

class12_report

```
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

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

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,
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
```

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

Q1. How many genes in the dataset?

```
nrow(counts)
```

```
[1] 38694
```

38694 genes

Q2. How many control cell lines?

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

```
control treated
        4       4
```

There are 4 control cell lines.

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

ENSG00000000003	ENSG00000000005	ENSG000000000419	ENSG000000000457	ENSG000000000460
900.75	0.00	520.50	339.75	97.25
ENSG000000000938				
	0.75			

Same thing with Dplyr:

```
library(dplyr)
```

Attaching package: 'dplyr'

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

combine

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

count

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

intersect, setdiff, union

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

intersect

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

collapse, desc, intersect, setdiff, slice, union

The following objects are masked from 'package:S4Vectors':

first, intersect, rename, setdiff, setequal, union

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

```
combine, intersect, setdiff, union
```

```
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 <- rowSums(control.counts)/4
head(control.mean)
```

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

Q3. How would you make the above code in either approach more robust?

```
control.mean <- rowSums(control.counts)/sum(metadata$dex == "control")
head(control.mean)
```

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

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)

```
treated <- metadata %>% filter(dex=="treated")
treated.counts <- counts %>% select(treated$id)
treated.mean <- rowSums(treated.counts)/sum(metadata$dex == "treated")
head(treated.mean)
```

```

ENSG000000000003 ENSG000000000005 ENSG000000000419 ENSG000000000457 ENSG000000000460
       658.00          0.00        546.00        316.50        78.75
ENSG000000000938
       0.00

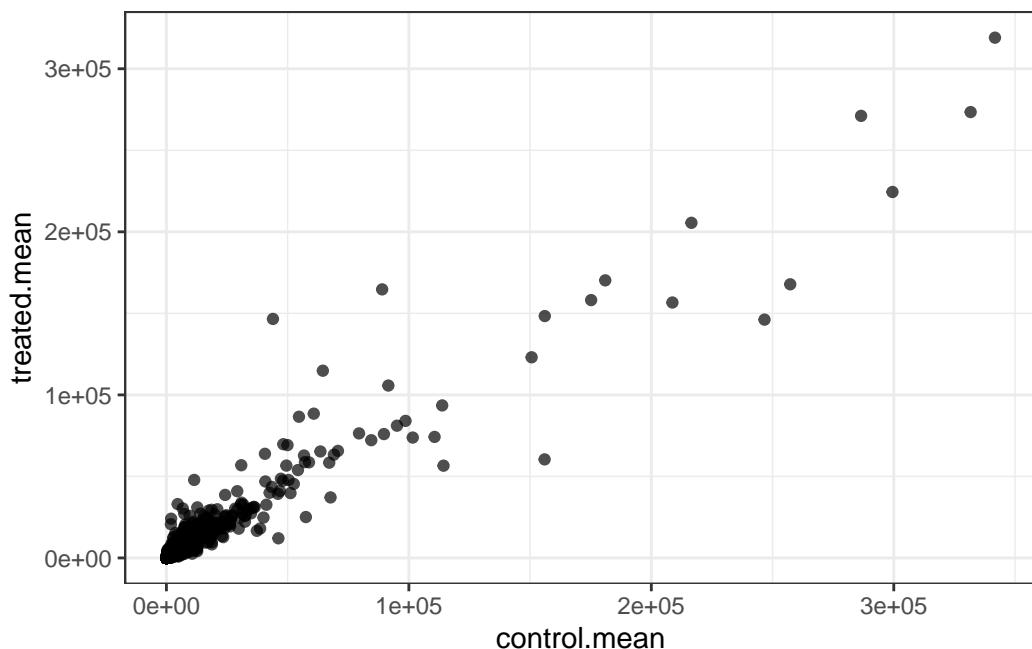
```

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

```

library(ggplot2)
ggplot(data = meancounts) +
  aes(x = control.mean, y = treated.mean) +
  geom_point(alpha = 0.7) +
  theme_bw()

```



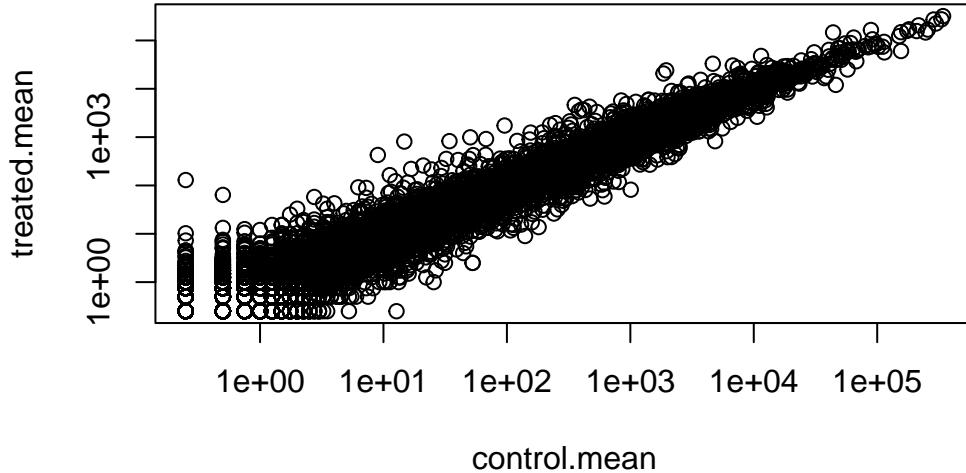
Q6. Plotting In base R with log scale:

```
plot(control.mean, treated.mean, 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



Getting log2FC

```
meancounts$log2fc <- log2(meancounts[, "treated.mean"] / meancounts[, "control.mean"])
head(meancounts)
```

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000005	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

Fix the NaN's, etc.

```
zero.vals <- which(meancounts[, 1:2]==0, arr.ind=TRUE)

to.rm <- unique(zero.vals[, 1])
mycounts <- meancounts[-to.rm,]
```

```
head(mycounts)
```

	control.mean	treated.mean	log2fc
ENSG000000000003	900.75	658.00	-0.45303916
ENSG000000000419	520.50	546.00	0.06900279
ENSG000000000457	339.75	316.50	-0.10226805
ENSG000000000460	97.25	78.75	-0.30441833
ENSG000000000971	5219.00	6687.50	0.35769358
ENSG00000001036	2327.00	1785.75	-0.38194109

Q7. What is the purpose of the arr.ind argument in the which() function call above? Why would we then take the first column of the output and need to call the unique() function?

```
up.ind <- mycounts$log2fc > 2  
down.ind <- mycounts$log2fc < (-2)
```

Q8. How many genes are upregulated Log2FC >2?

```
sum(up.ind)
```

[1] 250

There are 250 genes with log2FC >2.

Q9. How many genes are downregulated log2FC <2?

```
sum(down.ind)
```

[1] 367

Q10. Do you trust these results?

If we are asking whether these genes have a >2 log fold change? Yes, but we are more interested in whether the changes are statistically significant, so we need a statistical test.

Deseq2

```

library(DESeq2)
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

class: DESeqDataSet
dim: 38694 8
metadata(1): version
assays(1): counts
rownames(38694): ENSG000000000003 ENSG000000000005 ... ENSG00000283120
ENSG00000283123
rowData names(0):
colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
colData names(4): id dex celltype geo_id

#results(dds) this doesnt work
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

log2 fold change (MLE): dex treated vs control
Wald test p-value: dex treated vs control
DataFrame with 38694 rows and 6 columns
  baseMean log2FoldChange    lfcSE     stat   pvalue
  <numeric>      <numeric> <numeric> <numeric> <numeric>
ENSG000000000003  747.1942    -0.3507030  0.168246 -2.084470 0.0371175
ENSG000000000005   0.0000       NA        NA        NA        NA
ENSG000000000419  520.1342    0.2061078  0.101059  2.039475 0.0414026
ENSG000000000457  322.6648    0.0245269  0.145145  0.168982 0.8658106
ENSG000000000460   87.6826    -0.1471420  0.257007 -0.572521 0.5669691
...
...
ENSG00000283115   0.000000       NA        NA        NA        NA
ENSG00000283116   0.000000       NA        NA        NA        NA
ENSG00000283119   0.000000       NA        NA        NA        NA
ENSG00000283120   0.974916    -0.668258   1.69456 -0.394354 0.693319
ENSG00000283123   0.000000       NA        NA        NA        NA
  padj
  <numeric>
ENSG000000000003  0.163035
ENSG000000000005   NA
ENSG000000000419  0.176032
ENSG000000000457  0.961694
ENSG000000000460  0.815849
...
...
ENSG00000283115   NA
ENSG00000283116   NA
ENSG00000283119   NA
ENSG00000283120   NA
ENSG00000283123   NA

```

Convert to dataframe:

```

res <- as.data.frame(res)

summary(res)

baseMean      log2FoldChange      lfcSE      stat

```

```

Min.    : 0.0   Min.   :-6.030   Min.   :0.057   Min.   :-15.894
1st Qu.: 0.0   1st Qu.:-0.425   1st Qu.:0.174   1st Qu.:-0.643
Median  : 1.1   Median :-0.009   Median :0.445   Median : -0.027
Mean    : 570.2  Mean   :-0.011   Mean   :1.136   Mean   :  0.045
3rd Qu.: 201.8  3rd Qu.: 0.306   3rd Qu.:1.848   3rd Qu.:  0.593
Max.    :329280.4 Max.   : 8.906   Max.   :3.534   Max.   : 18.422
                           NA's   :13436   NA's   :13436   NA's   :13436

      pvalue          padj
Min.   :0.000   Min.   :0.000
1st Qu.:0.168   1st Qu.:0.203
Median :0.533   Median :0.606
Mean   :0.495   Mean   :0.539
3rd Qu.:0.800   3rd Qu.:0.866
Max.   :1.000   Max.   :1.000
NA's   :13578   NA's   :23549

```

```

res05 <- results(dds, alpha=0.05)
summary(res05)

```

```

out of 25258 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)      : 1236, 4.9%
LFC < 0 (down)     : 933, 3.7%
outliers [1]       : 142, 0.56%
low counts [2]     : 9033, 36%
(mean count < 6)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

```

Annotating

```

# BiocManager::install("AnnotationDbi")
# BiocManager::install("org.Hs.eg.db")
library(org.Hs.eg.db)

```

```
Loading required package: AnnotationDbi
```

```
Warning: package 'AnnotationDbi' was built under R version 4.1.1
```

```
Attaching package: 'AnnotationDbi'
```

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

```
select
```

```
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"
```

```
res$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="SYMBOL")
```

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

```
head(res)
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue
ENSG000000000003	747.1941954	-0.35070302	0.1682457	-2.0844697	0.03711747
ENSG000000000005	0.0000000		NA	NA	NA
ENSG00000000419	520.1341601	0.20610777	0.1010592	2.0394752	0.04140263
ENSG00000000457	322.6648439	0.02452695	0.1451451	0.1689823	0.86581056
ENSG00000000460	87.6826252	-0.14714205	0.2570073	-0.5725210	0.56696907
ENSG00000000938	0.3191666	-1.73228897	3.4936010	-0.4958463	0.62000288
	padj	symbol			
ENSG000000000003	0.1630348	TSPAN6			
ENSG000000000005	NA	TNMD			
ENSG00000000419	0.1760317	DPM1			
ENSG00000000457	0.9616942	SCYL3			
ENSG00000000460	0.8158486	C1orf112			
ENSG00000000938	NA	FGR			

```
Add Entrez, Uniprot ID and gene name:
```

```

res$entrez <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="ENTREZID")

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

res$uniprot <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="UNIPROTID")

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

res$genename <- mapIds(org.Hs.eg.db, keys=row.names(res), keytype="ENSEMBL", column="GENENAME")

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

ord <- order(res$padj)
head(res[ord,])

      baseMean log2FoldChange      lfcSE      stat     pvalue
ENSG00000152583    954.7709       4.368359  0.23712679   18.42204 8.744898e-76
ENSG00000179094    743.2527       2.863889  0.17556931   16.31201 8.107836e-60
ENSG00000116584   2277.9135      -1.034701  0.06509844  -15.89440 6.928546e-57
ENSG00000189221    2383.7537       3.341544  0.21240579   15.73189 9.144326e-56
ENSG00000120129    3440.7038       2.965211  0.20369513   14.55710 5.264243e-48
ENSG00000148175    13493.9204      1.427168  0.10038904   14.21638 7.251278e-46

      padj      symbol      entrez      uniprot
ENSG00000152583 1.324415e-71 SPARCL1      8404 A0A024RDE1
ENSG00000179094 6.139658e-56 PER1        5187 015534
ENSG00000116584 3.497761e-53 ARHGEF2      9181 Q92974
ENSG00000189221 3.462270e-52 MAOA        4128 P21397
ENSG00000120129 1.594539e-44 DUSP1        1843 B4DU40
ENSG00000148175 1.830344e-42 STOM        2040 F8VSL7

      genename
ENSG00000152583                      SPARC like 1
ENSG00000179094                      period circadian regulator 1
ENSG00000116584 Rho/Rac guanine nucleotide exchange factor 2
ENSG00000189221                         monoamine oxidase A
ENSG00000120129                         dual specificity phosphatase 1
ENSG00000148175                         stomatin

```

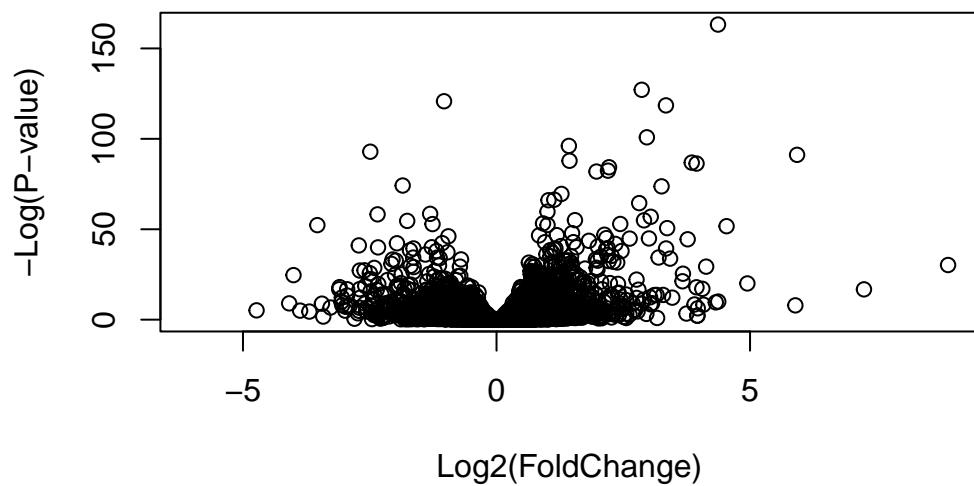
Save results to csv file:

```
write.csv(res[ord,],"deseq_results.csv")
```

Data Vis

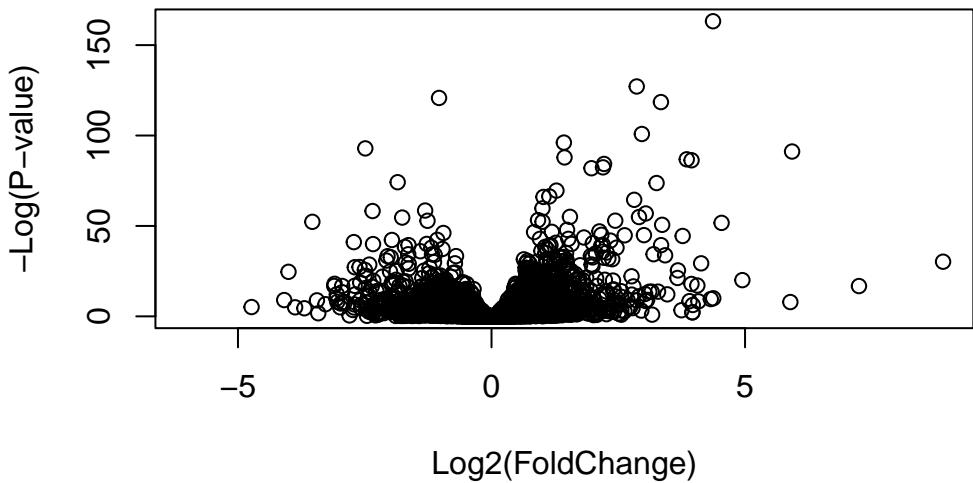
Volcano Plot

```
plot(res$log2FoldChange, -log(res$padj), xlab="Log2(FoldChange)", ylab="-Log(P-value)")
```

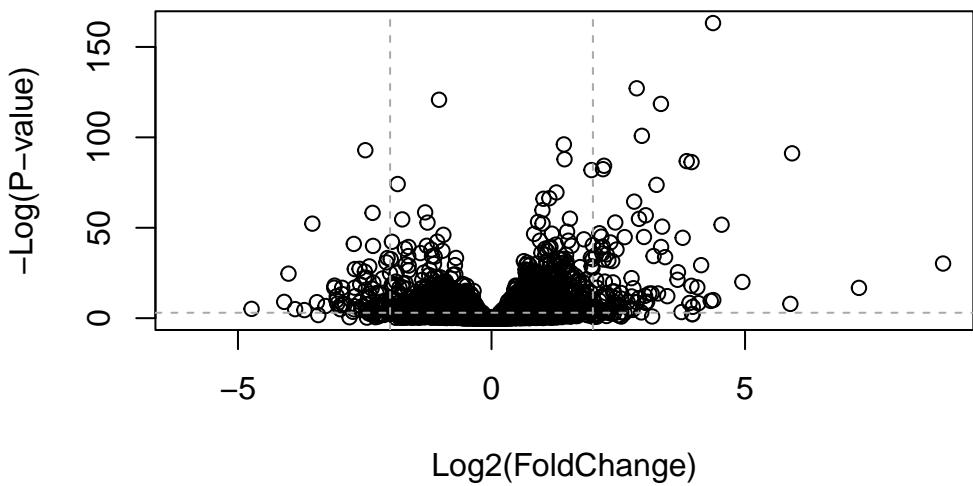


Labels in Base R

```
plot(res$log2FoldChange, -log(res$padj),
ylab="-Log(P-value)", xlab="Log2(FoldChange)")
```



```
plot(res$log2FoldChange, -log(res$padj), ylab="-Log(P-value)", xlab="Log2(FoldChange)") +  
abline(h=-log(0.05), col="darkgray", lty=2)
```



```

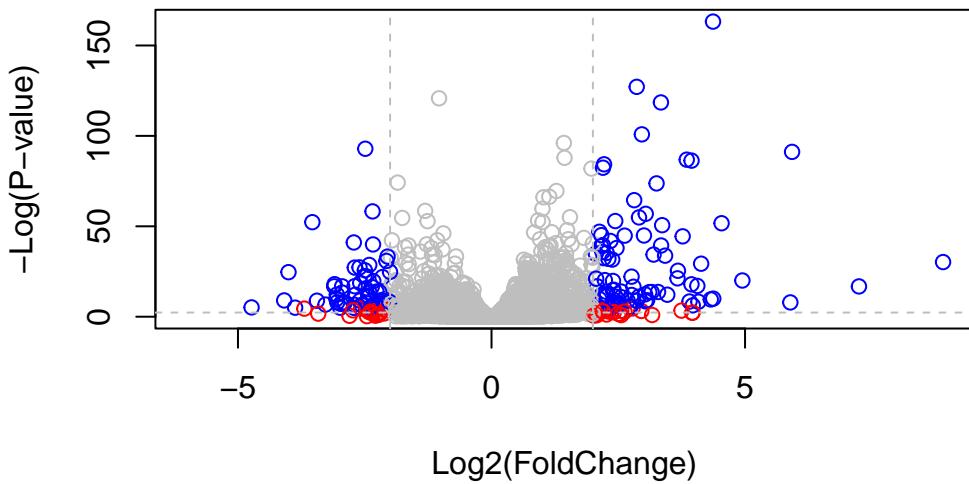
integer(0)

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

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

plot( res$log2FoldChange, -log(res$padj), col=mycols, ylab="-Log(P-value)", xlab="Log2(FoldChange)"

```



```
integer(0)
```

Using EnhanceVolcano

```
library(EnhancedVolcano)
```

```
Warning: package 'EnhancedVolcano' was built under R version 4.1.1
```

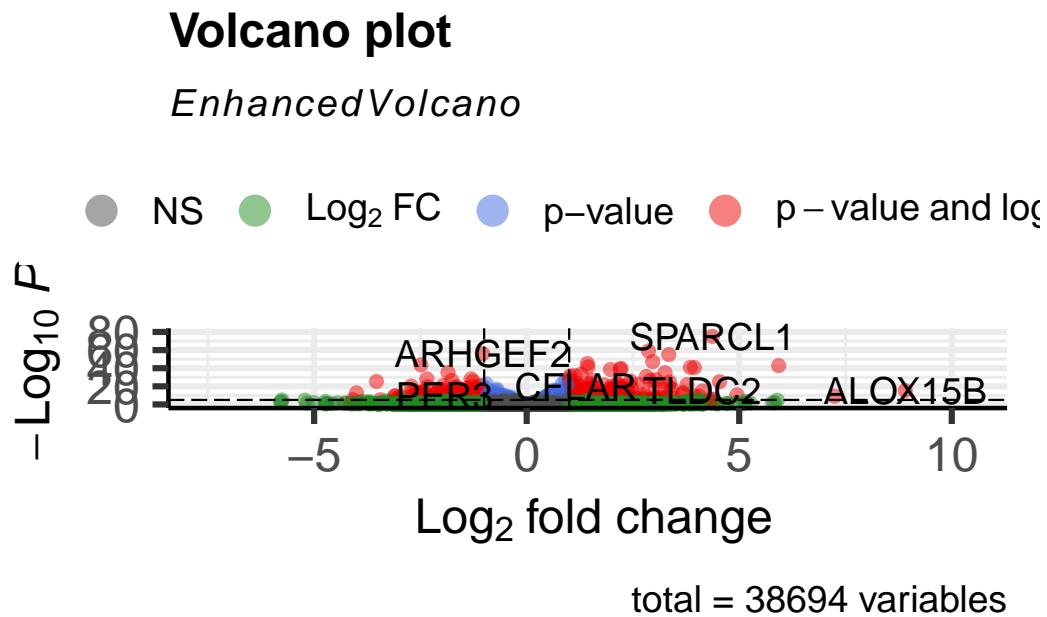
```
Loading required package: ggrepel
```

```
Warning: package 'ggrepel' was built under R version 4.1.1
```

```
Registered S3 methods overwritten by 'ggalt':
```

```
  method           from
grid.draw.absoluteGrob  ggplot2
grobHeight.absoluteGrob ggplot2
grobWidth.absoluteGrob  ggplot2
grobX.absoluteGrob     ggplot2
grobY.absoluteGrob     ggplot2
```

```
x <- as.data.frame(res)
EnhancedVolcano(x, lab = x$symbol, x = 'log2FoldChange', y = 'pvalue')
```



Pathway Analysis

```
library(pathview)
```

```
Warning: package 'pathview' was built under R version 4.1.1
```

```
#####
# 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)
```

```
Warning: package 'gage' was built under R version 4.1.1
```

```
library(gageData)
```

```
data(kegg.sets.hs)
head(kegg.sets.hs, 2)
```

```
$`hsa00232 Caffeine metabolism`
[1] "10"    "1544"   "1548"   "1549"   "1553"   "7498"   "9"
$`hsa00983 Drug metabolism - other enzymes`
[1] "10"    "1066"   "10720"  "10941"  "151531" "1548"   "1549"   "1551"
[9] "1553"  "1576"   "1577"   "1806"   "1807"   "1890"   "221223" "2990"
[17] "3251"  "3614"   "3615"   "3704"   "51733"  "54490"  "54575"  "54576"
[25] "54577" "54578"  "54579"  "54600"  "54657"  "54658"  "54659"  "54963"
[33] "574537" "64816"  "7083"   "7084"   "7172"   "7363"   "7364"   "7365"
[41] "7366"  "7367"   "7371"   "7372"   "7378"   "7498"   "79799" "83549"
[49] "8824"  "8833"   "9"      "978"
```

```
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

Gage pathway analysis

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

attributes(keggres)

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

head(keggres$less, 3)

          p.geomean stat.mean      p.val
hsa05332 Graft-versus-host disease 0.0004250461 -3.473346 0.0004250461
hsa04940 Type I diabetes mellitus 0.0017820293 -3.002352 0.0017820293
hsa05310 Asthma                 0.0020045888 -3.009050 0.0020045888

          q.val set.size      exp1
hsa05332 Graft-versus-host disease 0.09053483      40 0.0004250461
hsa04940 Type I diabetes mellitus 0.14232581      42 0.0017820293
hsa05310 Asthma                 0.14232581      29 0.0020045888
```

Pathview for Asthma:

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

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

Info: Working in directory C:/Users/lhodg/Documents/Research/BGGN213/class12

Info: Writing image file hsa05310.pathview.png

pathview(gene.data=foldchanges, pathway.id="hsa05310", kegg.native=FALSE)

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

Info: Working in directory C:/Users/lhodg/Documents/Research/BGGN213/class12
```

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

Q12.

Top downregulated pathway:

```
pathview(gene.data=foldchanges, pathway.id="hsa05332", kegg.native=FALSE)

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

Warning in .subtypeDisplay(object): Given subtype 'missing interaction' is not found!

Info: Working in directory C:/Users/lhodg/Documents/Research/BGGN213/class12

Info: Writing image file hsa05332.pathview.pdf
```

Second most down-regulated pathway

```
pathview(gene.data=foldchanges, pathway.id="hsa04940", kegg.native=FALSE)

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

Info: Working in directory C:/Users/lhodg/Documents/Research/BGGN213/class12

Info: Writing image file hsa04940.pathview.pdf
```

Optional

```
i <- grep("CRISPLD2", res$symbol)
res[i,]

  baseMean log2FoldChange      lfcSE      stat      pvalue
ENSG00000103196 3096.159      2.626034 0.2674445 9.818988 9.327474e-23
                  padj    symbol entrez     uniprot
ENSG00000103196 3.363443e-20 CRISPLD2  83716 A0A140VK80
                                         genename
ENSG00000103196 cysteine rich secretory protein LCCL domain containing 2
```

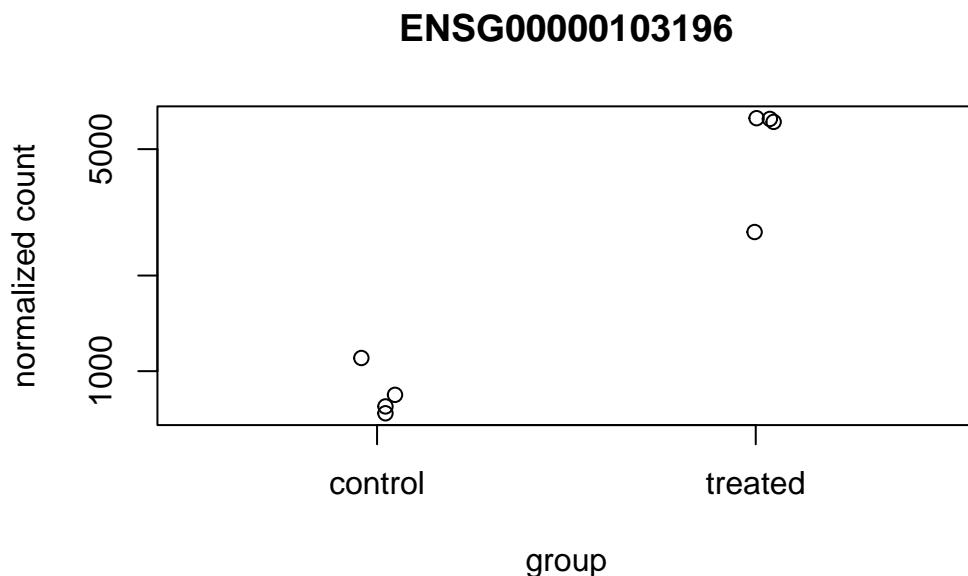
```

rownames(res[i,])

[1] "ENSG00000103196"

plotCounts(dds, gene="ENSG00000103196", intgroup="dex")

```



```

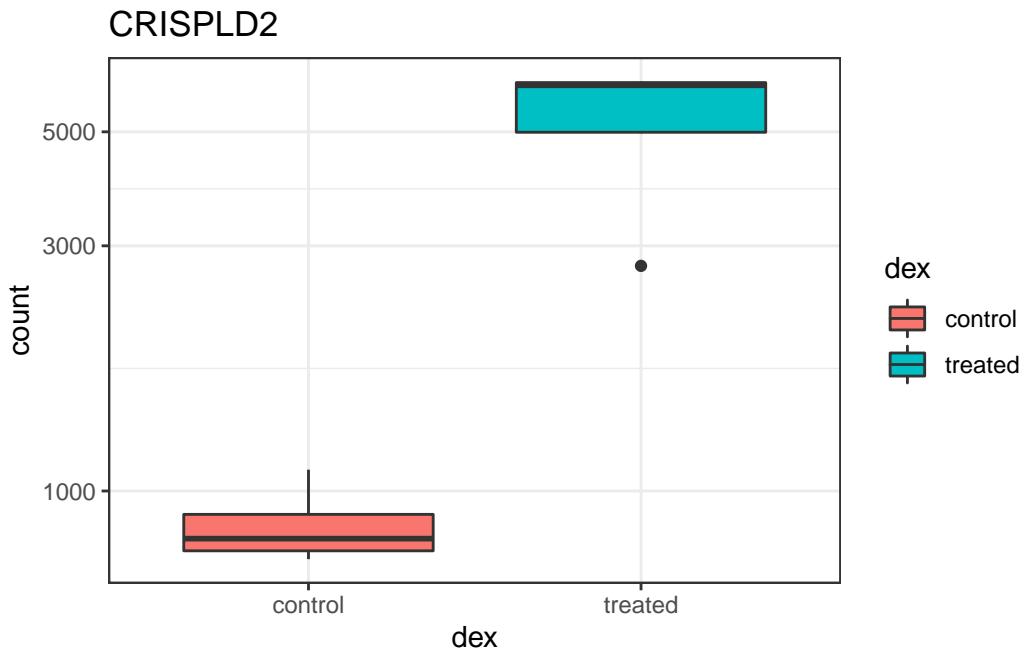
# Return the data
d <- plotCounts(dds, gene="ENSG00000103196", intgroup="dex", returnData=TRUE)
head(d)

```

	count	dex
SRR1039508	774.5002	control
SRR1039509	6258.7915	treated
SRR1039512	1100.2741	control
SRR1039513	6093.0324	treated
SRR1039516	736.9483	control
SRR1039517	2742.1908	treated

GGplot

```
ggplot(d, aes(dex, count, fill=dex)) + geom_boxplot() + scale_y_log10() + ggtitle("CRISPLD2")
```



```
sessionInfo()
```

```
R version 4.1.0 (2021-05-18)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252
```

```
attached base packages:
```

```
[1] stats4      stats       graphics   grDevices  utils      datasets   methods
[8] base
```

other attached packages:

```
[1] gageData_2.32.0           gage_2.44.0
[3] pathview_1.34.0          EnhancedVolcano_1.12.0
[5] ggrepel_0.9.1            org.Hs.eg.db_3.14.0
[7] AnnotationDbi_1.56.2     ggplot2_3.3.5
[9] dplyr_1.0.7              DESeq2_1.34.0
[11] SummarizedExperiment_1.24.0 Biobase_2.54.0
[13] MatrixGenerics_1.6.0      matrixStats_0.61.0
[15] GenomicRanges_1.46.1      GenomeInfoDb_1.30.0
[17] IRanges_2.28.0           S4Vectors_0.32.3
[19] BiocGenerics_0.40.0
```

loaded via a namespace (and not attached):

```
[1] bitops_1.0-7             bit64_4.0.5          ash_1.0-15
[4] RColorBrewer_1.1-2       httr_1.4.2            Rgraphviz_2.38.0
[7] tools_4.1.0              utf8_1.2.2            R6_2.5.1
[10] vipor_0.4.5             KernSmooth_2.23-20   DBI_1.1.2
[13] colorspace_2.0-2         withr_2.5.0           ggrastr_1.0.1
[16] tidyselect_1.1.1          ggalt_0.4.0           bit_4.0.4
[19] compiler_4.1.0           extrafontdb_1.0       graph_1.72.0
[22] cli_3.6.0                DelayedArray_0.20.0    labeling_0.4.2
[25] KEGGgraph_1.54.0          scales_1.1.1           proj4_1.0-10.1
[28] genefilter_1.76.0          stringr_1.4.0          digest_0.6.27
[31] rmarkdown_2.11             XVector_0.34.0        pkgconfig_2.0.3
[34] htmltools_0.5.4           extrafont_0.17         fastmap_1.1.0
[37] maps_3.4.0                rlang_1.0.6            rstudioapi_0.13
[40] RSQLite_2.2.9              generics_0.1.1         farver_2.1.0
[43] jsonlite_1.7.3            BiocParallel_1.28.3   RCurl_1.98-1.5
[46] magrittr_2.0.1             GO.db_3.14.0          GenomeInfoDbData_1.2.7
[49] Matrix_1.3-3              ggbeeswarm_0.6.0       Rcpp_1.0.7
[52] munsell_0.5.0              fansi_0.5.0            lifecycle_1.0.1
[55] stringi_1.7.6             yaml_2.2.2             MASS_7.3-54
[58] zlibbioc_1.40.0            grid_4.1.0             blob_1.2.2
[61] parallel_4.1.0             crayon_1.4.2           lattice_0.20-44
[64] Biostrings_2.62.0          splines_4.1.0          annotate_1.72.0
[67] KEGGREST_1.34.0            locfit_1.5-9.4         knitr_1.37
[70] pillar_1.6.5               geneplotter_1.72.0     XML_3.99-0.8
[73] glue_1.4.2                 evaluate_0.14          png_0.1-7
[76] vctrs_0.3.8                Rttf2pt1_1.3.9         gtable_0.3.0
[79] purrrr_0.3.4               assertthat_0.2.1       cachem_1.0.6
[82] xfun_0.29                  xtable_1.8-4           survival_3.2-11
[85] tibble_3.1.3                beeswarm_0.4.0         memoise_2.0.1
[88] ellipsis_0.3.2
```