NGS Final

2024-05-11

```
library(tximport)
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, setdiff, sort,
##     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
```

```
## 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
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::%within%() masks IRanges::%within%()
## x dplyr::collapse()     masks IRanges::collapse()
## x dplyr::combine()      masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::count()        masks matrixStats::count()
## x dplyr::desc()         masks IRanges::desc()
## x tidyr::expand()       masks S4Vectors::expand()
## x dplyr::filter()       masks stats::filter()
## x dplyr::first()        masks S4Vectors::first()
## x dplyr::lag()          masks stats::lag()
## x ggplot2::Position()   masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()       masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()       masks S4Vectors::rename()
## x lubridate::second()   masks S4Vectors::second()
```

```
## x lubridate::second<-() masks S4Vectors::second<-()
## x dplyr::slice()        masks IRanges::slice()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
#Define samples and filepaths
netid <- 'jh8755'
sample_names <- c('control1','control2','control3','treated1','treated2','treated3')
sample_condition <- c(rep('control',3),rep('treated',3))
files <- file.path("/scratch",netid,"ngs.final","res","salmon",sample_names,'quant.sf')
names(files) <- sample_names

#Import data
tx2gene <- read.table(file.path("/scratch",netid,"ngs.final","res","salmon","tx2gene.tsv"),header=F,sep
#Generate gene-level counts
txi <- tximport(files, type="salmon", tx2gene=tx2gene)
```

```
## reading in files with read_tsv
## 1 2 3 4 5 6
## summarizing abundance
## summarizing counts
## summarizing length
```

Define the metadata

```
metadata.df <- data.frame(sample = factor(sample_names),
                          condition = factor(sample_condition,levels = c('control','treated')),
                          replicate = factor(c('replicate1','replicate2','replicate3','replicate1','repl

row.names(metadata.df) <- sample_names
metadata.df
```

```
##            sample condition  replicate
## control1 control1   control replicate1
## control2 control2   control replicate2
## control3 control3   control replicate3
## treated1 treated1   treated replicate1
## treated2 treated2   treated replicate2
## treated3 treated3   treated replicate3
```

Create a DESeqDataSet object with gene-level counts

```
dds <- DESeqDataSetFromTximport(txi,
                                colData = metadata.df,
                                design = ~ condition)
```

```
## using counts and average transcript lengths from tximport
```

```
#Inspect the counts before filtering
counts(dds) %>%
  dim()
```

```
## [1] 62812     6
```

```
#Prefilter very low reads. remove any genes that have less than 8 counts across all samples
keep <- rowSums(counts(dds)) >= 8
dds <- dds[keep,]
counts(dds) %>%
  dim()
```

```
## [1] 23669     6
```

```
dds = DESeq(dds)
```

```
## estimating size factors
## using 'avgTxLength' from assays(dds), correcting for library size
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

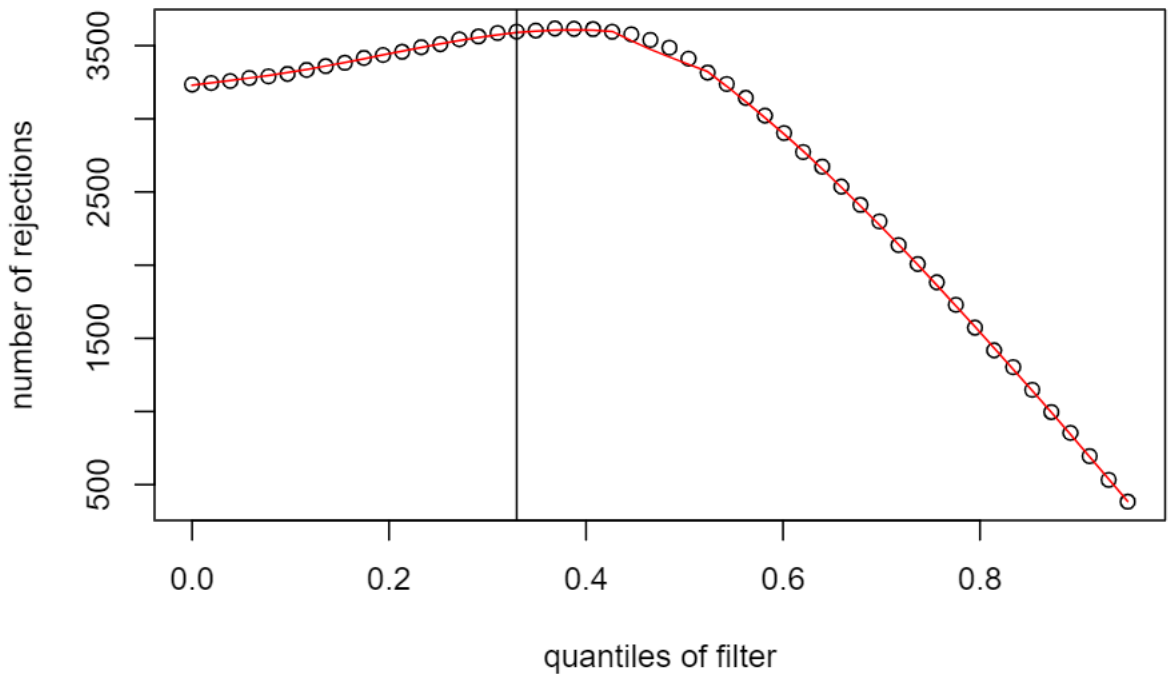
```
# Extract results table with unshrunken LFCs with alpha set to 0.05
res <- results(dds, alpha = 0.05)

# Create a results object with alpha = 0.05 and shrunken LFC estimates.
res.lfcShrink <- lfcShrink(dds,
                           res = res,
                           coef = 'condition_treated_vs_control',type = 'apeglm')
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
## Warning in nbinomGLM(x = x, Y = YNZ, size = size, weights = weightsNZ, offset =
## offsetNZ, : the line search routine failed, unable to sufficiently decrease the
## function value
```

```
#Make tibble of shrunken LFC results
library(tidyverse)
res.lfcShrink %>%
  as_tibble() %>%
  summarise(padj_NA = sum(is.na(padj)),
            padj_notNA = sum(!is.na(padj)))
```

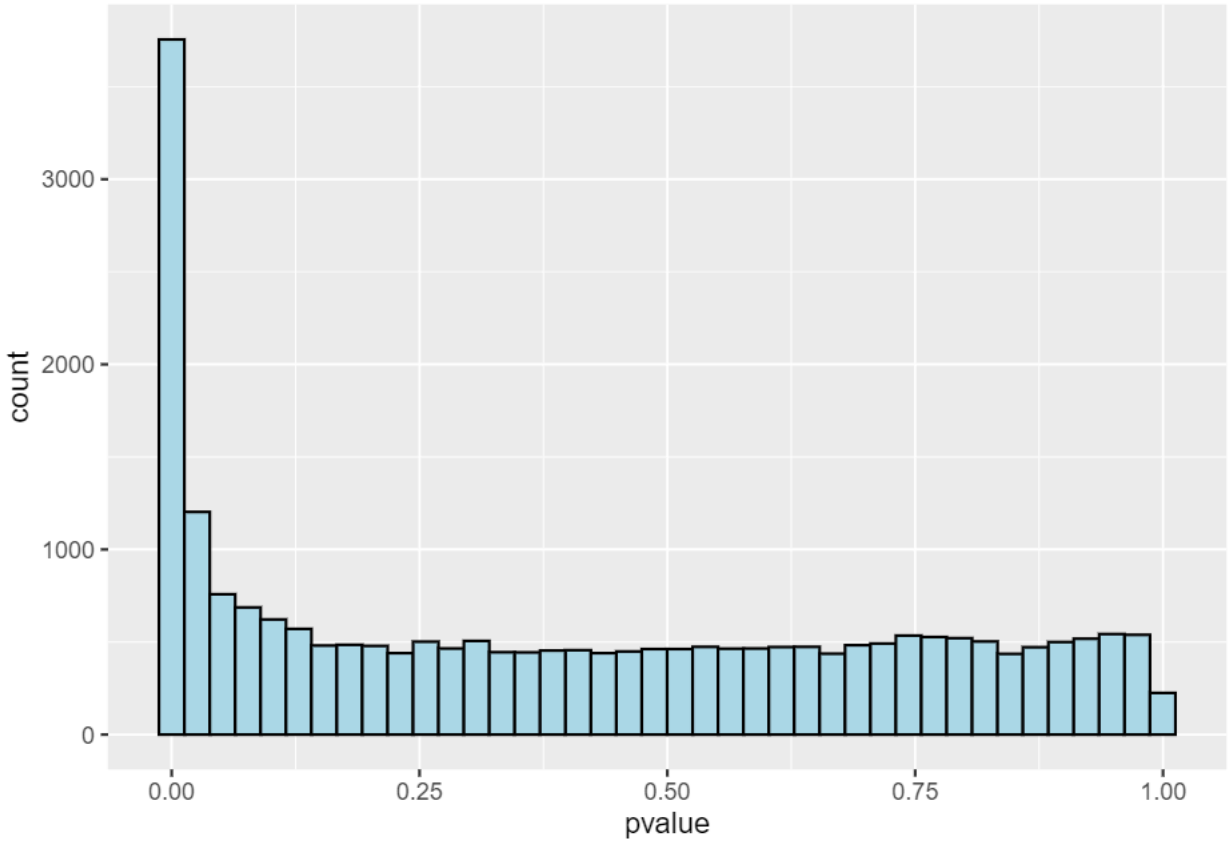
```
## # A tibble: 1 x 2
##   padj_NA padj_notNA
##     <int>      <int>
## 1    7819      15850
```

```
#Plot the number of positive predictions
plot(metadata(res.lfcShrink)$filterNumRej,
     type="b", ylab="number of rejections",
     xlab="quantiles of filter")
lines(metadata(res)$lo.fit, col="red")
abline(v=metadata(res)$filterTheta)
```




```
#Make histogram of adjusted pvalues
res.lfcShrink %>%
  as_tibble() %>% # coerce to tibble
  ggplot(aes(pvalue)) +
  geom_histogram(fill="light blue",color='black',bins = 40)
```

```
## Warning: Removed 22 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Sort the genes based on the adjusted p-value.

```
#Tidy the data
res.lfcShrink.tbl_df <- res.lfcShrink %>%
  as.data.frame() %>%
  rownames_to_column(var = "feature_id") %>%
  as_tibble()

#Sort tibble by adjusted p value
res.lfcShrink.tbl_df %>%
  arrange(padj)
```

```
## # A tibble: 23,669 x 6
##    feature_id      baseMean log2FoldChange  lfcSE    pvalue      padj
##    <chr>              <dbl>          <dbl>  <dbl>     <dbl>     <dbl>
##  1 ENSG00000175334    6423.          1.66  0.0659 1.03e-140 1.63e-136
##  2 ENSG00000163041    7972.          1.66  0.0719 1.12e-119 8.91e-116
##  3 ENSG00000196396    6618.          1.15  0.0531 6.08e-105 3.21e-101
##  4 ENSG00000105976    9566.          1.57  0.0760 6.70e- 96 2.65e- 92
##  5 ENSG00000128595   22967.          1.48  0.0721 5.51e- 95 1.75e- 91
##  6 ENSG00000101384   11784.          1.31  0.0664 3.72e- 88 8.41e- 85
##  7 ENSG00000124333    2742.          1.48  0.0750 3.66e- 88 8.41e- 85
##  8 ENSG00000117632   16768.          1.34  0.0706 1.46e- 81 2.89e- 78
##  9 ENSG00000180398   20588.         0.908  0.0519 1.06e- 69 1.86e- 66
## 10 ENSG00000213281    6962.          1.18  0.0678 4.42e- 69 7.00e- 66
## # i 23,659 more rows
```

```r
#Get number of genes with FDR < 0.05

res.lfcShrink.tbl_df %>%
  summarise(`FDR < 0.05` = sum(padj < 0.05,na.rm = T))
```

```
## # A tibble: 1 x 1
##   `FDR < 0.05`
##          <int>
## 1         3596
```

Look at number of statistically signficant LFCs

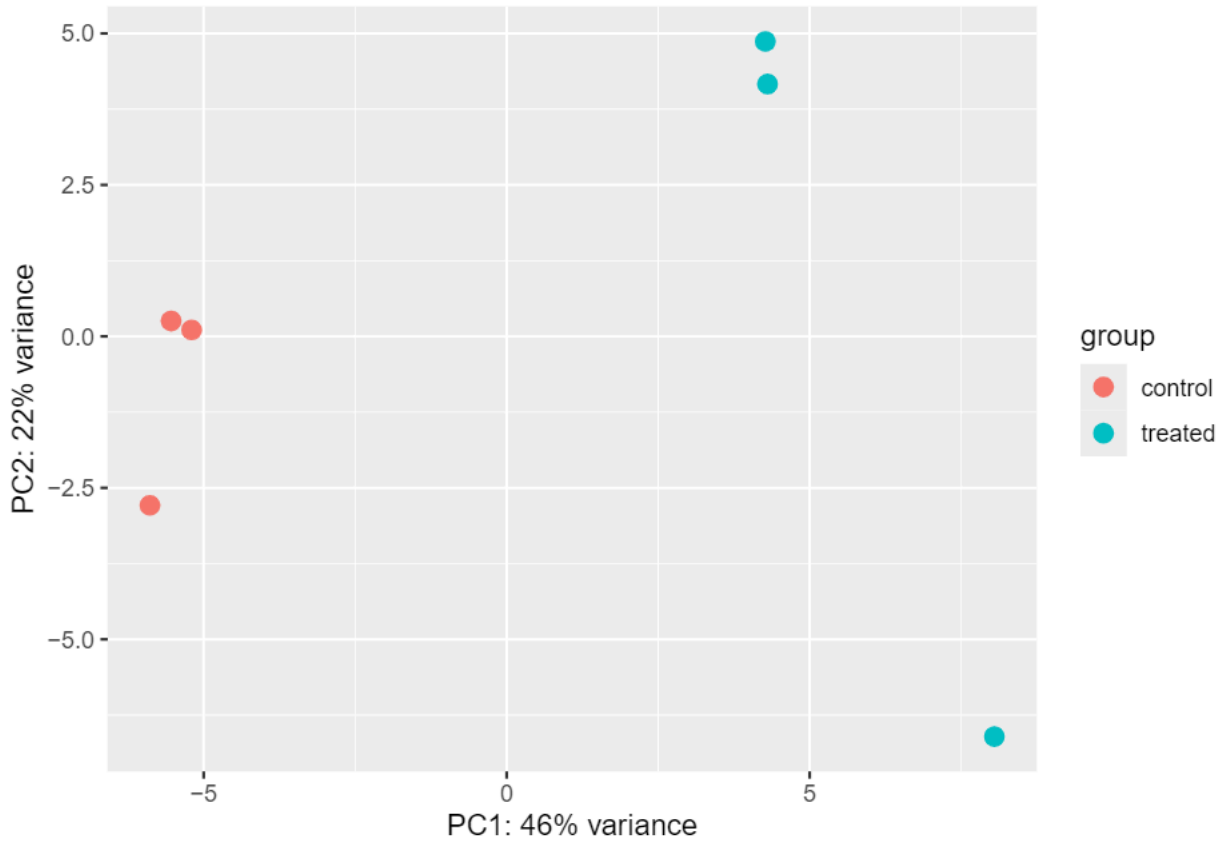
```r
res.lfcShrink.tbl_df %>%
  mutate(`LFC < 0` = case_when(log2FoldChange < 0 & padj < 0.05 ~ 1,
                               TRUE ~ 0)) %>%
  mutate(`LFC > 0` = case_when(log2FoldChange > 0 & padj < 0.05 ~1,
                               TRUE ~ 0)) %>%
  summarise(`LFC < 0 count`= sum(`LFC < 0`),
            `LFC > 0 count` = sum( `LFC > 0` ))
```

```
## # A tibble: 1 x 2
##   `LFC < 0 count` `LFC > 0 count`
##             <dbl>           <dbl>
## 1            1666            1930
```

PCA plot

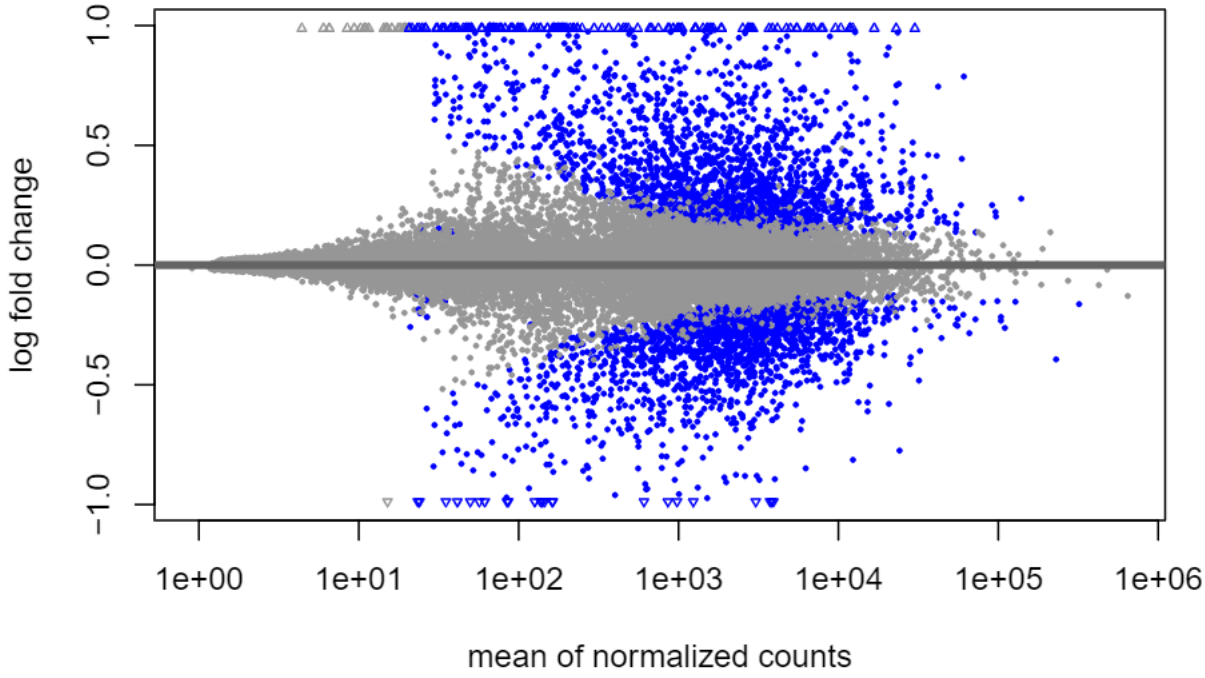
```r
#Stabilize variance
rld <- rlog(dds)
plotPCA(rld)
```

```
## using ntop=500 top features by variance
```



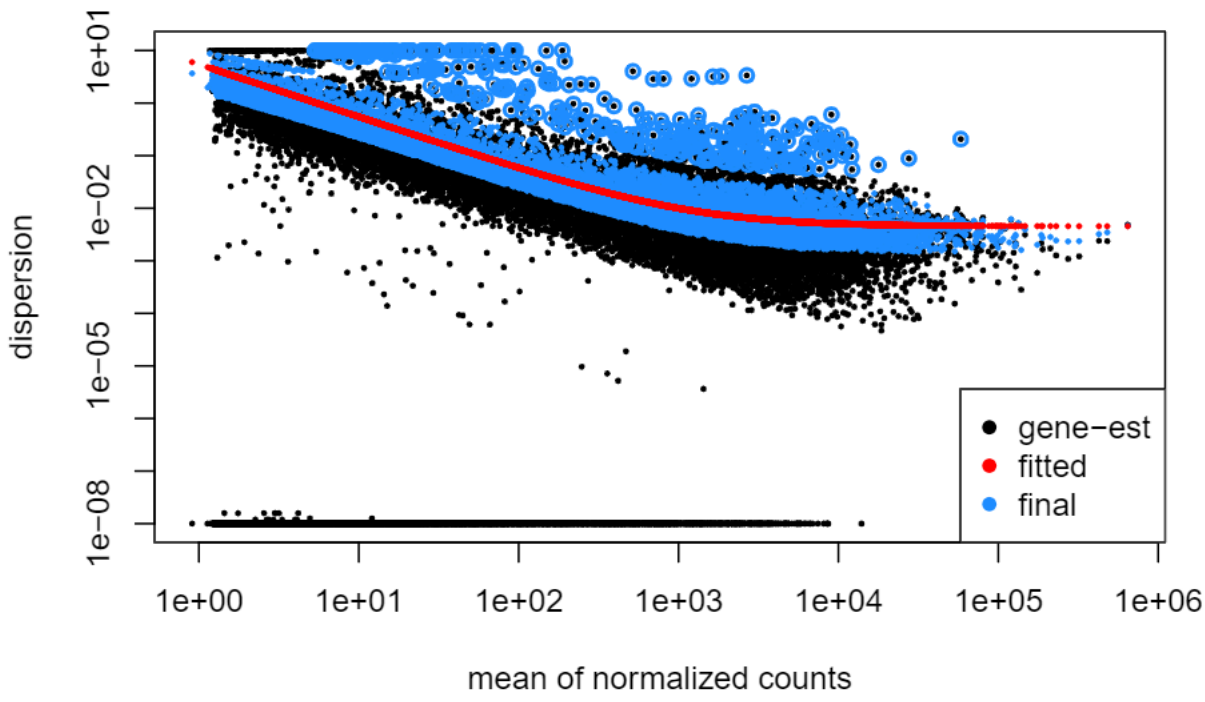
MA plot

```
plotMA(res.lfcShrink)
```



Dispersion by mean plot

```
plotDispEsts(dds)
```

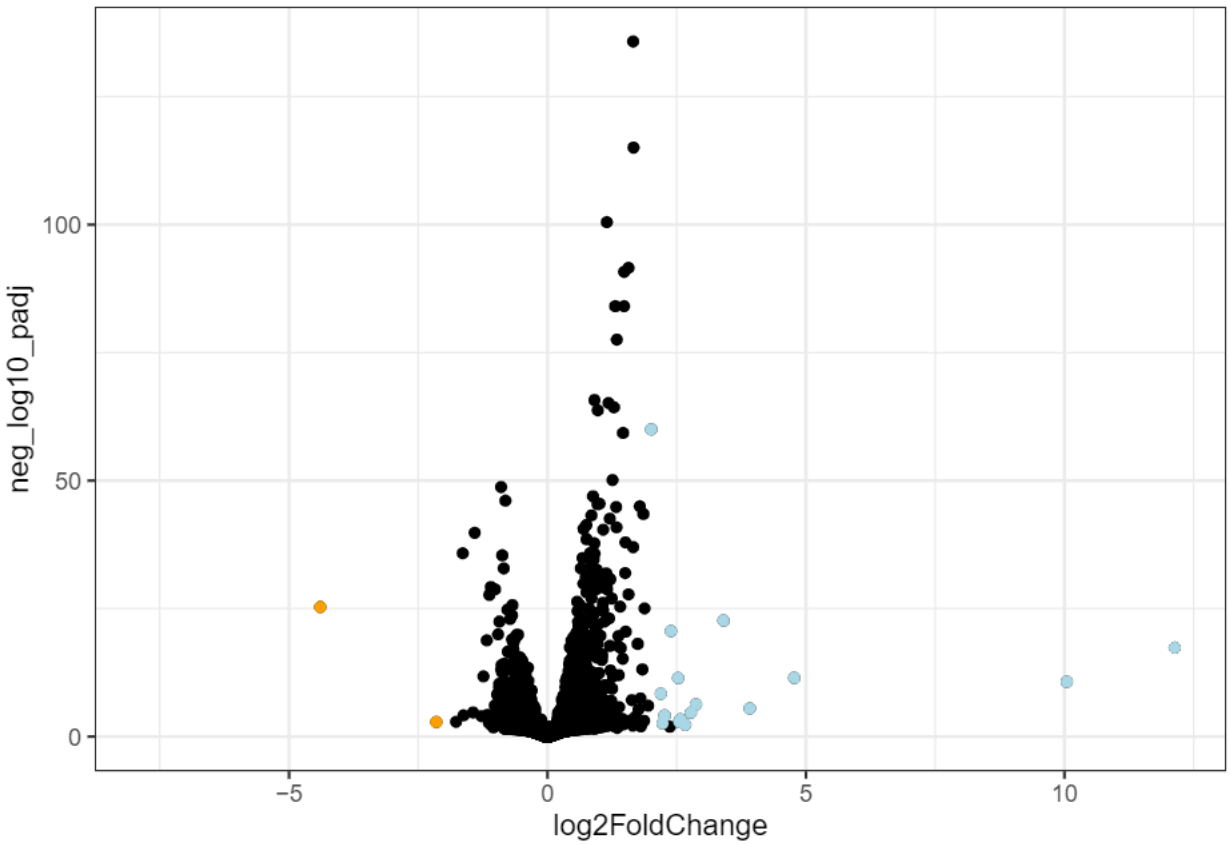


Volcano plot

```
res.lfcShrink.tbl_df %>%
  mutate(neg_log10_padj = -1*log10(padj)) %>%
  ggplot(aes(x = log2FoldChange, y = neg_log10_padj)) +
  geom_point(colr = 'gray') + # add gray points
  geom_point( data = ~.x %>% filter(log2FoldChange < -2 & neg_log10_padj > 2), color = 'orange') +
  geom_point( data = ~.x %>% filter(log2FoldChange > 2 & neg_log10_padj > 2), color = 'light blue') +
  theme_bw()
```

```
## Warning in geom_point(colr = "gray"): Ignoring unknown parameters: `colr`
```

```
## Warning: Removed 7819 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



GO Term enrichment

```
#Get significant
library(clusterProfiler)
```

```
##
## clusterProfiler v4.10.1  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.
##
## Attaching package: 'clusterProfiler'
## The following object is masked from 'package:purrr':
##
##     simplify
## The following object is masked from 'package:IRanges':
##
##     slice
## The following object is masked from 'package:S4Vectors':
##
##     rename
## The following object is masked from 'package:stats':
##
##     filter
```

```r
sig_genes = res.lfcShrink.tbl_df[res.lfcShrink.tbl_df$padj<0.05,]
sig_genes = na.omit(sig_genes)
#perform GO term enrichment
enrich_result <- enrichGO(gene = sig_genes$feature_id,
                          OrgDb = 'org.Hs.eg.db',
                          keyType = "ENSEMBL",
                          ont = "BP",
                          pAdjustMethod = "fdr",
                          pvalueCutoff = 0.05,
                          universe = res.lfcShrink.tbl_df$feature_id)
```

```
##
```

```r
dotplot(enrich_result,
        title = "GO Term Enrichment Analysis",
        showCategory = 15)
```