Analyzing_RNAseq_data_DESeq2

Analyzing RNA-seq data with DESeq2

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")
```

Preparation

Set the Directory

The folder has been uploaded to Google Drive

```
directory <- "/Users/margery/Desktop/data/New_Lobular_Duct/"


#direction<- "~/510_Final_Project_Data/New_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)
```

Set `sampleTable`

Only consider the `condition` as Factor

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

Build the DESeqDataSet Only Factor is "condition")

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

```
## class: DESeqDataSet
## dim: 60483 264
## metadata(1): version
## assays(1): counts
## rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSGR0000280767.1 ENSGR0000281849.1
## rowData names(0):
## colnames(264): Ductgroup1.gz Ductgroup10.gz ... Lobulargroup98.gz
##   Lobulargroup99.gz
## colData names(1): condition
```

Remove the suffix of fileName in `sampleTable`

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
sampleTable <- sampleTable %>%
  tidyr::separate(fileName,into = c("fileName"),sep = "\\.")
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 264 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

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

```
head(dds$condition)
```

```
## [1] Ductgroup Ductgroup Ductgroup Ductgroup Ductgroup Ductgroup
## 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)

```
Col_Duct_Lobular <- read.csv("Col_Duct_Lobular.csv")
```

Build sampleTable with 2 Factors

Both `condition` & `Stage` are Factors

```
colnames(Col_Duct_Lobular)[2] <- "fileName"
sampleTable <- merge(sampleTable,Col_Duct_Lobular,by="fileName")
sampleTableselect <- sampleTable
```

###Add multiple factors “Stage”

Select Necessary Columns

```
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
```

```
sampleTableselect <- sampleTableselect%>%
  dplyr::select(fileName,sampleName,condition,ajcc_pathologic_stage)
```

```
sampleTableselect$condition <- factor(sampleTableselect$condition)
sampleTableselect$Stage <- factor(sampleTableselect$ajcc_pathologic_stage)
```

Build the DESeqDataSet Factors are “condition” and “Stage”)

```
ddsMF <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTableselect,
                                    directory = directory,
                                    design= ~ condition + Stage)
```

```
##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
ddsMF
```

```
## class: DESeqDataSet
## dim: 60483 264
## metadata(1): version
## assays(1): counts
## rownames(60483): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSGR0000280767.1 ENSGR0000281849.1
## rowData names(0):
## colnames(264): Ductgroup1 Ductgroup10 ... Lobulargroup98 Lobulargroup99
## colData names(3): condition ajcc_pathologic_stage Stage
```

```
ddsMF <- DESeq(ddsMF)
```

```
## estimating size factors

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]

## final dispersion estimates

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]

## fitting model and testing

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
## -- replacing outliers and refitting for 7533 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]

## fitting model and testing

##   Note: levels of factors in the design contain characters other than
##   letters, numbers, '_' and '.'. It is recommended (but not required) to use
##   only letters, numbers, and delimiters '_' or '.', as these are safe characters
##   for column names in R. [This is a message, not a warning or an error]
```

```
head(ddsMF)
```

```
## class: DESeqDataSet
## dim: 6 264
## metadata(1): version
## assays(6): counts mu ... replaceCounts replaceCooks
## rownames(6): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSG00000000460.15 ENSG00000000938.11
## rowData names(39): baseMean baseVar ... maxCooks replace
## colnames(264): Ductgroup1 Ductgroup10 ... Lobulargroup98 Lobulargroup99
## colData names(5): condition ajcc_pathologic_stage Stage sizeFactor
##   replaceable
```

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,]
dds
```

```
## class: DESeqDataSet
## dim: 50442 264
## metadata(1): version
## assays(1): counts
## rownames(50442): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSG00000281918.1 ENSG00000281920.1
## rowData names(0):
## colnames(264): Ductgroup1.gz Ductgroup10.gz ... Lobulargroup98.gz
##   Lobulargroup99.gz
## colData names(1): condition
```

```
dds
```

```
## class: DESeqDataSet
## dim: 50442 264
## metadata(1): version
## assays(1): counts
## rownames(50442): ENSG00000000003.13 ENSG00000000005.5 ...
##   ENSG00000281918.1 ENSG00000281920.1
## rowData names(0):
## colnames(264): Ductgroup1.gz Ductgroup10.gz ... Lobulargroup98.gz
##   Lobulargroup99.gz
## colData names(1): condition
```

After filtering, the number of elements has decreased from 60483 to 50442

Check Factors

```
head(ddsMF$condition)
```

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

```
head(ddsMF$Stage)
```

```
## [1] Stage I  Stage I  Stage I  Stage I  Stage I  Stage IA
## Levels: Stage I Stage IA Stage II Stage IIA Stage IIB
```

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 9875 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 50442 rows and 6 columns
##                      baseMean log2FoldChange     lfcSE      stat      pvalue
##                     <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000000003.13 2989.3855      0.0482327 0.0996736  0.483907 6.28452e-01
## ENSG00000000005.5    54.0173      1.0471680 0.2288546  4.575691 4.74651e-06
## ENSG00000000419.11 2067.3363     -0.3369834 0.0629565 -5.352636 8.66819e-08
## ENSG00000000457.12 2087.5527      0.1159625 0.0640323  1.811002 7.01406e-02
## ENSG00000000460.15  743.5515     -0.1198142 0.0824957 -1.452369 1.46399e-01
## ...                      ...            ...       ...       ...         ...
## ENSG00000281909.1   0.849721      0.4665818 0.2681242  1.740170 8.18291e-02
## ENSG00000281910.1   0.342130      0.3502987 0.5512240  0.635492 5.25107e-01
## ENSG00000281912.1  97.282091     -0.0305586 0.0971704 -0.314485 7.53153e-01
## ENSG00000281918.1   2.345900      0.5859584 0.2593639  2.259213 2.38701e-02
## ENSG00000281920.1   5.961378      0.7547860 0.1663856  4.536366 5.72319e-06
##                          padj
##                     <numeric>
## ENSG00000000003.13 7.61261e-01
## ENSG00000000005.5  7.65734e-05
## ENSG00000000419.11 2.73642e-06
## ENSG00000000457.12 1.64740e-01
## ENSG00000000460.15 2.82971e-01
## ...                      ...
## ENSG00000281909.1  1.85083e-01
## ENSG00000281910.1           NA
## ENSG00000281912.1  8.49651e-01
## ENSG00000281918.1  7.28337e-02
## ENSG00000281920.1  8.95306e-05
```

Export the results to csv file

```
write.csv(res,file="Res_Lobular_Duct_All.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 50442 rows and 5 columns
##                     baseMean log2FoldChange     lfcSE      pvalue        padj
##                    <numeric>      <numeric> <numeric>   <numeric>   <numeric>
## ENSG00000000003.13 2989.3855      0.0454708 0.0915236 6.28452e-01 7.61261e-01
## ENSG00000000005.5    54.0173      0.9483953 0.2390311 4.74651e-06 7.65734e-05
## ENSG00000000419.11 2067.3363     -0.3235617 0.0629601 8.66819e-08 2.73642e-06
## ENSG00000000457.12 2087.5527      0.1086935 0.0622578 7.01406e-02 1.64740e-01
## ENSG00000000460.15  743.5515     -0.1072662 0.0788147 1.46399e-01 2.82971e-01
## ...                      ...            ...       ...         ...         ...
## ENSG00000281909.1   0.849721     0.25126298 0.2450287 8.18291e-02 1.85083e-01
## ENSG00000281910.1   0.342130    -0.00727492 0.2112771 5.25107e-01          NA
## ENSG00000281912.1  97.282091    -0.02591254 0.0893922 7.53153e-01 8.49651e-01
## ENSG00000281918.1   2.345900     0.36878319 0.2765818 2.38701e-02 7.28337e-02
## ENSG00000281920.1   5.961378     0.68697840 0.1735369 5.72319e-06 8.95306e-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))
```

Define the Differential Expressed Gene and Export the results

Define DEG as padj <= 0.05 & abs(log2FoldChange) >= 1.5

```
diff_gene_deseq2 <-subset(res, padj <= 0.05 & abs(log2FoldChange) >= 1.5)
dim(diff_gene_deseq2)
```

```
## [1] 463   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>
## ENSG00000006377.10    12.31083       -2.38141  0.269931  -8.82230 1.12135e-18
## ENSG00000010438.15    15.18604       -1.53442  0.254719  -6.02398 1.70179e-09
## ENSG00000019186.8     45.64159       -1.67105  0.256573  -6.51298 7.36752e-11
## ENSG00000036473.6      2.26701        1.54749  0.332245   4.65767 3.19802e-06
## ENSG00000039068.17 21560.11773       -1.86367  0.151579 -12.29503 9.63256e-35
## ENSG00000047457.12  3116.28392       -1.70154  0.268040  -6.34807 2.18034e-10
##                          padj
##                      <numeric>
## ENSG00000006377.10 1.05850e-15
## ENSG00000010438.15 1.07080e-07
## ENSG00000019186.8  7.70643e-09
## ENSG00000036473.6  5.50087e-05
## ENSG00000039068.17 6.21332e-31
## ENSG00000047457.12 1.90482e-08
```

```
summary(diff_gene_deseq2)
```

```
##
## out of 463 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 104, 22%
## LFC < 0 (down)     : 359, 78%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

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

```
diff_gene_deseq2 <- read.csv("/Users/margery/Desktop/Results/New_DEG_Lobular_Duct.csv")
```

ID Transfer

Change column name

```
colnames(diff_gene_deseq2)[1] <- "gene_id"
```

Remove the version number

```
library(tidyr)
diff_gene_deseq2 <- diff_gene_deseq2 %>%
  tidyr::separate(gene_id,into = c("gene_id"),sep = "\\.")
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 463 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
library(AnnotationDbi)
```

```
## Warning: package 'AnnotationDbi' was built under R version 4.1.2
```

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

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

```
library(org.Hs.eg.db)
```

```
##
```

```
diff_gene_deseq2$symbol <- mapIds(org.Hs.eg.db,
                     keys=diff_gene_deseq2$gene_id,
                     column="SYMBOL",
                     keytype="ENSEMBL",
                     multiVals="first")
```

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

Remove Duplicated and NA

```
library(dplyr)
diff_gene_deseq2 <- diff_gene_deseq2 %>%
  ## Remove NA
  filter(symbol!="NA") %>%
  ## Remove Duplicate
  distinct(symbol,.keep_all = T)
```

Extract UP-regulated & Down-regulated

```
Up_diff <- data.frame(subset(diff_gene_deseq2,log2FoldChange > 0) )
Up_diff$gene_reg <- "Up"
Down_diff <- data.frame(subset(diff_gene_deseq2,log2FoldChange < 0) )
Down_diff$gene_reg <- "Down"
```

```
reg_diff <- rbind(Up_diff,Down_diff)
```

```
write.csv(Up_diff,"Up_diff.csv",row.names = F)
write.csv(Down_diff,"Down_diff.csv",row.names = F)
write.csv(reg_diff,"All_diff_Reg.csv",row.names = F)
```

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"
```

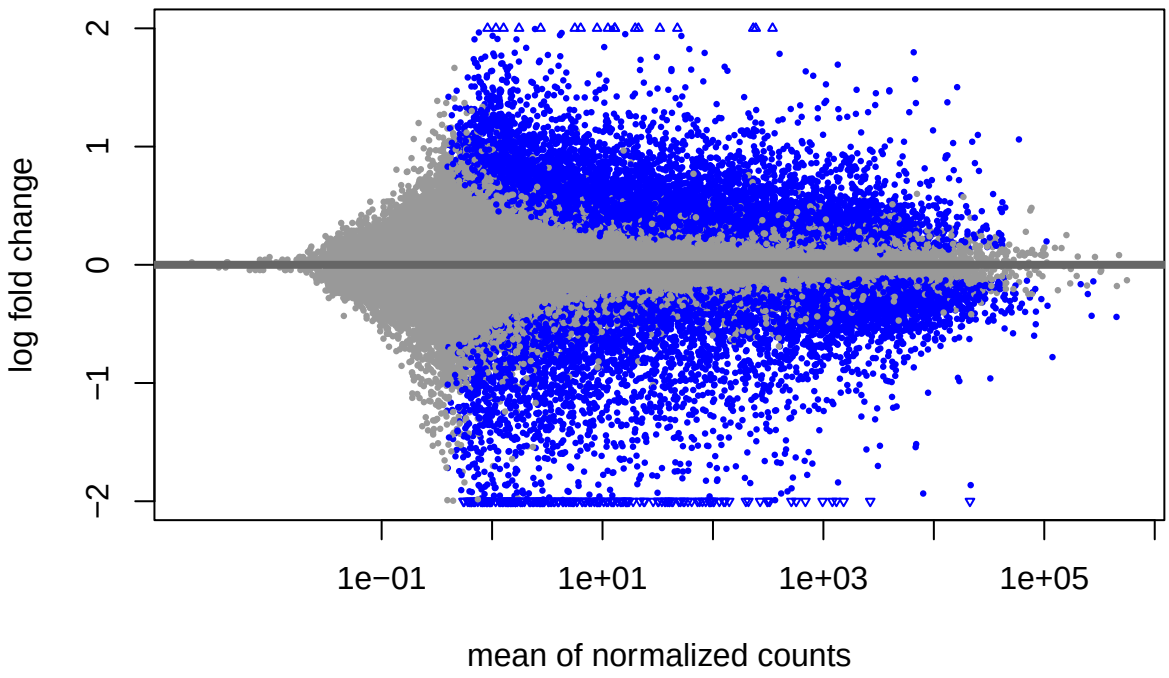
For a particular gene, a log2 fold change of -1 for `condition logroup vs ductgroup` means that the `logroup` induces a multiplicative change in observed gene expression level of 2^-1 = 0.5 compared to the untreated condition.

Visulization

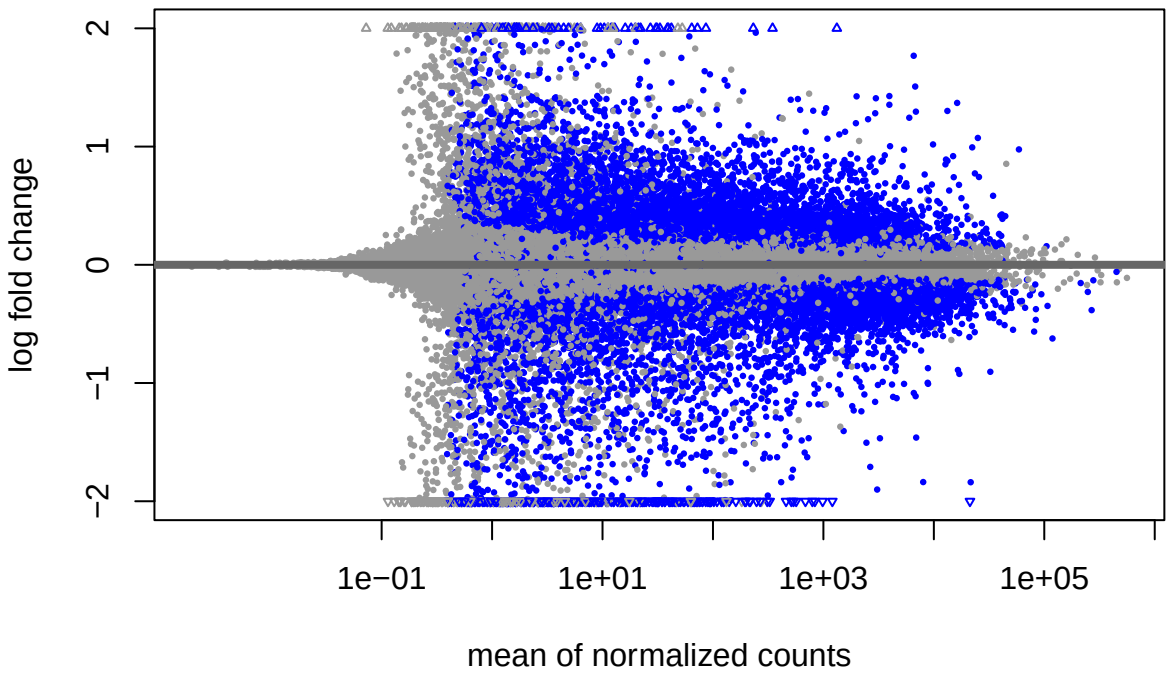
MA-plot

`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))
```

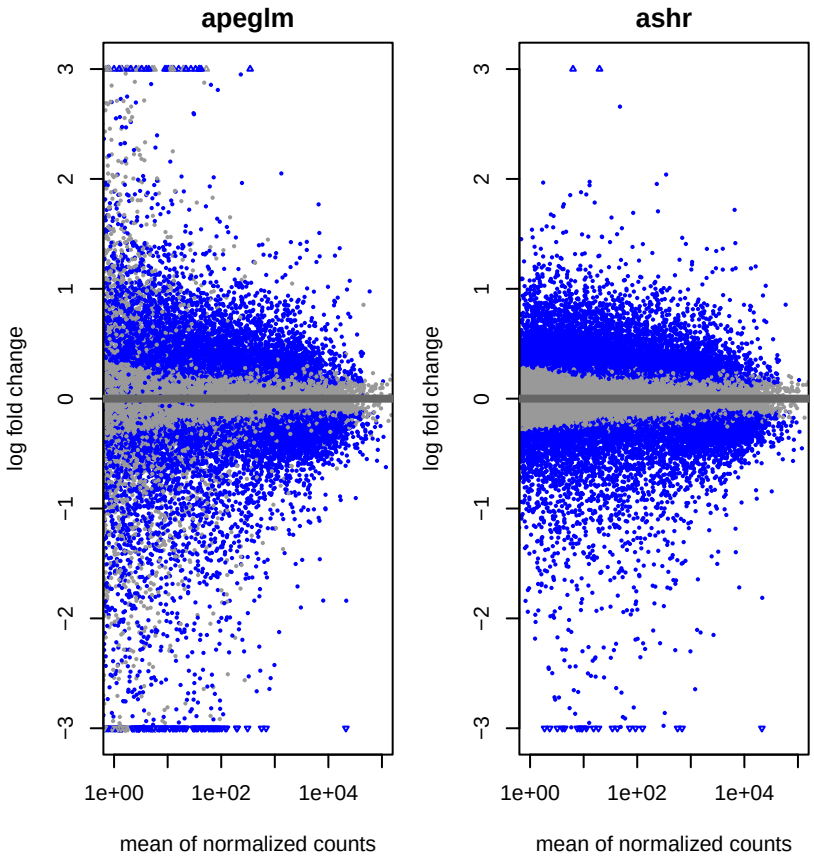



```
library(ashr)
resAsh <- lfcShrink(dds, coef=2, type="ashr")
```

```
## 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
```

```
# resNorm <- lfcShrink(dds, coef=2, type="normal") TOO Slow
```

```
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") too Slow
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.

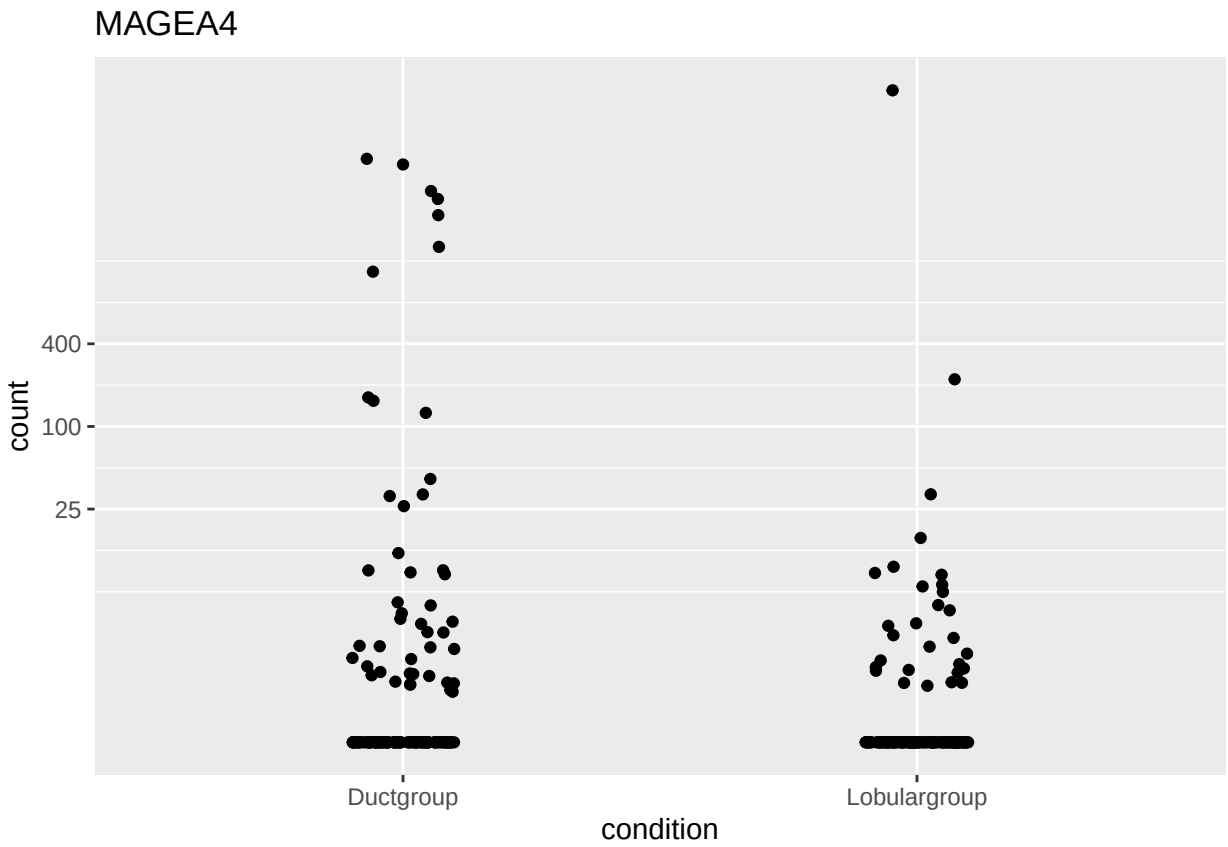
Here I sepcify the gene `MAGEA4` which is found Down-Regulataed

In IPA analysis, this gene is in the Pathway of disease "HER2 non-overexpressing breast carcinoma"

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

MAGEA4

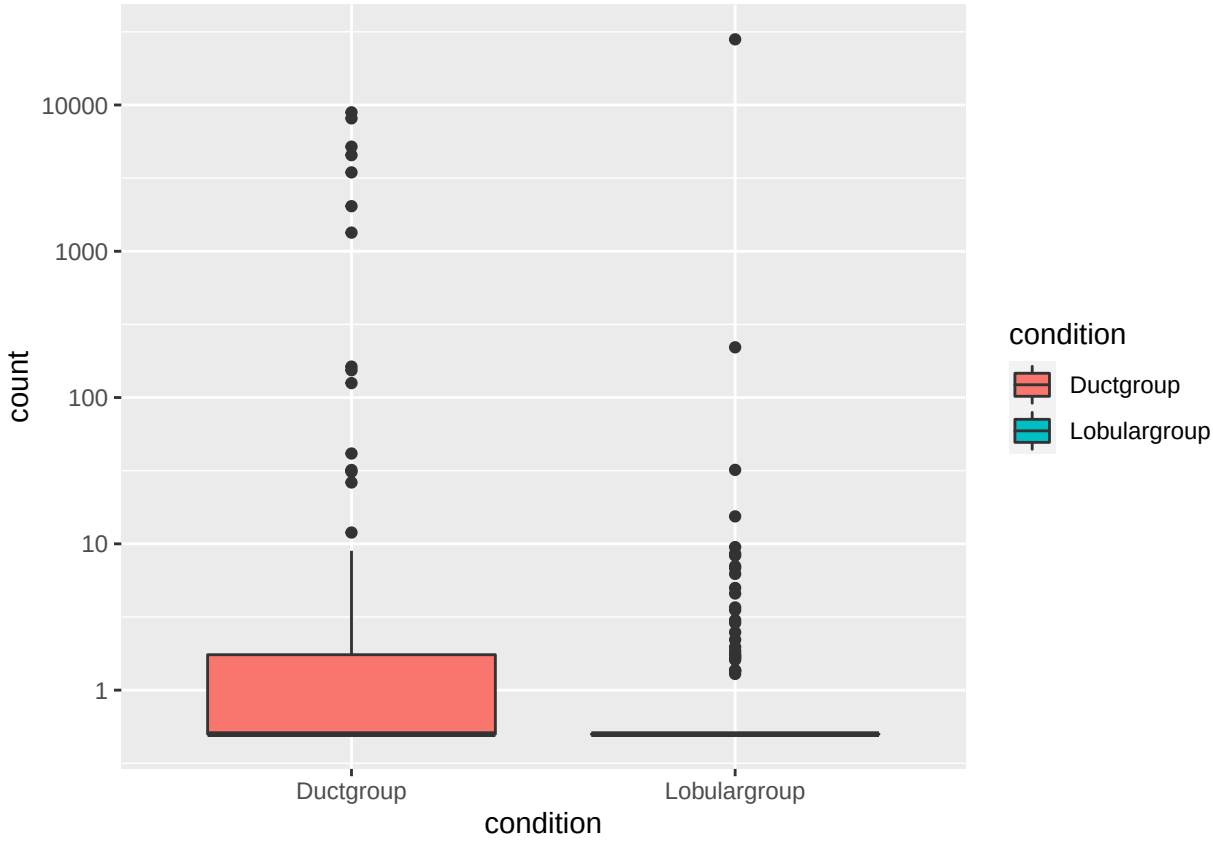
```
d <- plotCounts(dds, gene="ENSG00000147381.10", intgroup="condition",
                returnData=TRUE)
library("ggplot2")
ggplot(d, aes(x=condition, y=count)) + ggtitle("MAGEA4")+
  geom_point(position=position_jitter(w=0.1,h=0)) +
  scale_y_log10(breaks=c(25,100,400))
```



BOX plot

```
d1 <- plotCounts(dds,gene="ENSG00000147381.10", intgroup="condition",returnData = T)

ggplot(d1,aes(condition, count)) + geom_boxplot(aes(fill=condition)) + scale_y_log10()
```

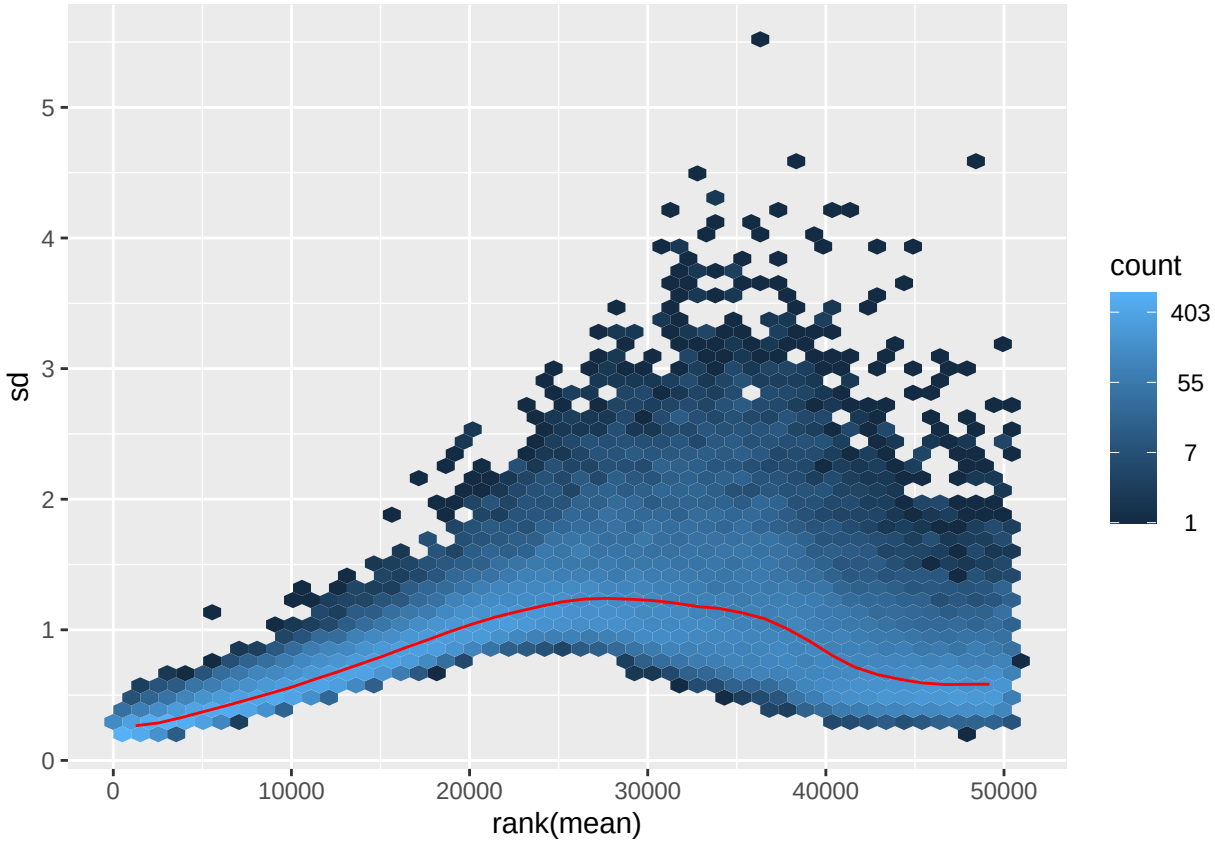


Count data Transformation

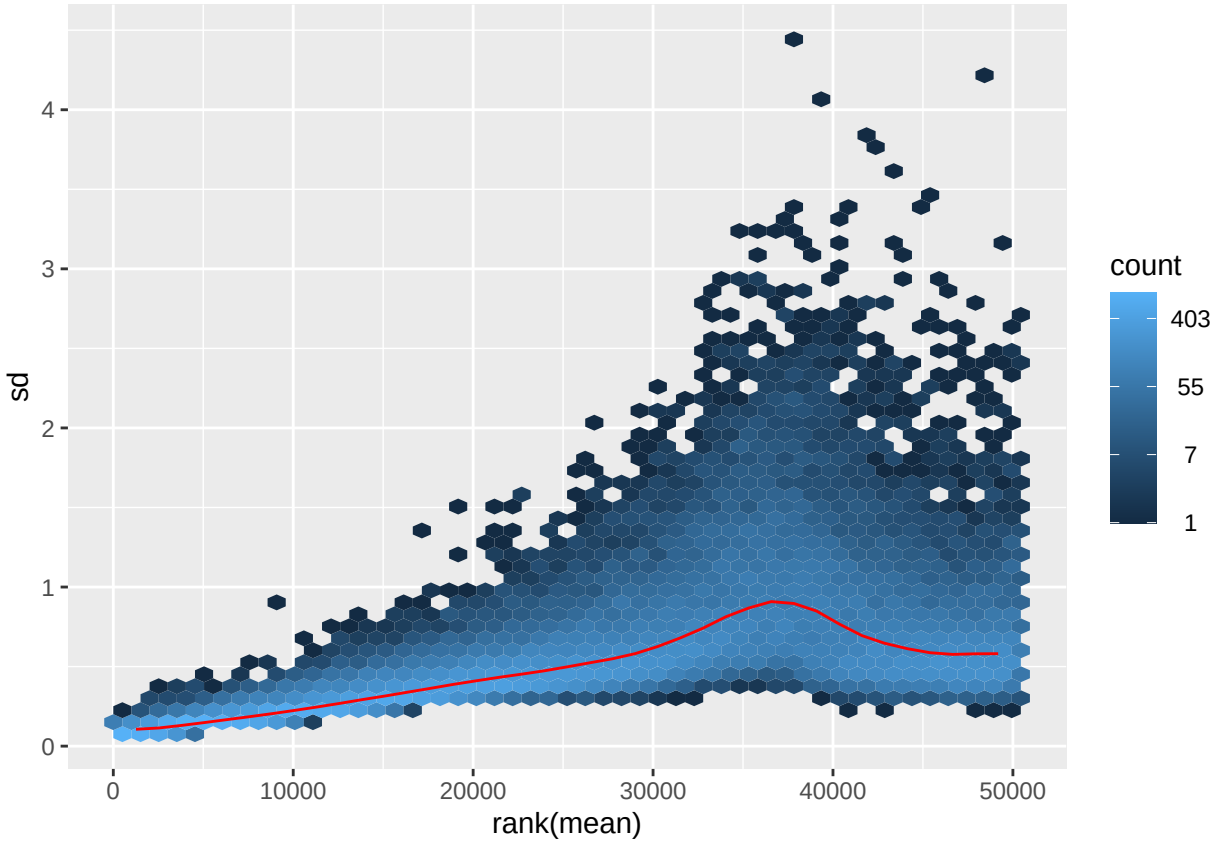
```
vsd <- vst(dds, blind=FALSE)
```

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

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




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



Data transformations and visualization (Use ddsMF)

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

Usually use `vsd` if the number of samples > 50

```
vsdMF <- vst(ddsMF, blind=FALSE)
```

```
ntdMF <- normTransform(ddsMF)
```

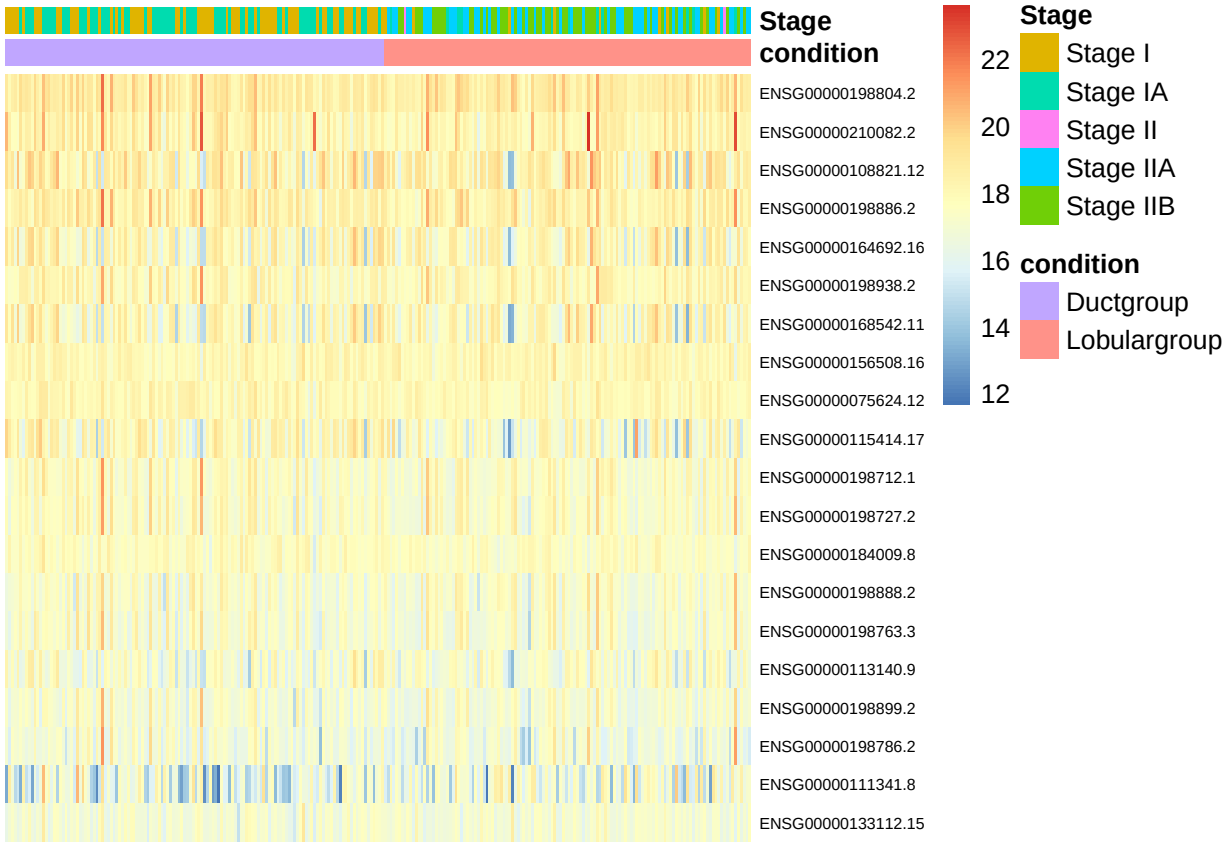
```
df <- sampleTableselect %>%
  dplyr::select(fileName,condition,Stage)
```

```
rownames(df)<-df[,1]
 df <- df[,-1]
```

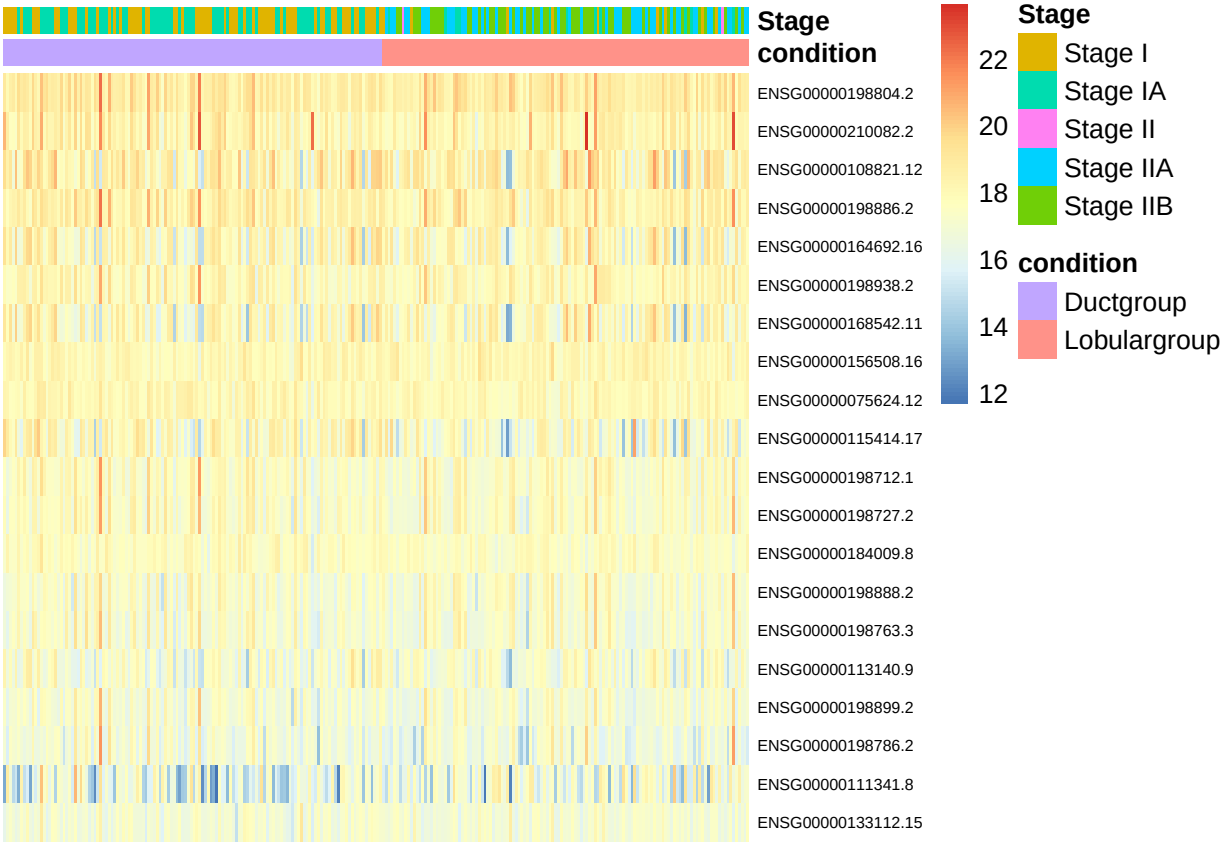
Heatmap

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

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




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



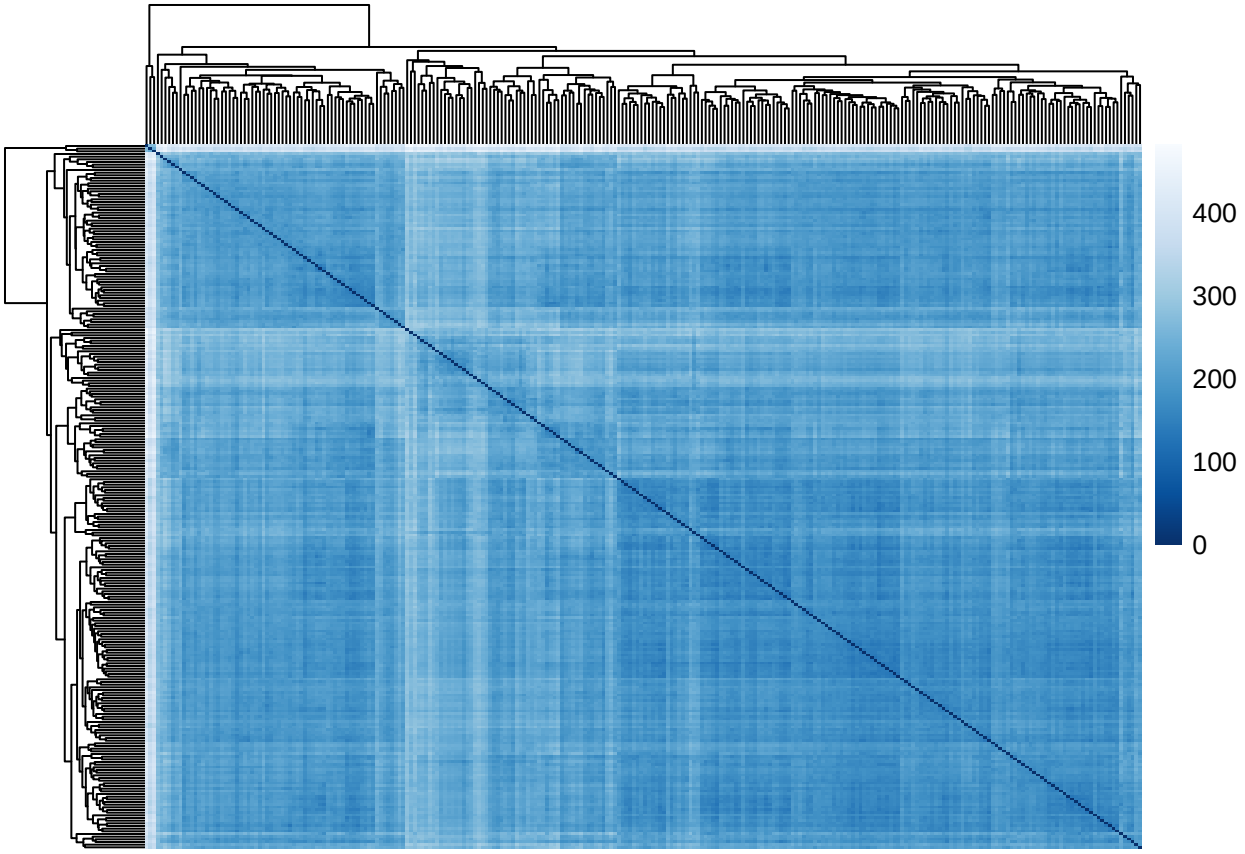
Heatmap of the sample-to-sample distances

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

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

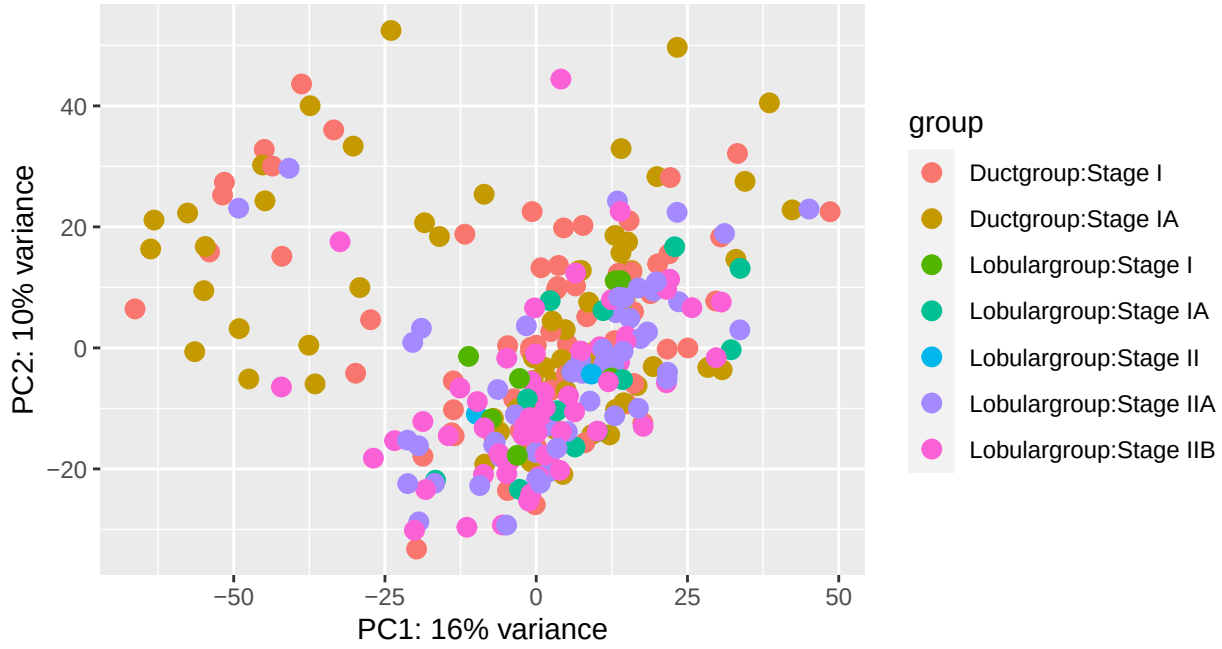
```
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsdMF$condition, vsdMF$type,vsdMF$Stage, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
```

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



PCA plot

```
plotPCA(vsdMF, intgroup=c("condition","Stage"))
```



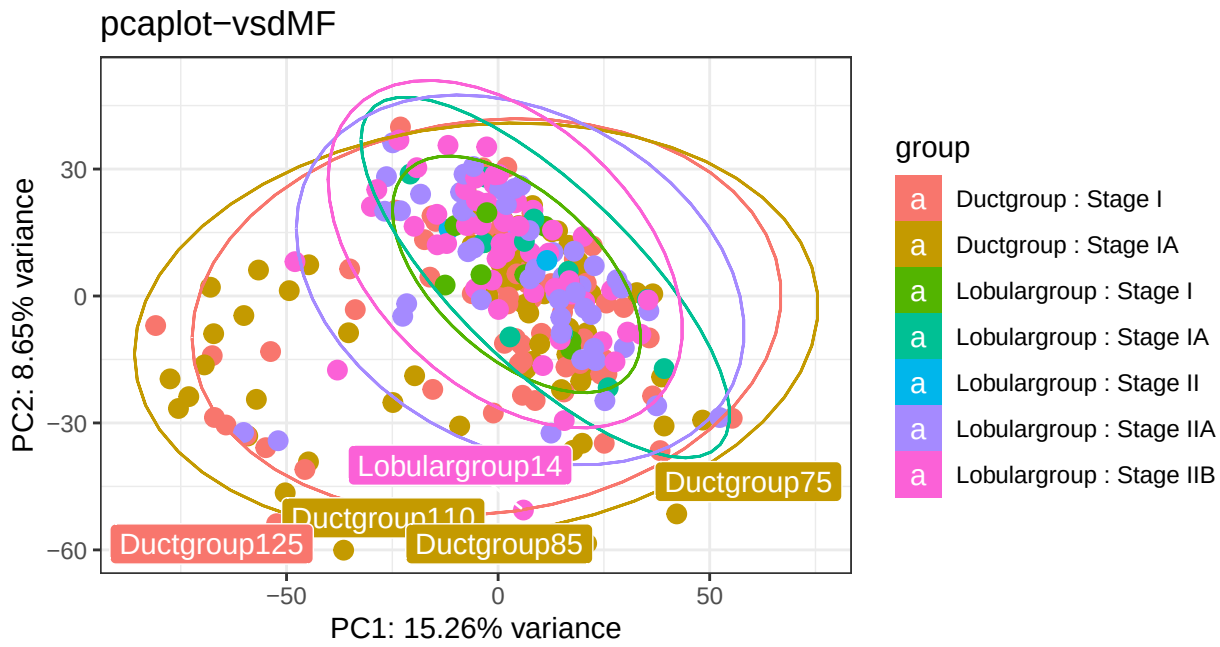

```
library(pcaExplorer)
```

```
##
```

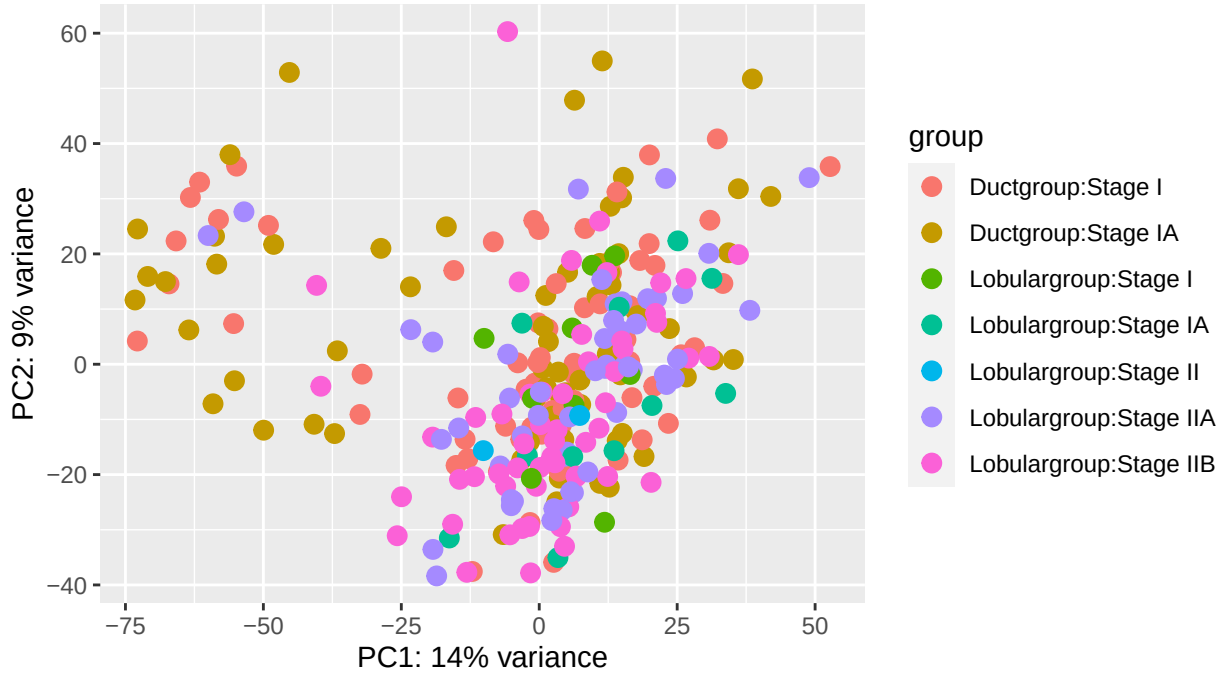
```
## Welcome to pcaExplorer v2.20.0
##
## If you use pcaExplorer in your work, please cite:
##
## Federico Marini, Harald Binder
## pcaExplorer: an R/Bioconductor package for interacting with RNA-seq principal components
## BMC Bioinformatics, 2019 - https://doi.org/10.1186/s12859-019-2879-1
```

```
pcaplot(vsdMF,intgroup = c("condition","Stage"),ntop = 1000,
        pcX = 1, pcY = 2, title = "pcaplot-vsdMF",
        ellipse = TRUE)
```

```
## Warning: ggrepel: 259 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

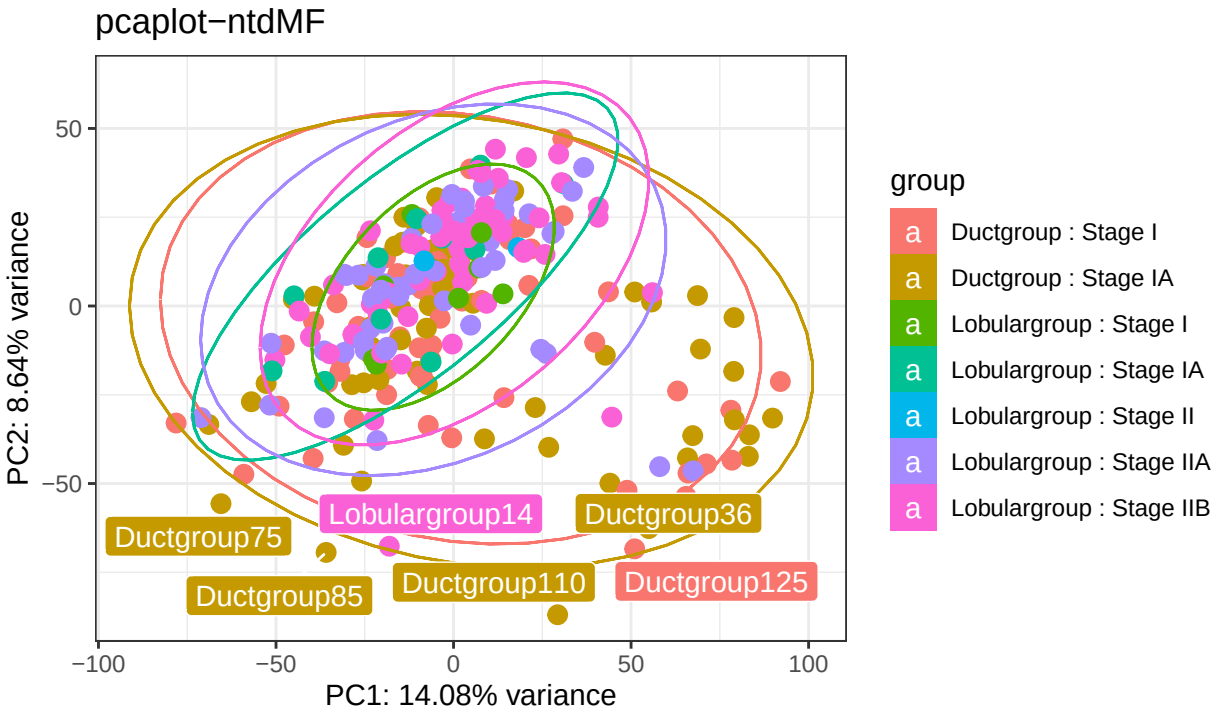



```
plotPCA(ntdMF, intgroup=c("condition","Stage"))
```




```
pcaplot(ntdMF,intgroup = c("condition","Stage"),ntop = 1000,
        pcX = 1, pcY = 2, title = "pcaplot-ntdMF",
        ellipse = TRUE)
```

```
## Warning: ggrepel: 258 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


pcaplot−ntdMF
PC2: 8.64% variance
PC1: 14.08% variance
50
0
−50
−100
−50
0
50
100
Ductgroup75
Lobulargroup14
Ductgroup36
Ductgroup85
Ductgroup110
Ductgroup125
group
a
Ductgroup : Stage I
a
Ductgroup : Stage IA
a
Lobulargroup : Stage I
a
Lobulargroup : Stage IA
a
Lobulargroup : Stage II
a
Lobulargroup : Stage IIA
a
Lobulargroup : Stage IIB