12: RNA-Seq Analysis

Lourd Yakoob (PID: A18543409)

Table of contents

1. Background 1
2. Bioconductor setup 1
3. Import countData and colData 2
4. Toy differential gene expression 3
Volcano Plot 11
Save our results 12
Add gene annotation 12
Pathway Analysis 14
Save our main results 16

1. Background

Today we will analyze some RNAseq data from Himes et al. on the effects of a common steroid (dexamethasone) on airway smooth muscle cells (ASM cells)

Our starting point is the "counts" data and "metadata" thatcontain the count values for each gene in their different experiments (i.e. cell lines with or without the drug).

2. Bioconductor setup

Bioconductor package was installed and set up

3. Import countData and colData

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

Let's have a wee peak at these objects:

```
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
```

```
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
```

A: There are 38694 genes in this dataset.

> Q. How many different experiments (columns in counts or rows in metadata) are there?

```
ncol(counts)
```

```
[1] 8
```

A: There is a total of 8 different experiments.

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

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

```
control treated
      4       4
```

A: There are 4 control cell lines.

4. Toy differential gene expression

To start our analysis, let's calculate the mean counts for all genes in the "control" experiments.

1. Extract all "control" columns from the `counts` object.
2. Calculate the mean for all rows (i.e. genes) of these "control" columns

3-4. Do the same for "treated"

5. Compare these `control.mean` and `treated.mean` values.

1 + 2:

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

3 + 4:

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

> Q3. How would you make the above code in either approach more robust? Is there a function that could help here?

A:

> Q4. Follow the same procedure for the treated samples (i.e. calculate the mean per gene across drug treated samples and assign to a labeled vector called treated.mean)

A:

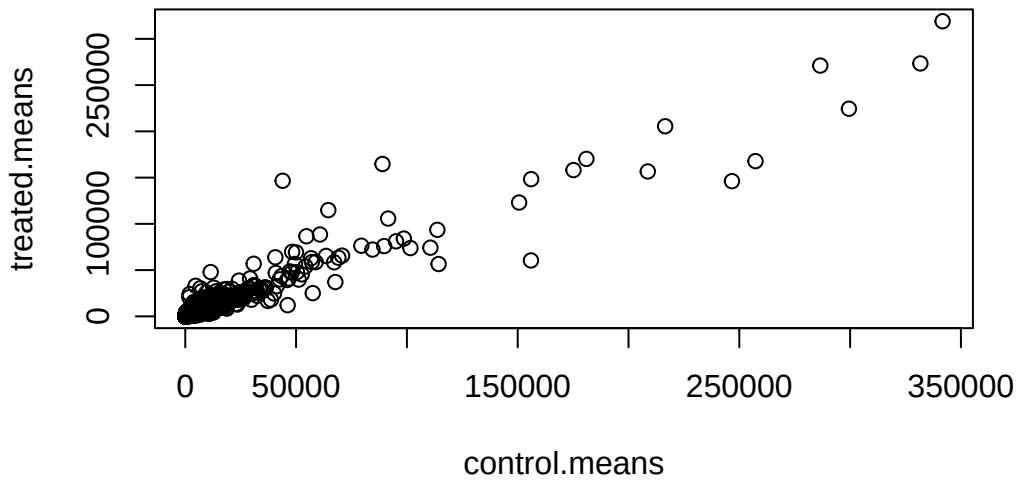
Store these together for ease of bookkeeping as `meancounts`

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

```
                control.means treated.means
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
```

Make a plot of control vs. treated mean values for all genes.

```
plot(meancounts)
```

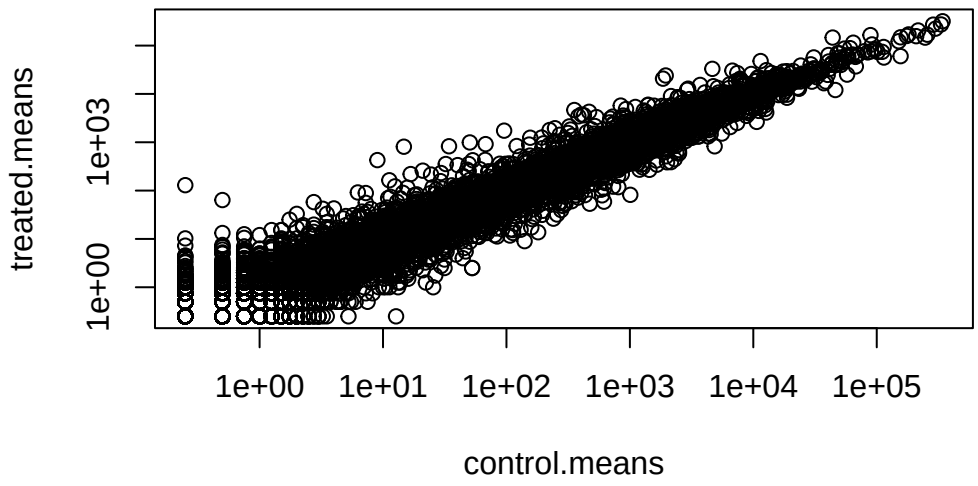


Make this a log log plot

```
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 often talk about metrics like "log2 fold-change"

```
# treated/control
log2(10/40)
```

```
[1] -2
```

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

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

A common "rule of thumb" is a log2 fold change cutoff of +2 and -2 to call gened "Up regulated" or "Down regulated" respectively.

Number of Up regulated genes:

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

```
[1] 1910
```

Number of Down regulated genes:

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

```
[1] 23046
```

For the inner nerd: These mean differences might not be as signficant as they look because of outliers in our data that skew the mean.

let's do this analysis properly and keep our inner stats nerd happy. i.e. are the difference we see between drug and no drug sigignificant given the replicate experiments.

```
library(DESeq2)
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```
Loading required package: generics
```

```

Attaching package: 'generics'
```

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

    as.difftime, as.factor, as.ordered, intersect, is.element, setdiff,
    setequal, union
```

```

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, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, saveRDS, table, tapply, unique,
    unsplit, which.max, which.min


Attaching package: 'S4Vectors'

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

    findMatches

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

    expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomicRanges

Loading required package: Seqinfo

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
```

For DESeq analysis we need three things:

- count values(`countData`)
- metadata telling us about the columns in `countData` (coldata)

- design of 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
```

The main function of DESeq that runs the analysis is called `DESeq`.

```
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)
```

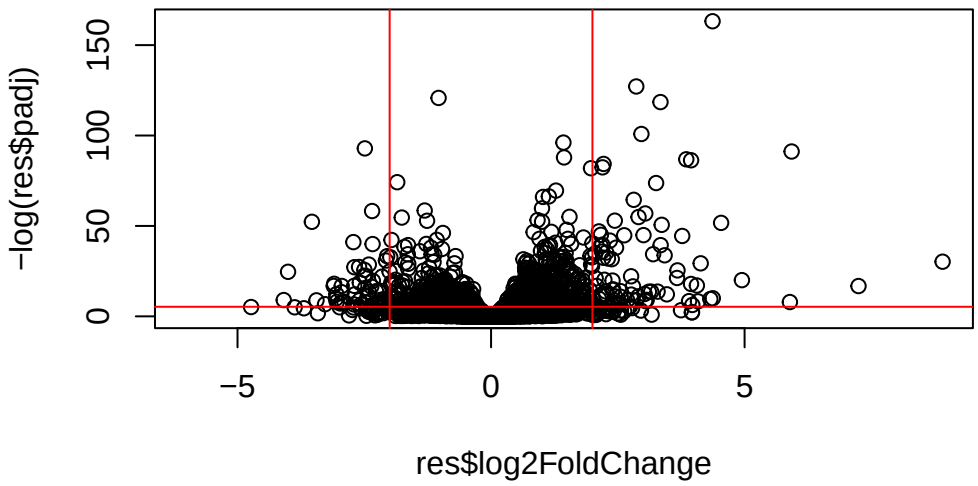
```
head(res)
```

```
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
```

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(0.005), col="red")
```



Save our results

```
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.

`BiocManager::install("AnnotationDbi")` & `BiocManager::install("org.Hs.eg.db")`- use this to instal the two packages below

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

Let's see what is in `org.Hs.eg.db` with the `columns()` function:

```
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
```

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")
```

```
'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")
```

```
'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 instal this if we don't have it already.

use `BiocManager::install("...")` to install packages below

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

To use **gage** I need two things

- a names vector of fold-change values for our DEGs (our geneset of interest)
- a set of pathways or genesets to use for annotation

```
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
```

```
data("kegg.sets.hs")

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

In our reults object we have:

```
attributes(keggres)
```

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

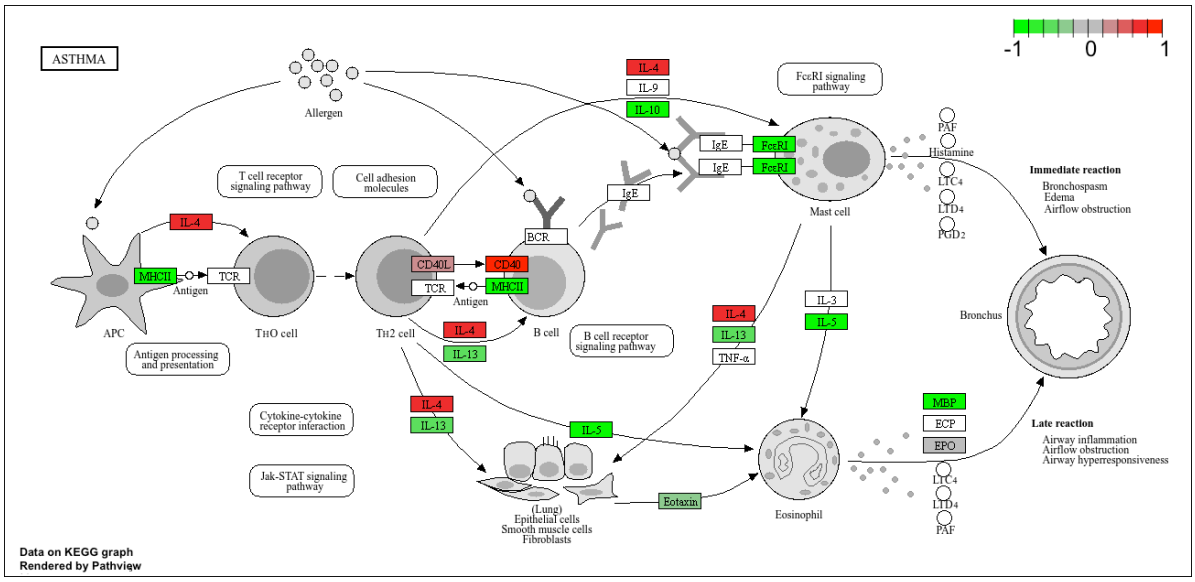
```
head(keggres$less, 5)
```

```
                                                          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
                                                              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
                                                      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
```

Let;s look at one of these pathways (hsa05310 Asthma) without genes colored up so we can see the overlap.

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

Add this pathway figure into our lab report



Save our main results

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