

# Untitled

Mthande\_Fezile

8/14/2020

Importing our data

```
sample= read.table("D:/Matrix/counts1.Rmatrix.txt", header= TRUE, sep="\t", row.names = 1)
```

renaming the headers of the file

```
colnames(sample) <- gsub("Bam", "", colnames(sample))
colnames(sample) <- gsub(".sorted.bam", "", colnames(sample))
colnames(sample) <- gsub("\\.", "", colnames(sample))
head(sample)
```

```
##           sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972      2       0       0       0       0       0
## ENSG00000227232     70      58     210      65      66      72
## ENSG00000278267      7      11      36       2       2       4
## ENSG00000243485      0       0       0       1       0       0
## ENSG00000284332      0       0       0       0       0       0
## ENSG00000237613      0       0       0       1       1       0
```

importing the metadata

```
meta= read.table("D:/Matrix/practice.dataset.metadata.tsv", header=TRUE, sep="\t", row.names = 1)
head(meta)
```

```
##           Condition
## sample37    normal
## sample38    normal
## sample39    normal
## sample40   disease
## sample41   disease
## sample42   disease
```

```
meta$Condition <- factor(meta$Condition)
```

confirming if all the colnames correspond

```
all(rownames(meta) %in% colnames(sample))
```

```
## [1] TRUE
```

```
all(rownames(meta) == colnames(sample))

## [1] TRUE

creating the dds

dds= DESeqDataSetFromMatrix(countData = sample, colData = meta, design = ~ Condition)

dds

## class: DESeqDataSet
## dim: 60683 6
## metadata(1): version
## assays(1): counts
## rownames(60683): ENSG00000223972 ENSG00000227232 ... ENSG00000277475
##   ENSG00000268674
## rowData names(0):
## colnames(6): sample37 sample38 ... sample41 sample42
## colData names(1): Condition
```

combining

```
info <- as.data.frame( colData( dds))
head(info)

##           Condition
## sample37      normal
## sample38      normal
## sample39      normal
## sample40    disease
## sample41    disease
## sample42    disease
```

Filtering rows with low reads( $\geq 10$ )

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

## class: DESeqDataSet
## dim: 28146 6
## metadata(1): version
## assays(1): counts
## rownames(28146): ENSG00000227232 ENSG00000278267 ... ENSG00000271254
##   ENSG00000277475
## rowData names(0):
## colnames(6): sample37 sample38 ... sample41 sample42
## colData names(1): Condition
```

specifying the levels

```
dds$Condition <- factor(dds$Condition, levels = c("disease", "normal"))
```

running deseq

```
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): Condition normal vs disease
```

```
## Wald test p-value: Condition normal vs disease
```

```
## DataFrame with 28146 rows and 6 columns
```

```
##           baseMean log2FoldChange      lfcSE       stat     pvalue
##           <numeric>     <numeric> <numeric> <numeric> <numeric>
## ENSG00000227232    83.90039     0.740435  0.486531  1.521867 0.12804239
## ENSG00000278267     9.14536     2.729447  1.055116  2.586869 0.00968525
## ENSG00000238009     2.37385     1.424879  3.109428  0.458245 0.64677669
## ENSG00000233750     9.03780     2.031699  1.314490  1.545617 0.12219702
## ENSG00000268903    24.70144     1.669259  0.971367  1.718463 0.08571219
## ...
##           ...          ...        ...        ...        ...
## ENSG00000278384    21.90520     0.377467  1.204880  0.313282 0.7540668
## ENSG00000276345    62.45027     1.381980  0.684952  2.017632 0.0436296
## ENSG00000275063    1.53714    -2.749421  2.538575 -1.083057 0.2787832
## ENSG00000271254   202.97804    -0.641319  0.481434 -1.332101 0.1828270
## ENSG00000277475    1.83458     4.442179  3.933014  1.129459 0.2587042
##           padj
##           <numeric>
## ENSG00000227232    0.3859254
## ENSG00000278267    0.0843189
## ENSG00000238009      NA
## ENSG00000233750    0.3771705
## ENSG00000268903    0.3102699
## ...
##           ...
## ENSG00000278384    0.899573
## ENSG00000276345    0.212869
## ENSG00000275063      NA
## ENSG00000271254    0.462395
## ENSG00000277475      NA
```

specifying comparison of interest using contrast

```

res <- results(dds, contrast=c("Condition","normal","disease"))
res

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 28146 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
## ENSG00000227232   83.90039    0.740435  0.486531  1.521867 0.12804239
## ENSG00000278267   9.14536    2.729447  1.055116  2.586869 0.00968525
## ENSG00000238009   2.37385    1.424879  3.109428  0.458245 0.64677669
## ENSG00000233750   9.03780    2.031699  1.314490  1.545617 0.12219702
## ENSG00000268903  24.70144    1.669259  0.971367  1.718463 0.08571219
## ...
##           ...
## ENSG00000278384  21.90520    0.377467  1.204880  0.313282 0.7540668
## ENSG00000276345  62.45027    1.381980  0.684952  2.017632 0.0436296
## ENSG00000275063  1.53714    -2.749421  2.538575 -1.083057 0.2787832
## ENSG00000271254 202.97804   -0.641319  0.481434 -1.332101 0.1828270
## ENSG00000277475  1.83458    4.442179  3.933014  1.129459 0.2587042
##           padj
##           <numeric>
## ENSG00000227232  0.3859254
## ENSG00000278267  0.0843189
## ENSG00000238009   NA
## ENSG00000233750  0.3771705
## ENSG00000268903  0.3102699
## ...
##           ...
## ENSG00000278384  0.899573
## ENSG00000276345  0.212869
## ENSG00000275063   NA
## ENSG00000271254  0.462395
## ENSG00000277475   NA

```

shrinkage and ranking of genes

```

library("apeglm")
resLFC <- lfcShrink(dds, coef="Condition_normal_vs_disease", 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 normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 28146 rows and 5 columns
##           baseMean log2FoldChange      lfcSE      pvalue     padj
## ENSG00000227232   83.90039    0.4668352  0.432686  0.12804239 0.3859254
## ENSG00000278267   9.14536    1.6130742  1.434049  0.00968525 0.0843189

```

```

## ENSG00000238009 2.37385 0.0399322 0.536663 0.64677669 NA
## ENSG00000233750 9.03780 0.3173183 0.629569 0.12219702 0.3771705
## ENSG00000268903 24.70144 0.4996786 0.725420 0.08571219 0.3102699
## ... ...
## ENSG00000278384 21.90520 0.0645867 0.498664 0.7540668 0.899573
## ENSG00000276345 62.45027 0.7658131 0.697028 0.0436296 0.212869
## ENSG00000275063 1.53714 -0.0995452 0.545751 0.2787832 NA
## ENSG00000271254 202.97804 -0.3930151 0.412986 0.1828270 0.462395
## ENSG00000277475 1.83458 0.0587682 0.545442 0.2587042 NA

```

ordering according to p-value

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

genes with the smalles pvalues

```
idx <- which.min(res$pvalue)
counts(dds)[idx, ]
```

```

## sample37 sample38 sample39 sample40 sample41 sample42
## 0 12 10 1200 511 1144

```

summarizing

```
summary(res)
```

```

##
## out of 28146 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up) : 1334, 4.7%
## LFC < 0 (down) : 1778, 6.3%
## outliers [1] : 289, 1%
## low counts [2] : 3820, 14%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

p-values less than 0.1/10% acceptable(FDR)

```
sum(res$padj < 0.1, na.rm=TRUE)
```

```
## [1] 3112
```

setting alpha

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

```

## 
## out of 28146 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 826, 2.9%
## LFC < 0 (down)    : 1223, 4.3%
## outliers [1]       : 289, 1%
## low counts [2]     : 4366, 16%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

sum(res05$padj < 0.05, na.rm=TRUE)

```

## [1] 2049

## INDEPENDENT HYPOTHESIS TESTING

```
library(IHW)
```

```
resIHW <- results(dds, filterFun=ihw)
summary(resIHW)
```

```

## 
## out of 28146 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1385, 4.9%
## LFC < 0 (down)    : 1794, 6.4%
## outliers [1]       : 289, 1%
## [1] see 'cooksCutoff' argument of ?results
## see metadata(res)$ihwResult on hypothesis weighting
```

```
sum(resIHW$padj < 0.1, na.rm=TRUE)
```

## [1] 3179

```
metadata(resIHW)$ihwResult
```

```

## ihwResult object with 28146 hypothesis tests
## Nominal FDR control level: 0.1
## Split into 18 bins, based on an ordinal covariate
```

genes with high upregulation

```
resSig <- subset(res, res$padj < 0.1 )
head( resSig[ order( -resSig$log2FoldChange ), ],10)
```

```

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 10 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue

```

```

## <numeric> <numeric> <numeric> <numeric> <numeric>
## ENSG00000248400 16.3679 20.93324 3.34497 6.25813 3.89628e-10
## ENSG00000257464 17.6559 7.68939 2.50049 3.07516 2.10390e-03
## ENSG00000270307 15.7386 7.52365 2.82006 2.66791 7.63256e-03
## ENSG00000231698 14.1635 7.37770 2.38187 3.09745 1.95196e-03
## ENSG00000105467 12.4239 7.18121 2.63895 2.72124 6.50373e-03
## ENSG00000238001 43.4823 6.97165 2.16199 3.22465 1.26128e-03
## ENSG00000154415 9.6672 6.81401 1.95235 3.49015 4.82742e-04
## ENSG00000258770 17.5600 6.70611 2.27994 2.94135 3.26785e-03
## ENSG00000281327 150.2967 6.49779 1.73355 3.74826 1.78064e-04
## ENSG00000136688 14.8737 6.46684 2.24240 2.88389 3.92791e-03
## padj
## <numeric>
## ENSG00000248400 1.37728e-07
## ENSG00000257464 3.18059e-02
## ENSG00000270307 7.25727e-02
## ENSG00000231698 3.01344e-02
## ENSG00000105467 6.56022e-02
## ENSG00000238001 2.22364e-02
## ENSG00000154415 1.13206e-02
## ENSG00000258770 4.22535e-02
## ENSG00000281327 5.52275e-03
## ENSG00000136688 4.75883e-02

```

genes with down regulation

```
head( resSig[ order( resSig$log2FoldChange ), ],10)
```

```

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 10 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
## <numeric>     <numeric> <numeric> <numeric> <numeric>
## ENSG00000168658 330.554    -11.7399  1.41201  -8.31434 9.22613e-17
## ENSG00000231738 240.840    -11.2831  1.48397  -7.60334 2.88587e-14
## ENSG00000169064 216.082    -11.1267  1.37649  -8.08343 6.29677e-16
## ENSG00000091181 178.718    -10.8531  1.32556  -8.18759 2.66503e-16
## ENSG00000153789 143.718    -10.5386  1.35477  -7.77892 7.31492e-15
## ENSG00000176601 139.470    -10.4951  1.51190  -6.94164 3.87561e-12
## ENSG00000182329 133.647    -10.4335  1.56155  -6.68155 2.36420e-11
## ENSG00000152611 129.380    -10.3870  1.37670  -7.54481 4.52957e-14
## ENSG00000176029 113.750    -10.2012  1.39987  -7.28726 3.16315e-13
## ENSG00000230873 102.302    -10.0482  1.36401  -7.36667 1.74947e-13
## padj
## <numeric>
## ENSG00000168658 1.84807e-13
## ENSG00000231738 3.46838e-11
## ENSG00000169064 1.08111e-12
## ENSG00000091181 4.92764e-13
## ENSG00000153789 9.76826e-12
## ENSG00000176601 2.38866e-09
## ENSG00000182329 1.13657e-08
## ENSG00000152611 4.94897e-11

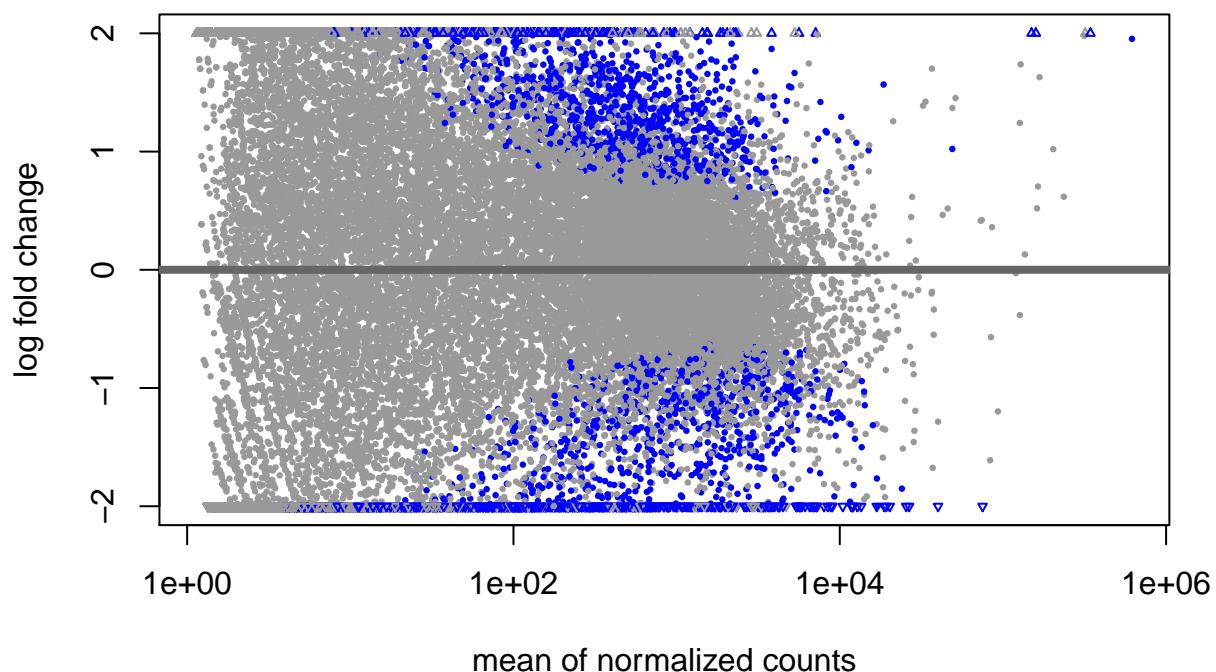
```

```
## ENSG00000176029 2.83691e-10  
## ENSG00000230873 1.75217e-10
```

PLOTS

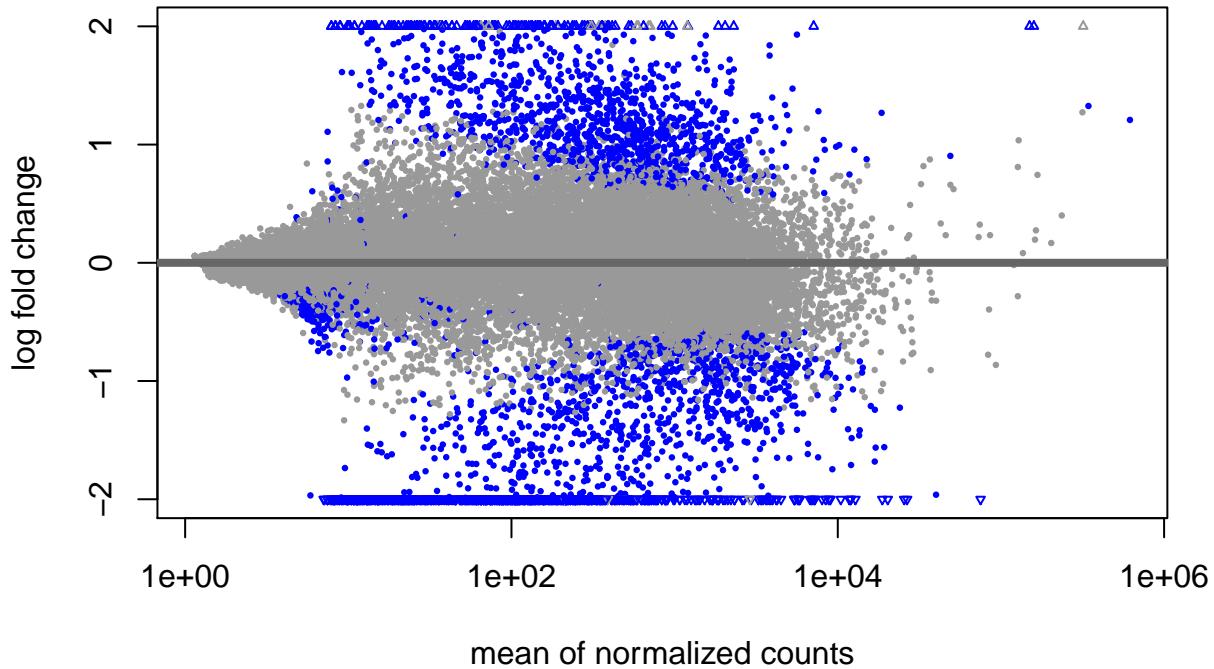
#EXPLORATION AND EXPORTING DATA 1.MA-PLOTS

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



SHRUNKEN FOLD CHAGES

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



```
#IDENTIFYING THE NUMBER OF GENES #idx <- identify(resbaseMean, reslog2FoldChange) #rownames(res)[idx]
```

```
SPECIFYING THE COEFFICIENT 2
```

```
resultsNames(dds)
```

```
## [1] "Intercept" "Condition_normal_vs_disease"
```

```
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='normal'.
```

```
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
```

```
## Reference: https://doi.org/10.1093/bioinformatics/bty895
```

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

```
## using 'ashr' for LFC shrinkage. If used in published research, please cite:
```

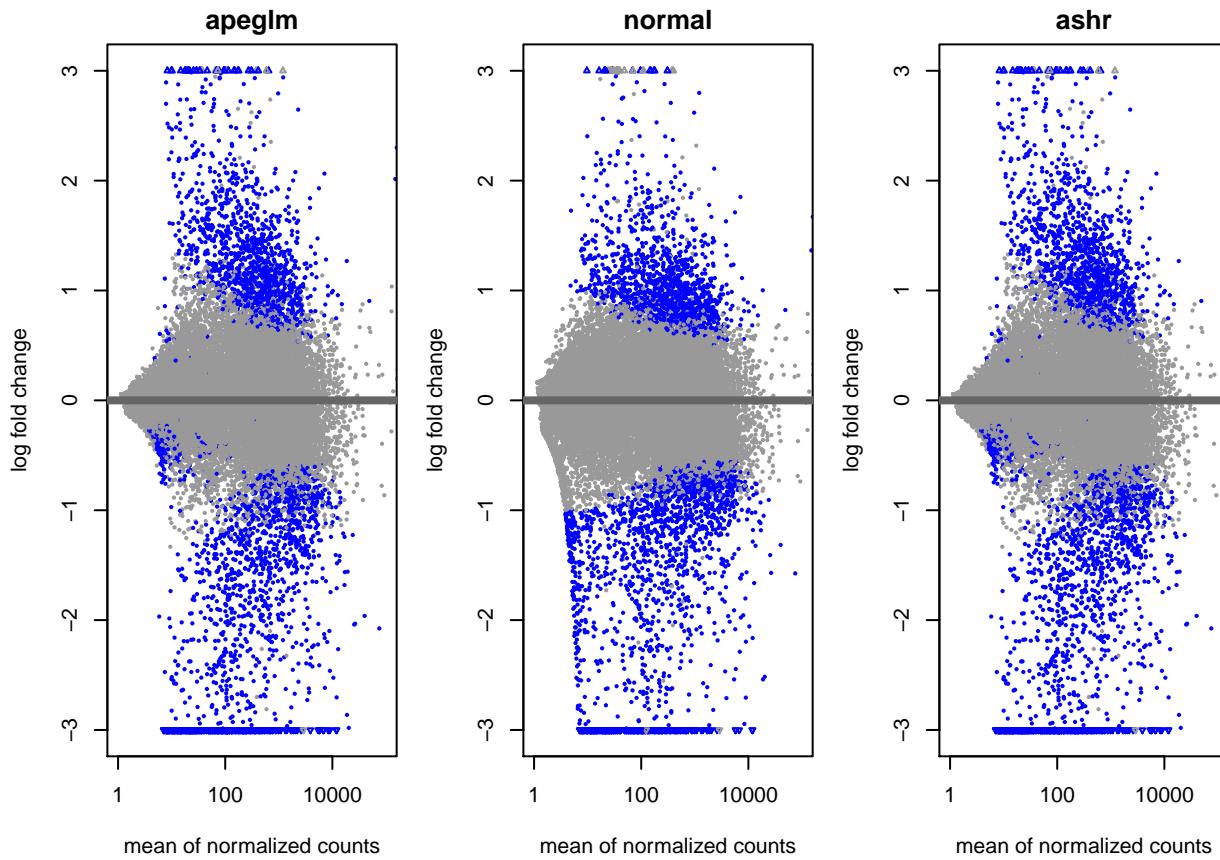
```
## Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
```

```
## https://doi.org/10.1093/biostatistics/kxw041
```

```
library("ashr")
```

the plots

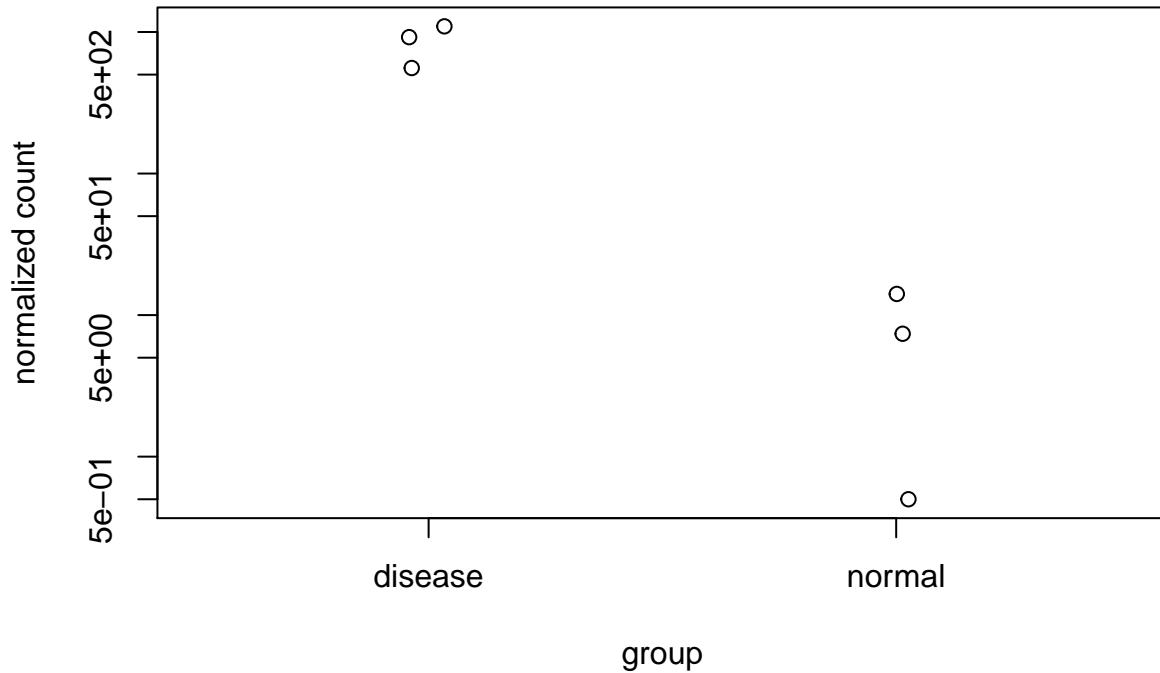
```
par(mfrow=c(1,3), mar=c(4,4,2,1))
xlim <- c(1,1e5); ylim <- c(-3,3)
plotMA(resLFC, xlim=xlim, ylim=ylim, main="apeglm")
plotMA(resNorm, xlim=xlim, ylim=ylim, main="normal")
plotMA(resLFC, xlim=xlim, ylim=ylim, main="ashr")
```



PLOT COUNTS

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

## ENSG00000039537



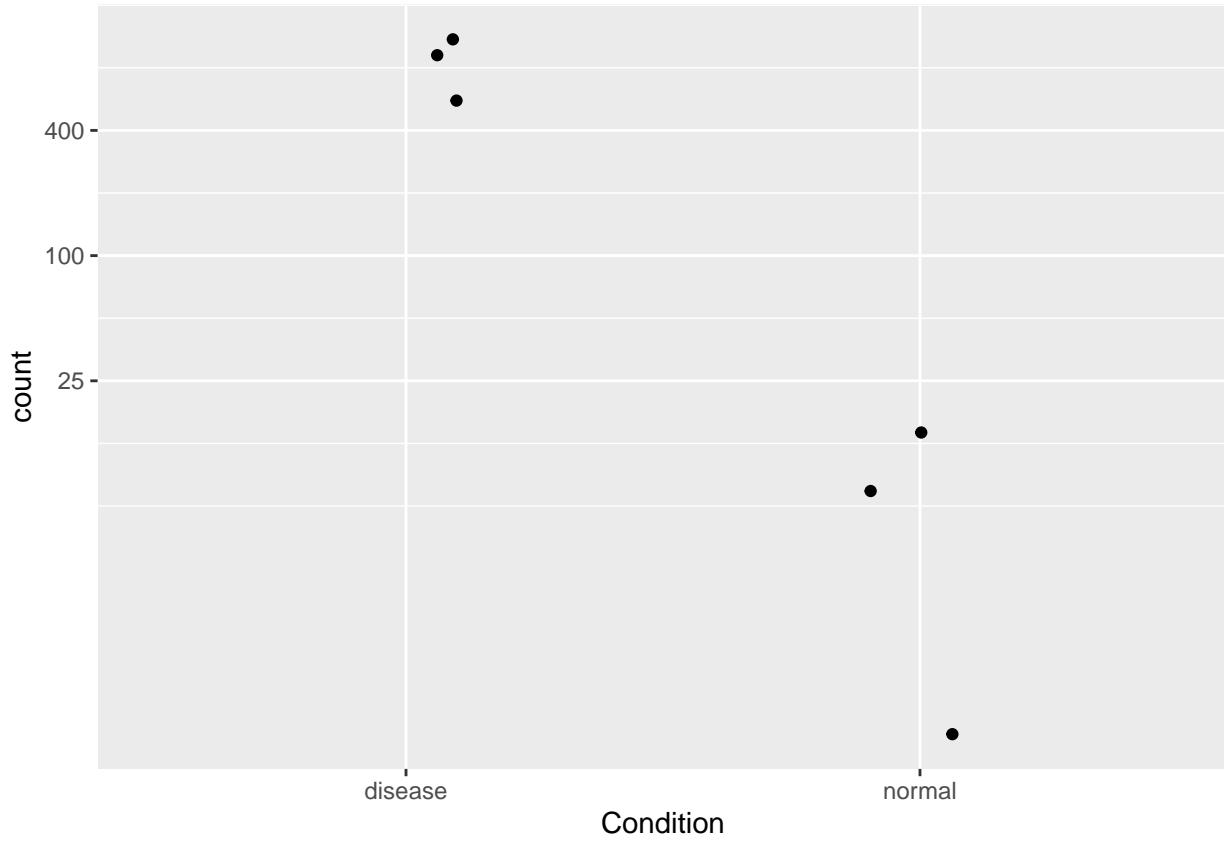
```
#using ggplot

d <- plotCounts(dds, gene=which.min(res$padj), intgroup="Condition",
                 returnData=TRUE)
library("ggplot2")

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:IHW':
##      alpha

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))
```



```
#results columns
```

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

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

## 2.VOLCANO PLOT

```
reset par
```

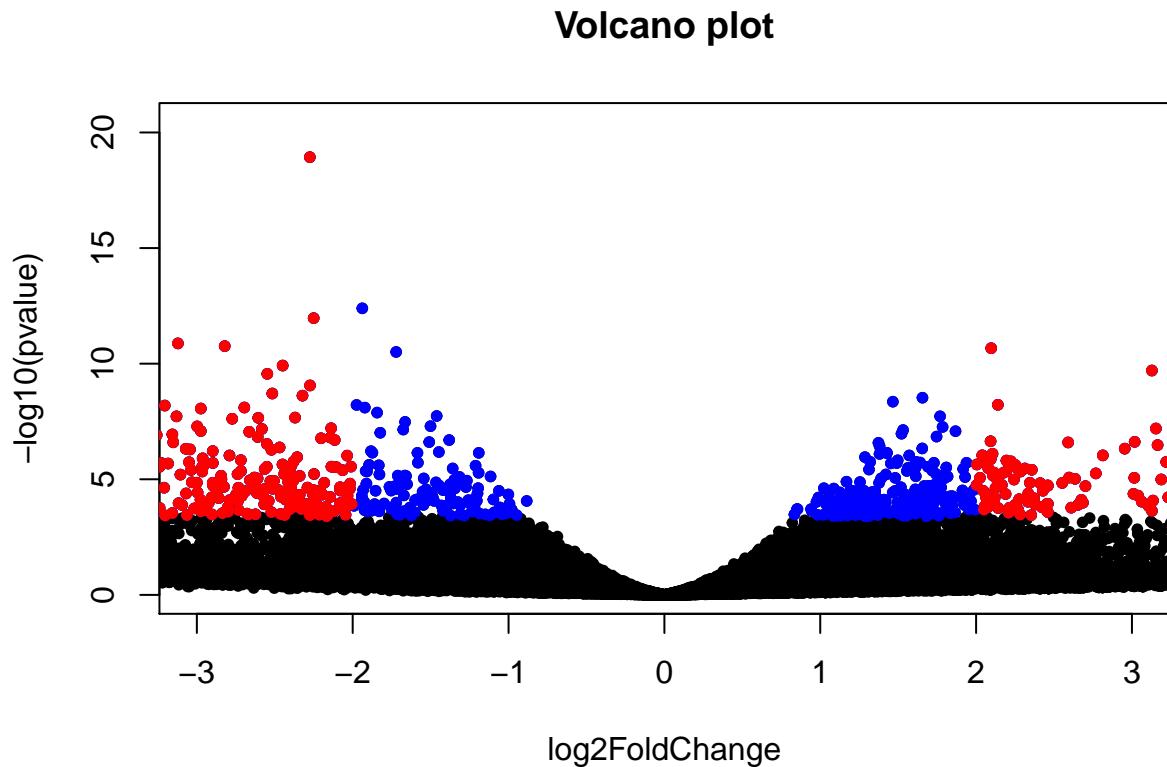
```
par(mfrow=c(1,1))
```

```
#Make a basic volcano plot
```

```
with(res, plot(log2FoldChange, -log10(pvalue), pch=20, main="Volcano plot", xlim=c(-3,3)))
```

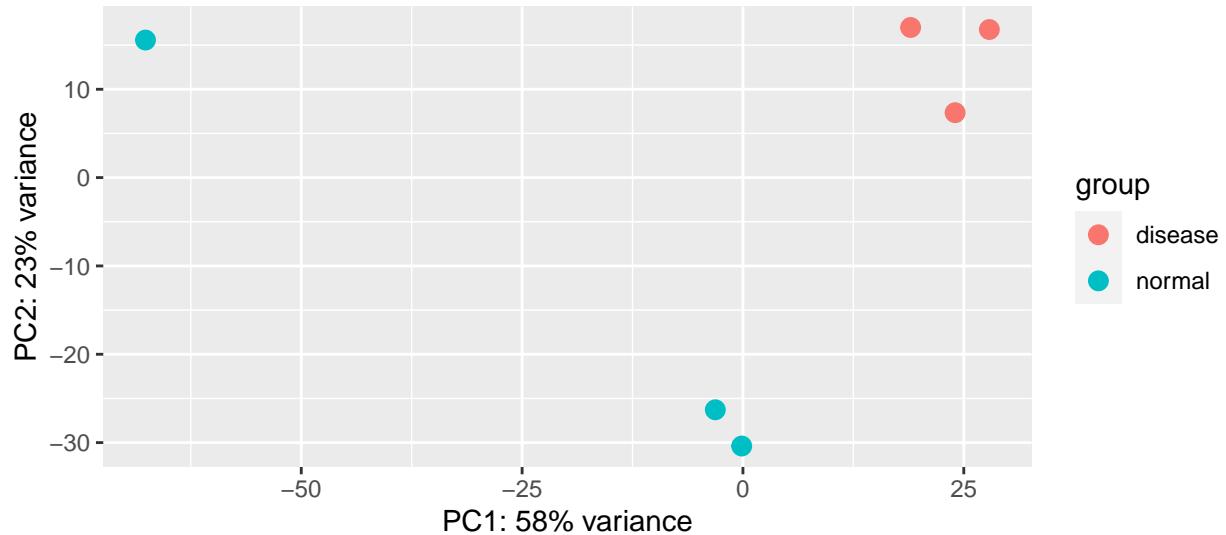
```
#Add colored points: blue if padj<0.01, red if log2FC>1 and padj<0.05)
```

```
with(subset(res, padj<.01 ), points(log2FoldChange, -log10(pvalue), pch=20, col="blue"))
with(subset(res, padj<.01 & abs(log2FoldChange)>2), points(log2FoldChange, -log10(pvalue), pch=20, col=
```



PCA

```
vsdata <- vst(dds, blind=FALSE)
plotPCA(vsdata, intgroup="Condition") #using the DESEQ2 plotPCA fxn we can
```



ANNOTATION

ADDING GENE NAMES

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

```
## Loading required package: AnnotationDbi
```

```
##
```

get the list of key types

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

```
## [1] "ACCNUM"      "ALIAS"        "ENSEMBL"       "ENSEMLPROT"    "ENSEMLTRANS"
## [6] "ENTREZID"     "ENZYME"       "EVIDENCE"      "EVIDENCEALL"   "GENENAME"
## [11] "GO"           "GOALL"        "IPI"          "MAP"          "OMIM"
## [16] "ONTOLOGY"     "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"       "UNIGENE"
## [26] "UNIPROT"
```

easy function for conversion

```

convertIDs <- function( ids, fromKey, toKey, db, ifMultiple=c( "putNA", "useFirst" ) ) {
  stopifnot( inherits( db, "AnnotationDb" ) )
  ifMultiple <- match.arg( ifMultiple )
  suppressWarnings( selRes <- AnnotationDbi::select(
    db, keys=ids, keytype=fromKey, columns=c(fromKey,toKey) ) )
  if( ifMultiple == "putNA" ) {
    duplicatedIds <- selRes[ duplicated( selRes[,1] ), 1 ]
    selRes <- selRes[ ! selRes[,1] %in% duplicatedIds, ] }
  return( selRes[ match( ids, selRes[,1] ), 2 ] )
}

```

converting the ideas into the names and symbols founf in the Annotation database

```
res$hgnc_symbol <- convertIDs( row.names(res), "ENSEMBL", "SYMBOL", org.Hs.eg.db )
```

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

```
res$entrezid <- convertIDs( row.names(res), "ENSEMBL", "ENTREZID", org.Hs.eg.db )
```

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

```
tail(res)
```

```

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat     pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000277196 193.61312      1.078092  0.401980  2.681955 0.00731934
## ENSG00000278384 21.90520      0.377467  1.204880  0.313282 0.75406676
## ENSG00000276345 62.45027      1.381980  0.684952  2.017632 0.04362960
## ENSG00000275063  1.53714     -2.749421  2.538575 -1.083057 0.27878323
## ENSG00000271254 202.97804     -0.641319  0.481434 -1.332101 0.18282695
## ENSG00000277475  1.83458      4.442179  3.933014  1.129459 0.25870424
##           padj hgnc_symbol      entrezid
##           <numeric> <character> <character>
## ENSG00000277196  0.070671 LOC102724788  102724788
## ENSG00000278384  0.899573          NA        NA
## ENSG00000276345  0.212869 LOC107987373  107987373
## ENSG00000275063            NA LOC102723407  102723407
## ENSG00000271254  0.462395 LOC102724250  102724250
## ENSG00000277475            NA          NA        NA

```

```
resSig$hgnc_symbol <- convertIDs( row.names(resSig), "ENSEMBL", "SYMBOL", org.Hs.eg.db )
```

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

```
tail(resSig)
```

```

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 7 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000210151    178.771      4.482046  0.933441  4.80164 1.57373e-06
## ENSG00000198712   159150.185     2.817353  0.822561  3.42510 6.14573e-04
## ENSG00000198840   11791.203      0.868517  0.300274  2.89241 3.82293e-03
## ENSG00000198886   343799.041      2.099021  0.837643  2.50587 1.22152e-02
## ENSG00000210194    21.674       3.706646  0.793641  4.67043 3.00565e-06
## ENSG00000277196   193.613      1.078092  0.401980  2.68195 7.31934e-03
##           padj hgnc_symbol
##           <numeric> <character>
## ENSG00000210151  0.000163757        NA
## ENSG00000198712  0.013565193      COX2
## ENSG00000198840  0.046621897      ND3
## ENSG00000198886  0.096427635      ND4
## ENSG00000210194  0.000259881        NA
## ENSG00000277196  0.070670958 LOC102724788

```

#### GENES THAT ARE UP REGULATED WITH THEIR NAMES AND SYMBOL

```

up_reg = head( resSig[ order( -resSig$log2FoldChange ), ], 10)
up_reg$hgnc_symbol <- convertIDs( row.names(up_reg), "ENSEMBL", "SYMBOL", org.Hs.eg.db )

```

```

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

```

```

up_reg$entrezid <- convertIDs( row.names(up_reg), "ENSEMBL", "ENTREZID", org.Hs.eg.db )

```

```

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

```

```

head(up_reg)

```

```

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000248400    16.3679      20.93324   3.34497   6.25813 3.89628e-10
## ENSG00000257464    17.6559      7.68939   2.50049   3.07516 2.10390e-03
## ENSG00000270307    15.7386      7.52365   2.82006   2.66791 7.63256e-03
## ENSG00000231698    14.1635      7.37770   2.38187   3.09745 1.95196e-03
## ENSG00000105467    12.4239      7.18121   2.63895   2.72124 6.50373e-03
## ENSG00000238001    43.4823      6.97165   2.16199   3.22465 1.26128e-03
##           padj hgnc_symbol      entrezid
##           <numeric> <character> <character>
## ENSG00000248400  1.37728e-07        NA        NA
## ENSG00000257464  3.18059e-02        NA        NA
## ENSG00000270307  7.25727e-02        NA        NA
## ENSG00000231698  3.01344e-02        NA        NA
## ENSG00000105467  6.56022e-02      SYNGR4    23546
## ENSG00000238001  2.22364e-02        NA        NA

```

Low regulated genes

```
down_reg = head( resSig[ order( -resSig$log2FoldChange ), ], 10)
down_reg$hgnc_symbol <- convertIDs( row.names(down_reg), "ENSEMBL", "SYMBOL", org.Hs.eg.db )

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

down_reg$entrezid <- convertIDs( row.names(down_reg), "ENSEMBL", "ENTREZID", org.Hs.eg.db )

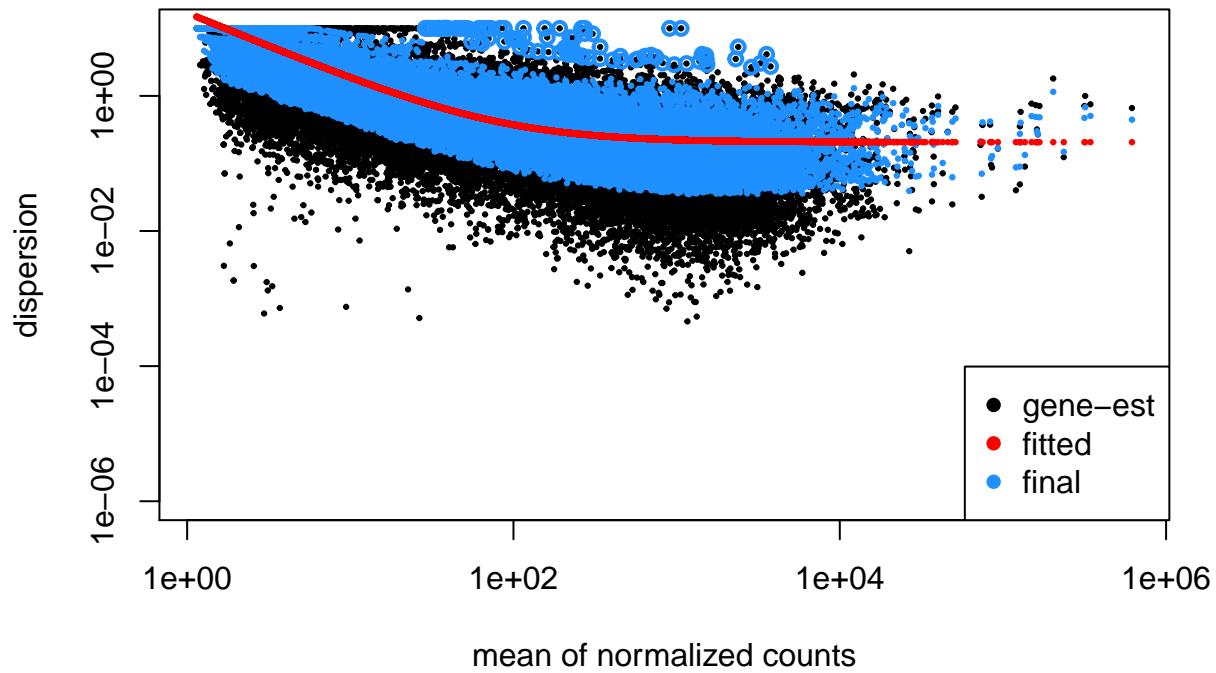
## 'select()' returned 1:1 mapping between keys and columns

head(down_reg)

## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>     <numeric> <numeric> <numeric>    <numeric>
## ENSG00000248400   16.3679    20.93324   3.34497   6.25813 3.89628e-10
## ENSG00000257464   17.6559    7.68939   2.50049   3.07516 2.10390e-03
## ENSG00000270307   15.7386    7.52365   2.82006   2.66791 7.63256e-03
## ENSG00000231698   14.1635    7.37770   2.38187   3.09745 1.95196e-03
## ENSG00000105467   12.4239    7.18121   2.63895   2.72124 6.50373e-03
## ENSG00000238001   43.4823    6.97165   2.16199   3.22465 1.26128e-03
##           padj hgnc_symbol      entrezid
##           <numeric> <character> <character>
## ENSG00000248400 1.37728e-07        NA       NA
## ENSG00000257464 3.18059e-02        NA       NA
## ENSG00000270307 7.25727e-02        NA       NA
## ENSG00000231698 3.01344e-02        NA       NA
## ENSG00000105467 6.56022e-02      SYNGR4    23546
## ENSG00000238001 2.22364e-02        NA       NA
```

Dispersion plot

```
plotDispEsts( dds, ylim = c(1e-6, 1e1) )
```



#### HISTOGRAM OF P -VALUES

```
hist( res$pvalue, breaks=20, col="red" )
```

**Histogram of res\$pvalue**

