 11
..$ family        : NULL
..$ face          : NULL
..$ colour        : NULL
..$ size          : NULL
..$ hjust         : NULL
..$ vjust         : num 0
..$ angle         : NULL
..$ lineheight    : NULL
..$ margin        : 'margin' num [1:4] 0points 0points 2.75points 0points
... .-. attr(*, "unit")= int 8
..$ debug         : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.x.bottom : NULL
$ axis.title.y   :List of 11

```

```

..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : NULL
..$ hjust        : NULL
..$ vjust        : num 1
..$ angle        : num 90
..$ lineheight   : NULL
..$ margin       : 'margin' num [1:4] 0points 2.75points 0points 0points
... .-. attr(*, "unit")= int 8
..$ debug        : NULL
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.title.y.left      : NULL
$ axis.title.y.right     :List of 11
..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : NULL
..$ hjust        : NULL
..$ vjust        : num 1
..$ angle        : num -90
..$ lineheight   : NULL
..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.75points
... .-. attr(*, "unit")= int 8
..$ debug        : NULL
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text      :List of 11
..$ family      : NULL
..$ face        : NULL
..$ colour      : chr "grey30"
..$ size         : 'rel' num 0.8
..$ hjust        : NULL
..$ vjust        : NULL
..$ angle        : NULL
..$ lineheight   : NULL
..$ margin       : NULL
..$ debug        : NULL
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x     :List of 11
..$ family      : NULL

```

```

..$ face      : NULL
..$ colour    : NULL
..$ size       : NULL
..$ hjust      : NULL
..$ vjust      : num 1
..$ angle      : NULL
..$ lineheight : NULL
..$ margin     : 'margin' num [1:4] 2.2points 0points 0points 0points
... .-. attr(*, "unit")= int 8
..$ debug      : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.top          :List of 11
..$ family     : NULL
..$ face       : NULL
..$ colour     : NULL
..$ size       : NULL
..$ hjust      : NULL
..$ vjust      : num 0
..$ angle      : NULL
..$ lineheight : NULL
..$ margin     : 'margin' num [1:4] 0points 0points 2.2points 0points
... .-. attr(*, "unit")= int 8
..$ debug      : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.x.bottom        : NULL
$ axis.text.y          :List of 11
..$ family     : NULL
..$ face       : NULL
..$ colour     : NULL
..$ size       : NULL
..$ hjust      : num 1
..$ vjust      : NULL
..$ angle      : NULL
..$ lineheight : NULL
..$ margin     : 'margin' num [1:4] 0points 2.2points 0points 0points
... .-. attr(*, "unit")= int 8
..$ debug      : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.y.left         : NULL
$ axis.text.y.right        :List of 11

```

```

..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : NULL
..$ hjust        : num 0
..$ vjust        : NULL
..$ angle        : NULL
..$ lineheight   : NULL
..$ margin       : 'margin' num [1:4] 0points 0points 0points 2.2points
... .-. attr(*, "unit")= int 8
..$ debug        : NULL
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.text.theta          : NULL
$ axis.text.r               :List of 11
..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : NULL
..$ hjust        : num 0.5
..$ vjust        : NULL
..$ angle        : NULL
..$ lineheight   : NULL
..$ margin       : 'margin' num [1:4] 0points 2.2points 0points 2.2points
... .-. attr(*, "unit")= int 8
..$ debug        : NULL
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_text" "element"
$ axis.ticks            :List of 6
..$ colour       : chr "grey20"
..$ linewidth    : NULL
..$ linetype     : NULL
..$ lineend      : NULL
..$ arrow         : logi FALSE
..$ inherit.blank: logi TRUE
... - attr(*, "class")= chr [1:2] "element_line" "element"
$ axis.ticks.x           : NULL
$ axis.ticks.x.top        : NULL
$ axis.ticks.x.bottom     : NULL
$ axis.ticks.y           : NULL
$ axis.ticks.y.left       : NULL
$ axis.ticks.y.right      : NULL
$ axis.ticks.theta        : NULL

```

```

$ axis.ticks.r : NULL
$ axis.minor.ticks.x.top : NULL
$ axis.minor.ticks.x.bottom : NULL
$ axis.minor.ticks.y.left : NULL
$ axis.minor.ticks.y.right : NULL
$ axis.minor.ticks.theta : NULL
$ axis.minor.ticks.r : NULL
$ axis.ticks.length : 'simpleUnit' num 2.75points
..- attr(*, "unit")= int 8
$ axis.ticks.length.x : NULL
$ axis.ticks.length.x.top : NULL
$ axis.ticks.length.x.bottom : NULL
$ axis.ticks.length.y : NULL
$ axis.ticks.length.y.left : NULL
$ axis.ticks.length.y.right : NULL
$ axis.ticks.length.theta : NULL
$ axis.ticks.length.r : NULL
$ axis.minor.ticks.length : 'rel' num 0.75
$ axis.minor.ticks.length.x : NULL
$ axis.minor.ticks.length.x.top : NULL
$ axis.minor.ticks.length.x.bottom: NULL
$ axis.minor.ticks.length.y : NULL
$ axis.minor.ticks.length.y.left : NULL
$ axis.minor.ticks.length.y.right : NULL
$ axis.minor.ticks.length.theta : NULL
$ axis.minor.ticks.length.r : NULL
$ axis.line : list()
..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ axis.line.x : NULL
$ axis.line.x.top : NULL
$ axis.line.x.bottom : NULL
$ axis.line.y : NULL
$ axis.line.y.left : NULL
$ axis.line.y.right : NULL
$ axis.line.theta : NULL
$ axis.line.r : NULL
$ legend.background :List of 5
..$ fill : NULL
..$ colour : logi NA
..$ linewidth : NULL
..$ linetype : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_rect" "element"

```

```

$ legend.margin           : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.5points
..- attr(*, "unit")= int 8
$ legend.spacing          : 'simpleUnit' num 11points
..- attr(*, "unit")= int 8
$ legend.spacing.x        : NULL
$ legend.spacing.y        : NULL
$ legend.key              : NULL
$ legend.key.size         : 'simpleUnit' num 1.2lines
..- attr(*, "unit")= int 3
$ legend.key.height       : NULL
$ legend.key.width        : NULL
$ legend.key.spacing      : 'simpleUnit' num 5.5points
..- attr(*, "unit")= int 8
$ legend.key.spacing.x    : NULL
$ legend.key.spacing.y    : NULL
$ legend.frame             : NULL
$ legend.ticks             : NULL
$ legend.ticks.length     : 'rel' num 0.2
$ legend.axis.line         : NULL
$ legend.text              :List of 11
..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : 'rel' num 0.8
..$ hjust        : NULL
..$ vjust        : NULL
..$ angle        : NULL
..$ lineheight   : NULL
..$ margin       : NULL
..$ debug        : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.text.position    : NULL
$ legend.title            :List of 11
..$ family      : NULL
..$ face        : NULL
..$ colour      : NULL
..$ size         : NULL
..$ hjust        : num 0
..$ vjust        : NULL
..$ angle        : NULL
..$ lineheight   : NULL
..$ margin       : NULL

```

```

..$ debug      : NULL
..$ inherit.blank: logi TRUE
..- attr(*, "class")= chr [1:2] "element_text" "element"
$ legend.title.position      : NULL
$ legend.position            : chr "right"
$ legend.position.inside     : NULL
$ legend.direction          : NULL
$ legend.byrow              : NULL
$ legend.justification       : chr "center"
$ legend.justification.top   : NULL
$ legend.justification.bottom: NULL
$ legend.justification.left  : NULL
$ legend.justification.right : NULL
$ legend.justification.inside: NULL
$ legend.location           : NULL
$ legend.box                 : NULL
$ legend.box.just            : NULL
$ legend.box.margin          : 'margin' num [1:4] 0cm 0cm 0cm 0cm
..- attr(*, "unit")= int 1
$ legend.box.background       : list()
..- attr(*, "class")= chr [1:2] "element_blank" "element"
$ legend.box.spacing          : 'simpleUnit' num 11points
..- attr(*, "unit")= int 8
[list output truncated]
- attr(*, "class")= chr [1:2] "theme" "gg"
- attr(*, "complete")= logi TRUE
- attr(*, "validate")= logi TRUE

```

Adding annotation data

At the minute all we know about the genes in our dataset is their ENSEMBLE database id

```
head(rownames(res))
```

```
[1] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457"
[5] "ENSG00000000460" "ENSG00000000938"
```

We can use a set of BioConductor packages to map these ENSEMBLE ids to things like GENE SYMBOL, REFSEQ id, ENTREZ id etc. In other words, what each gene is called in different databases that I might want to use for further analysis.

I install these packages with `BiocManager::install()`

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

The different formats that I can convert IDs between include:

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

```
[1] "ACCCNUM"      "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 use `mapIds()` function to do this “mapping”/conversion:

```
res$symbol <- mapIds(org.Hs.eg.db,
                      keys=row.names(res), # Our genenames
                      keytype="ENSEMBL",   # The format of our genenames
                      column="SYMBOL",    # The new format we want to add
                      multiVals="first")
```

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

```
res$genename <- mapIds(org.Hs.eg.db,
                        keys=row.names(res),
                        keytype="ENSEMBL",
                        column="GENENAME",   # The new format we want
                        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",   # The new format we want
                       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 9 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		genename	entrez
	<numeric>	<character>		<character>	<character>
ENSG000000000003	0.163035	TSPAN6	tetraspanin 6	7105	
ENSG000000000005	NA	TNMD	tenomodulin	64102	
ENSG000000000419	0.176032	DPM1 dolichyl-phosphate m..		8813	
ENSG000000000457	0.961694	SCYL3 SCY1 like pseudokina..		57147	
ENSG000000000460	0.815849	FIRRM FIGNL1 interacting r..		55732	
ENSG000000000938	NA	FGR FGR proto-oncogene, ..		2268	

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

Pathway Analysis

Let's use KEGG to see which pathways my gene sets overlap with - i.e. highlight the biology that may be influenced by the dex drug treatment.

We will use the following packages: `BiocManager::install(c("pathview", "gage", "gageData"))`

```
library(pathview)
```

```
#####
Pathview is an open source software package distributed under GNU General
Public License version 3 (GPLv3). Details of GPLv3 is available at
http://www.gnu.org/licenses/gpl-3.0.html. Particullary, users are required to
```

formally cite the original Pathview paper (not just mention it) in publications or products. For details, do citation("pathview") within R.

The pathview downloads and uses KEGG data. Non-academic uses may require a KEGG license agreement (details at <http://www.kegg.jp/kegg/legal.html>).

```
#####
```

```
library(gage)
```

```
library(gageData)
data(kegg.sets.hs)
```

The `gage` function wants as input a “named vector of importance”

```
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
```

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

```
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

We can have a quick look at one of the highlighted pathways e.g. hsa05310

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

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

```
Info: Working in directory C:/Users/yiyuw/Desktop/BGGN 213/class 13
```

```
Info: Writing image file hsa05310.pathview.png
```

