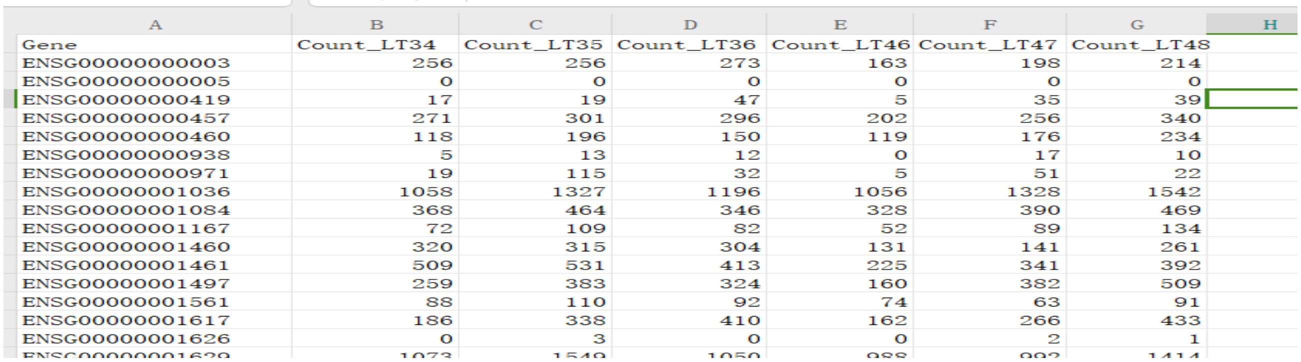
APPLIED SEQUENCING INFORMATICS
ASSIGNMENT-5

About the data:

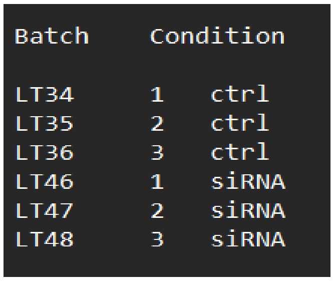
Datasets: HCMV-infected normal human dermal fibroblasts treated with either

- 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



Gene	Count_LT34	Count_LT35	Count_LT36	Count_LT46	Count_LT47	Count_LT48
ENSG00000000003	256	256	273	163	198	214
ENSG00000000005	0	0	0	0	0	0
ENSG00000000419	17	19	47	5	35	39
ENSG00000000457	271	301	296	202	256	340
ENSG00000000460	118	196	150	119	176	234
ENSG00000000938	5	13	12	0	17	10
ENSG00000000971	19	115	32	5	51	22
ENSG00000001036	1058	1327	1196	1056	1328	1542
ENSG00000001084	368	464	346	328	390	469
ENSG00000001167	72	109	82	52	89	134
ENSG00000001460	320	315	304	131	141	261
ENSG00000001461	509	531	413	225	341	392
ENSG00000001497	259	383	324	160	382	509
ENSG00000001561	88	110	92	74	63	91
ENSG00000001617	186	338	410	162	266	433
ENSG00000001626	0	3	0	0	2	1
ENSG00000001629	1073	1549	1050	988	992	1414

Fig1. counts data generated in the previous assignment.



```
Batch     Condition

LT34      1    ctrl
LT35      2    ctrl
LT36      3    ctrl
LT46      1    siRNA
LT47      2    siRNA
LT48      3    siRNA
```

Fig2: Sample.txt file content

1. Batch Correction

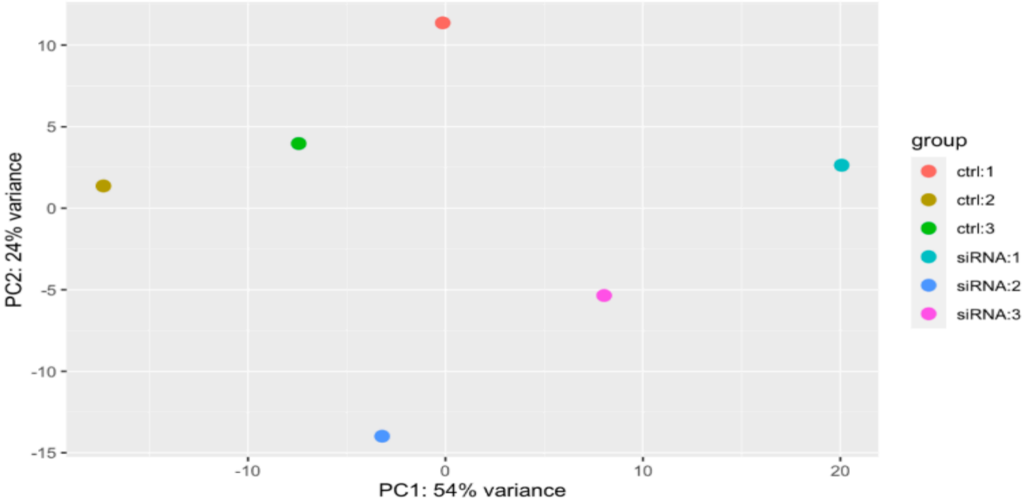


Fig3. Figure below represents a PCA plot before batch correction plot after DESeq analysis.
In the PCA plot, there was clear separation between the control group and the siRNA treated group. However, the control samples and treated samples didn't form distinct clusters together.

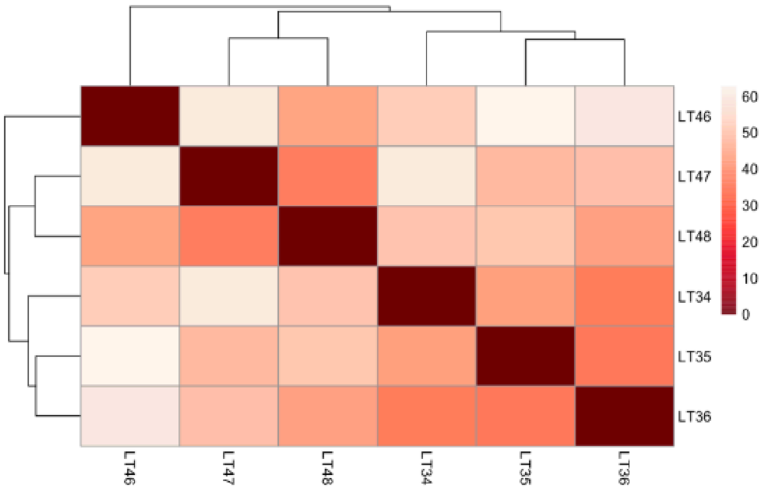


Fig.4: Heatmap of 6 datasets before batch correction.
A heatmap was generated to illustrate the correlation among individual samples. The heatmap revealed noticeable correlations among samples originating from corresponding batches (LT34 and LT46, LT35 and LT47, LT36 and LT48). Subsequently, batch correction was executed using the following code:

```
assay(vsd2) <- limma::removeBatchEffect(assay(vsd2), vsd2$batch)
```

PCA plot and heatmap were plotted after batch correction:

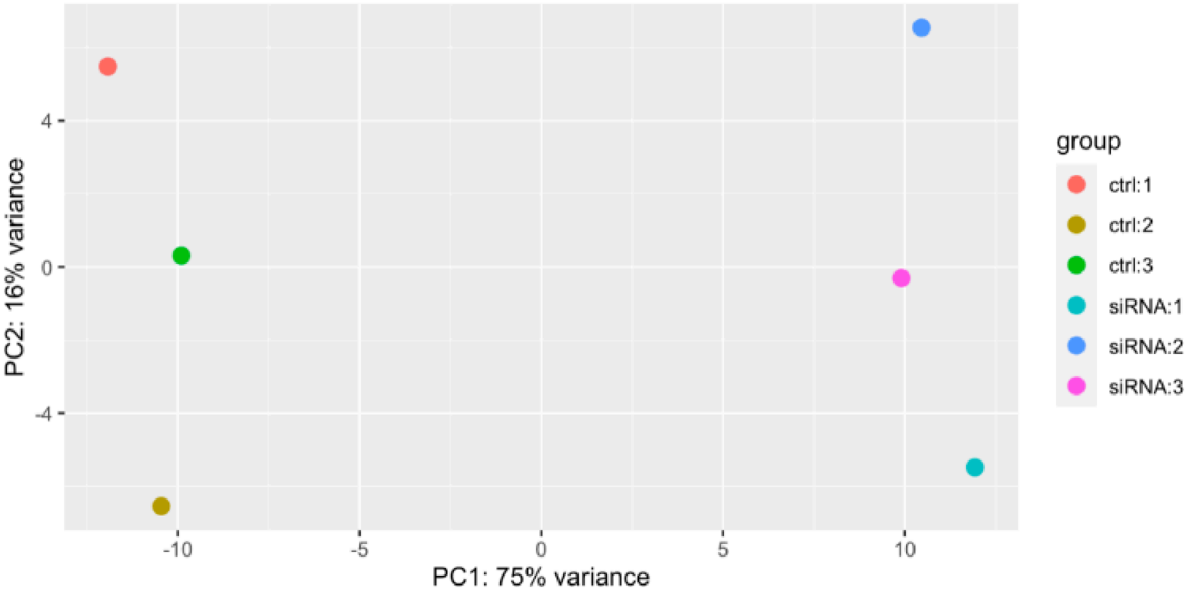


Fig.5: PCA plot after the batch correction

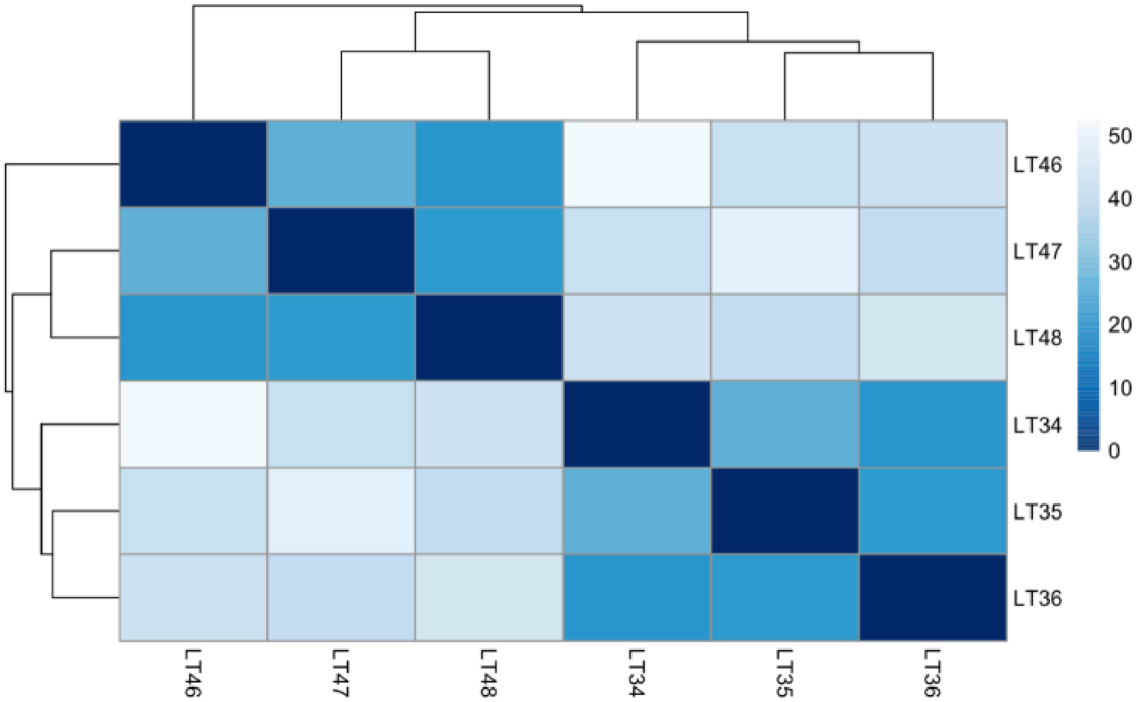


Fig6: Heatmap after the batch correction.
Following correction for batch effects, the controls and treated samples exhibited clear separation and clustering.

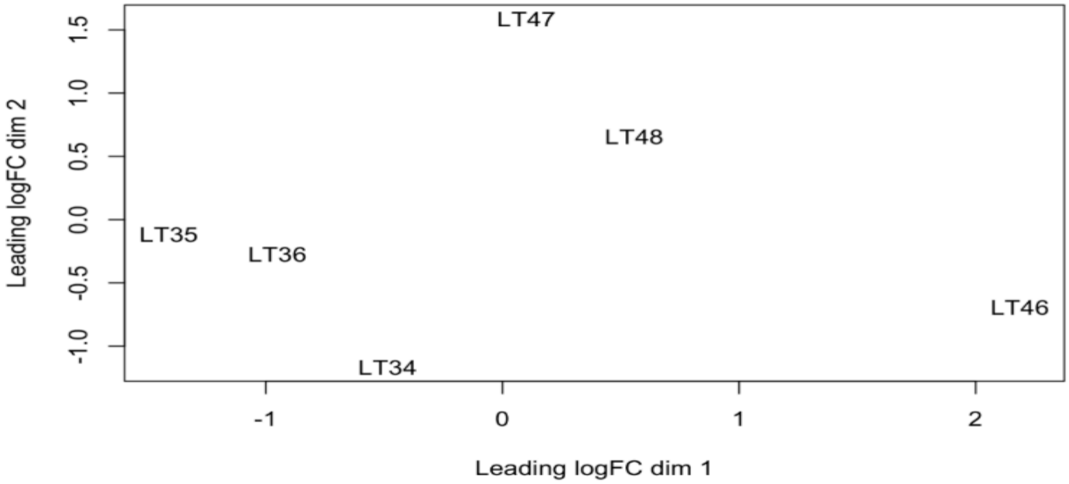


Fig.7: MDS plot of 6 samples before batch correction.

In the edgeR analysis, an MDS plot was generated to illustrate the root mean square of the maximum log2 fold change between samples. Generally, the controls were positioned below the treated samples, indicating a treatment effect. However, samples within the same batch tended to cluster closely together, suggesting the presence of a batch effect requiring correction.

To mitigate the batch effect, a design matrix was constructed using the experimental setup to delineate both batch and treatment condition details for subsequent analysis in edgeR. This includes tasks like estimating dispersion and examining genes for differential expression. Below are the commands for constructing the design matrix, estimating dispersion using it, and conducting tests for differential expression.

```
design <- model.matrix(~batch+condition) # build a design matrix with batch and

condition information
rownames(design) <- colnames(dgList)

disp <- estimateDisp(dgList, design) # taking account of batch and condition
information from the design matrix when estimating dispersion

et <- exactTest(disp) # test for DE genes
```

2. Differentially expressed genes

i) In DEseq2 analysis: **159 upregulated genes, and 171 downregulated genes**.

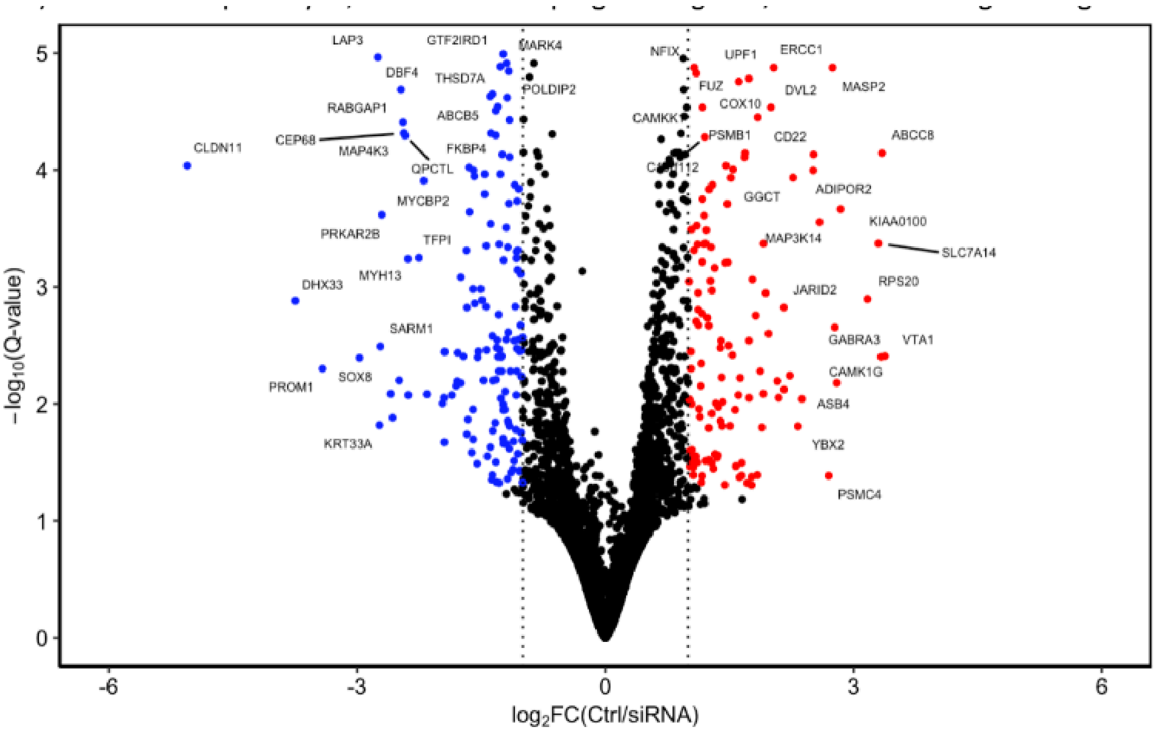


Fig.8: Volcano plot of differentially expressed genes in DEseq analysis.

ii) In edgeR analysis: **445 upregulated genes and 689 downregulated genes.**

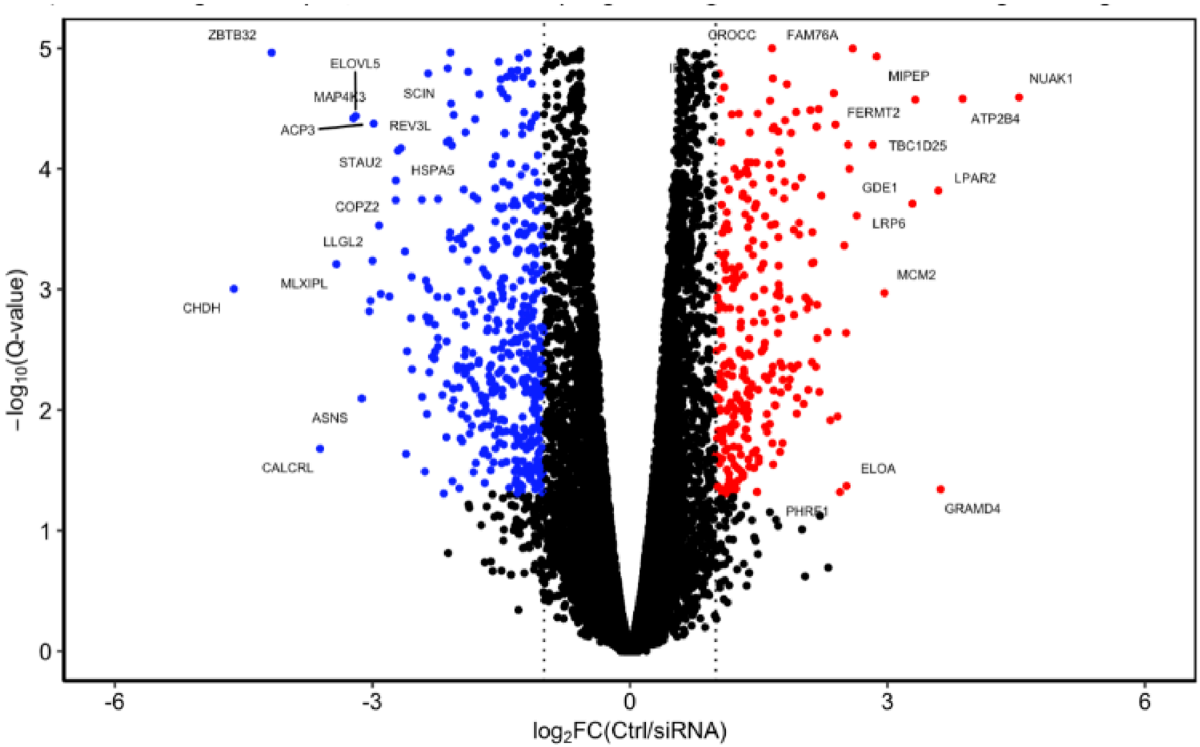


Fig.9: Volcano plot of differentially expressed genes in edgeR analysis.

iii) A significance threshold of **0.05 was applied to the p-values**, whereby any value exceeding 0.05 was deemed nonsignificant. **Genes exhibiting a log2 fold change exceeding 1 were categorized as upregulated, while those with a log2FC below -1 were classified as downregulated.** Subsequently, genes meeting these criteria in both analyses were filtered and exported to CSV files to identify commonalities between the two datasets.

Common Genes Overlap

Utilizing the "inner_join" function from the dplyr package, we extracted common differentially expressed genes present in both the DESeq2 and edgeR+ analyses. The results indicated that **all differentially expressed genes identified in the DESeq2 analysis, whether upregulated or downregulated, were also detected as differentially expressed in the edgeR analysis.** Additionally, edgeR revealed a larger set of uniquely differentially expressed genes, encompassing both upregulated and downregulated genes.

Scripts

```
i) DEseq2
# Load 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(ggrepel)
# Import counts data and sample file built in advance
setwd("/Users/apoorva/Desktop/Applied_Sequencing_Informatics/Assignment5/")
CountTable <- read.table("output.txt", header=TRUE, row.names=1)
samples <- read.table("sample.txt", header=TRUE)
# Load counts data to DEseq2 and build a normalized matrix
dds <- DESeqDataSetFromMatrix(countData = CountTable, colData=samples, design=~condition)
dds = DESeq(dds)
counts(dds)->raw_counts
raw_counts<-as.data.frame(counts(dds))
norm_counts = counts(dds, normalized = TRUE)
# Transform our raw count data using a variance stabilizing transformation (VST) that roughly
mirrors how DeSeq2 models the data, plot a PCA plot to visualize datasets before batch
correction according to batch and treatment condition
vsd1 <- varianceStabilizingTransformation(dds, blind=FALSE)
plotPCA(vsd1, c("condition","batch"))
# Visualize dataset with a heatmap
sampleDists <- dist( t( assay(vsd1) ) )
sampleDists

sampleDistMatrix <- as.matrix( sampleDists )
colors <- colorRampPalette( rev(brewer.pal(9, "Reds")) )(255)
pheatmap(sampleDistMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists,
col=colors)
# Create another VST transformed data and perform batch correction
vsd2 <- varianceStabilizingTransformation(dds, blind=FALSE)
assay(vsd2) <- limma::removeBatchEffect(assay(vsd2), vsd2$batch)
# Visualize datasets with PCA plots after batch correction
plotPCA(vsd2, c("condition","batch"))
data <- plotPCA(vsd2, c("condition","batch"),returnData=TRUE)
# Heatmap after batch correction
sampleDistsCorr <- dist( t( assay(vsd2) ) )
sampleDistsCorr
```

```
sampleDistCorrMatrix <- as.matrix( sampleDistsCorr )
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistCorrMatrix, clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists,
col=colors)
# Run DEseq2 on dataset
DatasetProcessed <- DESeq(dds)
par(mfrow=c(1,1))
# Set control as the base line for analysis of genes differentially expressed in siRNA treated
samples
DatasetProcessed$condition <- relevel(DatasetProcessed$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, coef=2)
# Add two further columns, the gene symbol and entrez
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")
# Set thresholds for filtering out differentially expressed genes: here I set threshold of 0.05 for
p-value (adjusted p value in DEseq2 analysis) which selects for statistically significant genes and
1 for log2 fold change which identifies up or down regulated genes.
Ctrl_siRNA<-res1
adj_p_val = 0.05
abs_log2fc = 1
# Calculate Q values with adjusted P value
rna_df<-as.data.frame(`Ctrl_siRNA`)
rna_df$log10.pvalue<-(-1*log10(rna_df$padj))
# Select genes with adjusted p value less than threshold and absolute log2 fold change greater than
threshold and
mark them as significant genes
rna_df$Significant_Gene<-"No"
row_number<-which(rna_df$padj<adj_p_val & abs(rna_df$log2FoldChange)>abs_log2fc)
rna_df$Significant_Gene[row_number]<-"Yes"
# Filter out differentially expressed genes with log2 fold change values
rna_df$Differential_Gene<-rna_df$Significant_Gene
rna_df$Differential_Gene[which(rna_df$padj< adj_p_val & rna_df$log2FoldChange>
abs_log2fc)]<-"Up-Regulated"

rna_df$Differential_Gene[which(rna_df$padj< adj_p_val & rna_df$log2FoldChange<
-abs_log2fc)]<-"Down-
Regulated"

# Filter out up/down regulated genes and separate them to different dataframes. Order by log2
fold change so that the list of genes are most up/down regulated to least up/downregulated.
Write as csv files.
upreg <- filter(rna_df, rna_df$Differential_Gene == "Up-Regulated")
upreg <- upreg[order(upreg$log2FoldChange, decreasing=TRUE),]
downreg <- filter(rna_df, rna_df$Differential_Gene == "Down-Regulated")
downreg <- downreg[order(downreg$log2FoldChange),]
write.csv(upreg, file="upreg_DEseq2.csv")
write.csv(downreg, file="downreg_DEseq2.csv")
# Plot a volcano plot to visualize differentially expressed genes.
rna_df$label<-"No"
rna_df$label[which(rna_df$padj<adj_p_val & abs(rna_df$log2FoldChange)>abs_log2fc)]<-"Yes"
pRNA <- ggplot(rna_df, aes(log2FoldChange, log10.pvalue))+
geom_point(aes(colour = Differential_Gene),size=1)+
scale_colour_manual(values=c("blue", "black","red"))+
xlim(-6,6)+
ylim(0,5)+
xlab(expression(paste(log[2], 'FC(Ctrl/siRNA)')))+
ylab(expression(paste(-log[10], '(Q-value)'))) +
theme_bw() +
theme(panel.background = element_blank())+
theme(panel.border = element_rect(colour = "black", fill=NA, size=1))+
theme(text = element_text(size=10)) +
theme(plot.title = element_blank())+
theme(axis.text = element_text( color = "black", size = 10))+
theme(panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), axis.line=element_line()) +
theme(legend.position="none")+
geom_vline(xintercept=-abs_log2fc,linetype="dotted")+
geom_vline(xintercept=abs_log2fc,linetype="dotted")+
geom_text_repel(
data = subset(rna_df, label=="Yes"),
aes(label = rna_df$symbol[which(label=="Yes")]),
size = 2,
box.padding = unit(0.35, "lines"),
point.padding = unit(0.5, "lines")
)
```

ii) EdgeR

```
# Load Libraries
library("edgeR")
library("pheatmap")
library("RColorBrewer")
library("vsn")
library("AnnotationDbi")
library("org.Hs.eg.db")
library("genefilter")
library("biomaRt")
library("IHW")
library("ggplot2")
library("statmod")
library("dplyr")
library(ggrepel)
# Import counts data and sample file built in advance
setwd("/Users/apoorva/Desktop/Applied_Sequencing_Informatics/Assignment5/")
CountTable <- read.table("output.txt", header=TRUE, row.names=1)
```

```
samples <- read.table("sample.txt", header=TRUE)
#Load counts data from kallisto output
dgList <- DGEList(counts=CountTable, samples = samples, group =condition)
# Filter out genes that have at leat 1 count per million in at least 2 samples. (EdgeR is unlike
DEseq2 which pre-filters genes.)
countsPerMillion <- cpm(dgList)
countCheck <- countsPerMillion > 1
keep <- which(rowSums(countCheck) >= 2)
dgList <- dgList[keep,]
# Perform TMM normalization (TMM is the default normalization method for edgeR)
dgList <- calcNormFactors(dgList, method="TMM")
# Plot an MDS plot to examine if the samples can cluster separately based on treatment
conditions and if there is a batch effect that should be corrected
plotMDS(dgList)
# Built a design matrix based on experiment design (to compare differential expression
between control and siRNA treated samples)
condition <- as.factor(samples$condition)
batch <- as.factor(samples$batch)
design <- model.matrix(~batch+condition)
rownames(design) <- colnames(dgList)
# Estimates the dispersion and plot a possible abundance trend
disp <- estimateDisp(dgList, design)
plotBCV(disp)
# Use exacttest to test gene expression levels
et <- exactTest(disp)
# Extract DGE data from output
dat <- et$table
# Add corresponding gene name and entrez ID
dat$symbol <- mapIds(org.Hs.eg.db, keys=row.names(dat), column="SYMBOL", keytype="ENSEMBL",
multiVals="first") # MAPS GENE IDs

dat$entrez <- mapIds(org.Hs.eg.db, keys=row.names(dat), column="ENTREZID", keytype="ENSEMBL",
multiVals="first")
# Set thresholds for filtering out differentially expressed genes: here I set threshold of 0.05 for
p-value (adjusted p value in DEseq2 analysis) which selects for statistically significant genes and
1 for log2 fold change which identifies up or down regulated genes
Ctrl_siRNA<-dat
adj_p_val = 0.05
abs_log2fc = 1
# Calculate Q values with adjusted P value
rna_df<-as.data.frame(`Ctrl_siRNA`)
rna_df$log10.pvalue<-(-1*log10(rna_df$padj))
# Select genes with adjusted p value less than threshold and absolute log2 fold change greater
than threshold and mark them as significant genes
rna_df$Significant_Gene<-"No"
row_number<-which(rna_df$padj<adj_p_val & abs(rna_df$log2FoldChange)>abs_log2fc)
rna_df$Significant_Gene[row_number]<-"Yes"
# Filter out differentially expressed genes with log2 fold change values
rna_df$Differential_Gene<-rna_df$Significant_Gene
```

```
rna_df$Differential_Gene[which(rna_df$padj< adj_p_val & rna_df$log2FoldChange>
abs_log2fc)]<-"Up-Regulated"

rna_df$Differential_Gene[which(rna_df$padj< adj_p_val & rna_df$log2FoldChange<
-abs_log2fc)]<-"Down-
Regulated"

# Filter out up/down regulated genes and separate them to different dataframes. Order by log2
fold change so that the list of genes are most up/down regulated to least up/downregulated.
Write as csv files.
upreg <- filter(rna_df, rna_df$Differential_Gene == "Up-Regulated")
upreg <- upreg[order(upreg$logFC, decreasing=TRUE),]
downreg <- filter(rna_df, rna_df$Differential_Gene == "Down-Regulated")
downreg <- downreg[order(downreg$logFC),]
write.csv(upreg, file="upreg_edgeR.csv")
write.csv(downreg, file="downreg_edgeR.csv")
# Plot a volcano plot to visualize differentially expressed genes.
rna_df$label<-"No"
rna_df$label[which(rna_df$padj<adj_p_val & abs(rna_df$log2FoldChange)>abs_log2fc)]<-"Yes"
pRNA <- ggplot(rna_df, aes(log2FoldChange, log10.pvalue))+
geom_point(aes(colour = Differential_Gene),size=1)+
scale_colour_manual(values=c("blue", "black","red"))+
xlim(-6,6)+
ylim(0,5)+
xlab(expression(paste(log[2], 'FC(Ctrl/siRNA)')))+
ylab(expression(paste(-log[10], '(Q-value)'))) +
theme_bw() +
theme(panel.background = element_blank())+
theme(panel.border = element_rect(colour = "black", fill=NA, size=1))+
theme(text = element_text(size=10)) +
theme(plot.title = element_blank())+
theme(axis.text = element_text( color = "black", size = 10))+
theme(panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), axis.line=element_line()) +
theme(legend.position="none")+
geom_vline(xintercept=-abs_log2fc,linetype="dotted")+

geom_vline(xintercept=abs_log2fc,linetype="dotted")+
geom_text_repel(
data = subset(rna_df, label=="Yes"),
aes(label = rna_df$symbol[which(label=="Yes")]),
size = 2,
box.padding = unit(0.35, "lines"),
point.padding = unit(0.5, "lines")
)
```

iii) Check for overlapping genes

```
# Load data from up/down regulated gene list filtered out from both analyses
library("dplyr")
setwd("/Users/apoorva/Desktop/Applied_Sequencing_Informatics/Assignment5/")
up_edgeR <- read.csv(file = "upreg_edgeR.csv")
```

```
down_edgeR <- read.csv(file = "downreg_edgeR.csv")
up_DEseq2 <- read.csv(file = "upreg_DEseq2.csv")
down_DEseq2 <- read.csv(file = "downreg_DEseq2.csv")
# Rename first column
names(up_edgeR)[1] <- "GeneID"
names(up_DEseq2)[1] <- "GeneID"
names(down_edgeR)[1] <- "GeneID"
names(down_DEseq2)[1] <- "GeneID"
# Inner join by Gene ID to filter out genes that are up/regulated in both analysis
up_common <- inner_join(up_DEseq2,up_edgeR,by = "GeneID")
down_common <- inner_join(down_DEseq2,down_edgeR,by = "GeneID")
```