Class12: RNASeq Analysis

Dani Weatherwax (PID: A17856408)

Table of contents

Quarto 1

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

##Background Today we will analyze some RNASeq data from Himes et al. on the effects of a common steroid on airway smooth muscle cells (ASM cells). Our starting pointis the "counts" data and "metadata" that contain the count values for each gene in their different experiements (i.e. cell lines iwth or without the drug)

##Data import

```
counts <- read.csv("~/Downloads/airway_scaledcounts.csv", row.names = 1)

metadata <- read.csv("~/Downloads/airway_metadata.csv")
```

```
nrow(counts)
```

```
[1] 38694
```

> Q1. How many different genes(columns in counts or rows in metadata) are there? There are 38,694 genes.

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

```
[1] 4
```

Q2. How many different 'control' cell lines do we have? 4

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
control <- metadata %>% filter(dex=="control")
control.counts <- counts %>% select(control$id)
control.mean <- rowMeans(control.counts)
head(control.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         900.75            0.00          520.50          339.75           97.25
ENSG00000000938
           0.75
```

```
library(dplyr)
treated <- metadata %>% filter(dex=="treated")
treated.counts <- counts %>% select(treated$id)
treated.mean <- rowMeans(treated.counts)
head(treated.mean)
```

```
ENSG00000000003 ENSG00000000005 ENSG00000000419 ENSG00000000457 ENSG00000000460
         658.00            0.00          546.00          316.50           78.75
ENSG00000000938
           0.00
```

Q3. You could change the divide by four into a "mean" function so that you don't have to correct if more data points are added.

1. Extract all "control" columns from the 'counts' object

2. Calculate the mean across all control genes 3-4. Do the same for "treated"
3. Compare these 'control.mean' and 'treated.mean' values.
4. Store these together for ease of bookkeeping

```
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
```

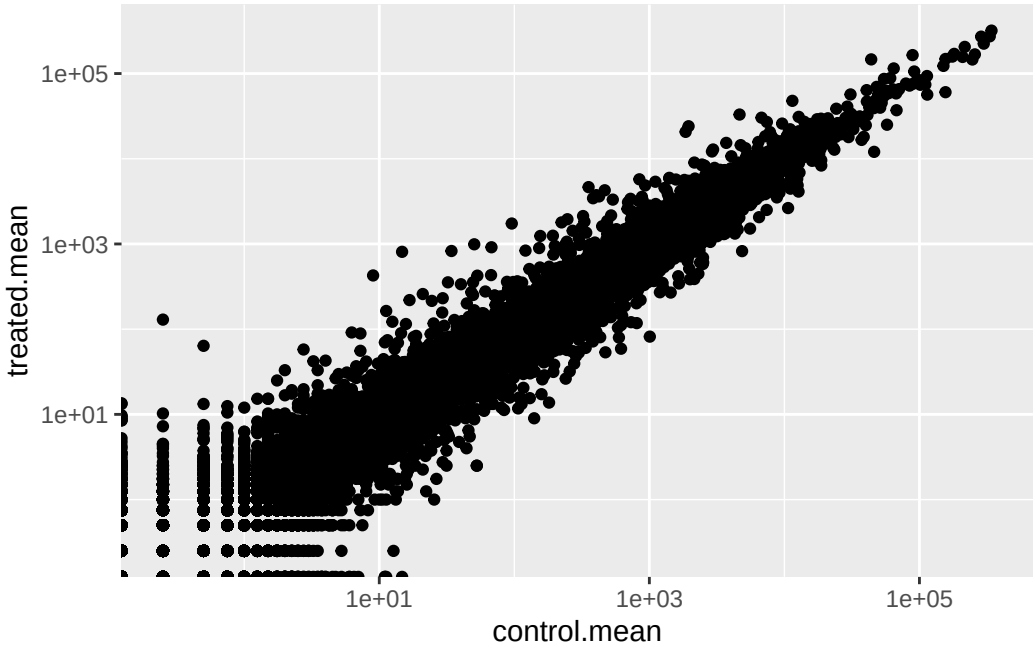
Q5b. You would use geom_point to create a scatter plot.

Q6. You add scale_x_log10() and scale_y_log10().

```
library(ggplot2)
ggplot(meancounts, aes(control.mean, treated.mean))+
  geom_point()+
scale_x_log10()+
scale_y_log10()
```

```
Warning in scale_x_log10(): log-10 transformation introduced infinite values.
```

```
Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```



Let's calculate the log2 fold change for our treated over control mean counts.

```
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
```

A common "rule of thumb" is a log2 fold change cutoff of +2 and -2 and to call genes "up regulated" or "down regulated"

```
sum(meancounts$log2fc>=2, na.rm=TRUE)
```

```
[1] 1910
```

```
sum(meancounts$log2fc<=-2, na.rm=TRUE)
```

```
[1] 2330
```

Q8. We have 1910 upregulated genes >2.

Q9. We have 2330 downregulated genes >2.

Q10. We don't know if these results are statistically significant or not.

Let's do this analysis properly and keep our inner stats nerd happy- i.e. are the differences we see between drug and no drug significant given the replicate experiments?

```
library(DESeq2)
```

For DESeq analysis we need 3 things: - Count values (`countdata`) -metadata telling us about in columns in `countdata`(`coldata`) -design of the experiment (i.e. what do you want to compare)

Our first function from DESeq2 will setup the input required for analysis by storing all these 3 things together.

```
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
```

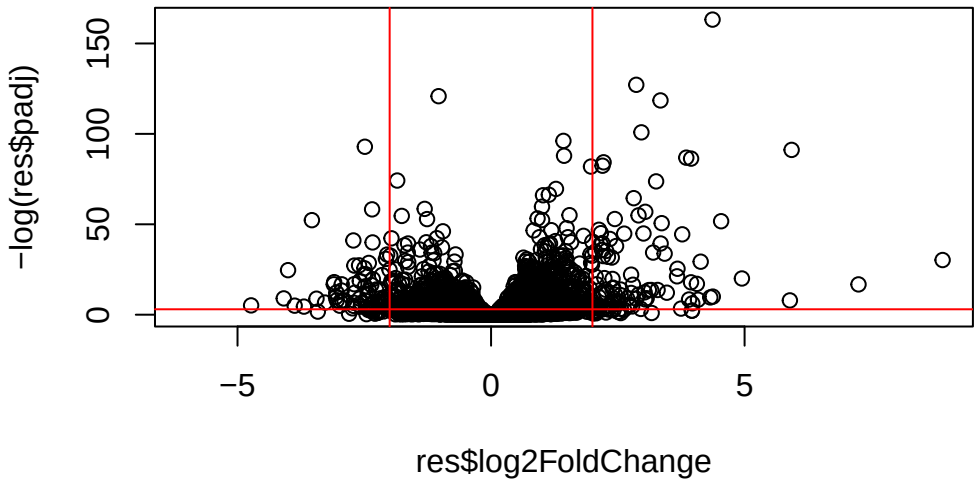
```
final dispersion estimates

fitting model and testing
```

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

##Volcano Plot

This is a common summary results figure from these types of experiments and plot the log2 fold-change vs. the adjusted p-value.

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




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

##Add gene annotation

To help make sense of our results and communicate them to other folks, we need to add some more annotation to our main `res` object.

We will use two bioconductor packages to first map IDs to different formats including the classic gene "symbol" gene name.

```
library("AnnotationDbi")
```

```
Attaching package: 'AnnotationDbi'

The following object is masked from 'package:dplyr':

    select
```

```
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 or "map" IDs between any of these 26 databases using the `mapIds()` function.

```
res$symbol <- mapIds(keys= row.names(res), #our current IDs
       keytype="ENSEMBL", #the format of our IDS
       x=org.Hs.eg.db, #where to get the mappings from
       column="SYMBOL" #the format/DB to map to
       )
```

```
'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      -0.350703  0.168242 -2.084514 0.0371134
ENSG00000000005   0.000000             NA        NA        NA        NA
ENSG00000000419 520.134160       0.206107  0.101042  2.039828 0.0413675
ENSG00000000457 322.664844       0.024527  0.145134  0.168996 0.8658000
ENSG00000000460  87.682625      -0.147143  0.256995 -0.572550 0.5669497
ENSG00000000938   0.319167      -1.732289  3.493601 -0.495846 0.6200029
                     padj      symbol
                <numeric> <character>
ENSG00000000003  0.163017      TSPAN6
ENSG00000000005        NA        TNMD
ENSG00000000419  0.175937        DPM1
ENSG00000000457  0.961682       SCYL3
ENSG00000000460  0.815805       FIRRM
ENSG00000000938        NA         FGR
```

Add the mappings for "GENENAME" and "ENTREZID" and store as `res$genename` and `res$entrez`

```
res$genename <- mapIds(keys= row.names(res), #our current IDs
       keytype="ENSEMBL", #the format of our IDS
       x=org.Hs.eg.db, #where to get the mappings from
       column="GENENAME" #the format/DB to map to
       )
```

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

```
res$entrez <- mapIds(keys= row.names(res), #our current IDs
       keytype="ENSEMBL", #the format of our IDS
       x=org.Hs.eg.db, #where to get the mappings from
       column="ENTREZID" #the format/DB to map to
       )
```

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

##Pathway Analysis

There are lots of bioconductor packages to do this type of analysis;for now, let's just try one called **gage** again we need to install this if we don't have it already.

```
library(gage)
```

```
library(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).
################################################################################
```

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

```
      7105       64102        8813       57147       55732        2268
-0.35070296          NA  0.20610728  0.02452701 -0.14714263 -1.73228897
```

```
data(kegg.sets.hs)
keggres<- gage(foldchanges, gsets=kegg.sets.hs)
```

```
attributes(keggres)
```

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

```
head(keggres$less, 5)
```

```
                                                      p.geomean stat.mean
hsa05332 Graft-versus-host disease                   0.0004250607 -3.473335
hsa04940 Type I diabetes mellitus                    0.0017820379 -3.002350
hsa05310 Asthma                                      0.0020046180 -3.009045
hsa04672 Intestinal immune network for IgA production 0.0060434609 -2.560546
hsa05330 Allograft rejection                         0.0073679547 -2.501416
                                                            p.val      q.val
hsa05332 Graft-versus-host disease                   0.0004250607 0.09053792
hsa04940 Type I diabetes mellitus                    0.0017820379 0.14232788
hsa05310 Asthma                                      0.0020046180 0.14232788
hsa04672 Intestinal immune network for IgA production 0.0060434609 0.31387487
hsa05330 Allograft rejection                         0.0073679547 0.31387487
                                                     set.size         exp1
hsa05332 Graft-versus-host disease                         40 0.0004250607
hsa04940 Type I diabetes mellitus                          42 0.0017820379
hsa05310 Asthma                                            29 0.0020046180
hsa04672 Intestinal immune network for IgA production      47 0.0060434609
hsa05330 Allograft rejection                               36 0.0073679547
```

Let's look at one of these pathways with our genes colored up so we can see the overlap

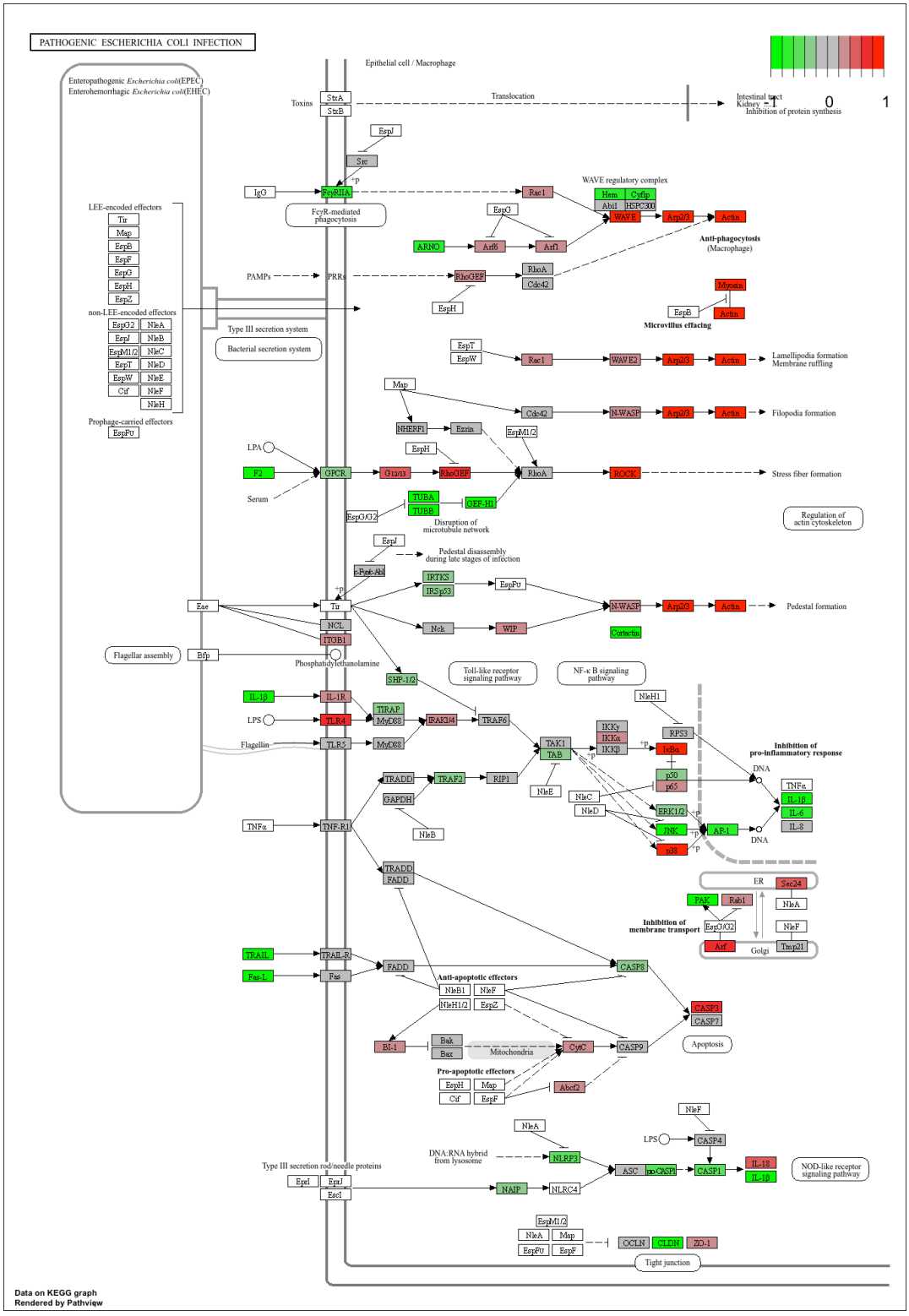
```
pathview(pathway.id="hsa05130", gene.data=foldchanges)
```

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

```
Info: Working in directory /Users/danielleweatherwax/Downloads/BIMM/Class12
```

```
Info: Writing image file hsa05130.pathview.png
```

Add this pathway figure to our lab report


PATHOGENIC ESCHERICHIA COLI INFECTION
Epithelial cell / Macrophage
Enteropathogenic Escherichia coli(EPEC)
Enterohemorrhagic Escherichia coli(EHEC)
Toxins
StxA
StxB
Translocation
Intestinal tract
Kidney
Inhibition of protein synthesis
EspJ
Src
IgG
FcγRIIA
FcγR-mediated phagocytosis
Rac1
WAVE regulatory complex
Hem
Cyfip
Abi1
HSPC300
WAVE
Arp2/3
Actin
EspG
ARNO
Arf6
Arf1
Anti-phagocytosis
(Macrophage)
LEE-encoded effectors
Tir
Map
EspB
EspF
EspG
EspH
EspZ
non-LEE-encoded effectors
EspG2
NleA
EspJ
NleB
EspM1/2
NleC
EspT
NleD
EspW
NleE
Cif
NleF
NleH
Prophage-carried effectors
EspFU
PAMPs
PRRs
RhoGEF
RhoA
Cdc42
EspH
Myosin
EspB
Actin
Microvillus effacing
Type III secretion system
Bacterial secretion system
EspT
EspW
Rac1
WAVE2
Arp2/3
Actin
Lamellipodia formation
Membrane ruffling
Map
Cdc42
N-WASP
Arp2/3
Actin
Filopodia formation
NHERF1
Ezrin
EspM1/2
LPA
F2
GPCR
G12/13
RhoGEF
RhoA
ROCK
Stress fiber formation
Serum
TUBA
TUBB
GEF-H1
EspG/G2
Disruption of microtubule network
Regulation of actin cytoskeleton
EspJ
Pedestal disassembly during late stages of infection
c-Pext-Abl
IRTKS
IRSp53
EspFU
Eae
Tir
NCL
ITGB1
N-WASP
Arp2/3
Actin
Pedestal formation
Nck
WIP
Cortactin
Flagellar assembly
Bfp
Phosphatidylethanolamine
SHP-1/2
Toll-like receptor signaling pathway
NF-κ B signaling pathway
NleH1
IL-1β
IL-1R
LPS
TLR4
TRAP
MyD88
IRAK3/4
TRAF6
IKKγ
IKKα
IKKβ
TAK1
TAB
RPS3
IκBα
p50
p65
Inhibition of pro-inflammatory response
DNA
TNFα
IL-1β
IL-6
IL-8
Flagellin
TLR5
MyD88
TRADD
TRAF2
RIP1
GAPDH
NleE
NleC
NleD
ERK1/2
JNK
p38
AP-1
DNA
TNFα
TNF-R1
NleB
TRADD
FADD
ER
Sec24
FAK
Rab1
NleA
Inhibition of membrane transport
EspG/G2
NleF
Golgi
Arf
Tmp21
TRAIL
TRAIL-R
Fas-L
Fas
FADD
CASP8
Anti-apoptotic effectors
NleB1
NleF
NleH1/2
EspZ
CASP3
CASP7
Bcl-1
Bak
Bax
Mitochondria
CytC
CASP9
Apoptosis
Pro-apoptotic effectors
EspH
Map
Cif
EspF
Abcf2
NleA
NleF
LPS
CASP4
DNA:RNA hybrid from lysosome
NLRP3
ASC
pro-CASP1
CASP1
IL-18
IL-1β
NOD-like receptor signaling pathway
Type III secretion rod/needle proteins
EprI
EprJ
EscI
NAIP
NLRC4
EspM1/2
NleA
Map
EspFU
EspF
OCLN
CLDN
ZO-1
Tight junction
Data on KEGG graph
Rendered by Pathview

##Save

our main results

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