Class_12

Max Strul

11-4-2022

Table of contents

Answer Q1-> 10

Class 12

Setting up the packages, without having any errors or output present

```
#Testing New codeline
library("BiocManager")
```

```
Bioconductor version '3.15' is out-of-date; the current release version '3.16'
  is available with R version '4.2'; see https://bioconductor.org/install
```

```
library("DESeq2")
```

```
Loading required package: S4Vectors
```

```
Loading required package: stats4
```

```
Loading required package: BiocGenerics
```

```
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, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min


Attaching package: 'S4Vectors'


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

    expand.grid, I, unname


Loading required package: IRanges


Loading required package: GenomicRanges


Loading required package: GenomeInfoDb


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
```

#Setting up the data First you must align reads to a reference genome or transcriptome.

First we use the abundance of each transcript was don through using `kallisto` Then it was summarized to the gene level to produce length-scaled counts using R `txImport`.

- 1.) Conesa, A. et al. "A survey of best practices for RNA-seq data analysis." Genome Biology 17:13 (2016).
- 2.) Soneson, C., Love, M. I. & Robinson, M. D. "Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences." F1000Res. 4:1521 (2016).
- 3.) Griffith, Malachi, et al. "Informatics for RNA sequencing: a web resource for analysis on the cloud." PLoS Comput Biol 11.8: e1004393 (2015).

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

Question 1 & Question 2

How many genes are in this data set and ho many control cell lines do we have?

```
length(metadata$id)
```

```
[1] 8
```

```
length(metadata$id[as.logical(metadata$dex=="control")])
```

```
[1] 4
```

8 total genes and 4 control lines #Q3

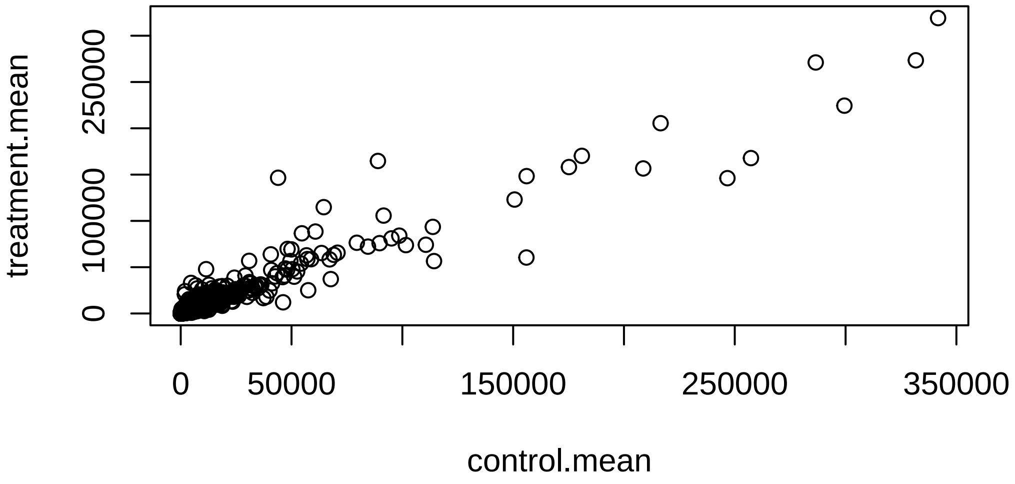
#Q4

```
control <- metadata[metadata$dex=="control",]
control.counts <- counts[,control$id]
control.mean <- rowSums(control.counts)/4

treatment <- metadata[metadata$dex=="treated",]
treatment.counts <- counts[,treatment$id]
treatment.mean <- rowSums(treatment.counts)/4
```

The columns of counts correspond to the rows of the meta data.

#Q5

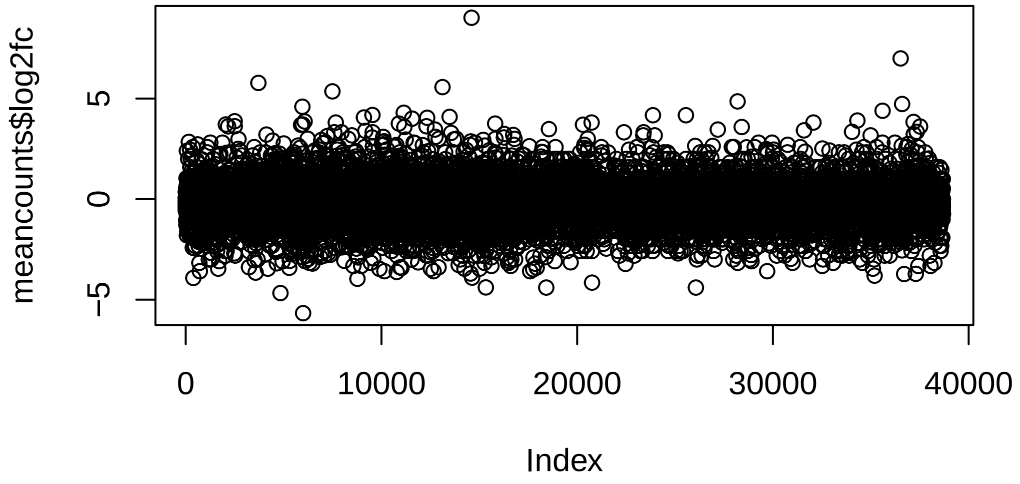
```
meancounts <- data.frame(control.mean, treatment.mean)

plot(meancounts)
```



```
library("ggplot2")
#ggplot(meancounts)+aes()+geom_point()
```

log2 fold change #Q6

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

plot(meancounts$log2fc)
```



Getting rid of zero values/ counts

#Q7

arr.ind means these will become the indices that we will extract from or use for future indices.

```
#non.zero.vals <- meancounts[,1:2]!=0
#mycounts <- meancounts[as.logical(rowSums(zero.vals!=0)),]

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

```
                control.mean treatment.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
```

Solving which genes are above a 2 fold change

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

```
[1] 314
```

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

```
[1] 485
```

#Q8 There are 314 genes that are unregulated by a fold change of 4 #Q9 There are 485 which are down regulated by a 4 fold change.

#Q10 No because we do not know the statistical significance of this though!

```r
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
```

```r
dds
```

```
class: DESeqDataSet
dim: 38694 8
metadata(1): version
assays(1): counts
rownames(38694): ENSG00000000003 ENSG00000000005 ... ENSG00000283120
  ENSG00000283123
rowData names(0):
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(4): id dex celltype geo_id
```

```r
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)
res <- as.data.frame(res)
```

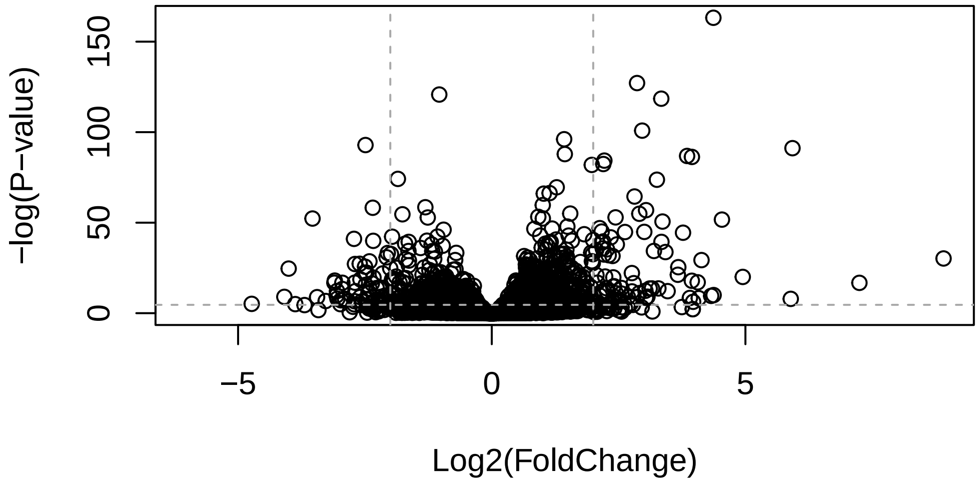
Finding the pvalue change difference between the two groups which is below 0.05

```
#res_counts <- res$baseMean[res$pvalue<= 0.05,]
```

Note: with each time you ask a question you are increasing the statistical chance that you've randomly selected something that happened by chance.

So we will be using adjusted p value

```
volcano_plot1 <- plot(res$log2FoldChange, -log(res$padj),ylab="-log(P-value)",xlab="Log2(F
abline(v=c(-2,2), col="darkgray", lty=2)
abline(h=-log(0.01), col="darkgray", lty=2)
```


−log(P−value)
0
50
100
150
−5
0
5
Log2(FoldChange)