Erica Chio, Assigned coursework #5

Due date: March 26th (two weeks), 11pm EST

Submission: PDF document by email to daniel.depledge@nyulangone.org

1. Datasets: HCMV-infected normal human dermal fibroblasts treated with either

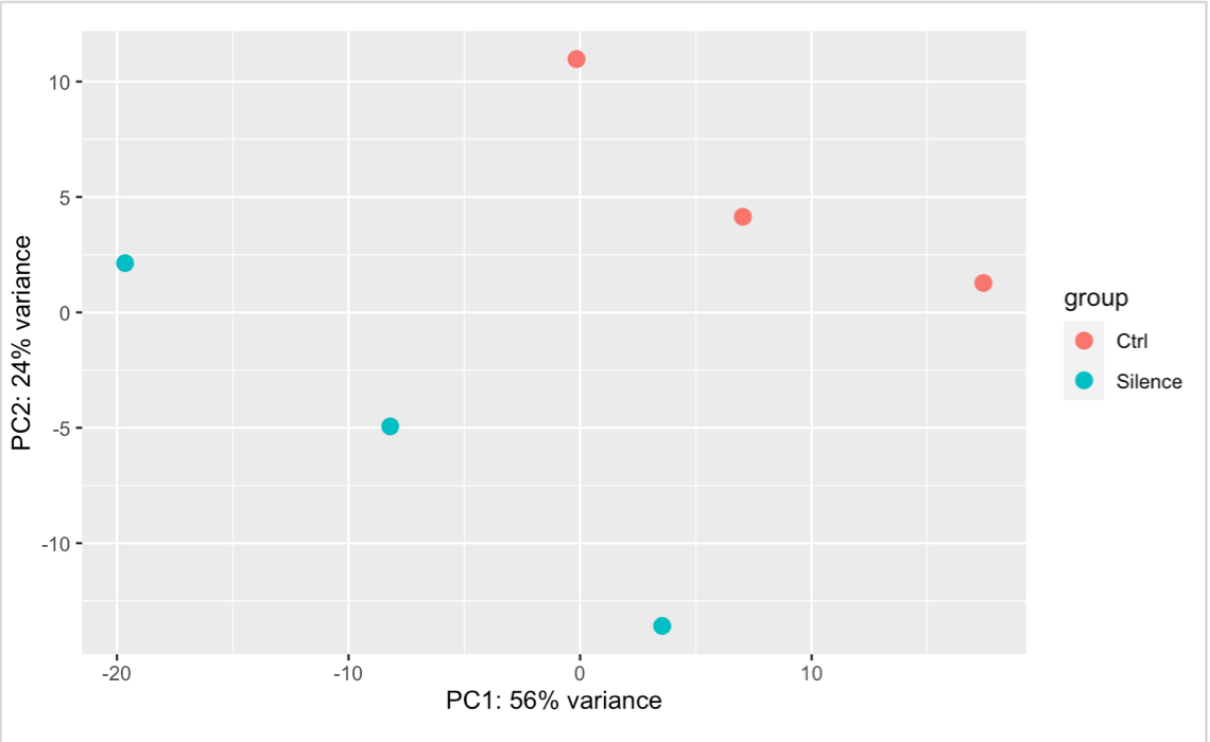
- a non-silencing control (LT34/LT35/LT36)
- an EIF3D-silencing siRNA (LT46/LT47/LT48)
- Batch #1 = LT34 & LT46
- Batch #2 = LT35 & LT47
- Batch #3 = LT36 & LT48

2. Use gene counts generated from transcriptome alignment [kallisto/salmon] you performed previously

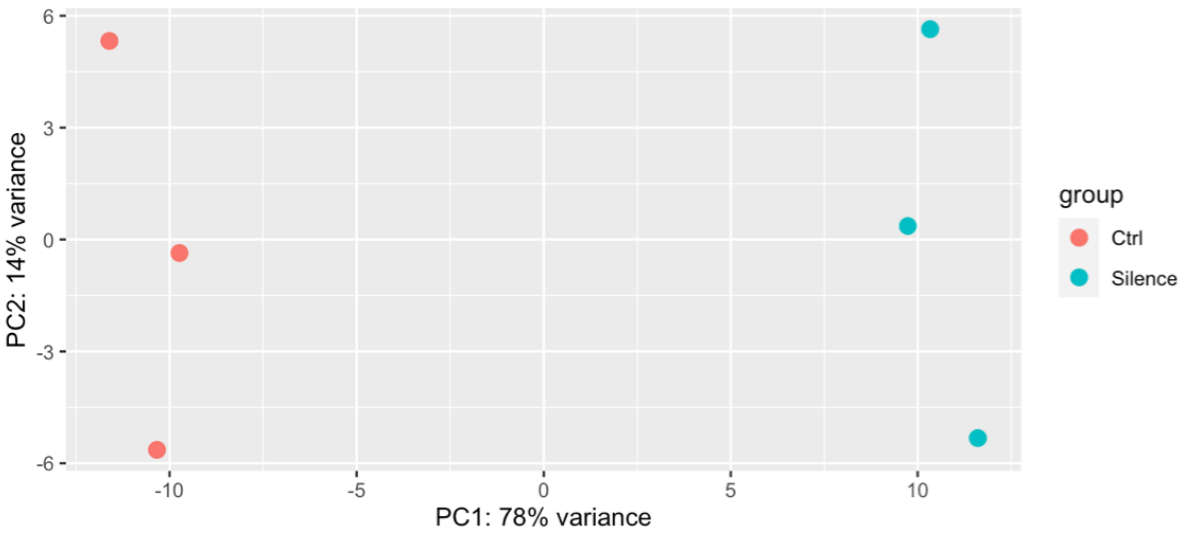
Preliminary Analysis on DATA

```
### LOAD REQUIRED LIBRARIES
library("DESeq2")
library("pheatmap")
library("RColorBrewer")
library("vsn")
library("AnnotationDbi")
library("org.Hs.eg.db")
library("genefilter")
library("biomaRt")
library("IHW")
library("ggplot2")

library("dplyr")
library("tibble")

library("edgeR")

### SET WORKING DIRECTORY
# note: this directory should be populated with the raw counts file
setwd("/Users/ericachio/Documents/sackler/applied informatics sequencing/
```

```
coursework5")

### Import count table and details on experimental design
# NB: Make sure column names in the sample(table) file and counts file are
exactly the same and in the same order
CountTable <- read.table("counts.txt", header=TRUE, row.names=1)
samples <- read.table("sample.txt", header=TRUE)
#Dataset <- DESeqDataSetFromMatrix(countData = CountTable, colData=samples,
design=~batch+condition)
Dataset <- DESeqDataSetFromMatrix(countData = CountTable, colData=samples,
design=~condition)

### PRELIMINARY ANALYSES ###
# The first steps in your analysis should focus on better understanding the
relationship of the datasets being studied. This can
# be simply achieved by generating a PCA plot showing the relationship of your
samples.
# First we transform our raw count data using a variance stabilizing
transformation (VST) that roughly mirrors how DeSeq2 models the data.
vsd1 <- varianceStabilizingTransformation(Dataset, blind=FALSE)

# Then we plot a PCA, grouping and coloring our datasets according to batch
plotPCA(vsd1, "condition")
```

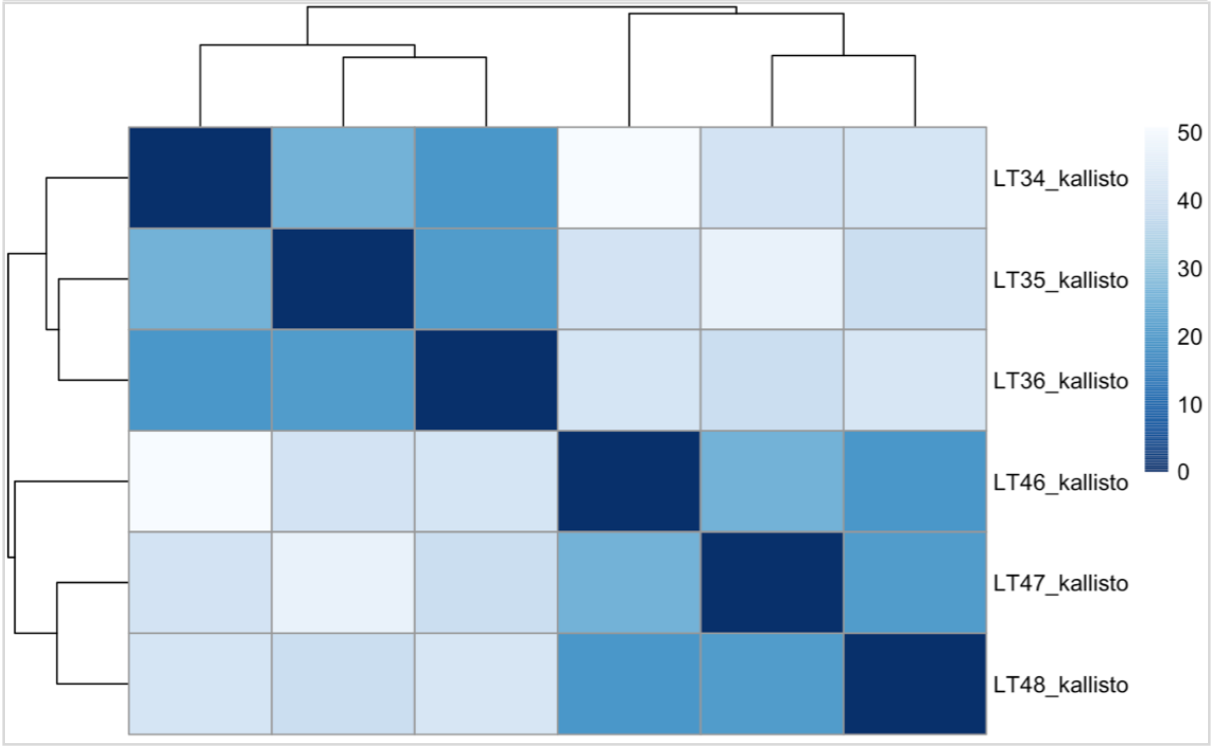


```
# note that you can attach additional information based on the column headers
in your sample table
plotPCA(vsd1, c("condition","batch"))

# we can also attempt to replicate the batch effect correction performed by
DeSeq2 using the limma::removeBatchEffect function
vsd2 <- varianceStabilizingTransformation(Dataset, blind=FALSE)
assay(vsd2) <- limma::removeBatchEffect(assay(vsd2), vsd2$batch)
plotPCA(vsd2, "condition")
```




```
# We can also calculate and plot sample distances using either the batch
corrected (vsd2) or uncorrected (vsd1) data.
# uncorrected
sampleDists <- dist( t( assay(vsd1) ) )
sampleDists
sampleDistMatrix <- as.matrix( sampleDists )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Reds")) )(255)
pheatmap(sampleDistMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)
```

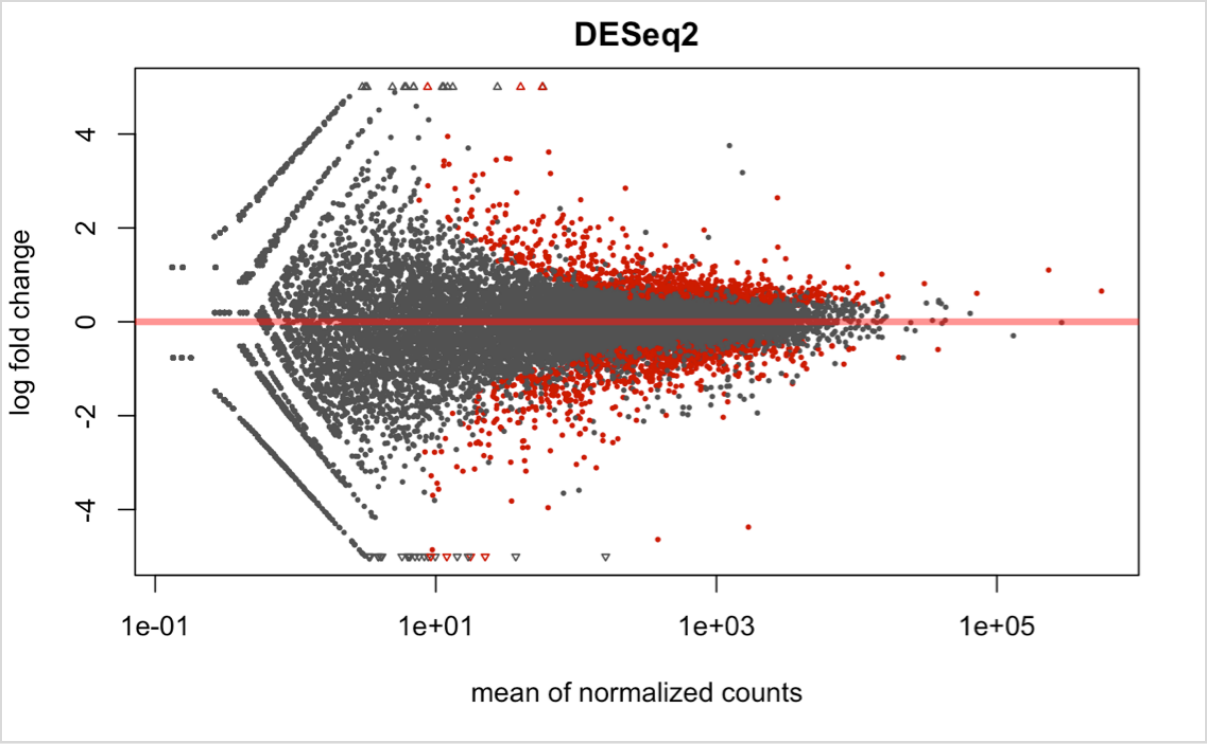
```
# corrected
sampleDistsCorr <- dist( t( assay(vsd2) ) )
sampleDistsCorr
sampleDistCorrMatrix <- as.matrix( sampleDistsCorr )
colnames(sampleDistCorrMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistCorrMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists, col=colors)
```



3. Analyze using both DeSeq2 and edgeR

DeSeq2

```
### BASIC DGE ANALYSIS USING DESEQ2 ###

# Run DESEQ and generate a simple plot showing the distribution of regulated
and unregulated genes
DatasetProcessed <- DESeq(Dataset) # runs DESEQ
par(mfrow=c(1,1))
DESeq2::plotMA(DatasetProcessed, main="DESeq2", ylim=c(-5,5))
```



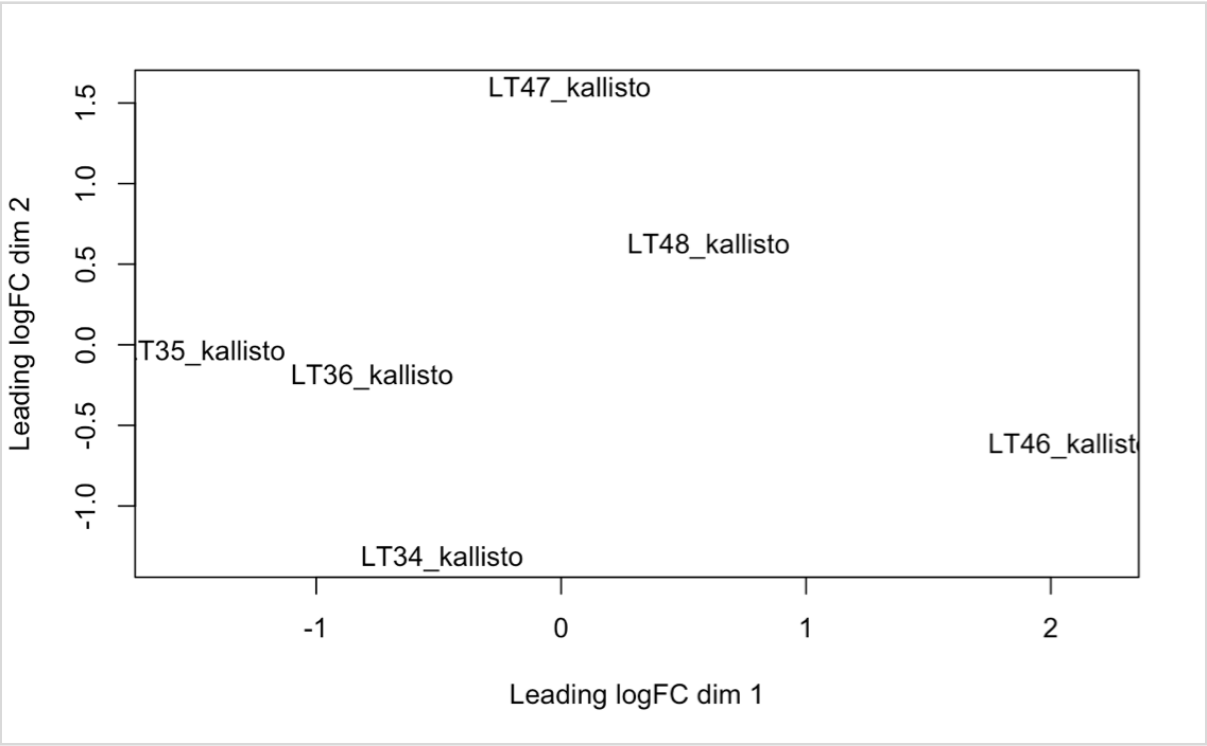

```r
# Next we perform a contrast analysis to produce a list of differentially
regulated genes between our two conditions
# First we set CTRL dataset as baseline
Dataset$condition <- relevel(Dataset$condition, "Ctrl")

# Next we create our results object while performing shrinkage of effect size
# (this reduces the impact of apparent gross changes in low expressed genes)
res1 <- lfcShrink(DatasetProcessed, contrast=c("condition","Silence","Ctrl"))

# Here we modify our output data to include two additional columns that contain
the baseMeans (a proxy for counts)
# This is useful for downstream filtering of lowly expressed genes
baseMeanCtrl = rowMeans(counts(DatasetProcessed,normalized=TRUE)
[,DatasetProcessed$condition == "Ctrl"])
baseMeanSilence = rowMeans(counts(DatasetProcessed,normalized=TRUE)
[,DatasetProcessed$condition == "Silence"])
res1 = cbind(as.data.frame(res1), baseMeanCtrl, baseMeanSilence)

# Here we add two further columns, the gene symbol (common name) and entrez ID
- both of which may be useful downstream
res1$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res1), column="SYMBOL",
keytype="ENSEMBL", multiVals="first") # MAPS GENE IDs
```

```
res1$entrez <- mapIds(org.Hs.eg.db, keys=row.names(res1), column="ENTREZID",
keytype="ENSEMBL", multiVals="first")

# Finally we write the complete results object to an outfile
write.csv(res1, "DGEanalysis.csv", row.names=TRUE)
```

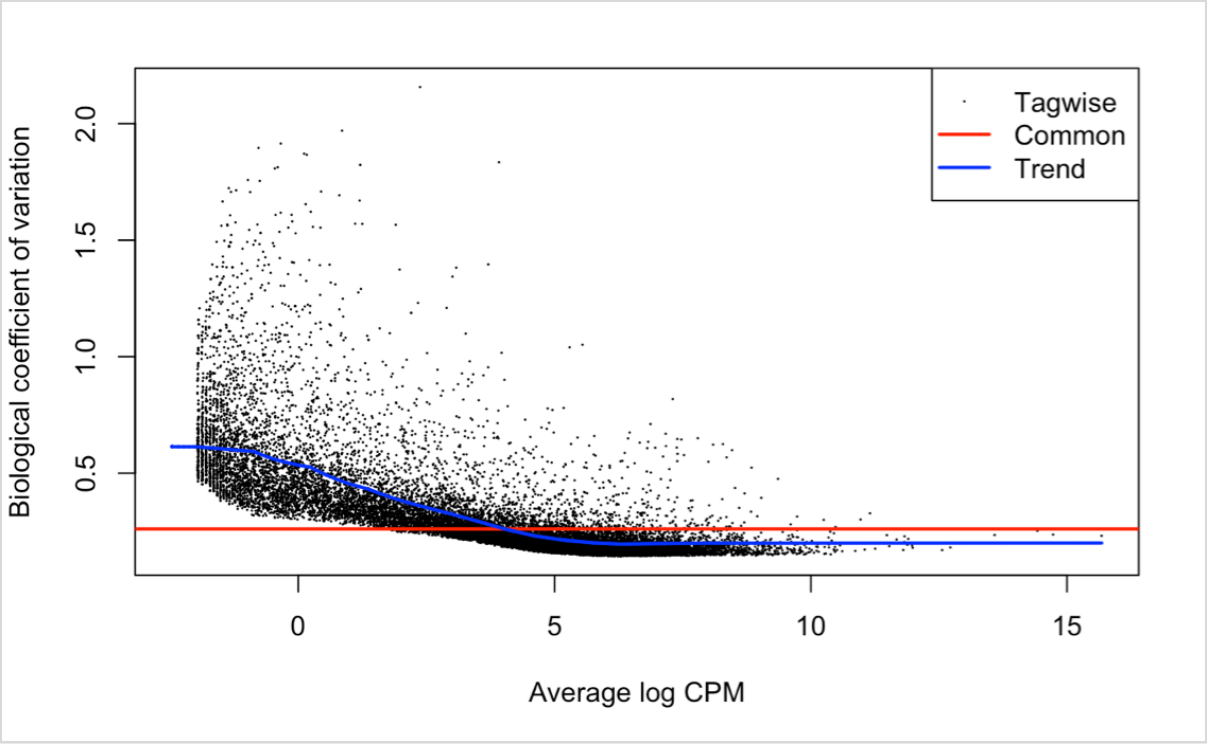
edgeR

followed: https://bioinformatics-core-shared-training.github.io/cruk-bioinf-sschool/Day3/rnaSeq_DE.pdf

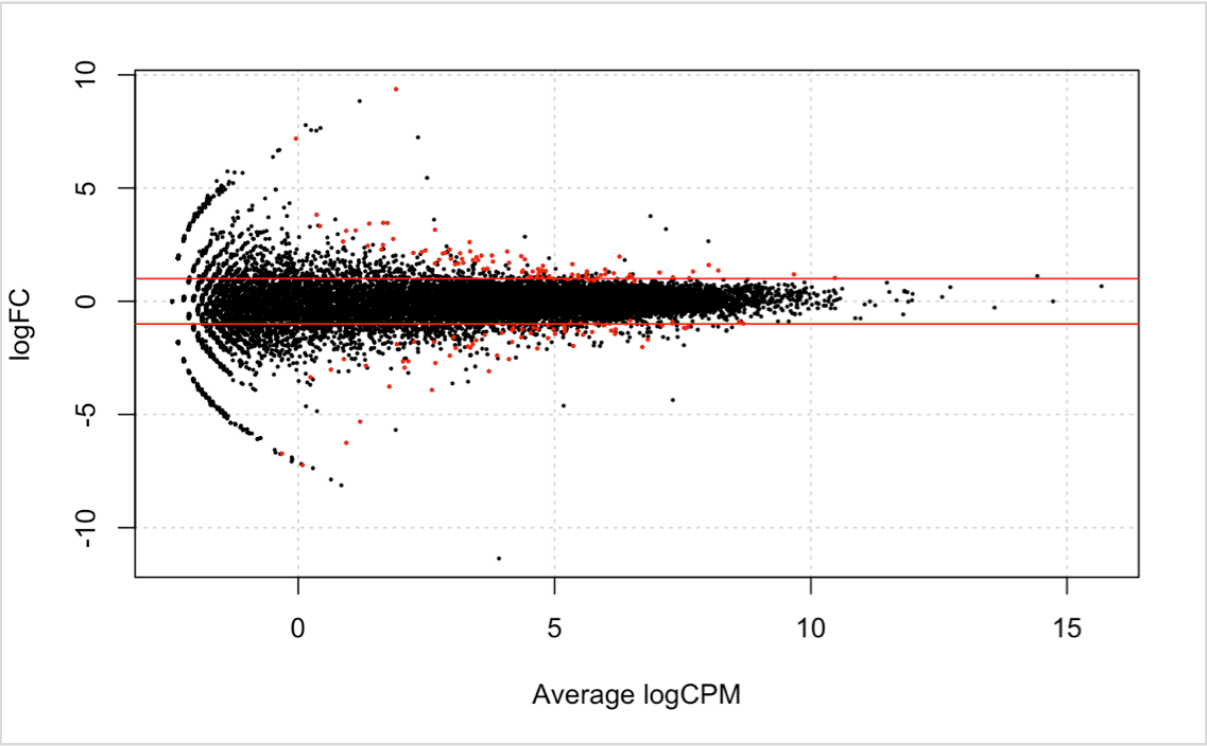
```
# Reading in the Data
counts <- read.table("counts.txt", header=TRUE, row.names=1)
sample <- read.table("sample.txt", header=TRUE)
condition <- sample$condition

# Creating a DGEList object
dgList <- DGEList(counts=counts, group = condition, genes = rownames(counts))
head(dgList$samples)

# Filtering: To ensure we only look at genes with counts.
countsPerMillion <- cpm(dgList)
summary(countsPerMillion)

# Normalisation - for both between/in samples
# tmm - trimmed mean of M values
dgList <- calcNormFactors(dgList, method="TMM")

# Data Exploration
plotMDS(dgList)
```




```r
# Model Set Up
designMat <- model.matrix(~condition+0, data = dgList$samples)
colnames(designMat) <- c("Ctrl", "Silence")

#Estimating Dispersions
dgList <- estimateDisp(dgList, designMat)
#estimateDisp is essentially the three commands below in one. But estimateDisp is better because it "provides protection against zero fitted values, gives a more stable likelihood landscape (due to hot-starting at every step of the grid search), gives a more graduated trend from local fitting compared to binning, and can estimate the necessary prior degrees of freedom for EB shrinkage from the data"
# source: (https://support.bioconductor.org/p/75970/#75973)
# dgList <- estimateGLMCommonDisp(dgList, design=designMat)
# dgList <- estimateGLMTrendedDisp(dgList, design=designMat)
# dgList <- estimateGLMTagwiseDisp(dgList, design=designMat)

#plot the estimates and see how they differ. The biological coefficient of variation (BCV) is the square root of the dispersion parameter in the negative binomial model
plotBCV(dgList)
```




```
fit <- glmFit(dgList, designMat)
contrast = makeContrasts(Silence - Ctrl, levels=colnames(designMat))
lrt <- glmLRT(fit, contrast=contrast )

#filter by adj p value 0.1 (to keep deseq / edgeR with same cutoffs)
edgeR_result <-  topTags(lrt, n = nrow(lrt$table), p.value = 0.1)

# there is an extra row of gene names, remove it.
edgeR_result$genes <- NULL

#plot all fold log change of genes to see which are differentially expressed.
deGenes <- decideTestsDGE(lrt, p=0.001)
deGenes <- rownames(lrt)[as.logical(deGenes)]
plotSmear(lrt, de.tags=deGenes)
abline(h=c(-1, 1), col=2)
```



4. How many genes are differentially regulated between conditions for each software?

- How many are upregulated
- How many are downregulated

```
# up / down reg of deseq
DeSeq_result <- read.csv("DGEanalysis.csv", header=TRUE, row.names=1)
DeSeq_result <- DeSeq_result %>%
  rownames_to_column('gene')  %>%
  filter(padj < 0.1) %>%
  column_to_rownames('gene')

DeSeq_result_Up <- DeSeq_result %>% filter(log2FoldChange > 0)
DeSeq_result_Down <- DeSeq_result %>% filter(log2FoldChange < 0)
dim(DeSeq_result_Up)
dim(DeSeq_result_Down)

#up / down reg of edgeR
edgeR_result <- as.data.frame(edgeR_result)
edgeR_result_Up <- edgeR_result %>% filter(logFC > 0)
```

```
edgeR_result_Down <- edgeR_result %>% filter(logFC < 0)
dim(edgeR_result_Up)
dim(edgeR_result_Down)
```

Analysis	UpRegulated	DownRegulated
DeSeq	944	862
edgeR	829	713

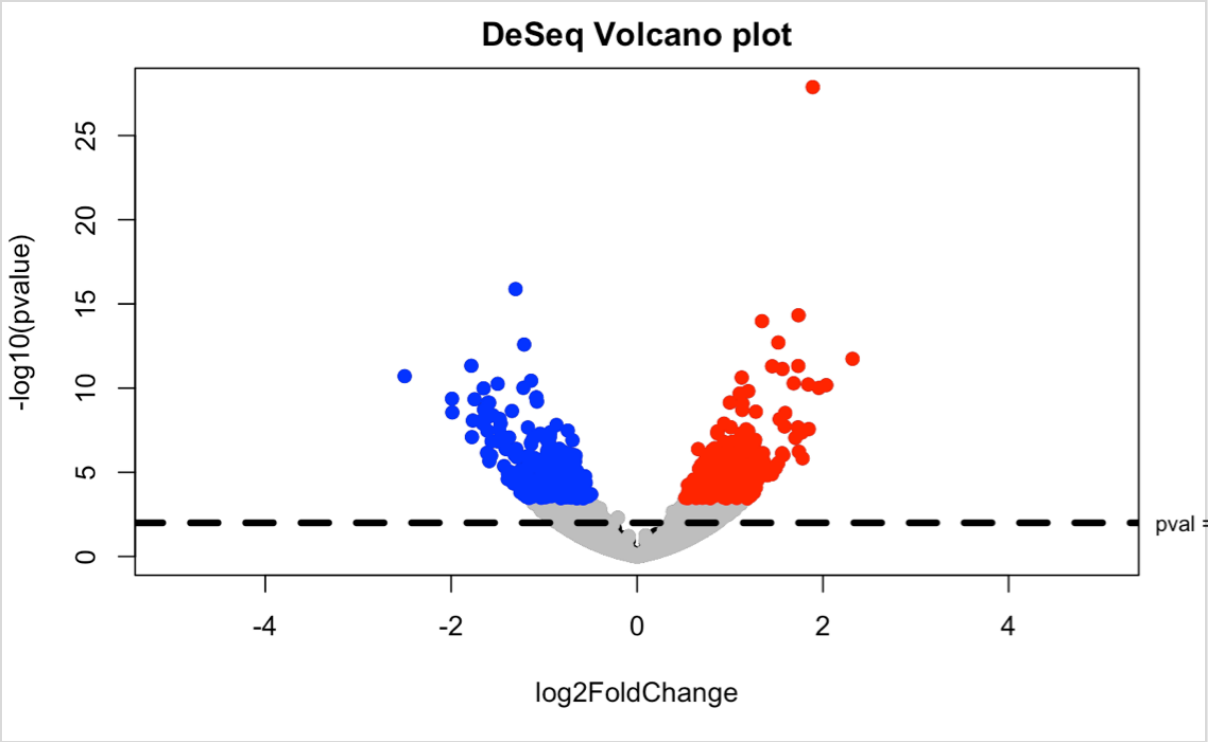
- What criteria did you use to decide this

I used logFC to determine if the gene was up regulated or down regulated.

Below are Volcano Plots of the logFC counts vs negativeLogPval

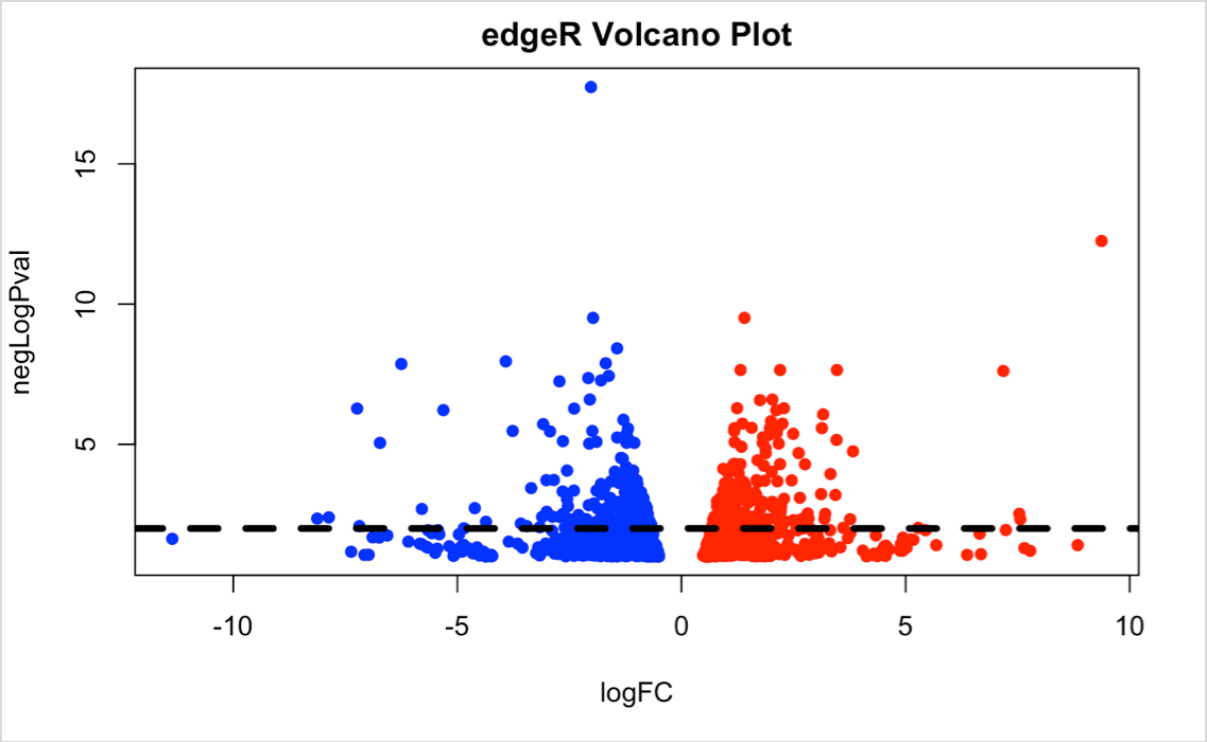
DeSeq

```
res1 <- lfcShrink(DatasetProcessed, contrast=c("condition","Ctrl","Silence"))
with(res1, plot(log2FoldChange, -log10(pvalue), pch=20, cex=1.5, main="DeSeq Volcano plot", xlim=c(-5,5)))
with(subset(res1, padj>0.01), points(log2FoldChange, -log10(pvalue), pch=20, cex=1.5, col="gray"))
with(subset(res1, padj<0.01 & log2FoldChange>0), points(log2FoldChange, -log10(pvalue), pch=20, cex=1.5, col="red"))
with(subset(res1, padj<0.01 & log2FoldChange<0), points(log2FoldChange, -log10(pvalue), pch=20, cex=1.5, col="blue"))
### ADD BELLS AND WHISTLES
pval = 0.01
abline(h = -log10(pval), col = "black", lty = 2, lwd=4)
mtext(paste("pval = 0.01", pval), side = 4, at = -log10(pval), cex = 0.8, line = 0.5, las = 1)
```



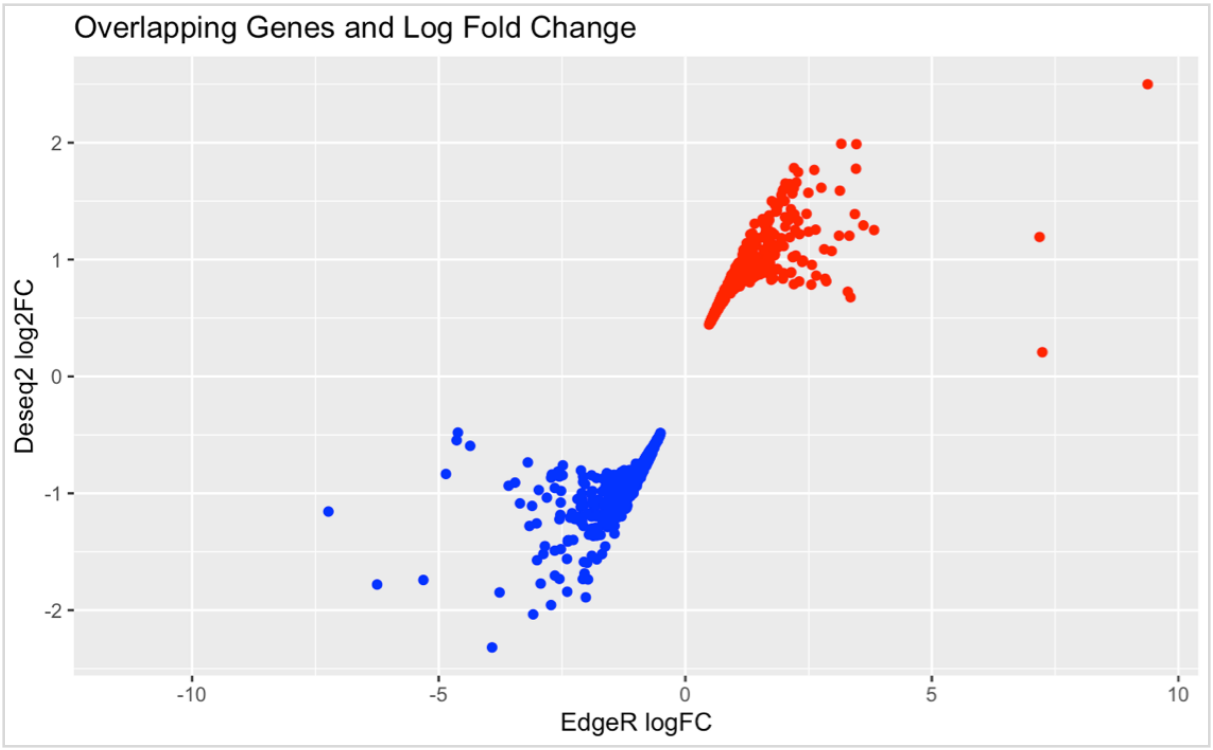
edgeR

```
edgeR_result <-  topTags(lrt, n = nrow(lrt$table), p.value = 0.1)
volcanoData <- cbind(edgeR_result$table$logFC, -log10(edgeR_result$table$FDR))
colnames(volcanoData) <- c("logFC", "negLogPval")
plot(volcanoData, pch=16, col = ifelse(edgeR_result$table$logFC <
0,'blue','red'), main = "edgeR Volcano Plot")
abline(h = -log10(pval), col = "black", lty = 2, lwd=4)
```



5. How do DeSeq2 and edgeR compare in terms of overlapping genes

```
#OVERLAPPING GENES
overlap <- merge(edgeR_result, DeSeq_result,by="row.names",all.x=TRUE)
ggplot(data = overlap, aes(x = logFC, y = log2FoldChange)) + geom_point() +
xlab("EdgeR logFC") + ylab("Deseq2 log2FC") + labs(title= "Overlapping Genes
and Log Fold Change")
dim(overlap)

overlap_Up <- overlap %>% filter(logFC > 0)
overlap_Down <- overlap %>% filter(logFC < 0)
dim(overlap_Up)
dim(overlap_Down)
```



All the genes that are considered up regulated by DeSeq are also considered up regulated in edgeR.

There was a total of **1542 overlapped** genes.

There was a total of **829 up regulated** genes

There was a total of **713 down regulated** genes

Provide full coding, justification for parameters chosen, pertinent figures and tables (both with legends) in a single pdf file

#appliedsequencinginformatics