R Notebook

Loading library

```
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
```

```
library("apeglm")
library("ggplot2")
library("vsn")
library("pheatmap")
library("RColorBrewer")
```

The names of files have been changed by bash:

add suffix “Ductgroup” in the htseqcounts files of Duct group

add suffix “Lobulargroup” in the htseqcounts files of Lobular group

put them together in one folder named `Lobular_Ductr` to point it to this directory

Preparation

Set the Directory

The folder has been uploaded to Google Drive

```
directory <- "/Users/margery/Desktop/data/Lobular_Duct/"
#in my computer

#direction<- "~/TRGN510_Final_Project_Data/Lobular_Duct"
```

Set the `sample condition` & `sampleFiles`

Extract the information of Condition directly from file names which could guarantee the one-to-one correspondence of expressed matrix and samples

Use `grep` to select those files containing string `group` Use `sub` to chop up the sample filename to obtain the condition status

```
sampleFiles <- grep("group",list.files(directory),value=TRUE)
```

```
sampleCondition <- sub("(.*group).*","\\1",sampleFiles)
```

Build the DESeqDataSet

```
sampleTable <- data.frame(sampleName = sampleFiles,
                          fileName = sampleFiles,
                          condition = sampleCondition)
sampleTable$condition <- factor(sampleTable$condition)
```

```
library("DESeq2")
dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                  directory = directory,
                                  design= ~ condition)
dds
```

```
## class: DESeqDataSet
## dim: 60483 265
## metadata(1): version
## assays(1): counts
## rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSGR0000280767.1 ENSGR0000281849.1
## rowData names(0):
## colnames(265): Ductgroup-TCGA-3C-AALK-01A-11R-A41B-07.htseq.counts.gz
##   Ductgroup-TCGA-A2-A04N-01A-11R-A115-07.htseq.counts.gz ...
##   Lobulargroup-TCGA-WT-AB44-01A-11R-A41B-07.htseq.counts.gz
##   Lobulargroup-TCGA-XX-A89A-01A-11R-A36F-07.htseq.counts.gz
## colData names(1): condition
```

60483

Pre-filtering

Remove the genes with few reads (less than 10) to reduce the memory size and increase the speed

```
keep <- rowSums(counts(dds)) >= 10
dds <- dds[keep,]
```

After filtering, the number of elements has decreased from 60483 to 50860

By default, R will choose a reference level for factors based on alphabetical order

```
dds$condition
```

```
##   [1] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##   [6] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [11] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [16] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [21] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [26] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [31] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [36] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [41] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [46] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [51] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [56] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [61] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [66] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [71] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [76] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [81] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [86] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [91] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
##  [96] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [101] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [106] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [111] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [116] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [121] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [126] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [131] Ductgroup    Ductgroup    Ductgroup    Ductgroup    Ductgroup
## [136] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [141] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [146] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [151] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [156] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [161] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [166] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [171] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [176] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [181] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [186] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [191] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [196] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [201] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [206] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [211] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [216] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [221] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [226] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [231] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [236] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [241] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [246] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [251] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## [256] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
```

```
## [261] Lobulargroup Lobulargroup Lobulargroup Lobulargroup Lobulargroup
## Levels: Ductgroup Lobulargroup
```

In this case , Ductgroup is the reference , define it manually to make sure

log2 fold change and Wald test p value： last level / reference level log2 fold change： log2 (Lobulargroupt / Ductgroup)

Differential expression analysis

Use function `DESeq` to do differential expression analysis Use function `results` to generate results tables with log2 fold changes, p values and adjusted p values.

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

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
## -- replacing outliers and refitting for 10467 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

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

```
res
```

```
## log2 fold change (MLE): condition Lobulargroup vs Ductgroup
## Wald test p-value: condition Lobulargroup vs Ductgroup
## DataFrame with 50860 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE       stat      pvalue
##                    <numeric>      <numeric> <numeric>  <numeric>   <numeric>
## ENSG00000000003.13 2994.2734      0.0420056 0.0994397   0.422423 6.72717e-01
## ENSG00000000005.5    53.8553      1.0533486 0.2281769   4.616369 3.90512e-06
## ENSG00000000419.11 2071.6910     -0.3423046 0.0628977  -5.442241 5.26143e-08
## ENSG00000000457.12 2086.8536      0.1151750 0.0637844   1.805693 7.09663e-02
## ENSG00000000460.15  744.0237     -0.1224067 0.0822178  -1.488810 1.36537e-01
## ...                      ...            ...       ...        ...         ...
## ENSG00000281909.1   0.854753      0.4452084 0.2666805  1.6694448 0.095029260
## ENSG00000281910.1   0.422694     -0.0534396 0.5809119 -0.0919927 0.926703856
## ENSG00000281912.1  97.306569     -0.0323205 0.0968381 -0.3337582 0.738561999
## ENSG00000281918.1   2.374675      0.5346585 0.2594000  2.0611354 0.039290120
## ENSG00000281920.1   5.935921      0.7656355 0.1668402  4.5890348 0.000004453
##                          padj
##                     <numeric>
## ENSG00000000003.13 7.99464e-01
## ENSG00000000005.5  6.50184e-05
## ENSG00000000419.11 1.81304e-06
## ENSG00000000457.12 1.69797e-01
## ENSG00000000460.15 2.74298e-01
## ...                       ...
## ENSG00000281909.1  2.10845e-01
## ENSG00000281910.1  9.60304e-01
## ENSG00000281912.1  8.44083e-01
## ENSG00000281918.1  1.08932e-01
## ENSG00000281920.1  7.23534e-05
```

Export the results to csv file

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

Log fold change shrinkage for visualization and ranking

Use function `lfcShrink` to shrink the LFC `apeglm` : (Zhu, Ibrahim, and Love 2018) effect size shrinkage,which improves on the previous estimator

```
resultsNames(dds)
```

```
## [1] "Intercept"                              "condition_Lobulargroup_vs_Ductgroup"
```

```
resLFC <- lfcShrink(dds, coef="condition_Lobulargroup_vs_Ductgroup", 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
```

```
resLFC
```

```
## log2 fold change (MAP): condition Lobulargroup vs Ductgroup
## Wald test p-value: condition Lobulargroup vs Ductgroup
## DataFrame with 50860 rows and 5 columns
##                     baseMean log2FoldChange     lfcSE      pvalue        padj
##                    <numeric>      <numeric> <numeric>   <numeric>   <numeric>
## ENSG00000000003.13 2994.2734      0.0398904 0.0911447 6.72717e-01 7.99464e-01
## ENSG00000000005.5    53.8553      0.9556161 0.2381667 3.90512e-06 6.50184e-05
## ENSG00000000419.11 2071.6910     -0.3288230 0.0629304 5.26143e-08 1.81304e-06
## ENSG00000000457.12 2086.8536      0.1079616 0.0620041 7.09663e-02 1.69797e-01
## ENSG00000000460.15  744.0237     -0.1095951 0.0785919 1.36537e-01 2.74298e-01
## ...                      ...            ...       ...         ...         ...
## ENSG00000281909.1   0.854753      0.2339201 0.2373185 0.095029260 2.10845e-01
## ENSG00000281910.1   0.422694     -0.0631372 0.2217272 0.926703856 9.60304e-01
## ENSG00000281912.1  97.306569     -0.0273729 0.0890516 0.738561999 8.44083e-01
## ENSG00000281918.1   2.374675      0.3136781 0.2636270 0.039290120 1.08932e-01
## ENSG00000281920.1   5.935921      0.6980221 0.1739988 0.000004453 7.23534e-05
```

`resLFC` is more compacted compared to `res` column `stat` is removed after shrinking

```
names(resLFC)
```

```
## [1] "baseMean"       "log2FoldChange" "lfcSE"          "pvalue"
## [5] "padj"
```

```
names(res)
```

```
## [1] "baseMean"       "log2FoldChange" "lfcSE"          "stat"
## [5] "pvalue"         "padj"
```

Speed up

Use `parallel=TRUE` and `BPPARAM=MulticoreParam(4)` to split the job over 4 cores

```
library("BiocParallel")
register(MulticoreParam(4))
```

p-values and adjusted p-values

order our results table by the smallest p value

```
resOrdered <- res[order(res$pvalue),]
summary(res)
```

```
##
## out of 50840 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 6735, 13%
## LFC < 0 (down)     : 6960, 14%
## outliers [1]       : 0, 0%
## low counts [2]     : 11848, 23%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Find out How many adjusted p-values were less than 0.05?

```
sum(res$padj < 0.05, na.rm =T)
```

```
## [1] 11017
```

By default the argument alpha is set to 0.1, set the alpha = 0.05 means set the statistical significance to be 0.05

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

```
##
## out of 50840 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 5318, 10%
## LFC < 0 (down)     : 5751, 11%
## outliers [1]       : 0, 0%
## low counts [2]     : 12834, 25%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Define the Differential Expressed Gene and Export the results

```
diff_gene_deseq2 <-subset(res, padj < 0.05 & abs(log2FoldChange) > 1)
dim(diff_gene_deseq2)
```

```
## [1] 1490    6
```

```
head(diff_gene_deseq2)
```

```
## log2 fold change (MLE): condition Lobulargroup vs Ductgroup
## Wald test p-value: condition Lobulargroup vs Ductgroup
## DataFrame with 6 rows and 6 columns
##                     baseMean log2FoldChange     lfcSE      stat      pvalue
##                    <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000000005.5    53.8553        1.05335  0.228177   4.61637 3.90512e-06
## ENSG00000001626.13   28.7633        1.19972  0.200847   5.97327 2.32548e-09
## ENSG00000003249.12 2114.7789       -1.00437  0.111452  -9.01169 2.02901e-19
## ENSG00000004799.7  4025.7035        1.03231  0.170746   6.04589 1.48586e-09
## ENSG00000006377.10   12.2752       -2.37282  0.269294  -8.81127 1.23735e-18
## ENSG00000010438.15   15.1358       -1.52541  0.254267  -5.99924 1.98239e-09
##                          padj
##                     <numeric>
## ENSG00000000005.5  6.50184e-05
## ENSG00000001626.13 1.38041e-07
## ENSG00000003249.12 1.97889e-16
## ENSG00000004799.7  9.36451e-08
## ENSG00000006377.10 1.04938e-15
## ENSG00000010438.15 1.21028e-07
```

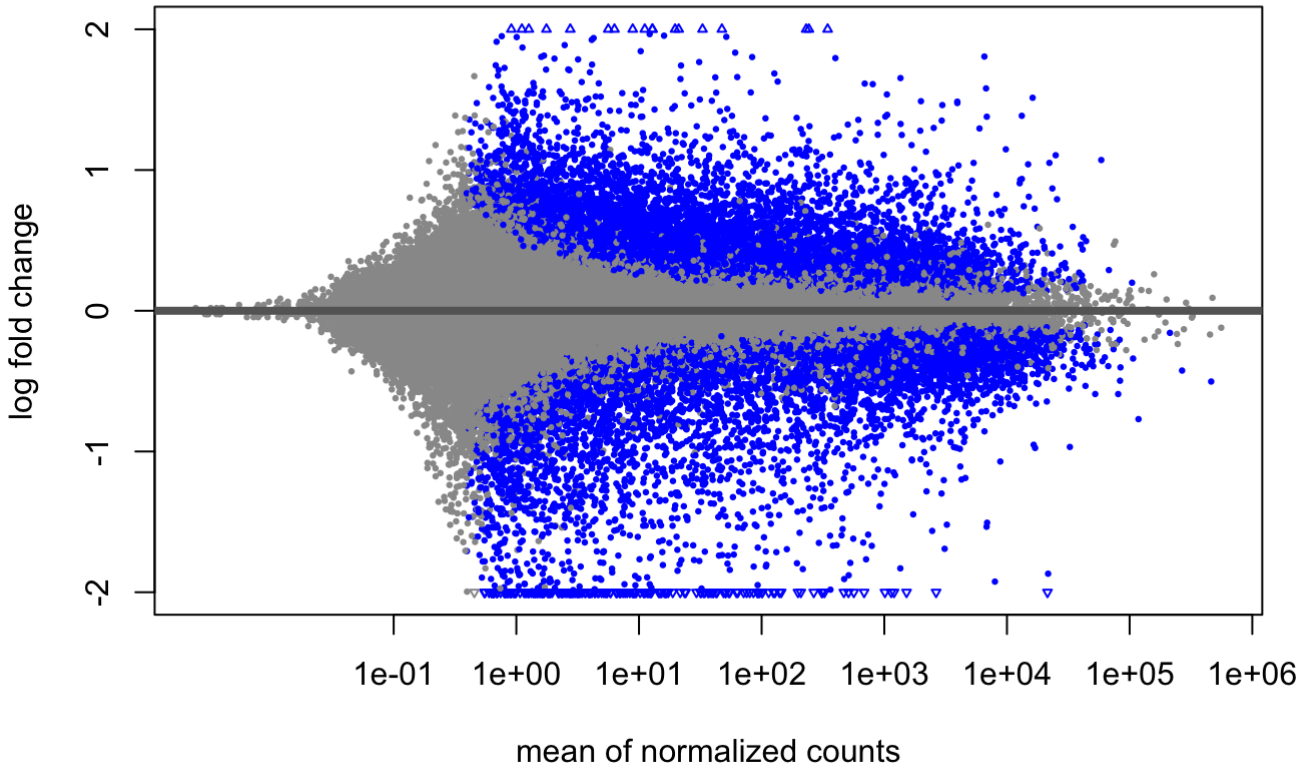
```
write.csv(diff_gene_deseq2,file = "DEG_Lobular_Duct.csv")
```

Visulization

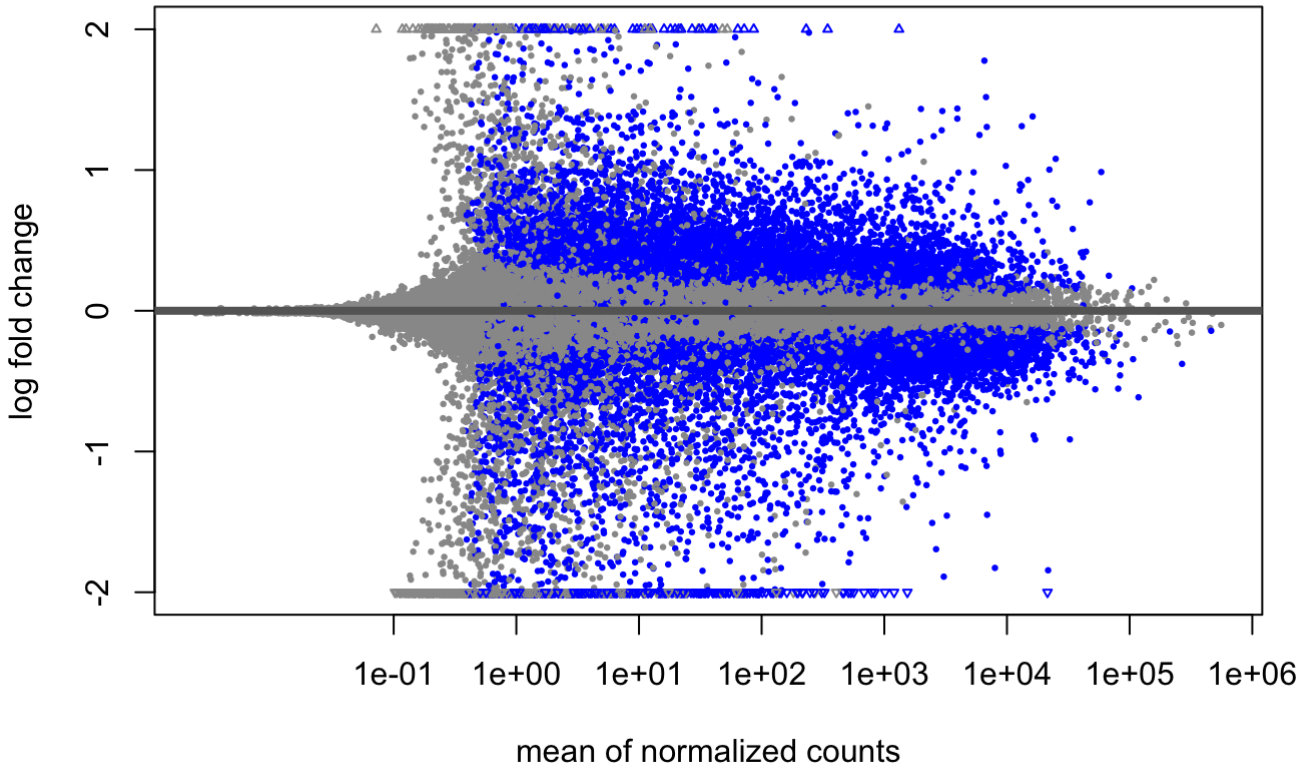
MA-plot

`plotMA` shows the log2 fold changes attributable to a given variable over the mean of normalized counts for all the samples in the DESeqDataSet.

```
plotMA(res, ylim=c(-2,2))
```




```
plotMA(resLFC, ylim=c(-2,2))
```



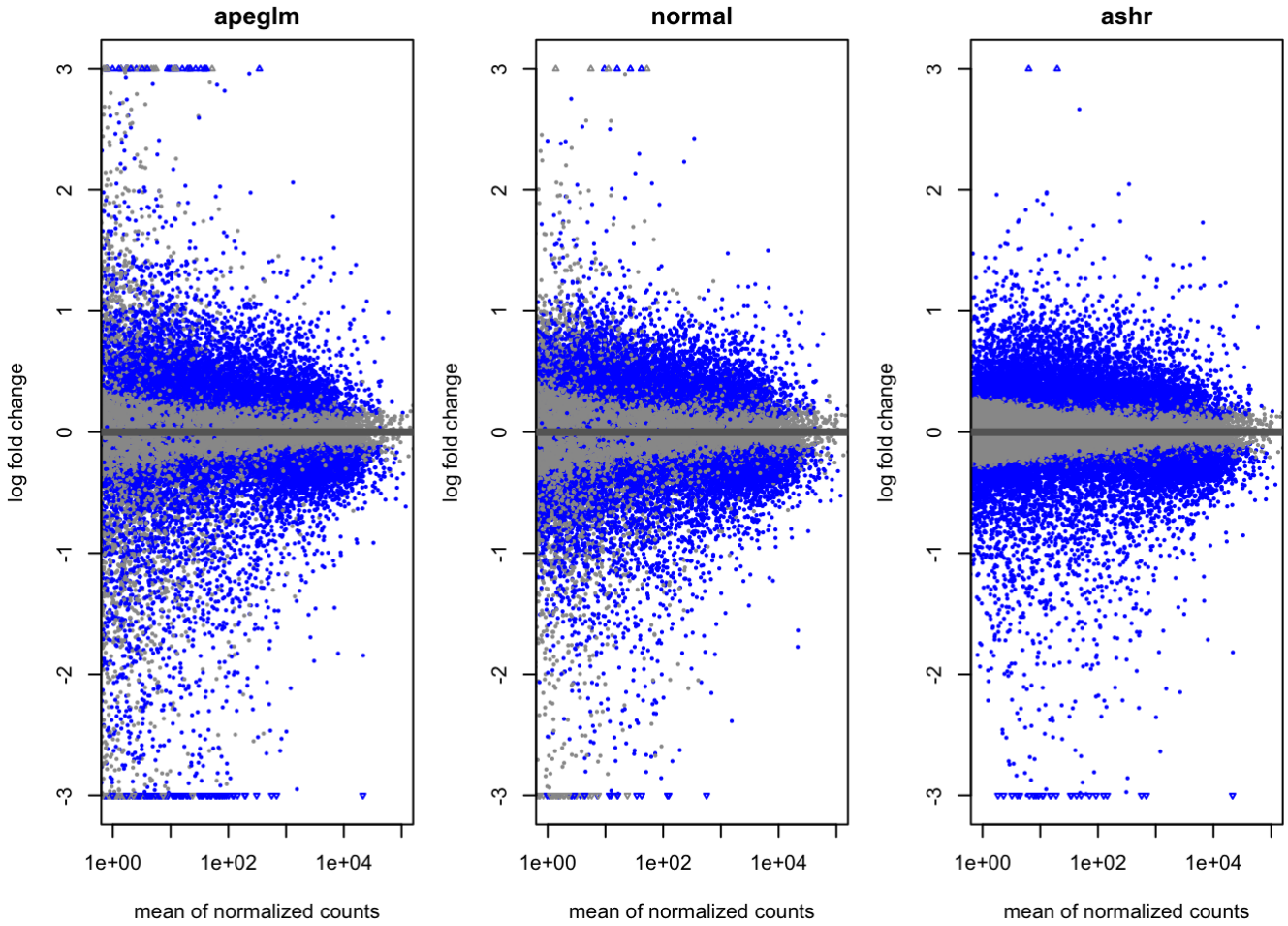

```
library(ashr)
resNorm  <- lfcShrink(dds, coef=2, type="normal")
```

```
## using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).
##
## Note that type='apeglm' and type='ashr' have shown to have less bias than type='norma
l'.
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
## Reference: https://doi.org/10.1093/bioinformatics/bty895
```

```
resAsh <- lfcShrink(dds, coef=2, type="ashr")
```

```
## using 'ashr' for LFC shrinkage. If used in published research, please cite:
##     Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
##     https://doi.org/10.1093/biostatistics/kxw041
```

```
par(mfrow=c(1,3), mar=c(4,4,2,1))
xlim <- c(1,1e5); ylim <- c(-3,3)
plotMA(resLFC, xlim=xlim, ylim=ylim, main="apeglm")
plotMA(resNorm, xlim=xlim, ylim=ylim, main="normal")
plotMA(resAsh, xlim=xlim, ylim=ylim, main="ashr")
```



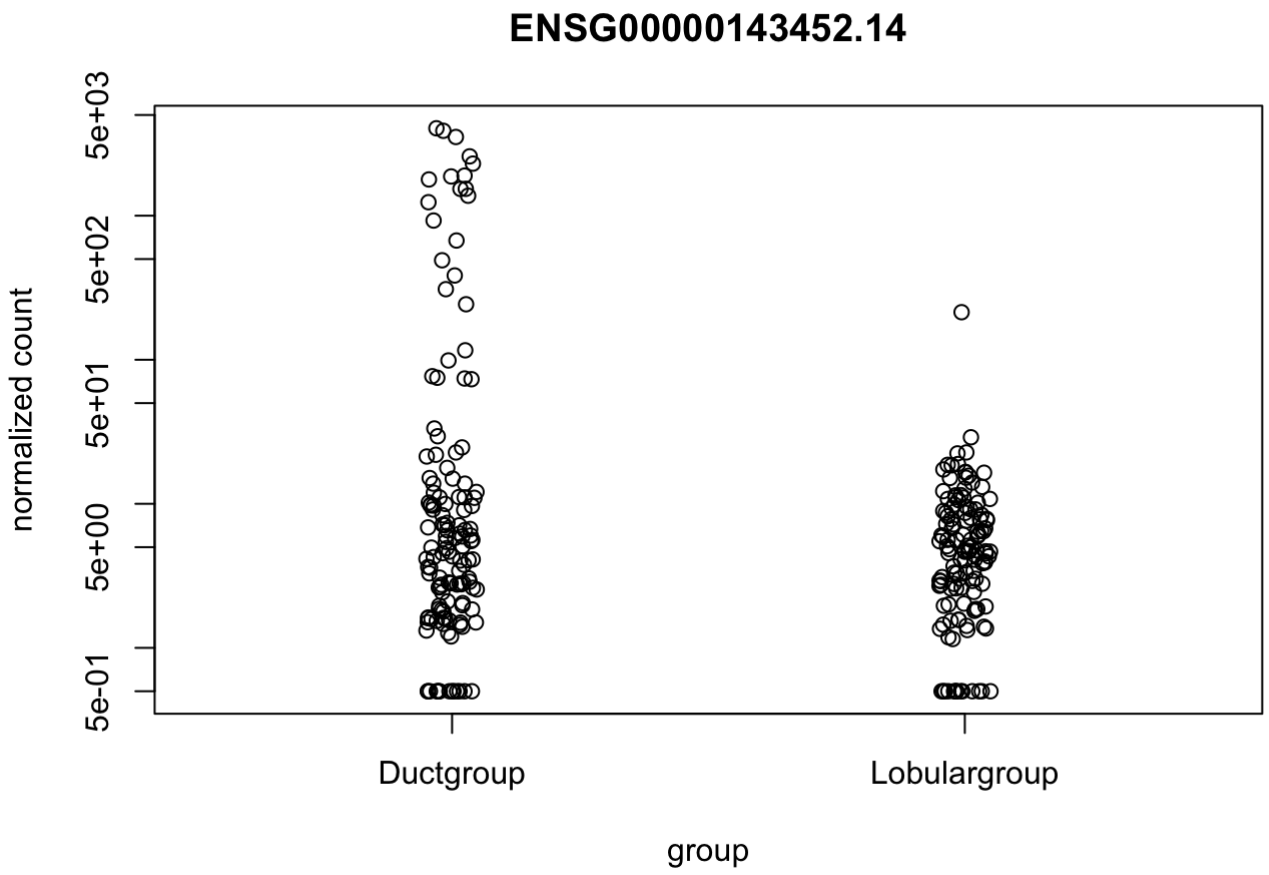
Plot counts

Examine the counts of reads for a single gene across the groups

Use function `plotCounts` to normalize counts by the estimated size factors and adds a pseudocount of 1/2 to allow for log scale plotting.

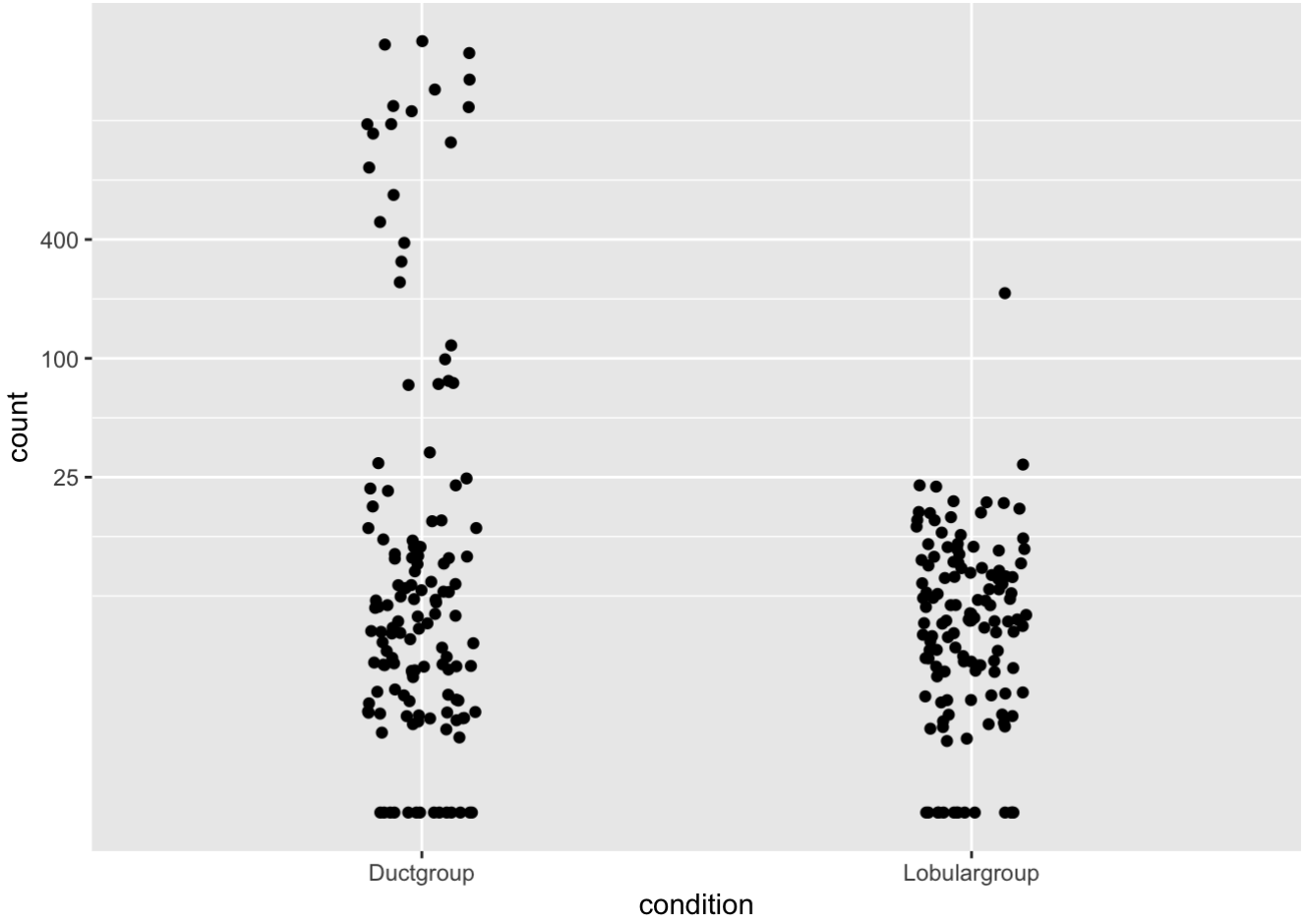
Here we specify the gene which had the **smallest p value** from the results table created above.

```
plotCounts(dds, gene=which.min(res$padj), intgroup="condition")
```



For customized plotting, an argument returnData specifies that the function should only return a data.frame for plotting with ggplot.

```
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="condition",
                returnData=TRUE)
library("ggplot2")
ggplot(d, aes(x=condition, y=count)) +
  geom_point(position=position_jitter(w=0.1,h=0)) +
  scale_y_log10(breaks=c(25,100,400))
```



More information on results columns

Use function `mcols` to find which variables and tests were used

```
mcols(res)$description
```

```
## [1] "mean of normalized counts for all samples"
## [2] "log2 fold change (MLE): condition Lobulargroup vs Ductgroup"
## [3] "standard error: condition Lobulargroup vs Ductgroup"
## [4] "Wald statistic: condition Lobulargroup vs Ductgroup"
## [5] "Wald test p-value: condition Lobulargroup vs Ductgroup"
## [6] "BH adjusted p-values"
```

For a particular gene, a log2 fold change of -1 for `condition logroup vs ductgroup` means that the `logroup` induces a multiplicative change in observed gene expression level of 2^-1 = 0.5 compared to the untreated condition.

Data transformations and visualization

Count data transformations

Use function `vsd` to remove the dependence of the variance on the mean instead of function `rlog` for it takes MUCH less time

```
vsd <- vst(dds, blind=FALSE)
```

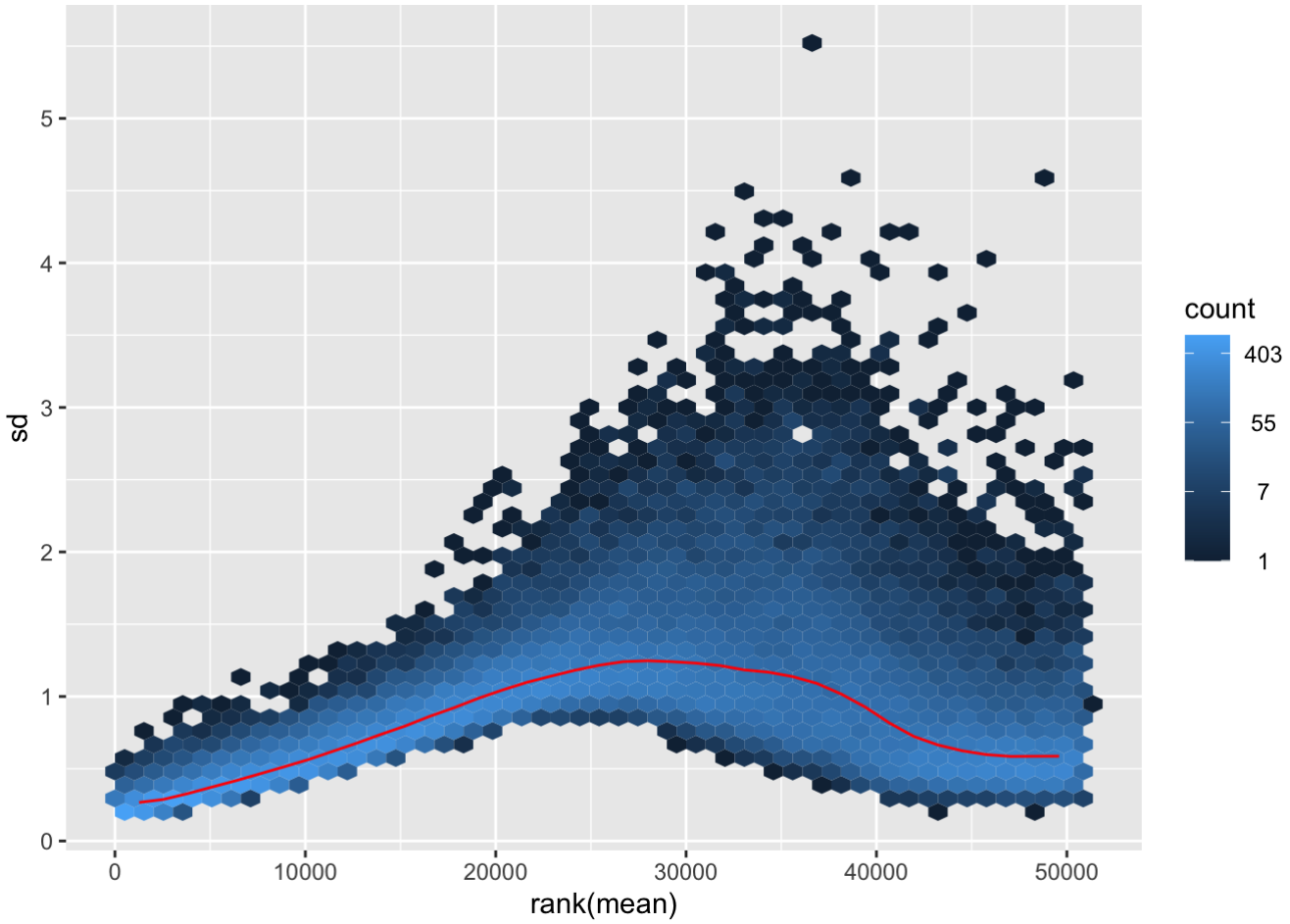
It takes more than 24hrs, so I cut it off

```
#rld <- rlog(dds, blind=FALSE)
```

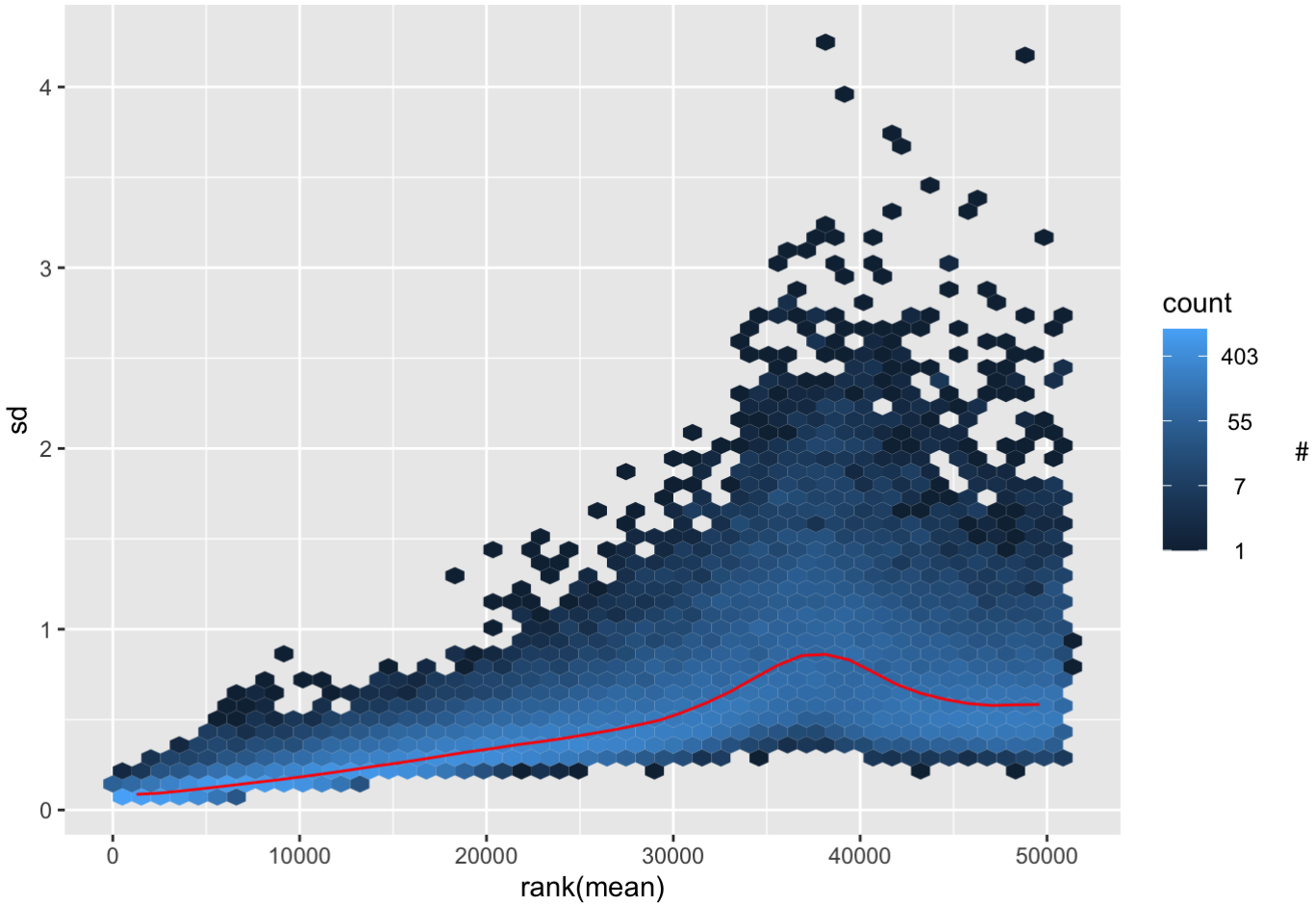
Extracting transformed values

```
# this gives log2(n + 1)
ntd <- normTransform(dds)
```

```
library("vsn")
meanSdPlot(assay(ntd))
```




```
meanSdPlot(assay(vsd))
```



Data quality assessment by sample clustering and visualization

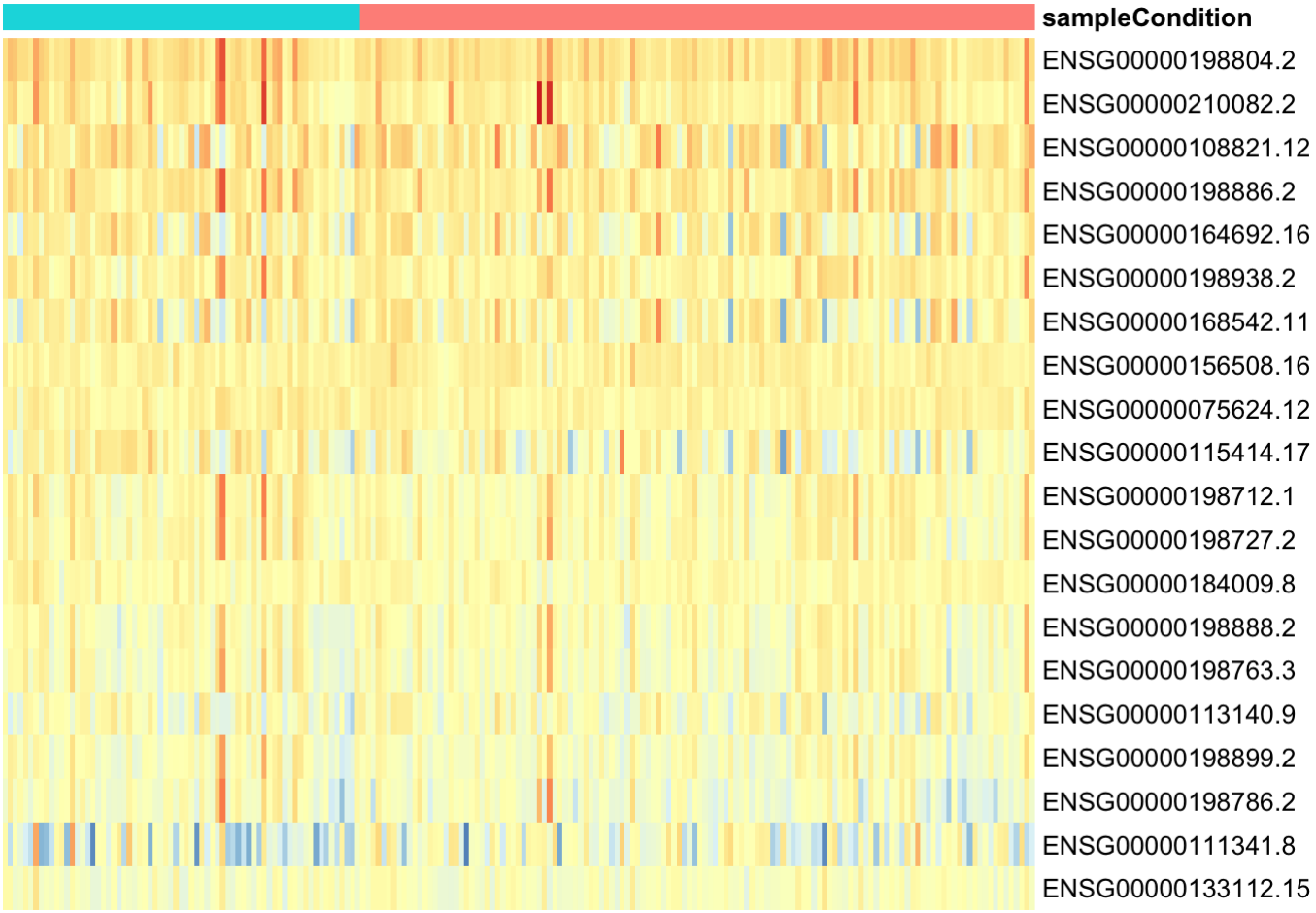
Heatmap of the count matrix

```
df <- as.data.frame(sampleCondition)
```

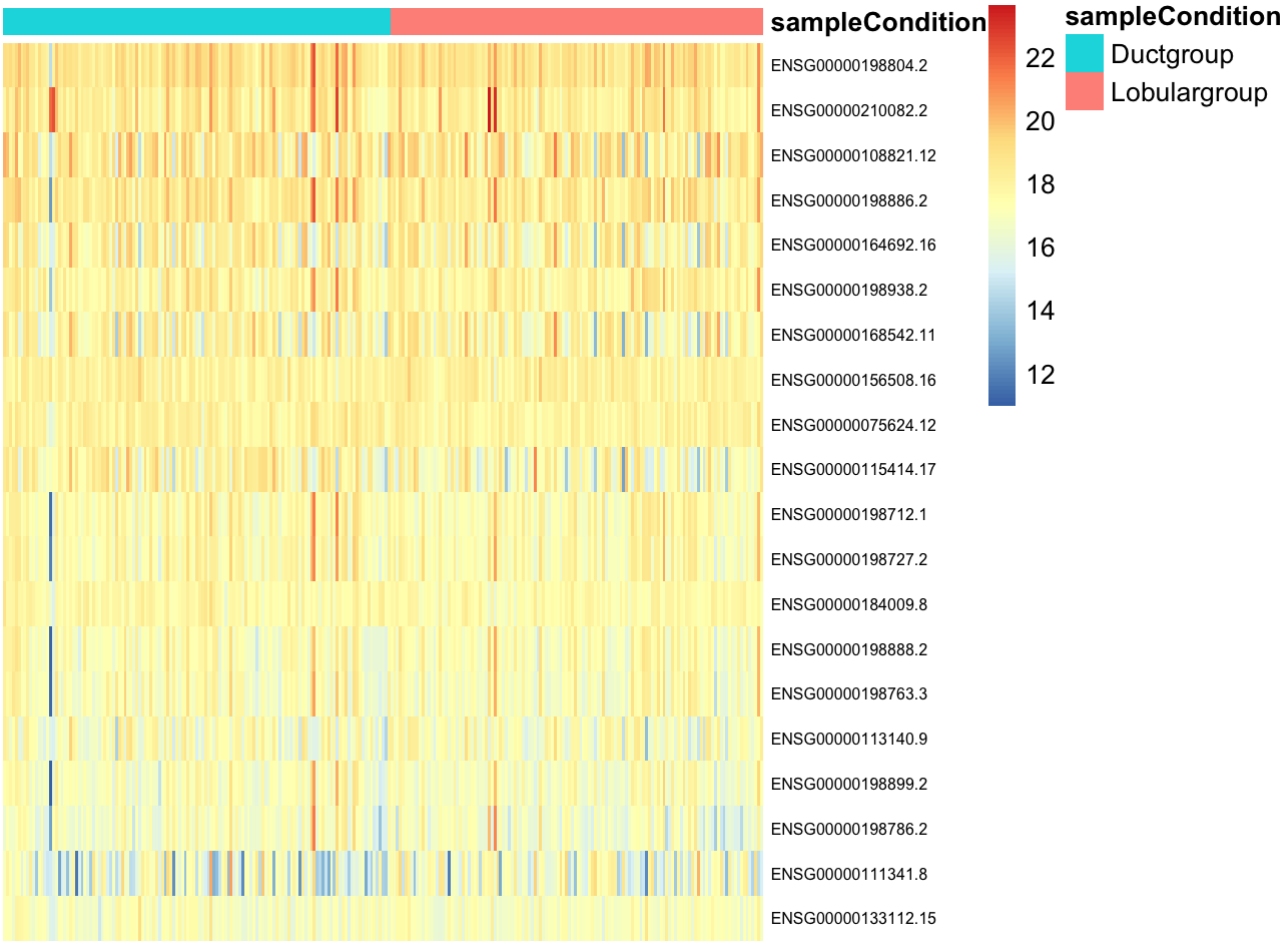
```
library("pheatmap")
select <- order(rowMeans(counts(dds,normalized=TRUE)),
                decreasing=TRUE)[1:20]
```

```
rownames(df) <- colnames(ntd)
```

```
pheatmap(assay(ntd)[select,], cluster_rows=FALSE,show_colnames=FALSE,
         cluster_cols=FALSE,annotation = df,cellwidth=2)
```




```
pheatmap(assay(vsd)[select,], cluster_rows=FALSE,show_colnames=FALSE,
         cluster_cols=FALSE,annotation = df,fontsize_row = 6)
```




```
normCounts <- counts(dds,normalized = T)
```

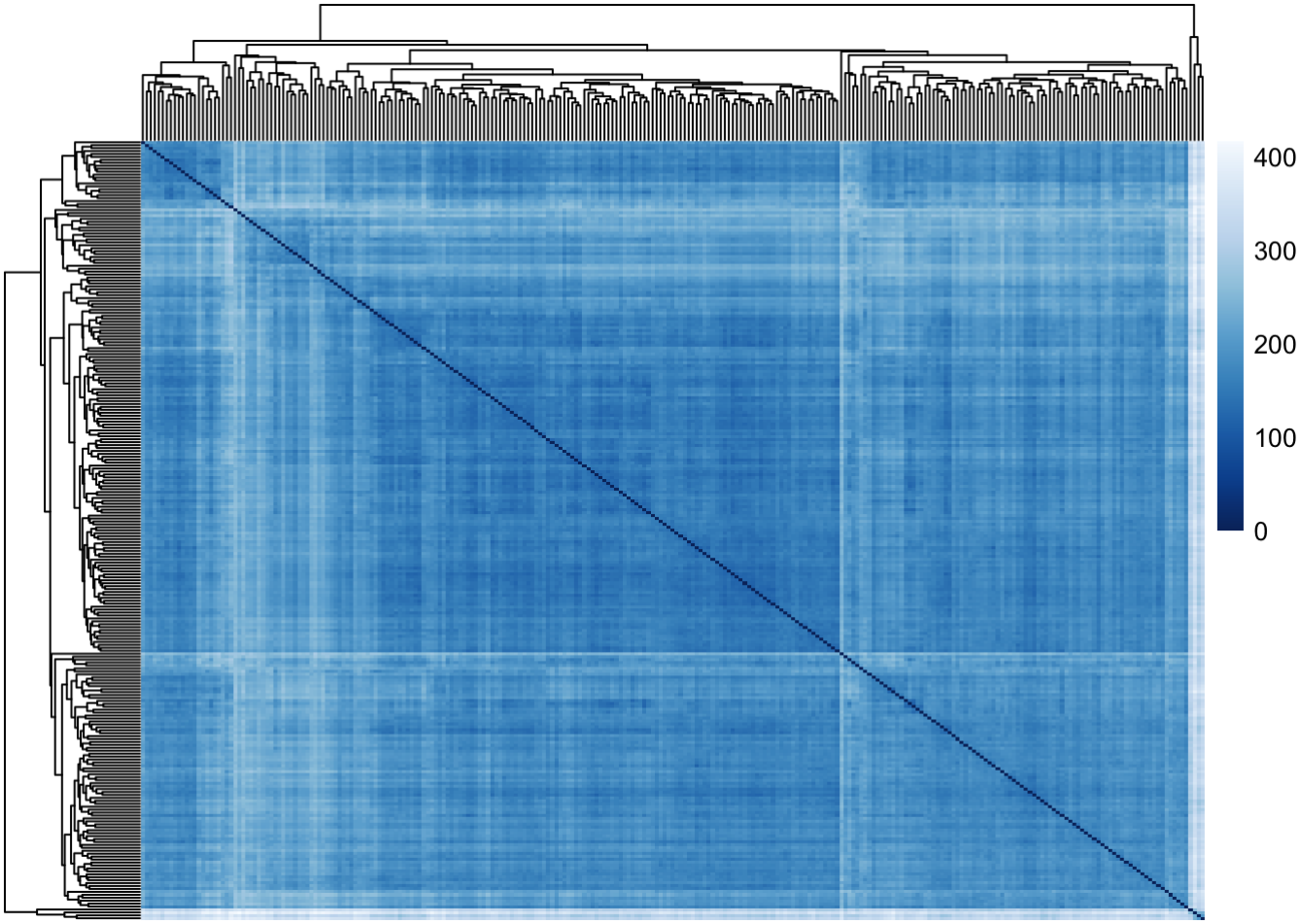
Heatmap of the sample-to-sample distances

Apply the dist function to the transpose of the transformed count matrix to get sample-to-sample distances.

```
sampleDists <- dist(t(assay(vsd)))
```

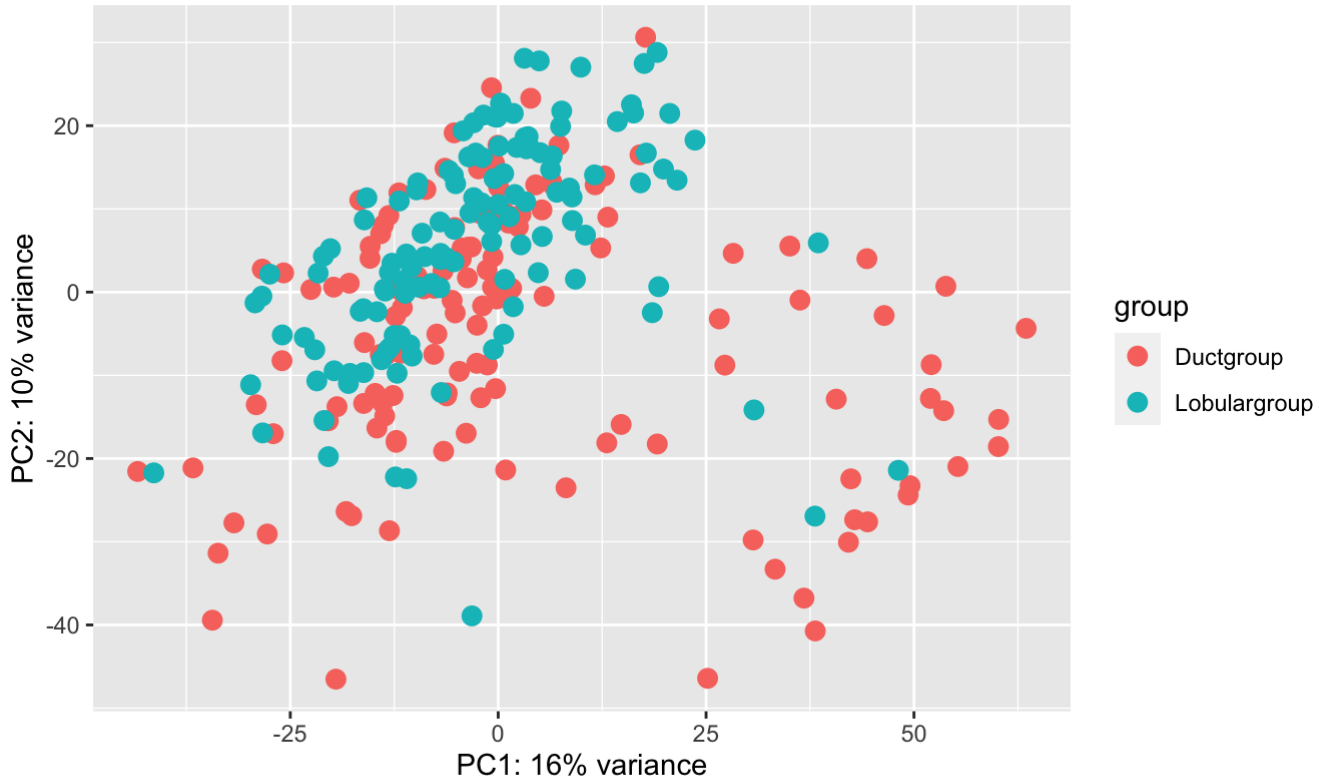
```
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, vsd$type, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
```

```
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors,show_rownames = F)
```



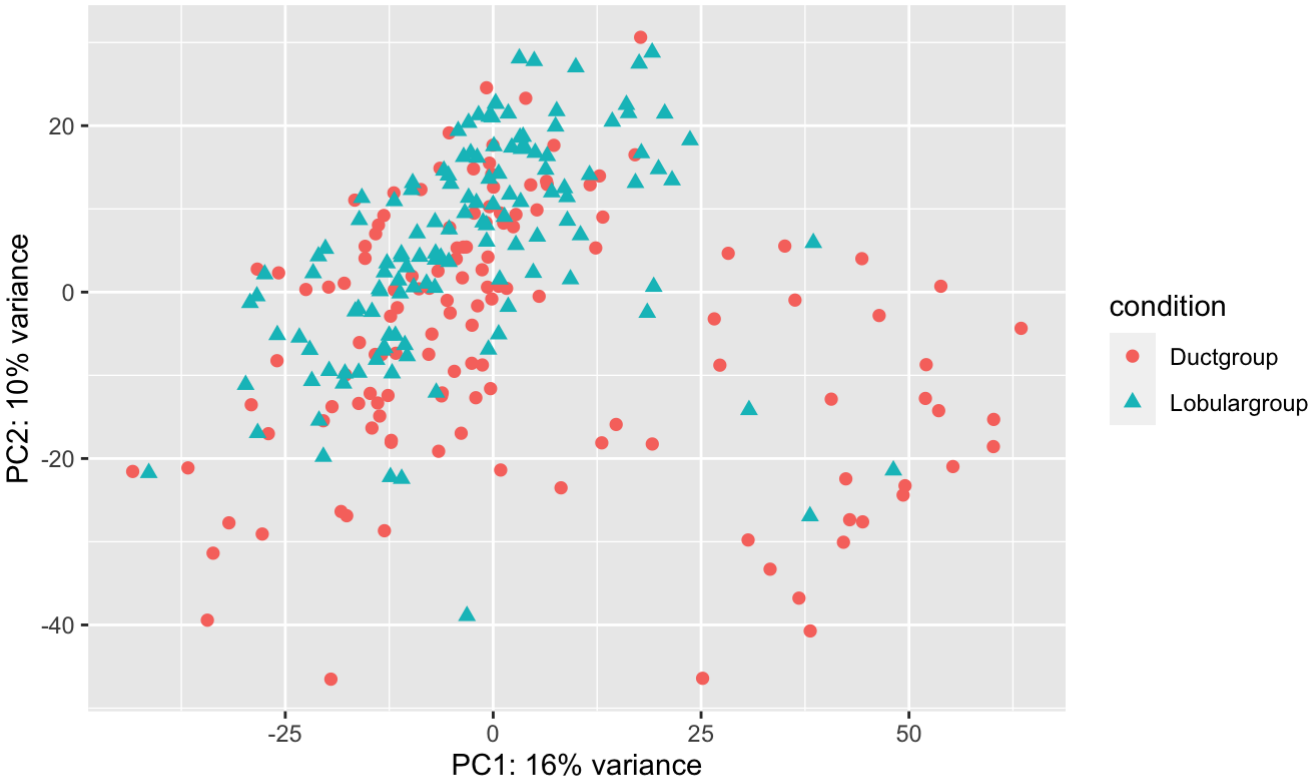
Principal component plot of the samples

```
plotPCA(vsd, intgroup=c("condition"))
```



customize the PCA plot using the ggplot function.

```
pcaData <- plotPCA(vsd, intgroup=c("condition"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=condition, shape=condition)) +
  geom_point(size=2) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  coord_fixed()
```


20
0
-20
-40
PC2: 10% variance
-25
0
25
50
PC1: 16% variance
condition
Ductgroup
Lobulargroup