Class13

Diana Furlan

#Import countData and colData

```r
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, aperm, 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, saveRDS, setdiff,
    table, tapply, union, 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


Attaching package: 'IRanges'

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

    windows

Loading required package: GenomicRanges

Loading required package: GenomeInfoDb

Loading required package: SummarizedExperiment

Loading required package: MatrixGenerics

Loading required package: matrixStats

Warning: package 'matrixStats' was built under R version 4.4.2


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
```

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

head(metadata)
```

```
               dex celltype     geo_id
SRR1039508 control   N61311 GSM1275862
SRR1039509 treated   N61311 GSM1275863
SRR1039512 control  N052611 GSM1275866
SRR1039513 treated  N052611 GSM1275867
SRR1039516 control  N080611 GSM1275870
SRR1039517 treated  N080611 GSM1275871
```

```r
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
```

> Q1. How many genes are in this dataset?

```r
nrow(counts)
```

```
[1] 38694
```

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

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

```
[1] 4
```

Toy differential gene expression

Calculate mean per gene count for all control samples, treated and compare

Find all control in counts

```r
control.inds <- metadata$dex == "control"
control.counts <- counts[,control.inds]
```

Find the mean across all control cols.

```r
treated.mean <- apply(counts[, metadata$dex == "treated"],2, mean)
```

Find the treated.mean

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

#{r} meancounts <- data.frame(control.mean, treated.mean)

plot #{r} plot(meancounts)

#“‘{r} library(ggplot2)

ggplot(meancounts) + aes(control.mean, treated.mean) + geom_point()

````
#```{r}
plot(meancounts[,1], meancounts[,2],log ="xy")
xlab= "log control counts", ylab "log treated"
````

log2 transformation for this type of data for easy interpretation of a fold-change and a rul of thumb

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

```
[1] 2
```

Calculate the fold change and add it to meancounts

#{r} meancounts$log2fc <- log2(meancounts$treated/meancounts$control.mean) head(meancounts)

To filter zero values

#“‘{r} to.rm <- rowSums(meancounts[, 1:2] == 0) > 0

mycounts <- meancounts[!to.rm,]

````
>How many genes left?

#```{r}
nrow(mycounts)
````

##Fold change

> How many genes are “up” regulated upon drug treatment at a threshold of +2 log2-fold-change?

1.extreact log2fc 2.find values above +2 3.count them

#{r} sum(mycounts$log2fc > 2)

> How many gnees are”down” regulated upon drup treatment at a threschold of -2 log2-fold change?

#{r} sum(mycounts$log2fc < -2) ##DESeq2 Analysis Adding Stats package DEseq to do analysis

```r
library (DESeq2)
```

Format function

```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
```

Main function in package is DESeq(), we run in dds obj

```r
dds <-DESeq(dds)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

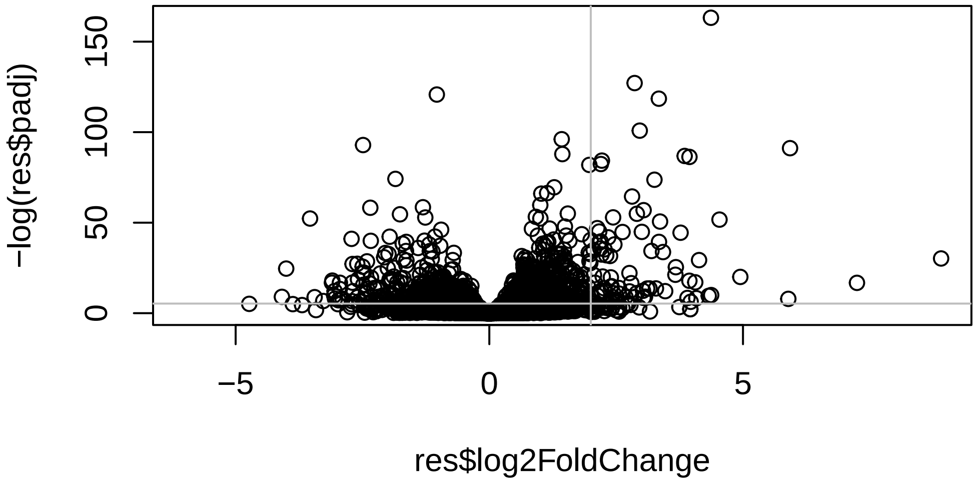
```
res<- results(dds)
head(res)
```

```
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
```

Results Fig. Volcano plot, shows fold change and stats

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

#Add line to thresholds or two with v=c(-2,2)
abline(v=2,col="gray")
abline(h=-log(0.005), col="gray")
```



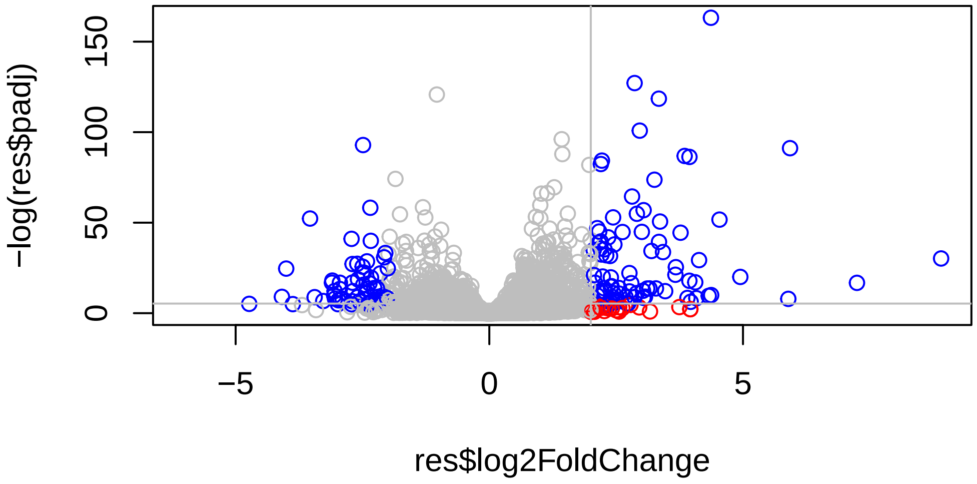
Adding color

```
mycols <- rep("grey", nrow(res))
mycols[res$log2FoldChange > 2 ]<- "red"


inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

plot(res$log2FoldChange,
-log(res$padj), col=mycols)

#Add line to thresholds or two with v=c(-2,2)
abline(v=2,col="gray")
abline(h=-log(0.005), col="gray")
```



the more neg, the smaller the pvalue

```r
log(0.0005)
```

```
[1] -7.600902
```

Save myresults to date out to disc

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