Differential Expression Analysis Script for all the Outputs

Fredrick E. Kakembo & Ruth Nanjala

8/13/2020

```
setwd("~/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/")

#Loading the neccessary packages
library(DESeq2)
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## 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 object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## 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")'.

## Loading required package: DelayedArray

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

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

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
```

```
library(GenomicFeatures)
```

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

```
library("tximport")
library("readr")
```

Part 1: Importing data

1. Importing Output from hisat's FeatureCount

```
# HISAT

#Reading the metadata file
meta <- read.csv("practice.dataset.metadata.tsv", sep="\t", row.names = 1, header = T)
meta$Condition <- as.factor(meta$Condition)
meta
```

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

```
countdata <- read.csv("Hisat_featurecounts/hisat_counts.txt", sep="\t", header=T, row.names=1, comment.
head(countdata)
```

```
##                                  Chr
## ENSG00000223972     1;1;1;1;1;1;1;1;1
## ENSG00000227232 1;1;1;1;1;1;1;1;1;1;1
## ENSG00000278267                    1
## ENSG00000243485          1;1;1;1;1
## ENSG00000284332                    1
## ENSG00000237613          1;1;1;1;1
##                                                                                  Start
## ENSG00000223972             11869;12010;12179;12613;12613;12975;13221;13221;13453
## ENSG00000227232 14404;15005;15796;16607;16858;17233;17606;17915;18268;24738;29534
## ENSG00000278267                                                                17369
## ENSG00000243485                                      29554;30267;30564;30976;30976
## ENSG00000284332                                                                30366
## ENSG00000237613                                      34554;35245;35277;35721;35721
##                                                                                    End
## ENSG00000223972             12227;12057;12227;12721;12697;13052;13374;14409;13670
## ENSG00000227232 14501;15038;15947;16765;17055;17368;17742;18061;18366;24891;29570
## ENSG00000278267                                                                17436
## ENSG00000243485                                      30039;30667;30667;31109;31097
## ENSG00000284332                                                                30503
```

```
## ENSG00000237613                                    35174;35481;35481;36073;36081
##                              Strand Length sample37_hisat_sorted.bam
## ENSG00000223972     +;+;+;+;+;+;+;+;+;+   1735                         0
## ENSG00000227232 -;-;-;-;-;-;-;-;-;-;-   1351                        52
## ENSG00000278267                      -     68                         7
## ENSG00000243485             +;+;+;+;+;+   1021                         0
## ENSG00000284332                     +;+    138                         0
## ENSG00000237613             -;-;-;-;-;-   1219                         0
##                 sample38_hisat_sorted.bam sample39_hisat_sorted.bam
## ENSG00000223972                         0                         0
## ENSG00000227232                        48                       187
## ENSG00000278267                        11                        36
## ENSG00000243485                         0                         0
## ENSG00000284332                         0                         0
## ENSG00000237613                         0                         0
##                 sample40_hisat_sorted.bam sample41_hisat_sorted.bam
## ENSG00000223972                         0                         0
## ENSG00000227232                        56                        59
## ENSG00000278267                         2                         3
## ENSG00000243485                         1                         0
## ENSG00000284332                         0                         0
## ENSG00000237613                         1                         1
##                 sample42_hisat_sorted.bam
## ENSG00000223972                         0
## ENSG00000227232                        69
## ENSG00000278267                         4
## ENSG00000243485                         0
## ENSG00000284332                         0
## ENSG00000237613                         0
```

```
#Check the columns
colnames(countdata)
```

```
##  [1] "Chr"                         "Start"
##  [3] "End"                         "Strand"
##  [5] "Length"                      "sample37_hisat_sorted.bam"
##  [7] "sample38_hisat_sorted.bam" "sample39_hisat_sorted.bam"
##  [9] "sample40_hisat_sorted.bam" "sample41_hisat_sorted.bam"
## [11] "sample42_hisat_sorted.bam"
```

```
#Remove the unwanted columns
countdata[c("Chr", "Start", "End", "Strand", "Length")] <- NULL

#Renaming the Colnames (to remove the suffix "_hisat_sorted.bam")
colnames(countdata) <- gsub("_hisat_sorted.bam", "", colnames(countdata))
head(countdata)
```

```
##                 sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972        0        0        0        0        0        0
## ENSG00000227232       52       48      187       56       59       69
## ENSG00000278267        7       11       36        2        3        4
## ENSG00000243485        0        0        0        1        0        0
## ENSG00000284332        0        0        0        0        0        0
## ENSG00000237613        0        0        0        1        1        0
```

```
#Check if samples in the countdata matrix have a corresponding anotation in the metadata file
all(rownames(meta) %in% colnames(countdata))
```

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

```
#Check if the order of the samples in the matrix is similar to that in the metadata file.
all(colnames(countdata) == rownames(meta))
```

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

```
#Importing the data into DESeqDataSet Object
dds <- DESeqDataSetFromMatrix(countData = countdata,
                              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
```

2. Importing Output from Star's FeatureCounts

```
#STAR

#Reading the STAR count matrix
star_count <- read.csv("Featurecounts/STAR_counts.txt", sep = "\t", header = T, row.names = 1, comment.
head(star_count)
```

```
##                                  Chr
## ENSG00000223972     1;1;1;1;1;1;1;1;1
## ENSG00000227232 1;1;1;1;1;1;1;1;1;1;1
## ENSG00000278267                    1
## ENSG00000243485            1;1;1;1;1
## ENSG00000284332                    1
## ENSG00000237613            1;1;1;1;1
##                                                                                Start
## ENSG00000223972            11869;12010;12179;12613;12613;12975;13221;13221;13453
## ENSG00000227232 14404;15005;15796;16607;16858;17233;17606;17915;18268;24738;29534
## ENSG00000278267                                                            17369
## ENSG00000243485                                        29554;30267;30564;30976;30976
## ENSG00000284332                                                            30366
## ENSG00000237613                                        34554;35245;35277;35721;35721
##                                                                                  End
## ENSG00000223972            12227;12057;12227;12721;12697;13052;13374;14409;13670
```

```
## ENSG00000227232 14501;15038;15947;16765;17055;17368;17742;18061;18366;24891;29570
## ENSG00000278267                                                             17436
## ENSG00000243485                                         30039;30667;30667;31109;31097
## ENSG00000284332                                                             30503
## ENSG00000237613                                         35174;35481;35481;36073;36081
##                                 Strand Length STAR_Alignment.sample37_sorted.bam
## ENSG00000223972     +;+;+;+;+;+;+;+;+;+   1735                                   0
## ENSG00000227232 -;-;-;-;-;-;-;-;-;-;-   1351                                  96
## ENSG00000278267                      -     68                                   8
## ENSG00000243485             +;+;+;+;+;+   1021                                   0
## ENSG00000284332                      +    138                                   0
## ENSG00000237613             -;-;-;-;-;-   1219                                   0
##                 STAR_Alignment.sample38_sorted.bam
## ENSG00000223972                                  0
## ENSG00000227232                                 72
## ENSG00000278267                                 21
## ENSG00000243485                                  0
## ENSG00000284332                                  0
## ENSG00000237613                                  0
##                 STAR_Alignment.sample39_sorted.bam
## ENSG00000223972                                  0
## ENSG00000227232                                265
## ENSG00000278267                                 79
## ENSG00000243485                                  0
## ENSG00000284332                                  0
## ENSG00000237613                                  0
##                 STAR_Alignment.sample40_sorted.bam
## ENSG00000223972                                  0
## ENSG00000227232                                 59
## ENSG00000278267                                  1
## ENSG00000243485                                  0
## ENSG00000284332                                  0
## ENSG00000237613                                  0
##                 STAR_Alignment.sample41_sorted.bam
## ENSG00000223972                                  0
## ENSG00000227232                                 62
## ENSG00000278267                                  0
## ENSG00000243485                                  0
## ENSG00000284332                                  0
## ENSG00000237613                                  0
##                 STAR_Alignment.sample42_sorted.bam
## ENSG00000223972                                  0
## ENSG00000227232                                 61
## ENSG00000278267                                  6
## ENSG00000243485                                  0
## ENSG00000284332                                  0
## ENSG00000237613                                  0
```

```
#Check column names
colnames(star_count)
```

```
##  [1] "Chr"                                "Start"
##  [3] "End"                                "Strand"
##  [5] "Length"                             "STAR_Alignment.sample37_sorted.bam"
```

```
##  [7] "STAR_Alignment.sample38_sorted.bam" "STAR_Alignment.sample39_sorted.bam"
##  [9] "STAR_Alignment.sample40_sorted.bam" "STAR_Alignment.sample41_sorted.bam"
## [11] "STAR_Alignment.sample42_sorted.bam"
```

```
#Remove the unwanted columns
star_count[c("Chr", "Start", "End", "Strand", "Length")] <- NULL

#Renaming the Colnames (to remove the prefix "STAR_Alignment." and suffix "_sorted.bam")
colnames(star_count) <- gsub("STAR_Alignment.", "", colnames(star_count))
colnames(star_count) <- gsub("_sorted.bam", "", colnames(star_count))
head(star_count)
```

```
##                 sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972        0        0        0        0        0        0
## ENSG00000227232       96       72      265       59       62       61
## ENSG00000278267        8       21       79        1        0        6
## ENSG00000243485        0        0        0        0        0        0
## ENSG00000284332        0        0        0        0        0        0
## ENSG00000237613        0        0        0        0        0        0
```

```
#Importing the data into DESeqDataSet Object
star_dds <- DESeqDataSetFromMatrix(countData = star_count,
                                   colData = meta,
                                   design = ~ Condition)
star_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
```

3. Importing Output from Kallisto

```
# Kallisto

#read in a table that links transcripts to genes for this dataset.
#We use the GenomicFeatures for this to create the tx2gene

# We create the TxDb obeject from the transcript annotation in the gff3 file
txdb <- makeTxDbFromGFF("gencode.v34.annotation.gff3")
```

```
## Import genomic features from the file as a GRanges object ... OK
## Prepare the 'metadata' data frame ... OK
## Make the TxDb object ...
```

```
## Warning in .get_cds_IDX(mcols0$type, mcols0$phase): The "phase" metadata column contains non-NA valu
##   stop_codon. This information was ignored.
```

```
## OK
```

```
txdb
```

```
## TxDb object:
## # Db type: TxDb
## # Supporting package: GenomicFeatures
## # Data source: gencode.v34.annotation.gff3
## # Organism: NA
## # Taxonomy ID: NA
## # miRBase build ID: NA
## # Genome: NA
## # Nb of transcripts: 228048
## # Db created by: GenomicFeatures package from Bioconductor
## # Creation time: 2020-08-18 09:03:32 +0300 (Tue, 18 Aug 2020)
## # GenomicFeatures version at creation time: 1.41.2
## # RSQLite version at creation time: 2.2.0
## # DBSCHEMAVERSION: 1.2
```

```
gene_id <- keys(txdb, keytype = "GENEID")  #Extracct the GeneIDs
#Matching the transcripts with the geneIDs
tx2gene <- select(txdb, keys = gene_id, keytype = "GENEID", columns = "TXNAME")
```

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

```
head(tx2gene,10)
```

```
##                GENEID            TXNAME
## 1  ENSG00000000003.15 ENST00000373020.9
## 2  ENSG00000000003.15 ENST00000612152.4
## 3  ENSG00000000003.15 ENST00000614008.4
## 4  ENSG00000000003.15 ENST00000496771.5
## 5  ENSG00000000003.15 ENST00000494424.1
## 6   ENSG00000000005.6 ENST00000373031.5
## 7   ENSG00000000005.6 ENST00000485971.1
## 8  ENSG00000000419.12 ENST00000371588.9
## 9  ENSG00000000419.12 ENST00000466152.5
## 10 ENSG00000000419.12 ENST00000371582.8
```

```
#Dircetory for Kallisto counts
kal_dir <- "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto"
files <- file.path(kal_dir, rownames(meta),"abundance.h5" )
names(files) <- rownames(meta)
files
```

```
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample37
##
```

```
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample38/
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample39/
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample40/
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample41/
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample42/
```

```
library("tximport")
library("readr")
#Reading the files for the different files
txi <- tximport(files, type="kallisto",tx2gene = tx2gene, txOut = T)
```

```
## 1 2 3 4 5 6
```

```
names(txi)
```

```
## [1] "abundance"           "counts"              "infReps"
## [4] "length"              "countsFromAbundance"
```

```
#Construct the  DESeqDataSet object
kallisto_dds <- DESeqDataSetFromTximport(txi,
                                   colData = meta,
                                   design = ~ Condition)
```

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

```
kallisto_dds
```

```
## class: DESeqDataSet
## dim: 228048 6
## metadata(1): version
## assays(2): counts avgTxLength
## rownames(228048):
##   ENST00000456328.2|ENSG00000223972.5|OTTHUMG00000000961.2|OTTHUMT00000362751.1|DDX11L1-202|DDX11L1|
##   ENST00000450305.2|ENSG00000223972.5|OTTHUMG00000000961.2|OTTHUMT00000002844.2|DDX11L1-201|DDX11L1|
##   ...
##   ENST00000387460.2|ENSG00000210195.2|-|-|MT-TT-201|MT-TT|66|Mt_tRNA|
##   ENST00000387461.2|ENSG00000210196.2|-|-|MT-TP-201|MT-TP|68|Mt_tRNA|
## rowData names(0):
## colnames(6): sample37 sample38 ... sample41 sample42
## colData names(1): Condition
```

4. Import Output from Salmon

```
#Dircetory for Salmon counts
sal_dir <- "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon"
list.files(sal_dir)
```

```
## [1] "sample37_quant" "sample38_quant" "sample39_quant" "sample40_quant"
## [5] "sample41_quant" "sample42_quant"
```

```
files <- file.path(sal_dir, paste(rownames(meta),"_quant", sep = ""), "quant.sf" )
names(files) <- rownames(meta)
files
```

```
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample37_qu
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample38_qu
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample39_qu
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample40_qu
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample41_qu
##
## "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Salmon/sample42_qu
```

```
txi <- tximport(files, type="salmon",tx2gene = tx2gene, txOut = T)
```

```
## reading in files with read_tsv
```

```
## 1 2 3 4 5 6
```

```
head(txi$abundance)
```

```
##                   sample37 sample38 sample39 sample40 sample41 sample42
## ENST00000456328.2 0.313879 0.000000 0.000000 0.469614 0.144288 0.000000
## ENST00000450305.2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## ENST00000488147.1 3.835637 5.556041 9.204962 4.313274 5.721516 5.846142
## ENST00000619216.1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## ENST00000473358.1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## ENST00000469289.1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
```

```
#Construct the  DESeqDataSet object
salmon_dds <- DESeqDataSetFromTximport(txi,
                                       colData = meta,
                                       design = ~ Condition)
```

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

```
salmon_dds
```

```
## class: DESeqDataSet
## dim: 227201 6
## metadata(1): version
## assays(2): counts avgTxLength
## rownames(227201): ENST00000456328.2 ENST00000450305.2 ...
##   ENST00000387460.2 ENST00000387461.2
## rowData names(0):
## colnames(6): sample37 sample38 ... sample41 sample42
## colData names(1): Condition
```

Differential expression analysis

```
#Transforming the counts & Calculating pvalue using DESeq function
#The steps it performs are the estimation of size factors (which control for differences in the library
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
#Extracting the reults form the formed dds object
res <- results(dds)

#Alternative ways;
#res <- results(dds, name="Condition_normal_vs_disease")
#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 60683 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat    pvalue
##                  <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000223972  0.000000             NA        NA        NA        NA
## ENSG00000227232 71.877192       0.617389  0.547883  1.126864 0.2597999
## ENSG00000278267  9.328531       2.539811  1.067946  2.378220 0.0173964
## ENSG00000243485  0.152211      -0.780815  4.080473 -0.191354 0.8482482
## ENSG00000284332  0.000000             NA        NA        NA        NA
## ...                   ...            ...       ...       ...       ...
## ENSG00000271254 195.75986      -0.681117  0.510346  -1.33462  0.182001
## ENSG00000275405   0.00000             NA        NA        NA        NA
## ENSG00000275987   0.00000             NA        NA        NA        NA
## ENSG00000277475   1.83462       4.442474  3.933026   1.12953  0.258674
## ENSG00000268674   0.00000             NA        NA        NA        NA
##                      padj
##                 <numeric>
## ENSG00000223972        NA
## ENSG00000227232  0.573958
## ENSG00000278267  0.137215
## ENSG00000243485        NA
## ENSG00000284332        NA
```

```
## ...                   ...
## ENSG00000271254  0.484362
## ENSG00000275405        NA
## ENSG00000275987        NA
## ENSG00000277475        NA
## ENSG00000268674        NA
```

```
# Log fold change shrinkage for visualization and ranking
## useful for visualization and ranking of genes. We shall use the apeglm method
resultsNames(dds)
```

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

```
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 60683 rows and 5 columns
##                   baseMean log2FoldChange     lfcSE    pvalue      padj
##                  <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000223972  0.000000             NA        NA        NA        NA
## ENSG00000227232 71.877192      0.3295576  0.433939 0.2597999  0.573958
## ENSG00000278267  9.328531      1.1311198  1.572694 0.0173964  0.137215
## ENSG00000243485  0.152211     -0.0357337  0.535571 0.8482482        NA
## ENSG00000284332  0.000000             NA        NA        NA        NA
## ...                   ...            ...       ...       ...       ...
## ENSG00000271254 195.75986      -0.396262  0.431564  0.182001  0.484362
## ENSG00000275405   0.00000             NA        NA        NA        NA
## ENSG00000275987   0.00000             NA        NA        NA        NA
## ENSG00000277475   1.83462       0.057589  0.539882  0.258674        NA
## ENSG00000268674   0.00000             NA        NA        NA        NA
```

```
## p-values and adjusted p-values
#Reorder res based on pvalues
resOrdered <- res[order(res$pvalue),]
resOrdered
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 60683 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat      pvalue
##                  <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000039537   431.823       -6.92550  0.731458  -9.46809 2.85020e-21
## ENSG00000124237   241.613       -7.98885  0.855855  -9.33435 1.01613e-20
## ENSG00000160401   507.498       -6.03790  0.658602  -9.16775 4.83008e-20
```

```
## ENSG00000007908  1332.119       -6.21421  0.693271  -8.96361 3.14214e-19
## ENSG00000188817   409.116       -5.82887  0.667047  -8.73832 2.36603e-18
## ...                   ...            ...       ...       ...          ...
## ENSG00000276351         0             NA        NA        NA           NA
## ENSG00000275661         0             NA        NA        NA           NA
## ENSG00000275405         0             NA        NA        NA           NA
## ENSG00000275987         0             NA        NA        NA           NA
## ENSG00000268674         0             NA        NA        NA           NA
##                       padj
##                  <numeric>
## ENSG00000039537 7.13547e-17
## ENSG00000124237 1.27194e-16
## ENSG00000160401 4.03070e-16
## ENSG00000007908 1.96659e-15
## ENSG00000188817 1.18467e-14
## ...                     ...
## ENSG00000276351          NA
## ENSG00000275661          NA
## ENSG00000275405          NA
## ENSG00000275987          NA
## ENSG00000268674          NA
```

```
summary(res)
```

```
##
## out of 38260 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 1004, 2.6%
## LFC < 0 (down)     : 1498, 3.9%
## outliers [1]       : 294, 0.77%
## low counts [2]     : 12931, 34%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
#Interpreting the LFC, since its normal vs disease, if the LFC > 0 means activity of the gene is lower

#How many adjusted p-values were less than 0.1?
sum(res$padj < 0.1, na.rm=TRUE)
```

```
## [1] 2502
```

```
#By default cutoff alpha for padj is 0.1, however this can be changed so to 0.05 as shown below;
res05 <- results(dds, alpha=0.05)
summary(res05)
```

```
##
## out of 38260 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 609, 1.6%
## LFC < 0 (down)     : 1062, 2.8%
## outliers [1]       : 294, 0.77%
```

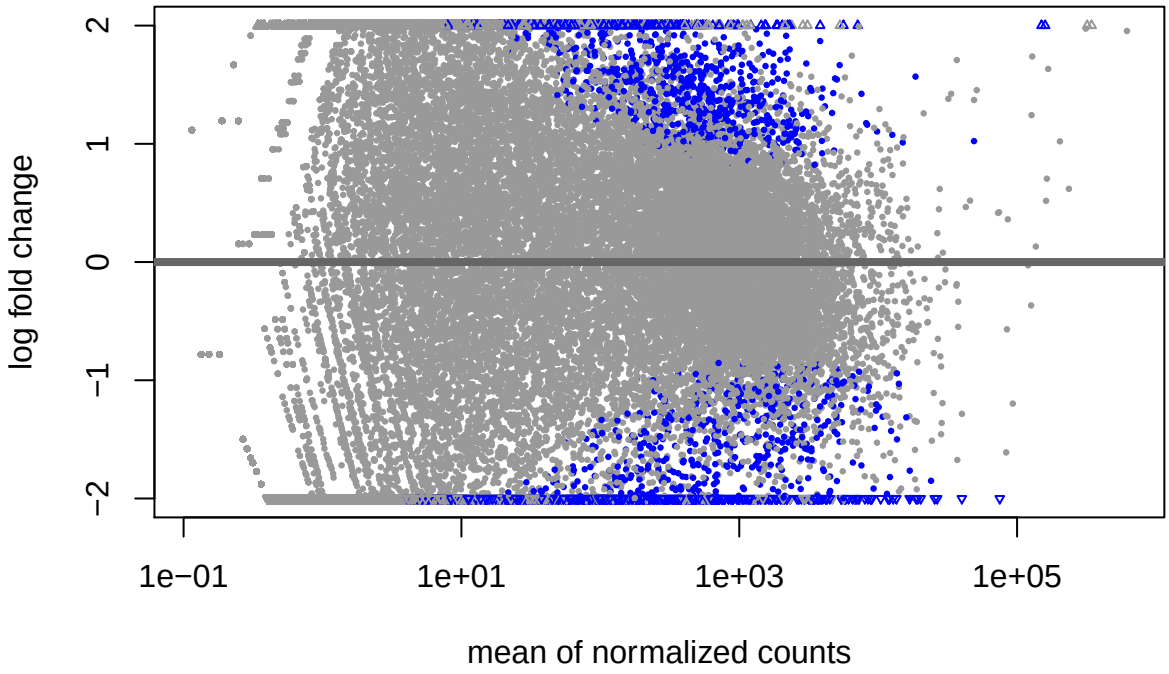
```
## low counts [2]     : 12931, 34%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
#The number of significant genes here decrease since 95% is much more stringient
sum(res05$padj < 0.05, na.rm=TRUE)
```

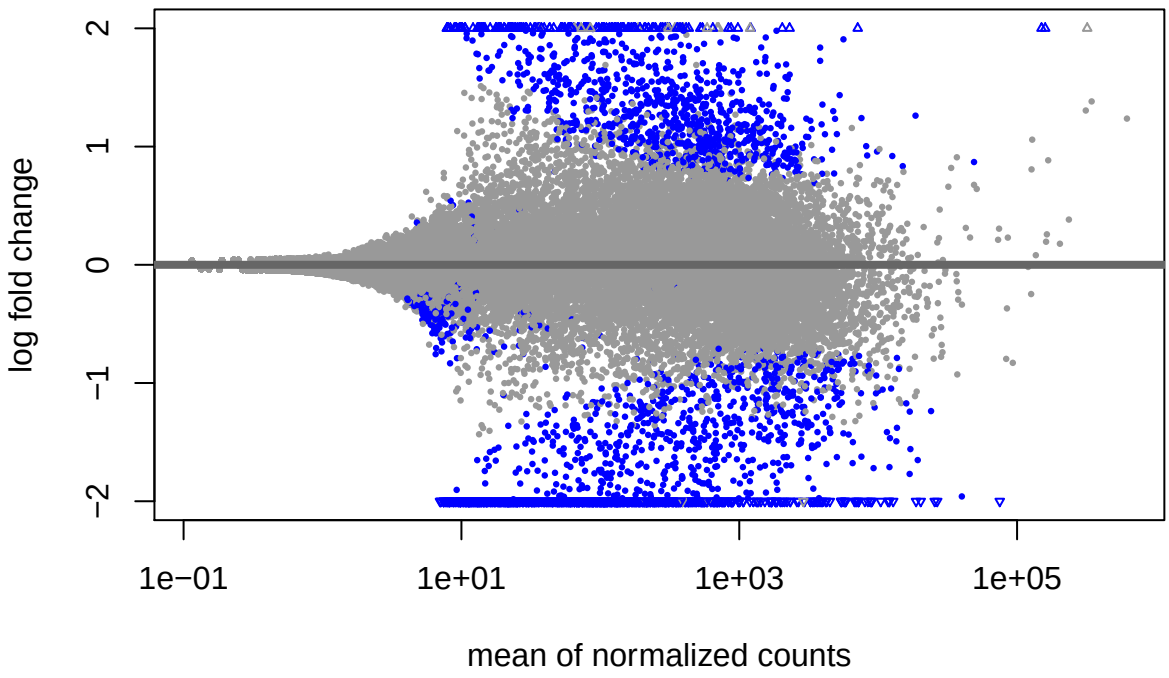
```
## [1] 1671
```

Exploring and exporting results

```
#MA-plot
#MA plot: The plot visualizes the differences between measurements taken in two samples, by transformin
plotMA(res, ylim=c(-2,2))
```




```
#It is more useful visualize the MA-plot for the shrunken log2 fold changes, which remove the noise ass
plotMA(resLFC, ylim=c(-2,2))
```



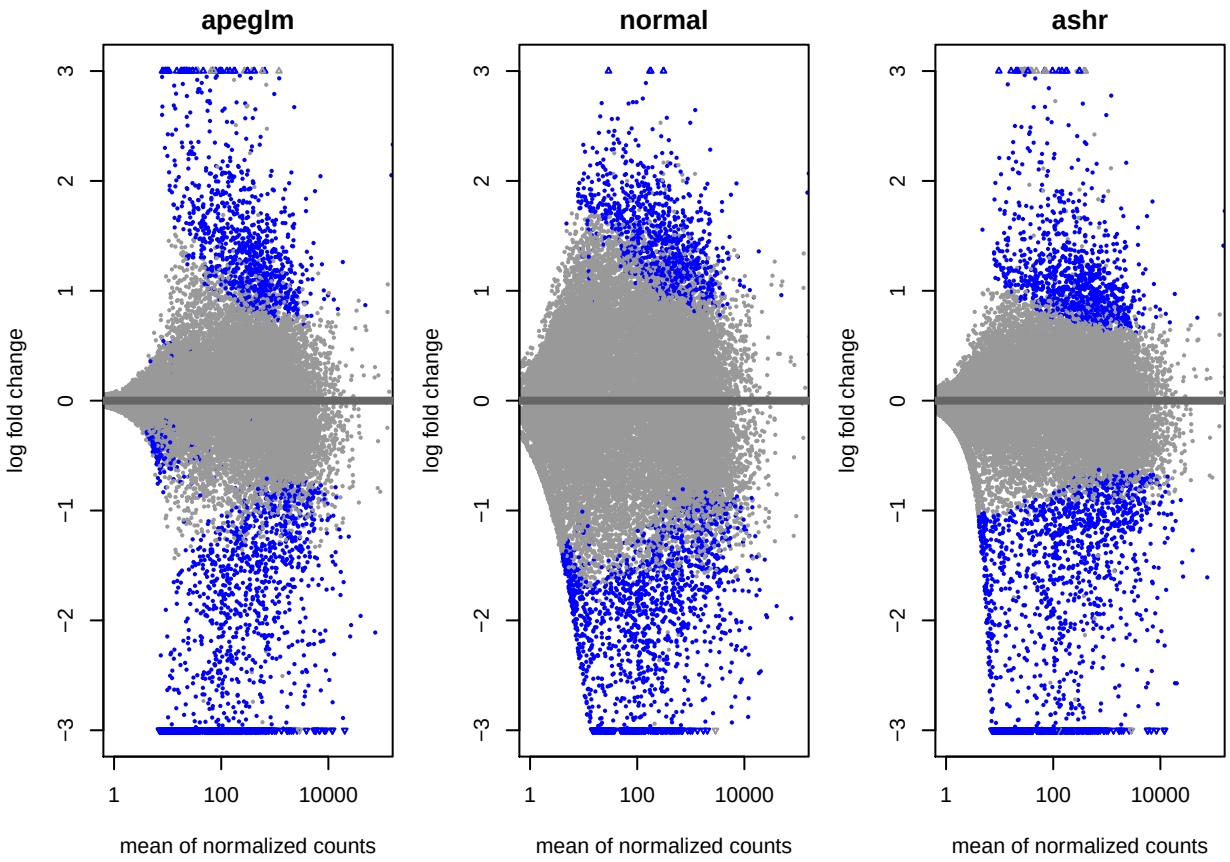

```
#Alternative shrinkage estimators apart from apeglm
# because we are interested in treated vs untreated, we set 'coef=2'
resNorm <- lfcShrink(dds, coef=2, type="normal")
```

```
## 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
```

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

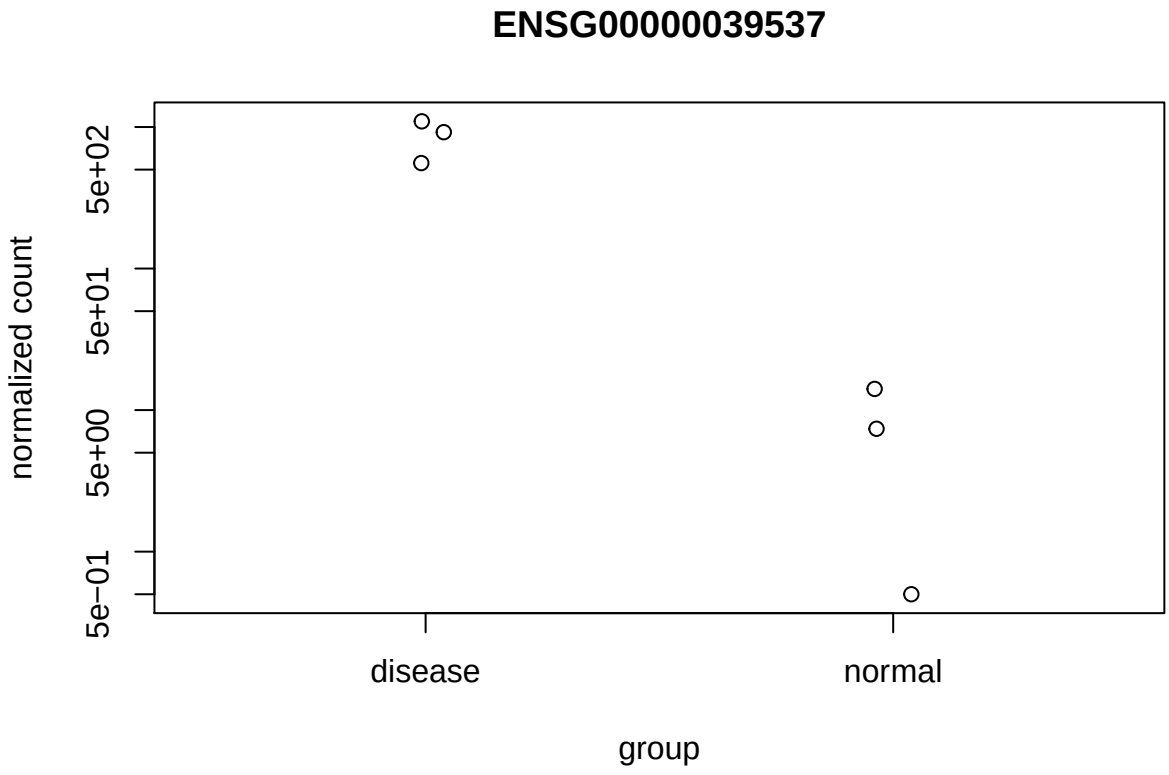
```
## using 'ashr' for LFC shrinkage. If used in published research, please cite:
##     Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
##     https://doi.org/10.1093/biostatistics/kxw041
```

```
par(mfrow=c(1,3), mar=c(4,4,2,1))  #mar sets the margin sizes in the following order: bottom, left, top
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

```
#This plots the normalized counts for each sample in the particular gene

#For the gene with the minimum padj
plotCounts(dds, gene=which.min(res$padj), intgroup="Condition")
```




```
#Or we could return a dataframe and plot it with ggplot2, we use the returnData=TRUE
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="Condition",
                returnData=TRUE)
d
```

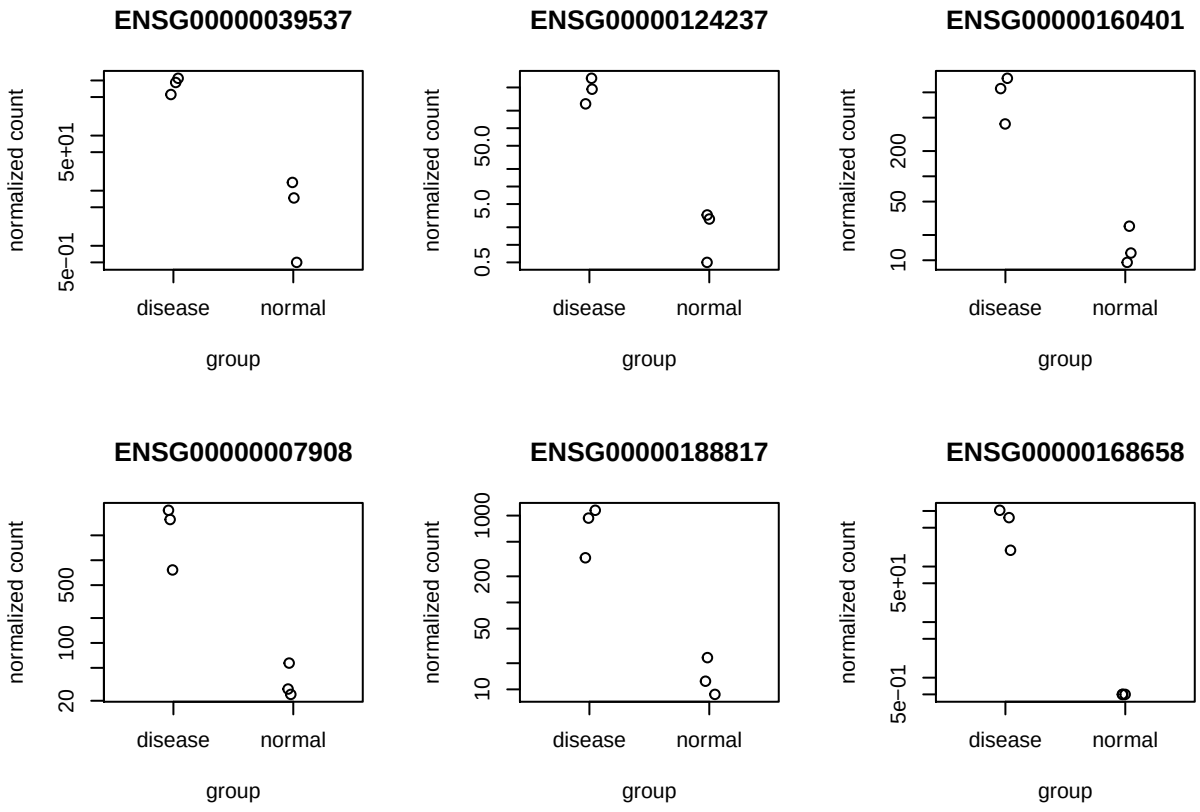
```
##                count Condition
## sample37    0.500000    normal
## sample38   14.112156    normal
## sample39    7.379807    normal
## sample40 1096.420711   disease
## sample41  556.222562   disease
## sample42  919.302136   disease
```

```
#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))



#Look at the 6 genes with the lowest adj
head(resOrdered)
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
```

```
## DataFrame with 6 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat      pvalue
##                  <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000039537   431.823       -6.92550  0.731458  -9.46809 2.85020e-21
## ENSG00000124237   241.613       -7.98885  0.855855  -9.33435 1.01613e-20
## ENSG00000160401   507.498       -6.03790  0.658602  -9.16775 4.83008e-20
## ENSG00000007908  1332.119       -6.21421  0.693271  -8.96361 3.14214e-19
## ENSG00000188817   409.116       -5.82887  0.667047  -8.73832 2.36603e-18
## ENSG00000168658   330.530      -11.73927  1.400479  -8.38233 5.18913e-17
##                         padj
##                    <numeric>
## ENSG00000039537 7.13547e-17
## ENSG00000124237 1.27194e-16
## ENSG00000160401 4.03070e-16
## ENSG00000007908 1.96659e-15
## ENSG00000188817 1.18467e-14
## ENSG00000168658 1.78295e-13
```

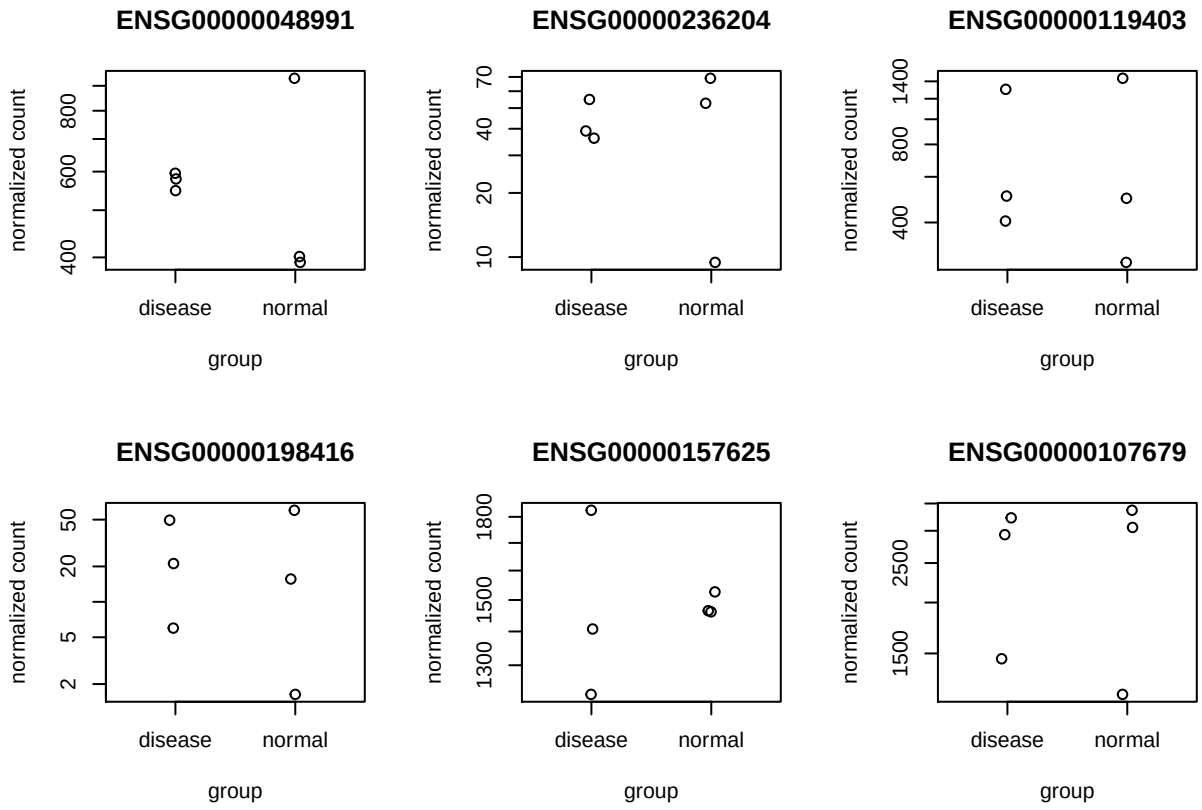
```
#Plot counts for these genes
par(mfrow=c(2,3))
plotCounts(dds, gene="ENSG00000039537", intgroup="Condition")
plotCounts(dds, gene="ENSG00000124237", intgroup="Condition")
plotCounts(dds, gene="ENSG00000160401", intgroup="Condition")
plotCounts(dds, gene="ENSG00000007908", intgroup="Condition")
plotCounts(dds, gene="ENSG00000188817", intgroup="Condition")
plotCounts(dds, gene="ENSG00000168658", intgroup="Condition")
```



```
#Compare this with non-significant p-values
tail(na.omit(resOrdered))
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE         stat    pvalue
##                 <numeric>      <numeric> <numeric>    <numeric> <numeric>
## ENSG00000048991  574.2415   -3.05142e-04  0.491974 -0.000620241  0.999505
## ENSG00000236204   43.0285    3.32590e-04  0.819029  0.000406079  0.999676
## ENSG00000119403  737.9646   -2.65767e-04  0.733961 -0.000362101  0.999711
## ENSG00000198416   25.1621    3.01625e-04  1.266748  0.000238110  0.999810
## ENSG00000157625 1484.0039   -4.74458e-05  0.346662 -0.000136865  0.999891
## ENSG00000107679 2538.6277   -3.13757e-05  0.559391 -0.000056089  0.999955
##                      padj
##                 <numeric>
## ENSG00000048991  0.999705
## ENSG00000236204  0.999831
## ENSG00000119403  0.999831
## ENSG00000198416  0.999890
## ENSG00000157625  0.999931
## ENSG00000107679  0.999955
```

```
par(mfrow=c(2,3))
plotCounts(dds, gene="ENSG00000048991", intgroup="Condition")
plotCounts(dds, gene="ENSG00000236204", intgroup="Condition")
plotCounts(dds, gene="ENSG00000119403", intgroup="Condition")
plotCounts(dds, gene="ENSG00000198416", intgroup="Condition")
plotCounts(dds, gene="ENSG00000157625", intgroup="Condition")
plotCounts(dds, gene="ENSG00000107679", intgroup="Condition")
```



A closer look at the columns for the result, res

```
head(res)
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat    pvalue
##                 <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000223972  0.000000             NA        NA        NA        NA
## ENSG00000227232 71.877192       0.617389  0.547883  1.126864 0.2597999
## ENSG00000278267  9.328531       2.539811  1.067946  2.378220 0.0173964
## ENSG00000243485  0.152211      -0.780815  4.080473 -0.191354 0.8482482
## ENSG00000284332  0.000000             NA        NA        NA        NA
## ENSG00000237613  0.333464      -1.771875  4.030220 -0.439647 0.6601927
##                      padj
##                 <numeric>
## ENSG00000223972        NA
## ENSG00000227232  0.573958
## ENSG00000278267  0.137215
## ENSG00000243485        NA
## ENSG00000284332        NA
## ENSG00000237613        NA
```

```
#Notes
# - if basemean for all samples is zero, the p values and lfc will be NA
# - If a row contains any sample with extreme count outlier, the p-value and padj are set to NA
# - If a row is filetered due to low counts then only the padj is set to NA
```

```
#Only subsettting genes with Significant padj value
resSig <- subset(resOrdered, padj < 0.1)
resSig
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 2502 rows and 6 columns
##                    baseMean log2FoldChange     lfcSE      stat      pvalue
##                   <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000039537    431.823       -6.92550  0.731458  -9.46809 2.85020e-21
## ENSG00000124237    241.613       -7.98885  0.855855  -9.33435 1.01613e-20
## ENSG00000160401    507.498       -6.03790  0.658602  -9.16775 4.83008e-20
## ENSG00000007908   1332.119       -6.21421  0.693271  -8.96361 3.14214e-19
## ENSG00000188817    409.116       -5.82887  0.667047  -8.73832 2.36603e-18
## ...                    ...            ...       ...       ...         ...
## ENSG00000162729  695.44558        1.00330  0.389301   2.57719  0.00996078
## ENSG00000213435    7.68007       -4.32709  1.679101  -2.57703  0.00996540
## ENSG00000134419 2448.41647        1.24898  0.484722   2.57669  0.00997507
## ENSG00000245954    9.28888       -3.69288  1.433305  -2.57648  0.00998121
## ENSG00000171695    5.04518       -5.70991  2.216485  -2.57611  0.00999182
##                       padj
##                  <numeric>
## ENSG00000039537 7.13547e-17
## ENSG00000124237 1.27194e-16
## ENSG00000160401 4.03070e-16
## ENSG00000007908 1.96659e-15
## ENSG00000188817 1.18467e-14
## ...                     ...
## ENSG00000162729   0.0998271
## ENSG00000213435   0.0998335
## ENSG00000134419   0.0998904
## ENSG00000245954   0.0999118
## ENSG00000171695   0.0999781
```

```
getwd()
```

```
## [1] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts"
```

Exporting Significant Results

```
resSig <- subset(resOrdered, padj < 0.1)
head(resSig)
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
```

```
## DataFrame with 6 rows and 6 columns
##                   baseMean log2FoldChange     lfcSE      stat      pvalue
##                  <numeric>      <numeric> <numeric> <numeric>   <numeric>
## ENSG00000039537   431.823       -6.92550  0.731458  -9.46809 2.85020e-21
## ENSG00000124237   241.613       -7.98885  0.855855  -9.33435 1.01613e-20
## ENSG00000160401   507.498       -6.03790  0.658602  -9.16775 4.83008e-20
## ENSG00000007908  1332.119       -6.21421  0.693271  -8.96361 3.14214e-19
## ENSG00000188817   409.116       -5.82887  0.667047  -8.73832 2.36603e-18
## ENSG00000168658   330.530      -11.73927  1.400479  -8.38233 5.18913e-17
##                        padj
##                   <numeric>
## ENSG00000039537 7.13547e-17
## ENSG00000124237 1.27194e-16
## ENSG00000160401 4.03070e-16
## ENSG00000007908 1.96659e-15
## ENSG00000188817 1.18467e-14
## ENSG00000168658 1.78295e-13
```

```
write.table(as.data.frame(resSig),
          file="condition_treated_significant_results.csv")
```

Data Transformation

Testing for DE operates on raw counts, however for visualizations and clusterings, its better to work with transformed count data. There are 2 common waays of transformation; which produce transformed data on the log2 scale which has been normalized with respect to library size or other normalization factors.

- Variance Stabilizing Transformations (VST) and
- regularized logarithm or rlog

The goal of transformation is to eliminate any variance arising when the mean is low. rlog might be slower for many samples compared to vst.

One common argument used during transformation is `blind` which tells whether transformation should be blind to sample information specified in the the design (ie in an unbiased manner), hence using only the intercept. However this is not the best choice especially if the counts difference are attributed to the design and if one is to use the transformed data for downstream analysis. If `blind = FALSE` is set, we take into account already estimated dispersion due to design as we are transforming the data, and make the process much faster.

Comparison: VST runs faster than rlog. If the library size of the samples and therefore their size factors vary widely, the rlog transformation is a better option than VST. Both options produce log2 scale data which has been normalized by the DESeq2 method with respect to library size.

```
#Count Transformations
vsd <- vst(dds, blind=FALSE)
rld <- rlog(dds, blind=FALSE)
head(assay(vsd))   #Assay extracts a matrix of normalized counts
```

```
##                 sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972 5.281749 5.281749 5.281749 5.281749 5.281749 5.281749
## ENSG00000227232 7.177809 6.902050 7.635498 6.856577 7.026329 6.915044
## ENSG00000278267 6.020474 6.088279 6.404361 5.593753 5.698105 5.694927
```

```
## ENSG00000243485 5.281749 5.281749 5.281749 5.502584 5.281749 5.281749
## ENSG00000284332 5.281749 5.281749 5.281749 5.281749 5.281749 5.281749
## ENSG00000237613 5.281749 5.281749 5.281749 5.502584 5.522688 5.281749
```

```
head(assay(rld))
```

```
##                  sample37  sample38  sample39  sample40  sample41  sample42
## ENSG00000223972  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
## ENSG00000227232  6.210644  5.891107  6.701883  5.835198  6.038031  5.905750
## ENSG00000278267  3.095538  3.190424  3.618631  2.546669  2.670309  2.660245
## ENSG00000243485 -1.979000 -1.982334 -1.988459 -1.960492 -1.982850 -1.986563
## ENSG00000284332  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
## ENSG00000237613 -1.503514 -1.510533 -1.523192 -1.464545 -1.460042 -1.519438
```

```
head(res)
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 6 rows and 6 columns
##                  baseMean log2FoldChange     lfcSE      stat    pvalue
##                 <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000223972  0.000000             NA        NA        NA        NA
## ENSG00000227232 71.877192       0.617389  0.547883  1.126864 0.2597999
## ENSG00000278267  9.328531       2.539811  1.067946  2.378220 0.0173964
## ENSG00000243485  0.152211      -0.780815  4.080473 -0.191354 0.8482482
## ENSG00000284332  0.000000             NA        NA        NA        NA
## ENSG00000237613  0.333464      -1.771875  4.030220 -0.439647 0.6601927
##                      padj
##                 <numeric>
## ENSG00000223972        NA
## ENSG00000227232  0.573958
## ENSG00000278267  0.137215
## ENSG00000243485        NA
## ENSG00000284332        NA
## ENSG00000237613        NA
```

```
head(countdata)
```

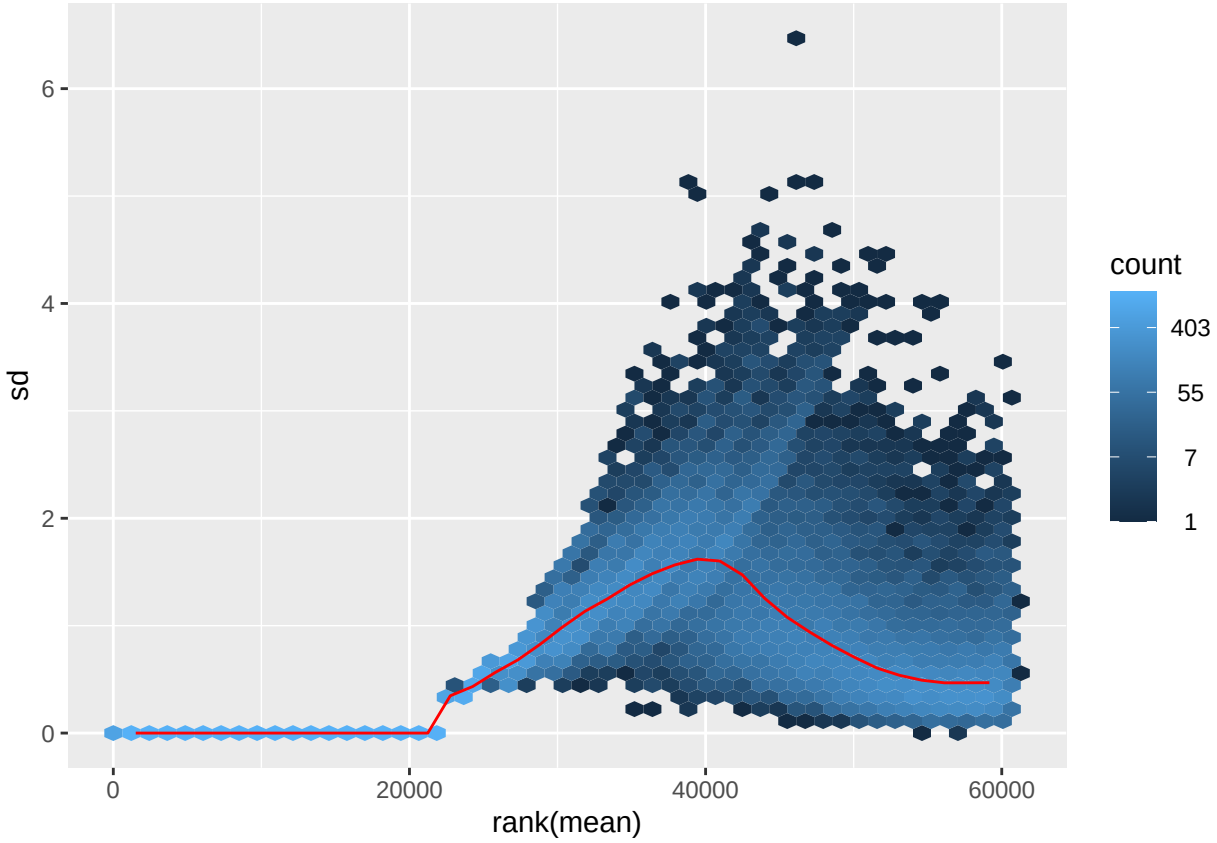
```
##                 sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972        0        0        0        0        0        0
## ENSG00000227232       52       48      187       56       59       69
## ENSG00000278267        7       11       36        2        3        4
## ENSG00000243485        0        0        0        1        0        0
## ENSG00000284332        0        0        0        0        0        0
## ENSG00000237613        0        0        0        1        1        0
```

Effects of transformations on the variance

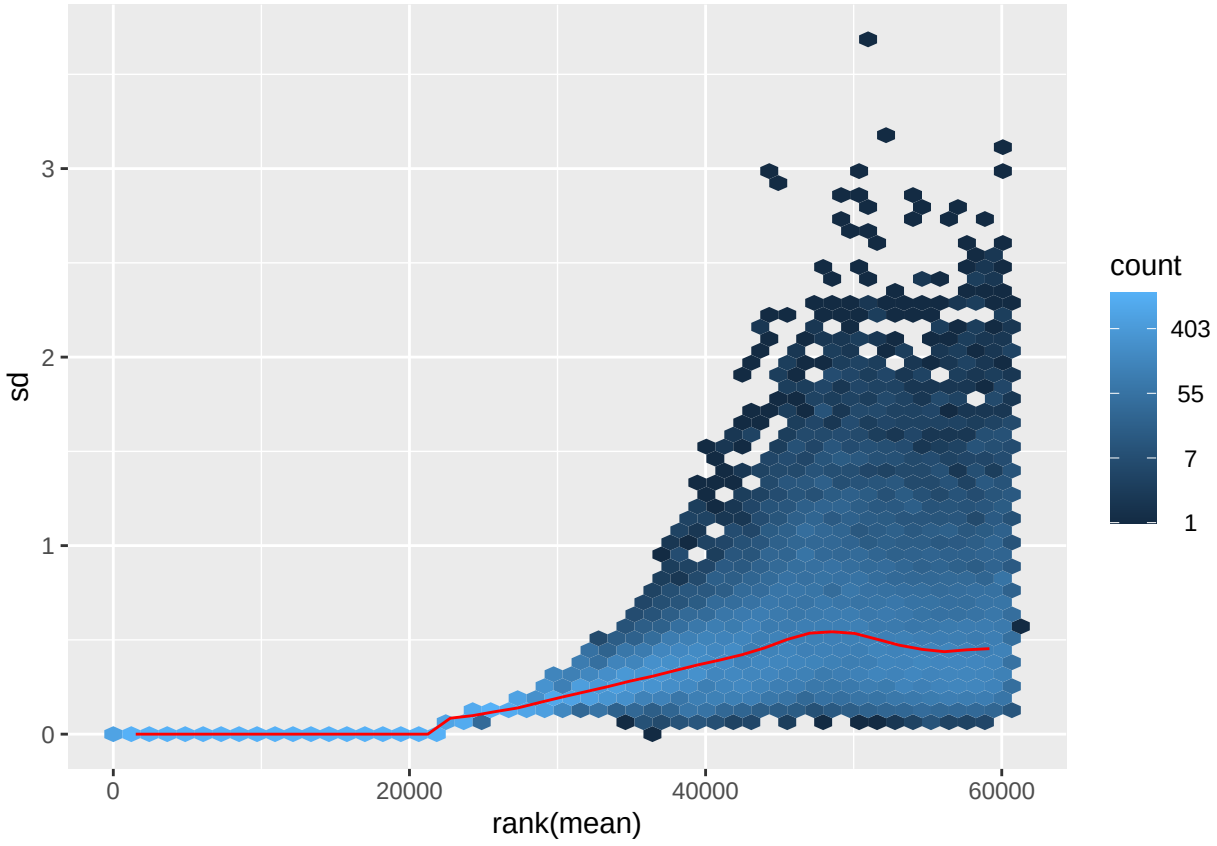
The figure below plots the standard deviation of the transformed data, across samples, against the mean, using the shifted logarithm transformation, the regularized log transformation and the variance stabilizing transformation.

What we expect is: The shifted logarithm has elevated standard deviation in the lower count range, and the regularized log to a lesser extent, while for the variance stabilized data the standard deviation is roughly constant along the whole dynamic range.

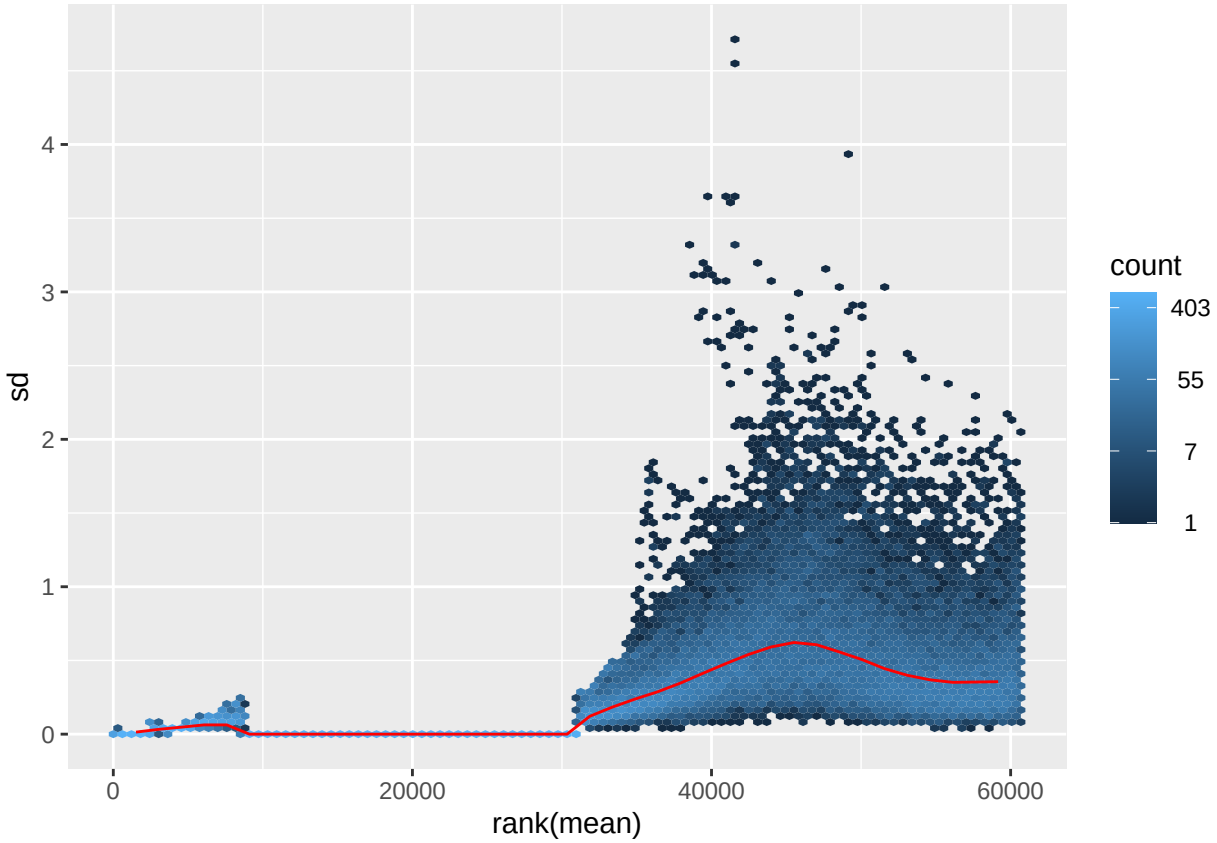
```
#BiocManager::install("vsn")
library("vsn")

#for the untransformed dds
ntd <- normTransform(dds)
meanSdPlot(assay(ntd))
```




```
#for vst
meanSdPlot(assay(vsd))
```




```
#For rlog
meanSdPlot(assay(rld), bins = 100)
```



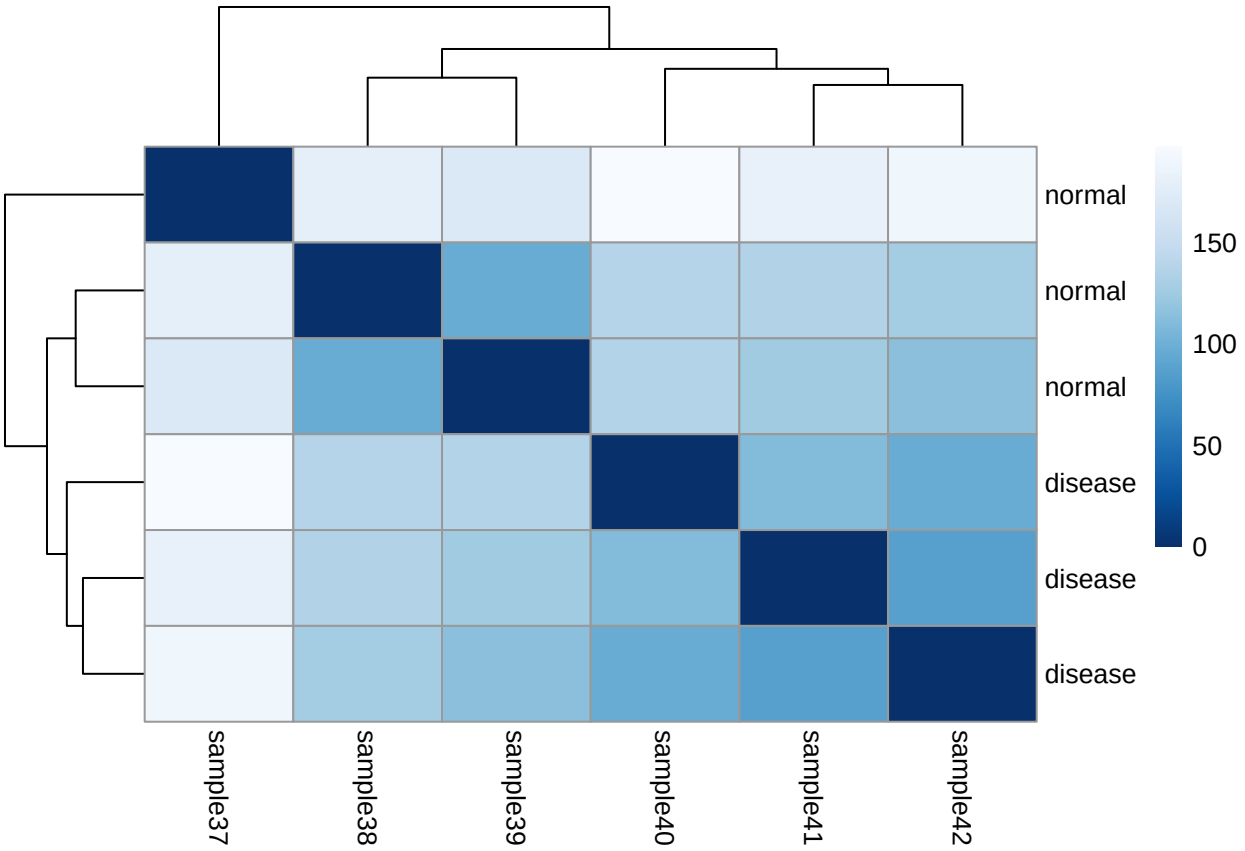
From the above plots i don't see much of the difference.

Data quality assessment by sample clustering and visualization

Heatmap of the sample-to-sample distances

A heatmap of this distance matrix gives us an overview over similarities and dissimilarities between samples.

```
#Calculate sample-2-sample distances
sampleDists <- dist(t(assay(vsd)))

library("RColorBrewer")
library(pheatmap)
sampleDistMatrix <- as.matrix(sampleDists)  #Converting the dist object to matrix
rownames(sampleDistMatrix) <- vsd$Condition
colnames(sampleDistMatrix) <- rownames(meta)  #My modification to match samples and Condition
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```

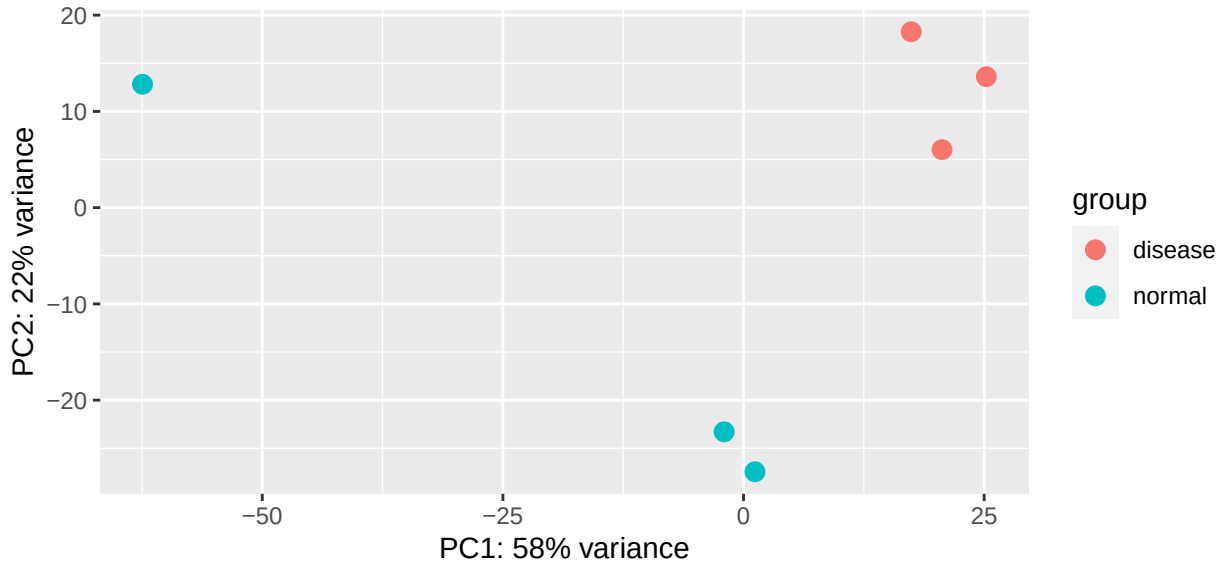


From this sample37 looks to be slightly different to other sanples (even those in the same Condition)

Principal component plot of the samples

This plot gives a visual of the overall relatedness between the samples in the different batches

```
plotPCA(vsd, intgroup="Condition")
```



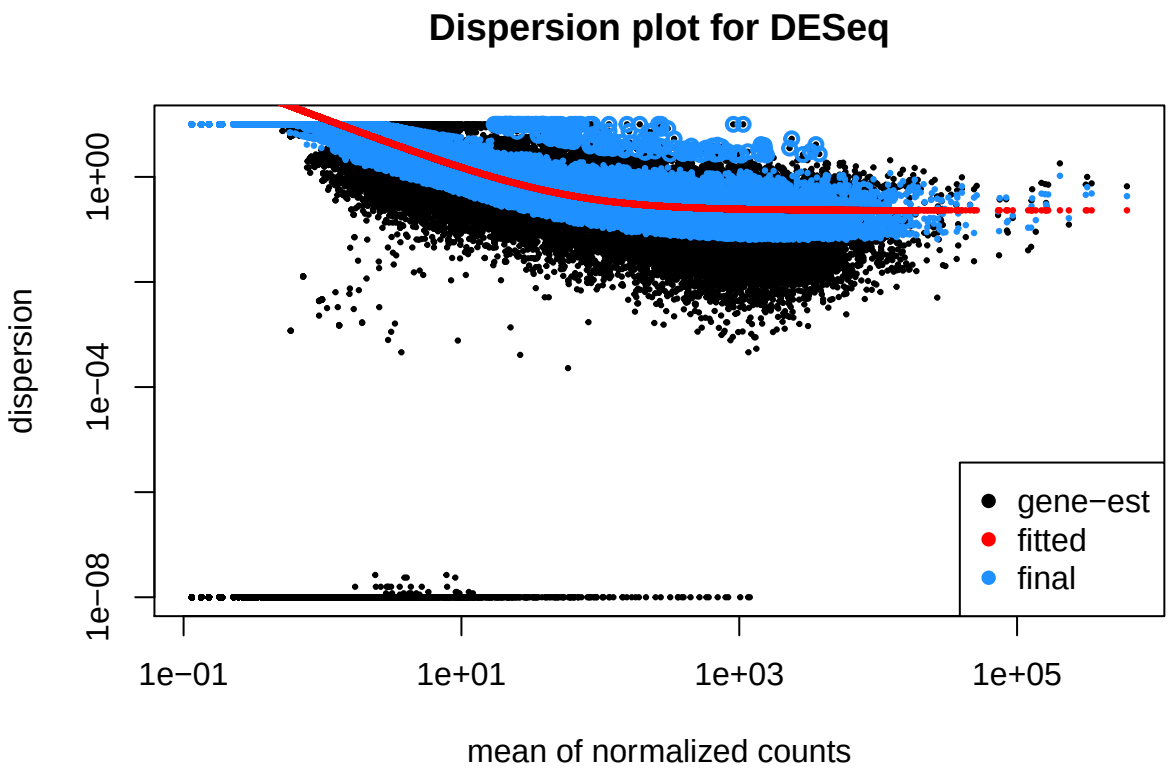
From this PCA plot, one sample seems to be off among the normal category, which i presume its the samnple37.

Dispersion plot

This plot generally show how gene expression data differs across samples in the same treatment group.

- The black dots are the dispersion estimates for each gene separately
- The red line is the fitted trend which shows the dispersion's dependence on the mean.
- The blue dots are genes that have been shrunk/fitted towards the red line.
- The outer blue circles with black dots inside are considered outliers and therefore not shrunk towards the fitted line

```
plotDispEsts(dds, main="Dispersion plot for DESeq")
```



Adding Gene names

source: http://www.sthda.com/english/wiki/rna-seq-differential-expression-work-flow-using-deseq2#running-the-deseq2-pipeline

The goal here is to use the Ensembl gene names to get the gene symbol and entryID which can then be used later to locate it function in the reactome database.

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

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

```
##
```

```
columns(org.Hs.eg.db)     #list of all available key types
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [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"
```

```
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 ] )
}

res1 <- res  #making a backup
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
```

```
head(res, 20)
```

```
## log2 fold change (MLE): Condition normal vs disease
## Wald test p-value: Condition normal vs disease
## DataFrame with 20 rows and 8 columns
##                  baseMean log2FoldChange     lfcSE       stat    pvalue
##                 <numeric>      <numeric> <numeric>  <numeric> <numeric>
## ENSG00000223972  0.000000             NA        NA         NA        NA
## ENSG00000227232 71.877192       0.617389  0.547883   1.126864 0.2597999
## ENSG00000278267  9.328531       2.539811  1.067946   2.378220 0.0173964
## ENSG00000243485  0.152211      -0.780815  4.080473  -0.191354 0.8482482
## ENSG00000284332  0.000000             NA        NA         NA        NA
## ...                   ...            ...       ...        ...       ...
## ENSG00000241860   5.20599      0.0130591  1.440118 0.00906811 0.9927648
## ENSG00000222623   1.43877      1.8915642  3.907910 0.48403470 0.6283612
## ENSG00000241599   0.00000             NA        NA         NA        NA
## ENSG00000279928   0.00000             NA        NA         NA        NA
## ENSG00000279457 113.60784      1.4705171  0.580297 2.53407700 0.0112744
##                      padj hgnc_symbol    entrezid
##                 <numeric> <character> <character>
## ENSG00000223972        NA     DDX11L1   100287102
## ENSG00000227232  0.573958          NA          NA
## ENSG00000278267  0.137215   MIR6859-1   102466751
## ENSG00000243485        NA          NA          NA
## ENSG00000284332        NA   MIR1302-2   100302278
## ...                   ...         ...         ...
## ENSG00000241860  0.997840          NA          NA
## ENSG00000222623        NA          NA          NA
## ENSG00000241599        NA          NA          NA
## ENSG00000279928        NA          NA          NA
## ENSG00000279457  0.107321          NA          NA
```

Gene-set enrichment analysis

The goal of this is to examine whether the observed genes with a strong/significant up- or down-regulation have either something in common or have something to do with the biological process causing the condition.

```
#BiocManager::install("reactome.db")
library("reactome.db" )

#resSig
#Extracting only genes that have corresponding data in reactome and padj value is not an NA
res2 <- res[ res$entrezid %in% keys( reactome.db, "ENTREZID" ) & !is.na( res$padj ) , ]
head(res2)
```

```
## 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>
## ENSG00000188976 1426.2300       0.472666  0.395878  1.193968 0.23249052
## ENSG00000187642   10.3347       1.514014  1.689232  0.896274 0.37010652
## ENSG00000187608  458.4554       1.542328  0.662844  2.326834 0.01997410
## ENSG00000188157 5994.5542       0.601311  0.341325  1.761699 0.07812021
## ENSG00000162571   77.5936      -2.362999  0.796479 -2.966807 0.00300910
## ENSG00000186891  119.8685       2.817714  0.929541  3.031295 0.00243508
##                      padj hgnc_symbol    entrezid
##                 <numeric> <character> <character>
## ENSG00000188976 0.5427388       NOC2L       26155
## ENSG00000187642 0.6758784       PERM1       84808
## ENSG00000187608 0.1492397       ISG15        9636
## ENSG00000188157 0.3184206        AGRN      375790
## ENSG00000162571 0.0469473      TTLL10      254173
## ENSG00000186891 0.0411628    TNFRSF18        8784
```

```
#Next we use select for AnnotationDbi to map entrez_id to Reactome_IDs
reactomeTable <- AnnotationDbi::select( reactome.db,
   keys=as.character(res2$entrezid), keytype="ENTREZID",
   columns=c("ENTREZID","REACTOMEID") )
```

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

```
head(reactomeTable, 10)
```

```
##    ENTREZID    REACTOMEID
## 1     26155  R-HSA-212436
## 2     26155 R-HSA-3700989
## 3     26155 R-HSA-5633007
## 4     26155 R-HSA-6804756
## 5     26155   R-HSA-73857
## 6     26155   R-HSA-74160
## 7     84808 R-HSA-1592230
## 8     84808 R-HSA-1852241
## 9     84808 R-HSA-2151201
## 10     9636  R-HSA-110313
```

```
tail(reactomeTable)
```

```
##       ENTREZID    REACTOMEID
## 97956     4541  R-HSA-611105
## 97957     4541 R-HSA-6799198
## 97958     4519 R-HSA-1428517
## 97959     4519 R-HSA-1430728
## 97960     4519  R-HSA-163200
## 97961     4519  R-HSA-611105
```

```
#We then tell which genes are members of which Reactome Paths.
incm <- do.call( rbind, with(reactomeTable, tapply(
  ENTREZID, factor(REACTOMEID), function(x) res2$entrezid %in% x ) ))
colnames(incm) <- res2$entrez
str(incm)
```

```
##  logi [1:2230, 1:9089] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:2230] "R-HSA-1059683" "R-HSA-109581" "R-HSA-109582" "R-HSA-109606" ...
##   ..$ : NULL
```

```
#We remove all rows corresponding to Reactome Paths with less than 20 or more than 80 assigned genes.
within <- function(x, lower, upper) (x>=lower & x<=upper)
incm <- incm[ within(rowSums(incm), lower=20, upper=80), ]

#We test whether the genes in a Reactome Path behave in a special way in our experiment,
testCategory <- function(reactomeID) {
  isMember <- incm[ reactomeID, ]
  data.frame(
     reactomeID  = reactomeID,
     numGenes    = sum( isMember ),
     avgLFC      = mean( res2$log2FoldChange[isMember] ),
     sdLFC       = sd( res2$log2FoldChange[isMember] ),
     zValue      = mean( res2$log2FoldChange[isMember] ) /sd( res2$log2FoldChange[isMember] ),
     strength    = sum( res2$log2FoldChange[isMember] ) / sqrt(sum(isMember)),
     pvalue      = t.test( res2$log2FoldChange[ isMember ] )$p.value,
     reactomeName = reactomePATHID2NAME[[reactomeID]],
     stringsAsFactors = FALSE ) }

#We call the function for all Paths in our incidence matrix and collect the results in a data frame:
reactomeResult <- do.call( rbind, lapply( rownames(incm), testCategory ) )
head(reactomeResult)
```

```
##     reactomeID numGenes     avgLFC     sdLFC     zValue   strength      pvalue
## 1 R-HSA-109606       53 -0.1293602 0.7695862 -0.1680906 -0.9417565 0.226574711
## 2 R-HSA-109704       33 -0.5389228 0.9878467 -0.5455531 -3.0958759 0.003678713
## 3 R-HSA-110313       38  0.2547932 0.7408378  0.3439257  1.5706505 0.040766495
## 4 R-HSA-110314       29  0.2207748 0.6433235  0.3431785  1.1889087 0.075180493
## 5 R-HSA-110328       34  0.5681210 1.4837583  0.3828932  3.3126861 0.032474254
## 6 R-HSA-110329       34  0.5681210 1.4837583  0.3828932  3.3126861 0.032474254
##                                                                                reactor
## 1                                                 Homo sapiens: Intrinsic Pathway for Apo
```

```
## 2                                                                    Homo sapiens: PI3K Ca
## 3         Homo sapiens: Translesion synthesis by Y family DNA polymerases bypasses lesions on DNA ter
## 4                           Homo sapiens: Recognition of DNA damage by PCNA-containing replication co
## 5 Homo sapiens: Recognition and association of DNA glycosylase with site containing an affected pyrim
## 6                                                  Homo sapiens: Cleavage of the damaged pyrimi
```

```
#Now we perform Multiple Testing Adjustment again
reactomeResult$padjust <- p.adjust( reactomeResult$pvalue, "BH" )

#We obtain the reactomeID with significant p.adj  values in our comparison of normal vs disease
reactomeResultSignif <- reactomeResult[ reactomeResult$padjust < 0.05, ]

print(head( reactomeResultSignif[ order(-reactomeResultSignif$strength), ] ,20))
```

```
##        reactomeID numGenes    avgLFC     sdLFC    zValue strength       pvalue
## 465  R-HSA-606279       48 0.7399235 1.3389488 0.5526152 5.126340 3.802590e-04
## 573  R-HSA-774815       48 0.7399235 1.3389488 0.5526152 5.126340 3.802590e-04
## 481 R-HSA-6799198       55 0.6506948 0.6785176 0.9589947 4.825682 2.709182e-09
## 124  R-HSA-174178       70 0.5710845 0.8587711 0.6650020 4.778035 4.673545e-07
## 132  R-HSA-176814       74 0.5460066 1.0587603 0.5157037 4.696927 3.177962e-05
## 131  R-HSA-176409       73 0.5274189 1.0538596 0.5004641 4.506269 5.752387e-05
## 130  R-HSA-176408       78 0.4706521 1.0156595 0.4633956 4.156687 1.042229e-04
## 137  R-HSA-179419       71 0.4877345 1.0394916 0.4692048 4.109724 1.818581e-04
## 123  R-HSA-174154       65 0.5055094 0.8218459 0.6150903 4.075547 5.506552e-06
## 472 R-HSA-6783310       37 0.6629791 1.0961700 0.6048141 4.032744 7.604850e-04
## 545   R-HSA-73884       64 0.4985099 1.1450059 0.4353776 3.988079 9.071153e-04
## 125  R-HSA-174184       70 0.4691639 1.0350665 0.4532694 3.925307 3.164487e-04
## 121  R-HSA-174084       63 0.4655583 0.7716272 0.6033462 3.695255 1.081917e-05
## 136  R-HSA-179409       25 0.7270055 0.9860425 0.7372963 3.635027 1.158797e-03
## 454 R-HSA-5693568       27 0.6889265 1.1526431 0.5976928 3.579767 4.546530e-03
## 448 R-HSA-5685942       59 0.4653526 0.9322903 0.4991498 3.574441 3.128783e-04
## 48   R-HSA-141405       20 0.7957718 1.0226913 0.7781153 3.558800 2.507457e-03
## 49   R-HSA-141430       20 0.7957718 1.0226913 0.7781153 3.558800 2.507457e-03
## 451 R-HSA-5693537       28 0.6600863 1.1413451 0.5783407 3.492848 4.953680e-03
## 120  R-HSA-174048       23 0.7266543 1.0419438 0.6974026 3.484912 2.933971e-03
##
## 465                                                        Homo sapiens: Deposition of new CENPA
## 573
## 481
## 124                    Homo sapiens: APC/C:Cdh1 mediated degradation of Cdc20 and other APC/C:Cdh1 ta
## 132                                             Homo sapiens: Activation of APC/C and APC/C:Cdc20
## 131                                                                Homo sapiens: APC/C:Cdc20
## 130                                                          Homo sapiens: Regulation of APC/C a
## 137                    Homo sapiens: APC:Cdc20 mediated degradation of cell cycle proteins prior to
## 123                                                                                Homo sapiens: APC
## 472
## 545
## 125                                                                      Homo sapiens: Cdc20:Phospl
## 121                                                                                  Homo sapien
## 136                                                                                 Homo sapiens
## 454                                                      Homo sapiens: Resolution of D-loop Structures
## 448                                                                                Homo sapiens: HD
## 48  Homo sapiens: Inhibition of the proteolytic activity of APC/C required for the onset of anaphase
## 49                                                         Homo sapiens: Inactivation of APC/C vi
```

```
## 451                                                                                                  Homo
## 120                                                                                 Homo sapiens: APC,
##          padjust
## 465 4.814222e-03
## 573 4.814222e-03
## 481 3.635722e-07
## 124 4.479927e-05
## 132 9.271358e-04
## 131 1.330983e-03
## 130 2.185424e-03
## 137 3.298021e-03
## 123 2.842228e-04
## 472 7.973210e-03
## 545 8.453810e-03
## 125 4.246741e-03
## 121 4.455002e-04
## 136 9.924930e-03
## 454 2.600589e-02
## 448 4.246741e-03
## 48  1.728553e-02
## 49  1.728553e-02
## 451 2.769933e-02
## 120 1.911354e-02
```

Extra Deseq

source: https://gif.biotech.iastate.edu/rnaseq-analysis-walk-through

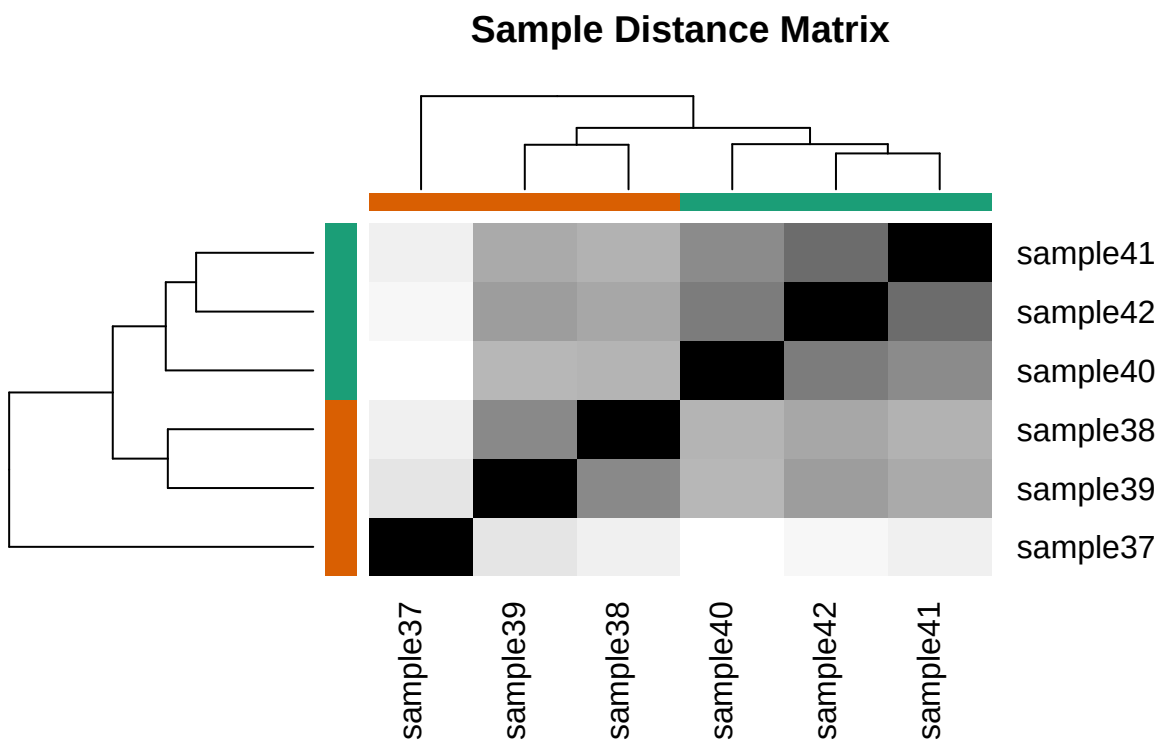
```
#Alternative Sample Distance matrix
(mycols <- brewer.pal(8, "Dark2")[1:length(unique(meta$Condition))])
```

```
## [1] "#1B9E77" "#D95F02"
```

```
# Sample distance heatmap
sampleDists <- as.matrix(dist(t(assay(rld))))
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

```
heatmap.2(as.matrix(sampleDists), key=F, trace="none",
col=colorpanel(100, "black", "white"),
ColSideColors=mycols[meta$Condition], RowSideColors=mycols[meta$Condition],
margin=c(10, 10), main="Sample Distance Matrix")
```




```
#PCA Analysis
rld_pca <- function (rld, intgroup = "Condition", ntop = 500, colors=NULL, legendpos="bottomleft", main=
require(genefilter)
require(calibrate)
require(RColorBrewer)
rv = rowVars(assay(rld))
select = order(rv, decreasing = TRUE)[seq_len(min(ntop, length(rv)))]
pca = prcomp(t(assay(rld)[select, ]))
fac = factor(apply(as.data.frame(colData(rld)[, intgroup, drop = FALSE]), 1, paste, collapse = " : "))
if (is.null(colors)) {
if (nlevels(fac) >= 3) {
colors = brewer.pal(nlevels(fac), "Paired")
} else {
colors = c("black", "red")
}
}
pc1var <- round(summary(pca)$importance[2,1]*100, digits=1)
pc2var <- round(summary(pca)$importance[2,2]*100, digits=1)
pc1lab <- paste0("PC1 (",as.character(pc1var),"%)")
pc2lab <- paste0("PC1 (",as.character(pc2var),"%)")
```

```
plot(PC2~PC1, data=as.data.frame(pca$x), bg=colors[fac], pch=21, xlab=pc1lab, ylab=pc2lab, main=main, .
with(as.data.frame(pca$x), textxy(PC1, PC2, labs=rownames(as.data.frame(pca$x)), cex=textcx))
legend(legendpos, legend=levels(fac), col=colors, pch=20)
# rldyplot(PC2 ~ PC1, groups = fac, data = as.data.frame(pca$rld),
# pch = 16, cerld = 2, aspect = "iso", col = colours, main = draw.key(key = list(rect = list(col = colo
# terldt = list(levels(fac)), rep = FALSE))
}
rld_pca(rld, colors=mycols, intgroup="Condition", xlim=c(-75, 35))
```

```
## Loading required package: genefilter
```

```
##
## Attaching package: 'genefilter'
```

```
## The following object is masked from 'package:readr':
##
##     spec
```

```
## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

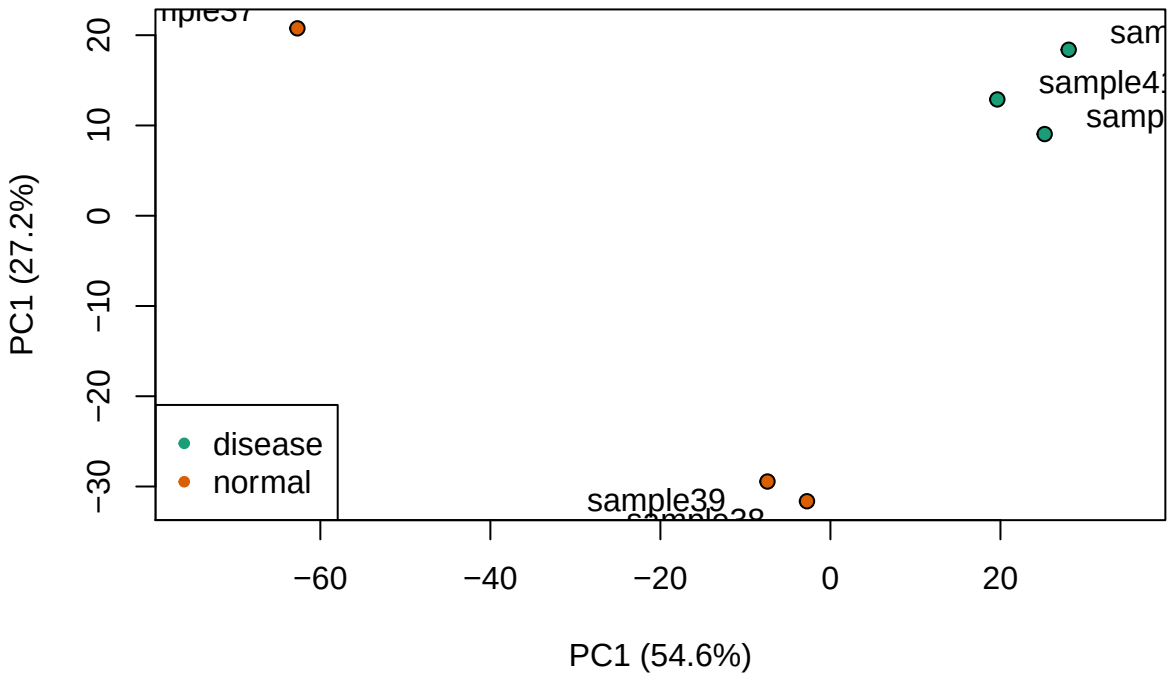
```
## Loading required package: calibrate
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:genefilter':
##
##     area
```

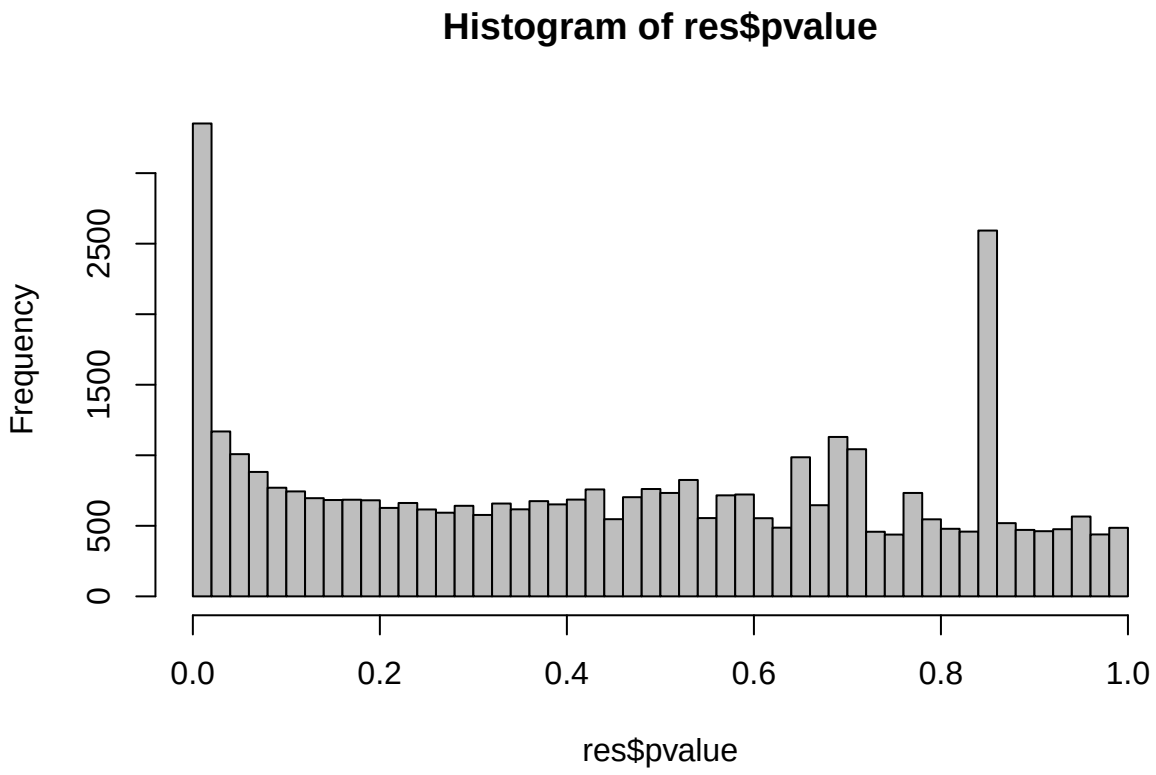
```
## The following object is masked from 'package:AnnotationDbi':
##
##     select
```




```
#Getting DE results
resdata <- merge(as.data.frame(res), as.data.frame(counts(dds, normalized=TRUE)), by="row.names", sort=F
names(resdata)[1] <- "Gene"
head(resdata)
```

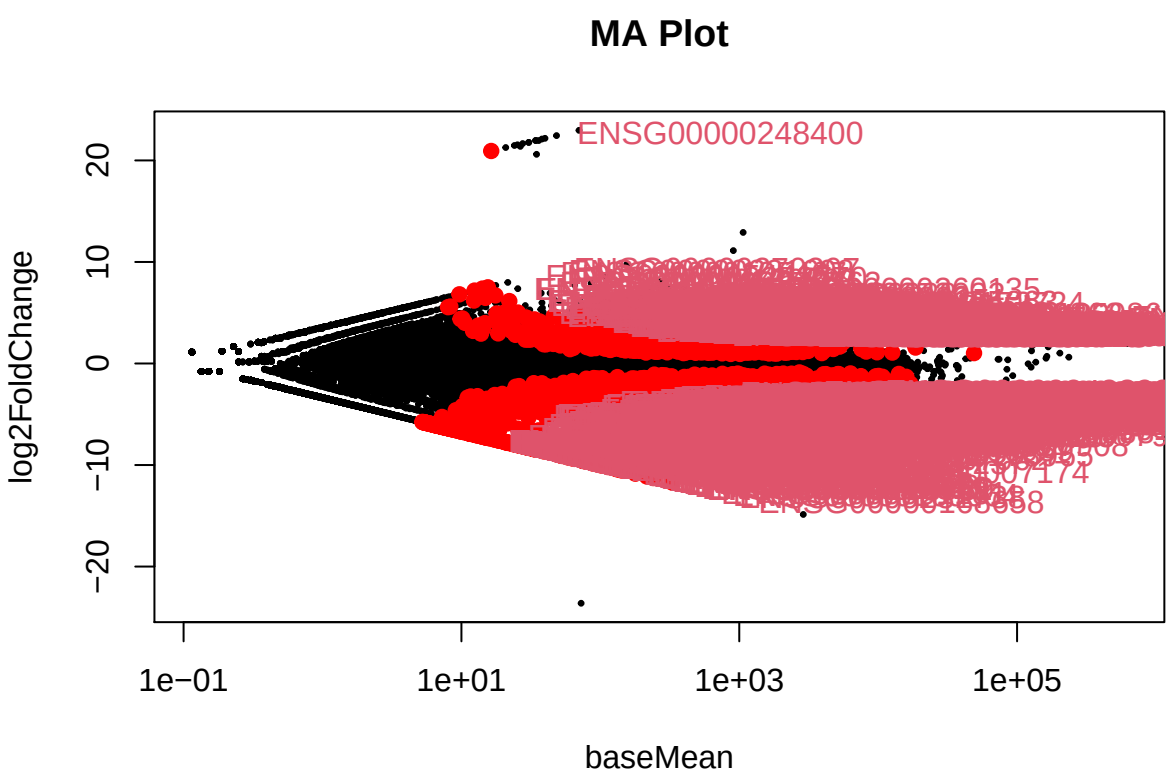
```
##              Gene   baseMean log2FoldChange     lfcSE       stat     pvalue
## 1 ENSG00000223972  0.0000000             NA        NA         NA         NA
## 2 ENSG00000227232 71.8771915      0.6173894 0.5478827  1.1268643 0.25979987
## 3 ENSG00000278267  9.3285305      2.5398109 1.0679460  2.3782204 0.01739642
## 4 ENSG00000243485  0.1522112     -0.7808149 4.0804729 -0.1913540 0.84824824
## 5 ENSG00000284332  0.0000000             NA        NA         NA         NA
## 6 ENSG00000237613  0.3334645     -1.7718750 4.0302204 -0.4396472 0.66019267
##        padj hgnc_symbol  entrezid sample37 sample38  sample39   sample40
## 1        NA     DDX11L1 100287102  0.00000  0.00000   0.00000  0.0000000
## 2 0.5739578        <NA>      <NA> 77.43823 54.44863 128.65240 51.1429665
## 3 0.1372147   MIR6859-1 102466751 10.42438 12.47781  24.76731  1.8265345
## 4        NA        <NA>      <NA>  0.00000  0.00000   0.00000  0.9132673
## 5        NA   MIR1302-2 100302278  0.00000  0.00000   0.00000  0.0000000
## 6        NA     FAM138A    645520  0.00000  0.00000   0.00000  0.9132673
##    sample41  sample42
## 1  0.000000  0.000000
## 2 64.163662 55.417262
## 3  3.262559  3.212595
## 4  0.000000  0.000000
## 5  0.000000  0.000000
## 6  1.087520  0.000000
```

```
#Examine plot for pvalue
hist(res$pvalue, breaks=50, col="grey")
```

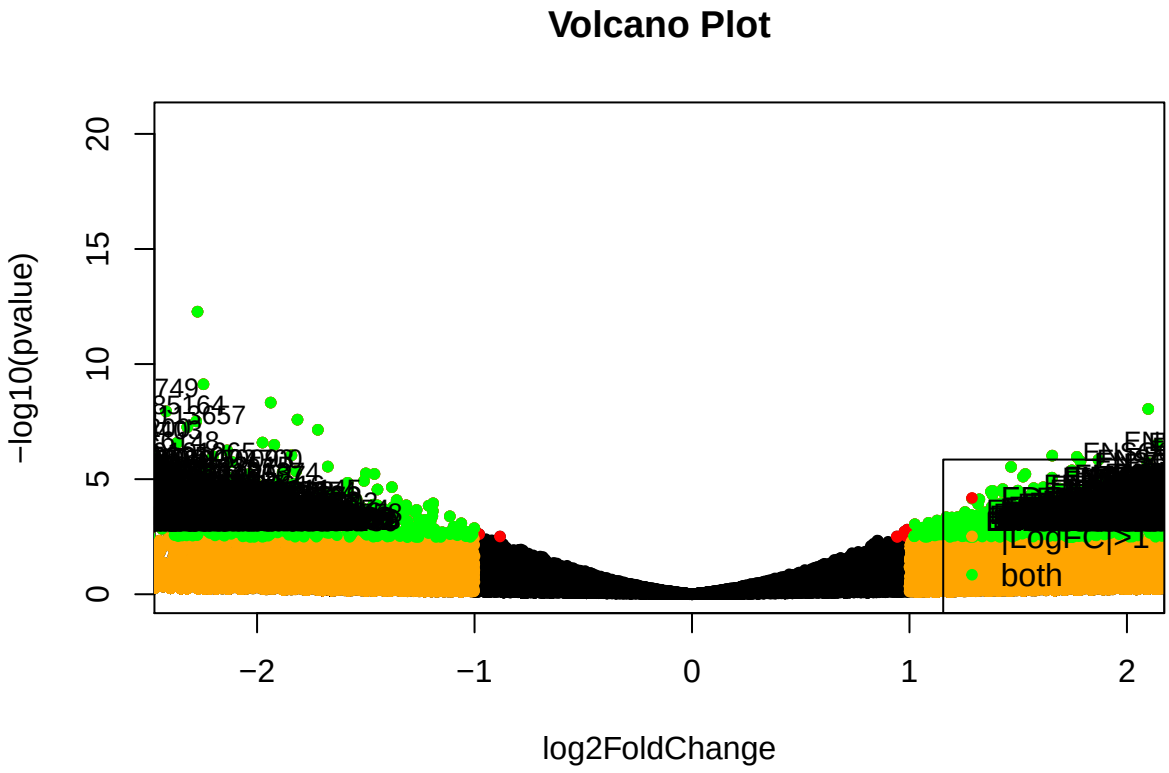



```
## MA plot
## Could do with built-in DESeq2 function:
## DESeq2::plotMA(dds, ylim=c(-1,1), cex=1)
## I like mine better:
maplot <- function (res, thresh=0.05, labelsig=TRUE, textcx=1, ...) {
with(res, plot(baseMean, log2FoldChange, pch=20, cex=.5, log="x", ...))
with(subset(res, padj<thresh), points(baseMean, log2FoldChange, col="red", pch=20, cex=1.5))
if (labelsig) {
require(calibrate)
with(subset(res, padj<thresh), textxy(baseMean, log2FoldChange, labs=Gene, cex=textcx, col=2))
}
}
maplot(resdata, main="MA Plot")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 22423 x values <= 0 omitted
## from logarithmic plot
```




```
# Volcano plot with "significant" genes labeled
volcanoplot <- function (res, lfcthresh=2, sigthresh=0.05, main="Volcano Plot", legendpos="bottomright"
with(res, plot(log2FoldChange, -log10(pvalue), pch=20, main=main, ...))
with(subset(res, padj<sigthresh ), points(log2FoldChange, -log10(pvalue), pch=20, col="red", ...))
with(subset(res, abs(log2FoldChange)>lfcthresh), points(log2FoldChange, -log10(pvalue), pch=20, col="or
with(subset(res, padj<sigthresh & abs(log2FoldChange)>lfcthresh), points(log2FoldChange, -log10(pvalue)
if (labelsig) {
require(calibrate)
with(subset(res, padj<sigthresh & abs(log2FoldChange)>lfcthresh), textxy(log2FoldChange, -log10(pvalue)
}
legend(legendpos, xjust=1, yjust=1, legend=c(paste("FDR<",sigthresh,sep=""), paste("|LogFC|>",lfcthresh
}
volcanoplot(resdata, lfcthresh=1, sigthresh=0.05, textcx=.8, xlim=c(-2.3, 2))
```



.

Working With Sleuth

Sleuth is a fast, lightweight tool that uses transcript abundance estimates output from pseudo-alignment algorithms that use bootstrap sampling, such as Sailfish, Salmon, and Kallisto, to perform differential expression analysis of gene isoforms.

```
#devtools::install_github("pachterlab/sleuth")
library(sleuth)

#suppressMessages({
#  library("sleuth")
#})

#Dir where kallisto results are stored
kal_dir <- "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto"
sample_id <- dir(file.path(kal_dir))  #This stores the directories in the kallisto workspace
sample_id
```

```
## [1] "sample37" "sample38" "sample39" "sample40" "sample41" "sample42"
```

```
#Specifying the path to the files
kal_dirs <- file.path(kal_dir, sample_id, "abundance.h5")
kal_dirs
```

```
## [1] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
## [2] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
## [3] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
## [4] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
## [5] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
## [6] "/Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sampl
```

```
#Appending sample names from meta to their paths
metadata <- read.table("practice.dataset.metadata.tsv", header = T)
metadata
```

```
##   SampleID Condition
## 1 sample37    normal
## 2 sample38    normal
## 3 sample39    normal
## 4 sample40   disease
## 5 sample41   disease
## 6 sample42   disease
```

```
s2c <- dplyr::mutate(metadata, path = kal_dirs)
print(s2c)
```

```
##   SampleID Condition
## 1 sample37    normal
## 2 sample38    normal
## 3 sample39    normal
## 4 sample40   disease
## 5 sample41   disease
## 6 sample42   disease
##
## 1 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample37
## 2 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample38
## 3 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample39
## 4 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample40
## 5 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample41
## 6 /Users/kakembo/Desktop/EANBiT_Documents/RT_2020_Program/RNASeq_miniproject/Counts/Kallisto/sample42
```

```
colnames(s2c)[1] <- "sample"  #Sleuth wants the sample names to be specifically as "sample"

#Creating the sleuth object
## Step 1: load the kallisto processed data into the object

#Here i had to rerun the kallisto output with bootstrap values specified ie adding    -b 100
so <- sleuth_prep(s2c, extra_bootstrap_summary = TRUE)
```

```
## Warning in check_num_cores(num_cores): It appears that you are running Sleuth from within Rstudio.
## Because of concerns with forking processes from a GUI, 'num_cores' is being set to 1.
## If you wish to take advantage of multiple cores, please consider running sleuth from the command lin
```

```
## reading in kallisto results

## dropping unused factor levels

## ......
## normalizing est_counts
## 70769 targets passed the filter
## normalizing tpm
## merging in metadata
## summarizing bootstraps
## ......
```

```
#Fiiting the full model
so <- sleuth_fit(so, ~Condition, 'full')
```

```
## fitting measurement error models
## shrinkage estimation
## 3 NA values were found during variance shrinkage estimation due to mean observation values outside o
## The LOESS fit will be repeated using exact computation of the fitted surface to extrapolate the missi
## These are the target ids with NA values: ENST00000304883.3|ENSG00000169836.5|OTTHUMG00000131124.2|OT
## computing variance of betas
```

```
#To test for transcripts that are differential expressed between the conditions, sleuth performs a seco
#The reduced model is fit
so <- sleuth_fit(so, ~1, 'reduced')
```

```
## fitting measurement error models
## shrinkage estimation
## 3 NA values were found during variance shrinkage estimation due to mean observation values outside o
## The LOESS fit will be repeated using exact computation of the fitted surface to extrapolate the missi
## These are the target ids with NA values: ENST00000304883.3|ENSG00000169836.5|OTTHUMG00000131124.2|OT
## computing variance of betas
```

```
# Perform a Likelihood Ratio Test for the 2 tests, so as to nest reduced model into full model
so <- sleuth_lrt(so, 'reduced', 'full')

#Checking the fitted models using model()
models(so)
```

```
## [  full  ]
## formula:  ~Condition
## data modeled:  obs_counts
## transform sync'ed:  TRUE
## coefficients:
##  (Intercept)
##      Conditionnormal
## [  reduced  ]
## formula:  ~1
## data modeled:  obs_counts
## transform sync'ed:  TRUE
## coefficients:
##  (Intercept)
```

```r
#Examing the results of the test
sleuth_table <- sleuth_results(so, 'reduced:full', 'lrt', show_all = FALSE)
head(sleuth_table)
```

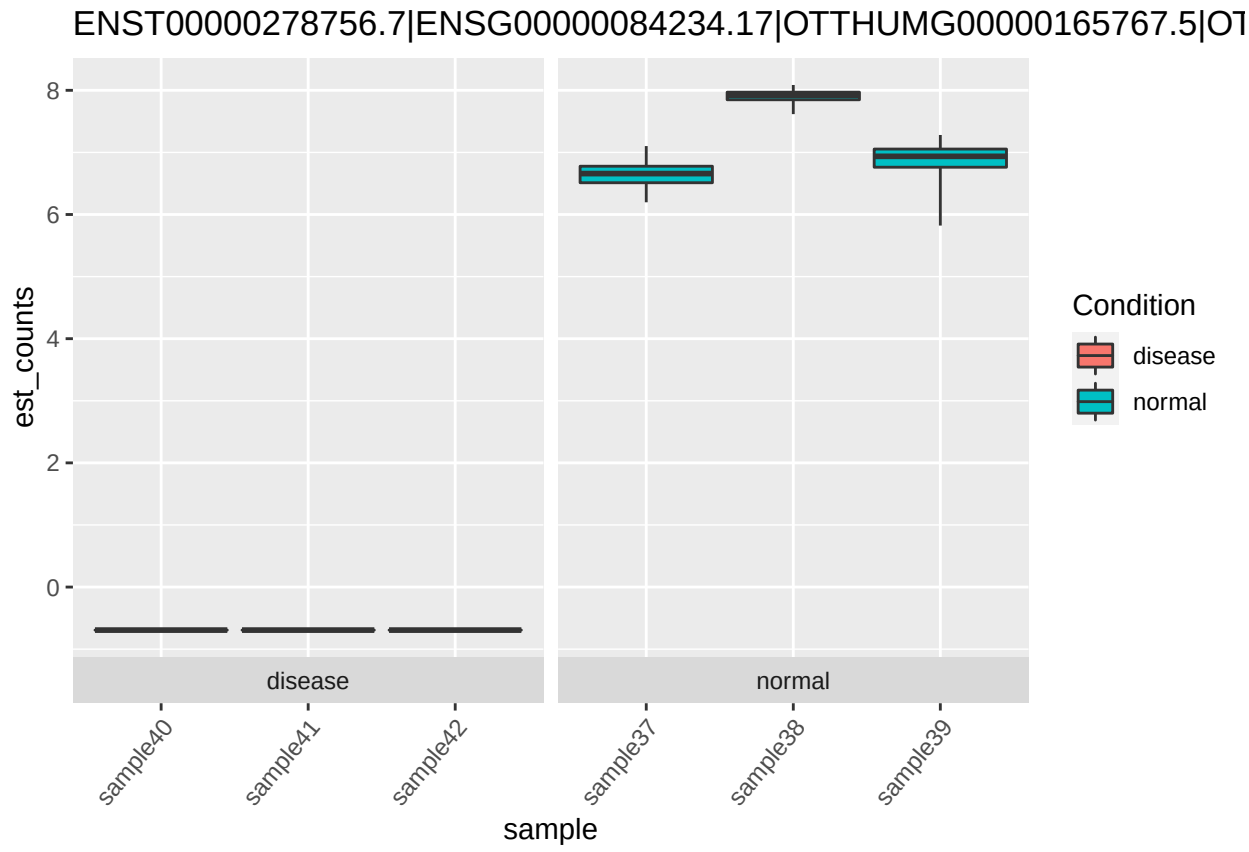
```
##
## 1  ENST00000278756.7|ENSG00000084234.17|OTTHUMG00000165767.5|OTTHUMT00000386112.2|APLP2-202|APLP2|368
## 2      ENST00000356005.8|ENSG00000115414.20|OTTHUMG00000133054.7|OTTHUMT00000337234.1|FN1-204|FN1|784
## 3    ENST00000381886.8|ENSG00000187555.16|OTTHUMG00000176903.7|OTTHUMT00000434623.2|USP7-202|USP7|358
## 4  ENST00000382329.1|ENSG00000107099.15|OTTHUMG00000078789.9|OTTHUMT00000192534.2|DOCK8-201|DOCK8|611
## 5    ENST00000383314.6|ENSG00000101654.17|OTTHUMG00000131718.5|OTTHUMT00000254636.1|RNMT-202|RNMT|625
## 6 ENST00000392318.8|ENSG00000128641.19|OTTHUMG00000132718.11|OTTHUMT00000334774.2|MYO1B-204|MYO1B|506
##           pval        qval test_stat       rss degrees_free mean_obs  var_obs
## 1 1.619914e-06 0.006377497  23.00012  93.16484            1 3.227860 18.63297
## 2 3.233072e-06 0.006377497  21.67302 140.09044            1 4.076839 28.01809
## 3 1.401548e-06 0.006377497  23.27855  88.63185            1 3.144242 17.72637
## 4 1.612249e-06 0.006377497  23.00924  87.60538            1 3.124519 17.52108
## 5 2.514820e-06 0.006377497  22.15514  80.41480            1 2.966328 16.08296
## 6 2.166691e-06 0.006377497  22.44126  82.95814            1 3.021751 16.59163
##      tech_var sigma_sq smooth_sigma_sq final_sigma_sq
## 1 0.018266598 18.61470       0.9985521       18.61470
## 2 0.032289882 27.98580       0.5096037       27.98580
## 3 0.002497197 17.72387       1.0721320       17.72387
## 4 0.039249552 17.48183       1.0905369       17.48183
## 5 0.001374773 16.08159       1.2482186       16.08159
## 6 0.007113947 16.58451       1.1914252       16.58451
```

```r
#Filtering only significant results
#sleuth_significant <- dplyr::filter(sleuth_table, qval <= 0.05)
sleuth_significant <- dplyr::filter(sleuth_table, qval <= 0.1)
#Top 20 significant genes with BH testing at q-value <= 0.1
head(sleuth_significant, 20)
```

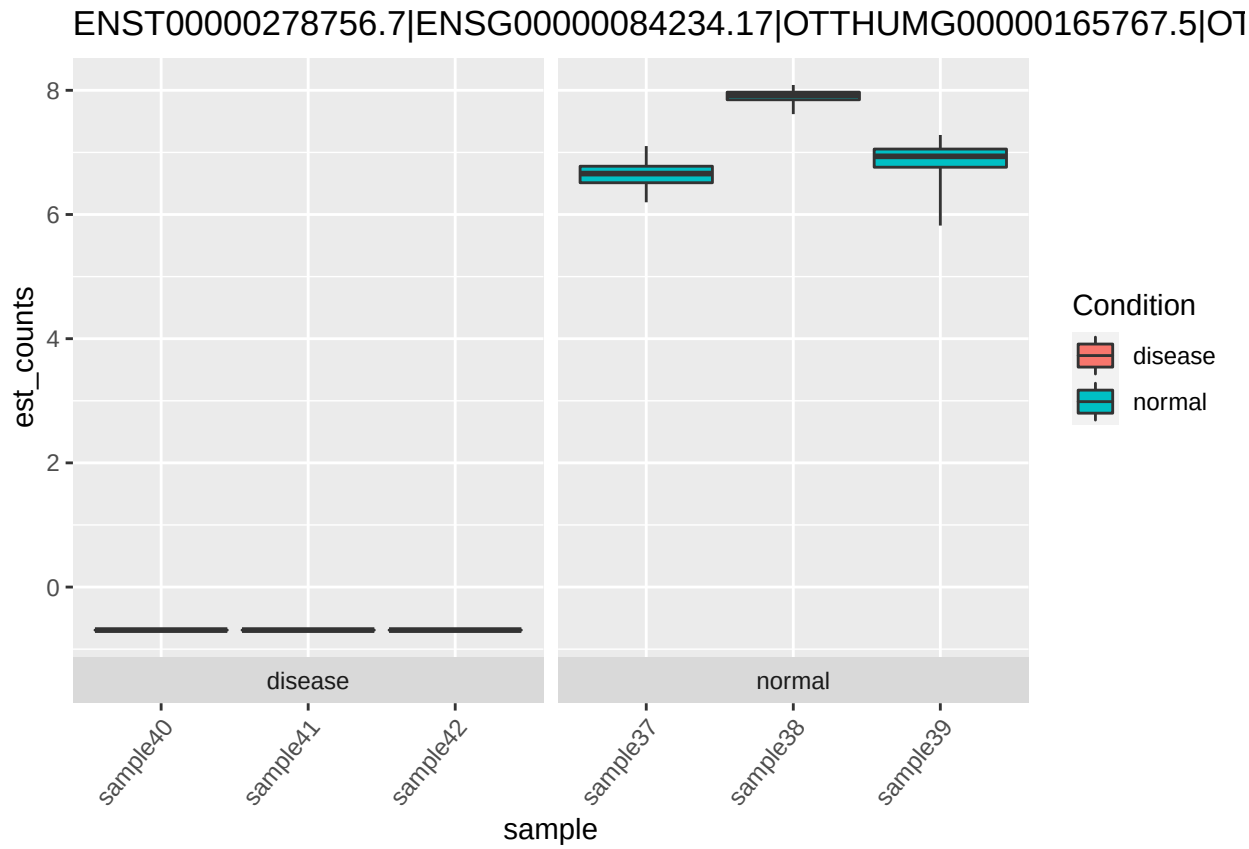
```
##
## 1            ENST00000278756.7|ENSG00000084234.17|OTTHUMG00000165767.5|OTTHUMT00000386112.2|APLP2-202
## 2               ENST00000356005.8|ENSG00000115414.20|OTTHUMG00000133054.7|OTTHUMT00000337234.1|FN1-2
## 3             ENST00000381886.8|ENSG00000187555.16|OTTHUMG00000176903.7|OTTHUMT00000434623.2|USP7-20
## 4            ENST00000382329.1|ENSG00000107099.15|OTTHUMG00000078789.9|OTTHUMT00000192534.2|DOCK8-201
## 5              ENST00000383314.6|ENSG00000101654.17|OTTHUMG00000131718.5|OTTHUMT00000254636.1|RNMT-20
## 6           ENST00000392318.8|ENSG00000128641.19|OTTHUMG00000132718.11|OTTHUMT00000334774.2|MYO1B-204
## 7           ENST00000395851.5|ENSG00000141027.21|OTTHUMG00000059309.12|OTTHUMT00000131753.3|NCOR1-204
## 8                          ENST00000399339.6|ENSG00000099290.17|OTTHUMG00000018225.5|-|WASHC2A-204|W
## 9             ENST00000440577.5|ENSG00000055917.15|OTTHUMG00000122098.4|OTTHUMT00000323918.1|PUM2-20
## 10           ENST00000441802.7|ENSG00000083857.14|OTTHUMG00000160320.10|OTTHUMT00000360209.4|FAT1-201
## 11 ENST00000444878.5|ENSG00000131051.24|OTTHUMG00000032358.21|OTTHUMT00000126527.4|RBM39-216|RBM39|21
## 12            ENST00000446688.5|ENSG00000079308.19|OTTHUMG00000133056.10|OTTHUMT00000339212.2|TNS1-20
## 13         ENST00000471714.5|ENSG00000154175.17|OTTHUMG00000159094.7|OTTHUMT00000353261.1|ABI3BP-206
## 14          ENST00000475937.5|ENSG00000114861.22|OTTHUMG00000158803.19|OTTHUMT00000352253.1|FOXP1-209
## 15          ENST00000482600.5|ENSG00000100129.18|OTTHUMG00000150671.8|OTTHUMT00000319553.1|EIF3L-213
## 16             ENST00000490680.5|ENSG00000065150.21|OTTHUMG00000017244.9|OTTHUMT00000354655.1|IPO5-21
## 17   ENST00000493560.5|ENSG00000143416.21|OTTHUMG00000012498.26|OTTHUMT00000034914.2|SELENBP1-218|SEL
## 18            ENST00000504734.5|ENSG00000127022.15|OTTHUMG00000130910.13|OTTHUMT00000371744.2|CANX-20
## 19          ENST00000507097.5|ENSG00000113013.15|OTTHUMG00000129206.7|OTTHUMT00000373736.1|HSPA9-207
## 20   ENST00000513303.5|ENSG00000250722.6|OTTHUMG00000162140.5|OTTHUMT00000367495.1|SELENOP-211|SELENO
```

```
##            pval        qval test_stat       rss degrees_free mean_obs   var_obs
## 1  1.619914e-06 0.006377497  23.00012  93.16484            1 3.227860 18.632969
## 2  3.233072e-06 0.006377497  21.67302 140.09044            1 4.076839 28.018087
## 3  1.401548e-06 0.006377497  23.27855  88.63185            1 3.144242 17.726371
## 4  1.612249e-06 0.006377497  23.00924  87.60538            1 3.124519 17.521075
## 5  2.514820e-06 0.006377497  22.15514  80.41480            1 2.966328 16.082961
## 6  2.166691e-06 0.006377497  22.44126  82.95814            1 3.021751 16.591628
## 7  2.748767e-06 0.006377497  21.98439  80.50778            1 2.967848 16.101555
## 8  2.838671e-06 0.006377497  21.92263  79.46068            1 2.943327 15.892135
## 9  3.604684e-06 0.006377497  21.46435  78.03064            1 2.906987 15.606129
## 10 2.903556e-06 0.006377497  21.87926  17.88933            1 5.451892  3.577866
## 11 3.066541e-06 0.006377497  21.77447  78.55607            1 2.923201 15.711213
## 12 7.602939e-07 0.006377497  24.45592 100.26942            1 3.384832 20.053883
## 13 2.758027e-06 0.006377497  21.97794  14.57780            1 6.100925  2.915560
## 14 3.140166e-06 0.006377497  21.72895  79.00810            1 2.930806 15.801620
## 15 3.356128e-06 0.006377497  21.60137  78.00471            1 2.912092 15.600942
## 16 3.165262e-06 0.006377497  21.71368  82.37154            1 3.005300 16.474308
## 17 2.536947e-06 0.006377497  22.13832  93.27118            1 3.217862 18.654236
## 18 6.478664e-07 0.006377497  24.76429 102.09808            1 3.418751 20.419615
## 19 3.353922e-06 0.006377497  21.60263  81.49928            1 2.989499 16.299856
## 20 5.603839e-07 0.006377497  25.04395 105.00910            1 3.475822 21.001820
##       tech_var  sigma_sq smooth_sigma_sq final_sigma_sq
## 1  0.018266598 18.614702       0.9985521      18.614702
## 2  0.032289882 27.985797       0.5096037      27.985797
## 3  0.002497197 17.723874       1.0721320      17.723874
## 4  0.039249552 17.481826       1.0905369      17.481826
## 5  0.001374773 16.081586       1.2482186      16.081586
## 6  0.007113947 16.584514       1.1914252      16.584514
## 7  0.025658308 16.075897       1.2466483      16.075897
## 8  0.002420824 15.889715       1.2720179      15.889715
## 9  0.010248796 15.595880       1.3096713      15.595880
## 10 0.033081305  3.544784       0.2205048       3.544784
## 11 0.006875747 15.704337       1.2928768      15.704337
## 12 0.044964827 20.008918       0.8833699      20.008918
## 13 0.039498220  2.876062       0.1725410       2.876062
## 14 0.003837949 15.797782       1.2849946      15.797782
## 15 0.032666960 15.568275       1.3043858      15.568275
## 16 0.081061238 16.393247       1.2081670      16.393247
## 17 0.027083301 18.627153       1.0069369      18.627153
## 18 0.004163799 20.415451       0.8609801      20.415451
## 19 0.108443653 16.191412       1.2243492      16.191412
## 20 0.001076709 21.000743       0.8238950      21.000743
```

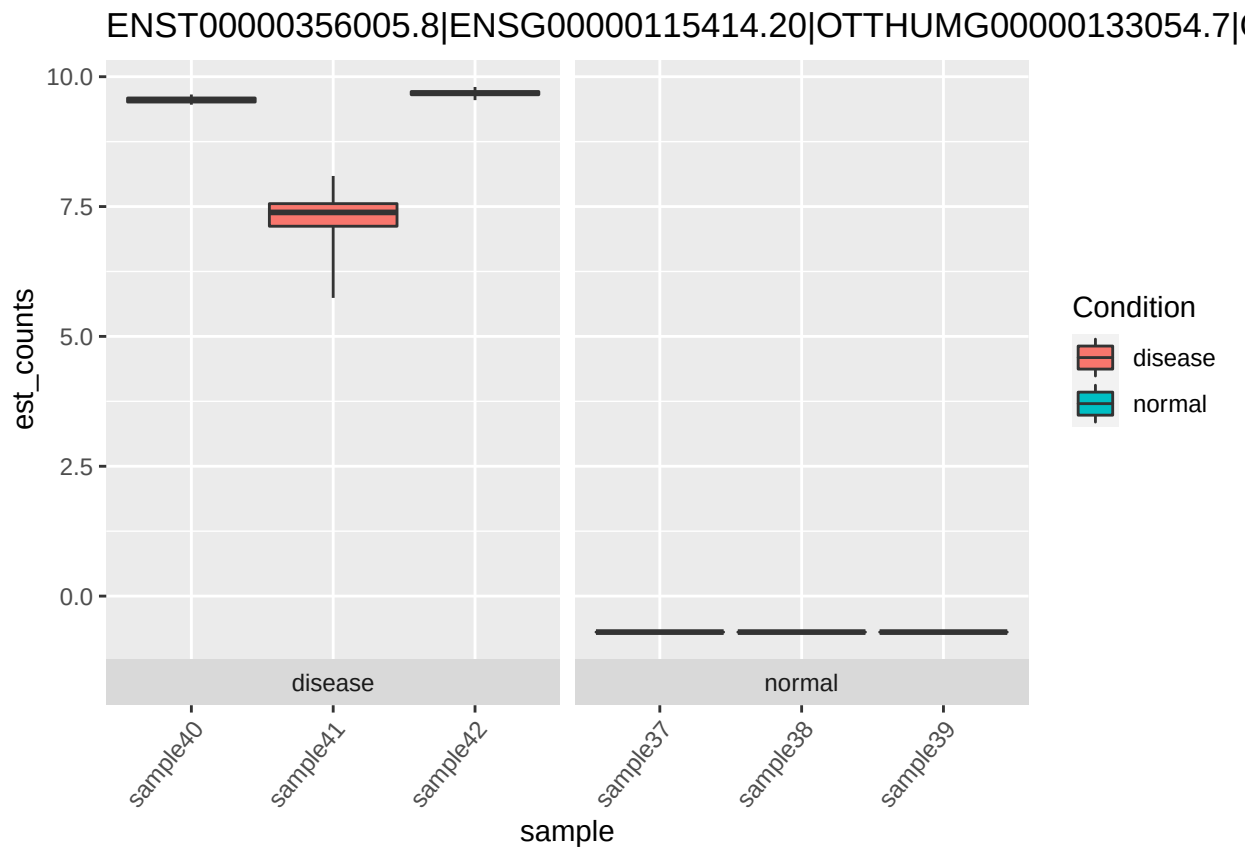
```
#Plots
plot_bootstrap(so, "ENST00000278756.7|ENSG00000084234.17|OTTHUMG00000165767.5|OTTHUMT00000386112.2|APLP
```



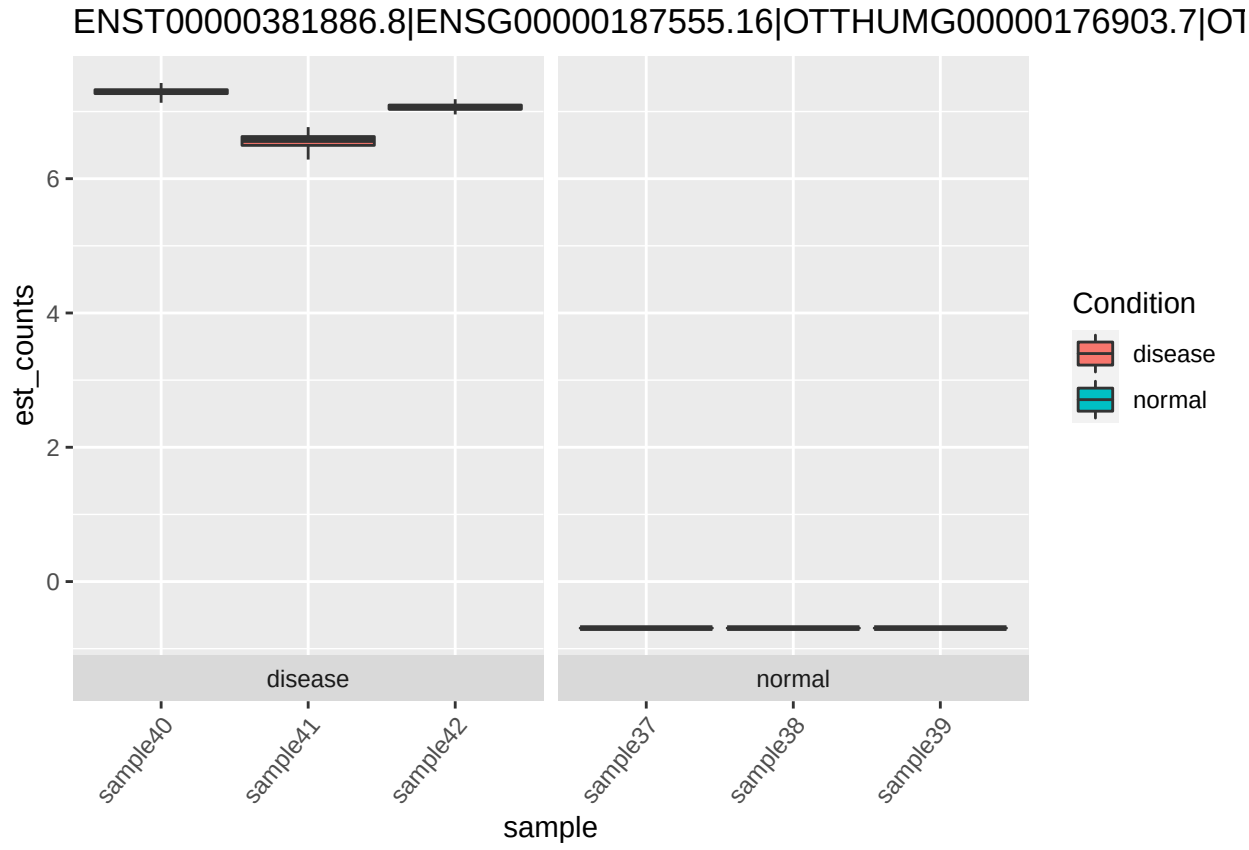

```
par(mfrow=c(2,3))
plot_bootstrap(so, sleuth_significant$target_id[1], units = "est_counts", color_by = "Condition")
```



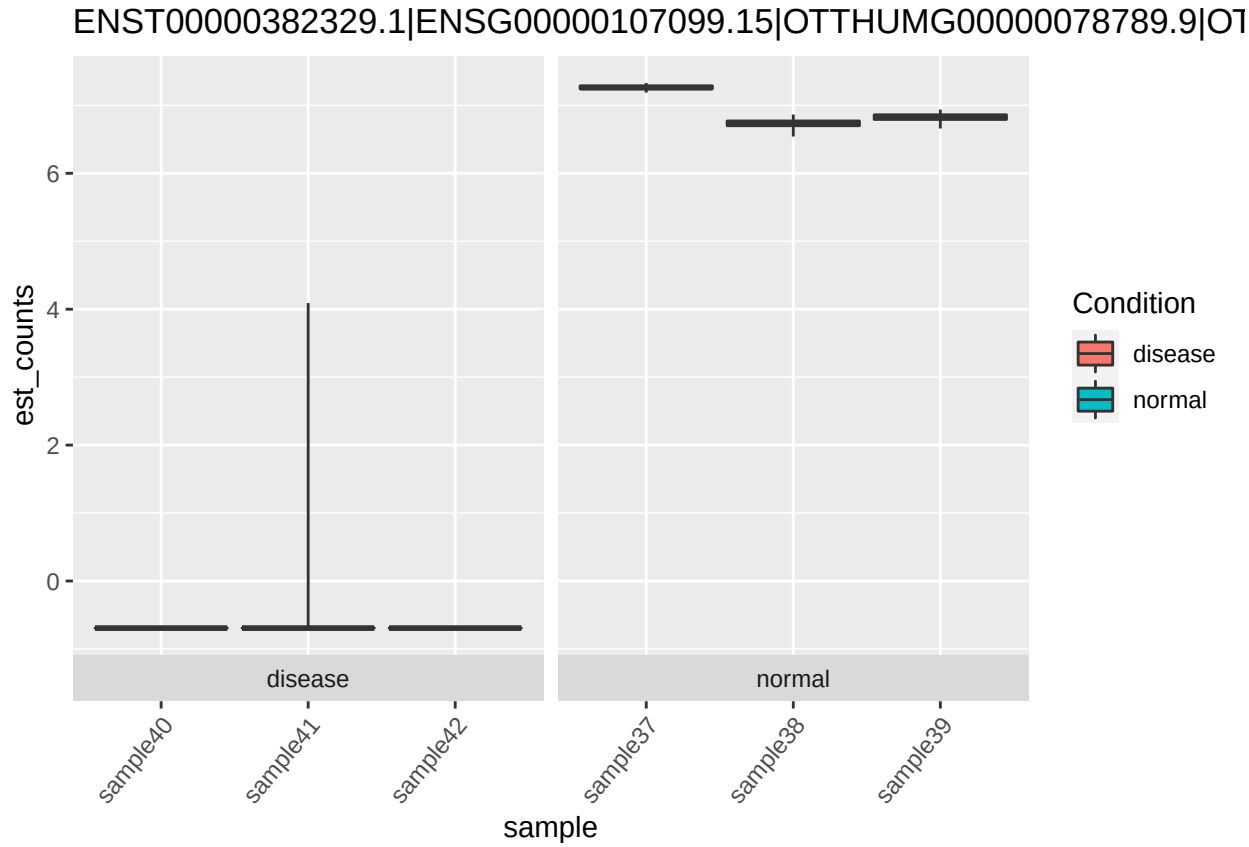

```
plot_bootstrap(so, sleuth_significant$target_id[2], units = "est_counts", color_by = "Condition")
```



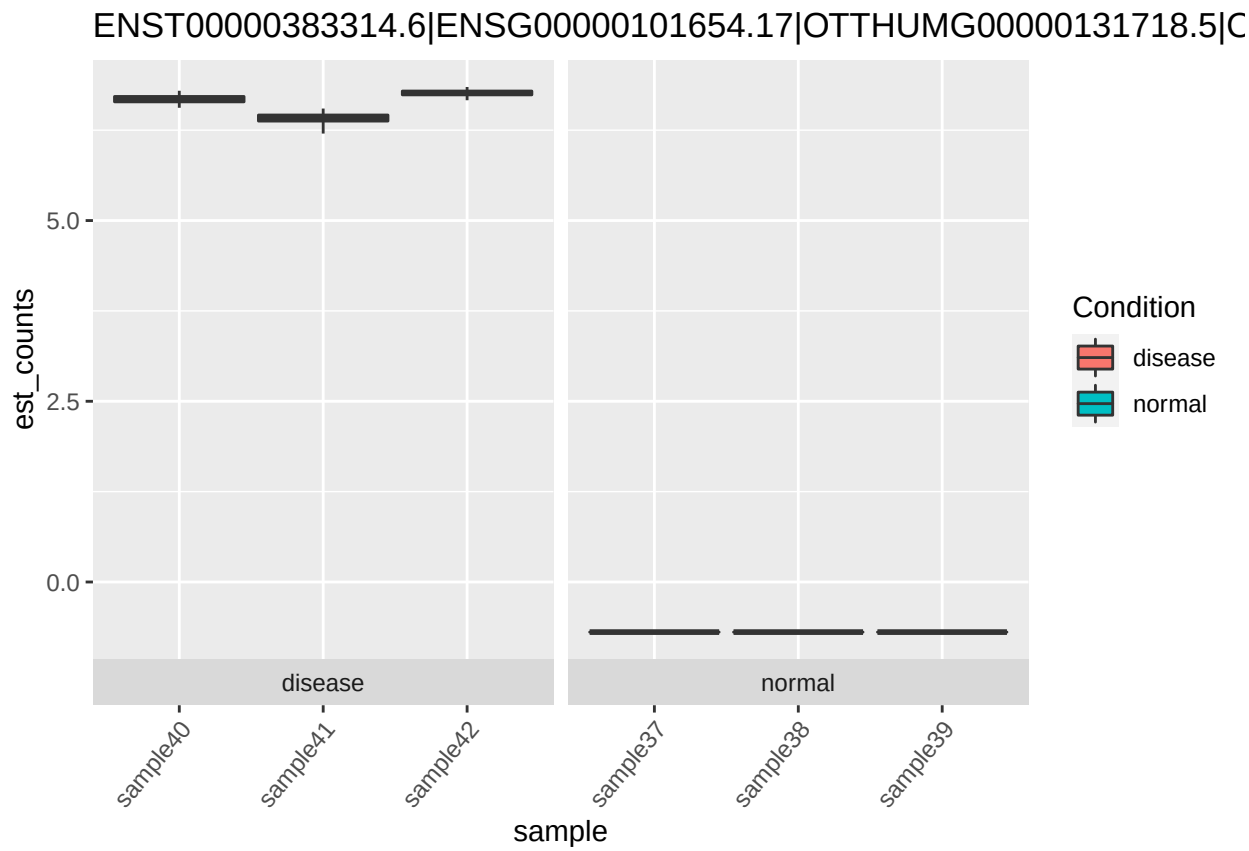

```
plot_bootstrap(so, sleuth_significant$target_id[3], units = "est_counts", color_by = "Condition")
```



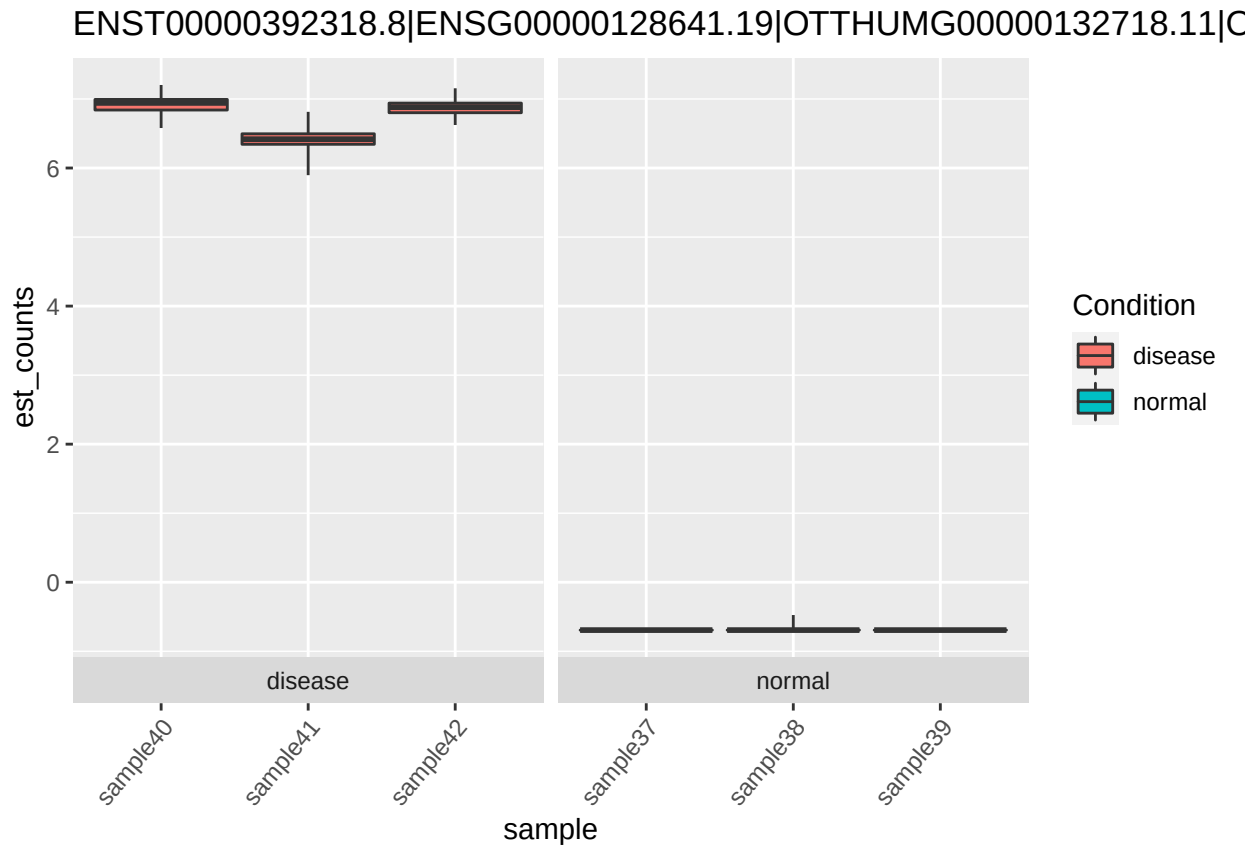

```
plot_bootstrap(so, sleuth_significant$target_id[4], units = "est_counts", color_by = "Condition")
```




```
plot_bootstrap(so, sleuth_significant$target_id[5], units = "est_counts", color_by = "Condition")
```




```
plot_bootstrap(so, sleuth_significant$target_id[6], units = "est_counts", color_by = "Condition")
```



Working with EdgeR

Source: https://gif.biotech.iastate.edu/rnaseq-analysis-walk-through I will implement output from hisat

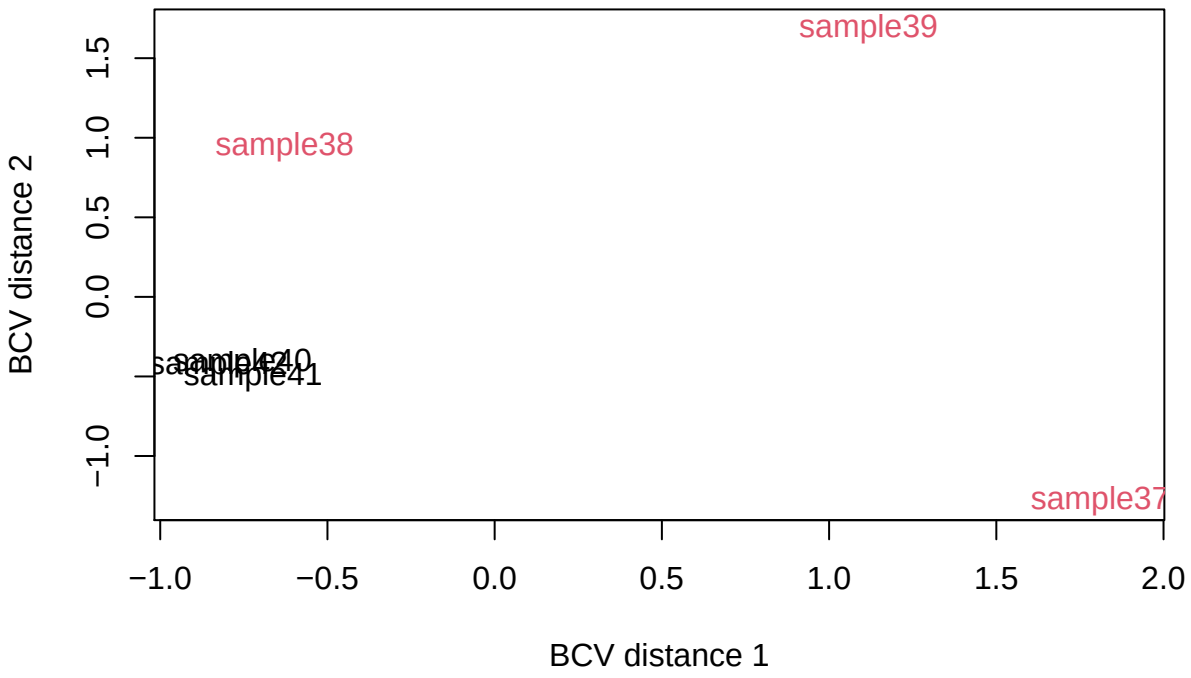
```
#BiocManager::install("edgeR")
library(edgeR)
```

```
## Loading required package: limma

##
## Attaching package: 'limma'

## The following object is masked _by_ '.GlobalEnv':
##
##     volcanoplot

## The following object is masked from 'package:DESeq2':
##
##     plotMA

## The following object is masked from 'package:BiocGenerics':
##
##     plotMA
```

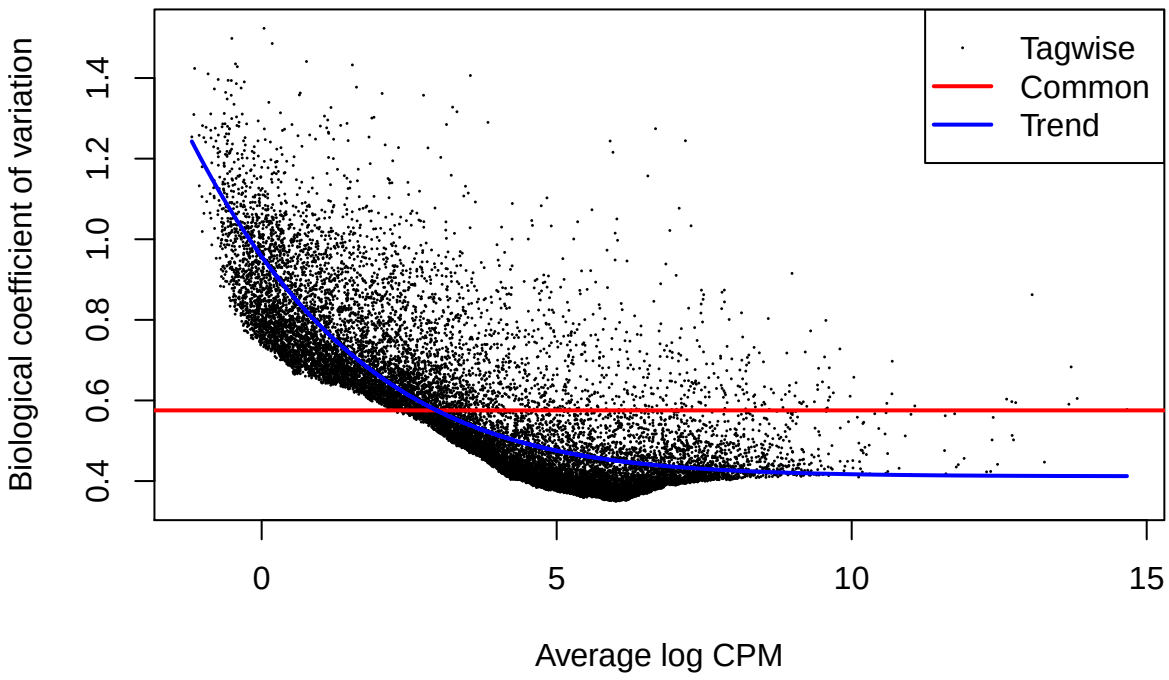
```r
# import data
#Using previously imported data for Hisat
datain <- countdata
head(datain)
```

```
##                 sample37 sample38 sample39 sample40 sample41 sample42
## ENSG00000223972        0        0        0        0        0        0
## ENSG00000227232       52       48      187       56       59       69
## ENSG00000278267        7       11       36        2        3        4
## ENSG00000243485        0        0        0        1        0        0
## ENSG00000284332        0        0        0        0        0        0
## ENSG00000237613        0        0        0        1        1        0
```

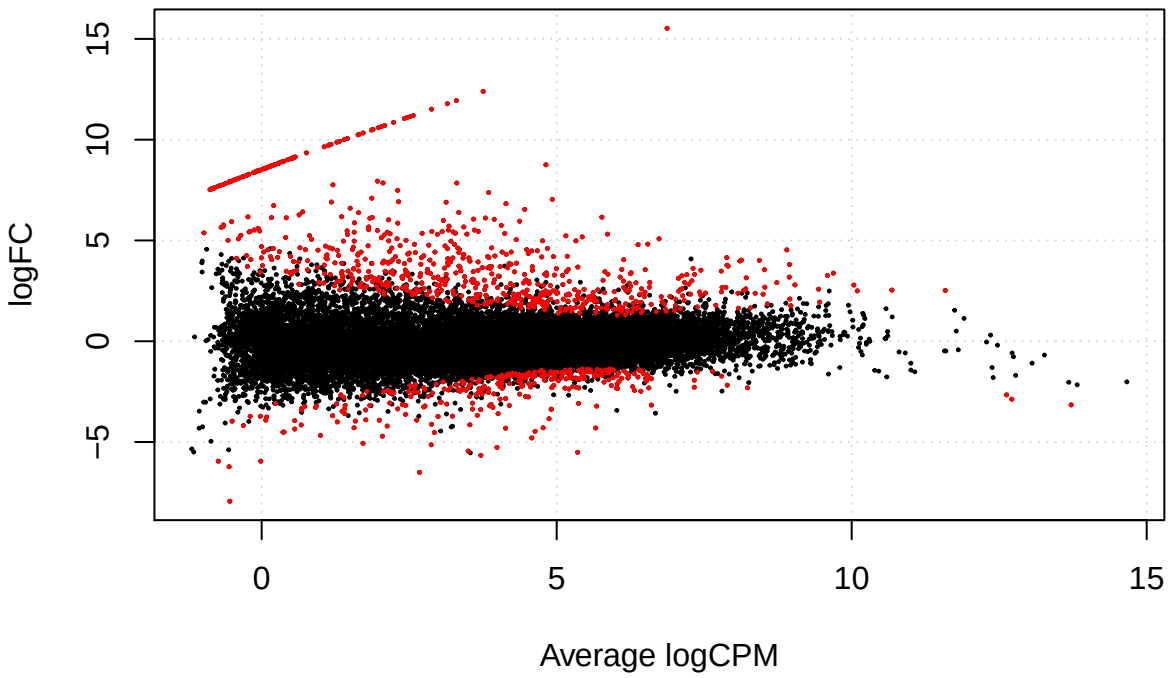
```r
# create DGE object of edgeR
dgList <- DGEList(counts=datain,group=factor(meta$Condition))

# filter data to retain genes that are represented at least 1 counts per million (cpm) in at least 2 sa
countsPerMillion <- cpm(dgList)   #Compute the counts per million
countCheck <- countsPerMillion > 1   #Check which has more than 1cpm
keep <- which(rowSums(countCheck) >= 2)  #Check if in a row atleast 2 samples have 1cpm
dgList <- dgList[keep,]  #Subset the dgList for only qualifying genes
dgList$samples$lib.size <- colSums(dgList$counts)

# normalization using TMM method
dgList <- calcNormFactors(dgList, method="TMM")

# data exploration
# Multi Dimension Scale, MDS plot
plotMDS(dgList, method="bcv", col=as.numeric(dgList$samples$group))
```



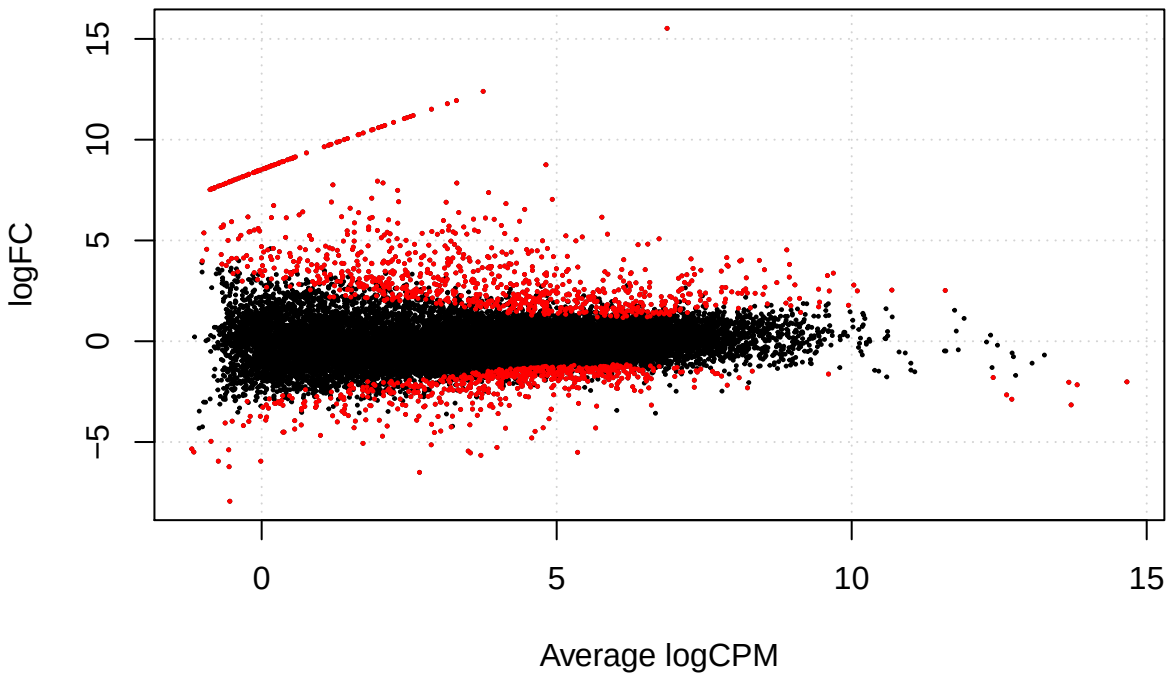

```
# Dispersion estimates
design.mat <- model.matrix(~ 0 + dgList$samples$group)
colnames(design.mat) <- levels(dgList$samples$group)
dgList <- estimateGLMCommonDisp(dgList,design.mat)
dgList <- estimateGLMTrendedDisp(dgList,design.mat, method="power")
dgList <- estimateGLMTagwiseDisp(dgList,design.mat)
plotBCV(dgList)
```




```
# Differentail expression analysis
fit <- glmFit(dgList, design.mat)  #fits genewise negative binomial glms
lrt <- glmLRT(fit, contrast=c(1,-1))  #conducts likelihood ratio tests for one or more coefficients in
edgeR_results <- topTags(lrt, n=Inf)

# plot log2FC of genes and highlight the DE genes at p = 0.05
deGenes <- decideTestsDGE(lrt, p=0.05)
deGenes <- rownames(lrt)[as.logical(deGenes)]
plotSmear(lrt, de.tags=deGenes)
```




```
# plot log2FC of genes and highlight the DE genes at p = 0.1
deGenes01 <- decideTestsDGE(lrt, p=0.1)
deGenes01 <- rownames(lrt)[as.logical(deGenes01)]
plotSmear(lrt, de.tags=deGenes01)
```


logFC
15
10
5
0
−5
0
5
10
15
Average logCPM

. . . . . .