APPENDIX C

RNA-sequencing\_analysis

Jasmine Speranza

Re-call the packages you need for the analysis

library('S4Vectors')  
library('DESeq2')  
library('pheatmap')  
library('factoextra')  
library('tidyverse')  
library('tidyr')  
library('usethis')  
library('ggplot2')  
library('devtools')   
library('ggrepel')  
library('EnhancedVolcano')   
library('topGO')  
library('org.Dm.eg.db')   
library('phyloseq')  
library('clusterProfiler')  
library ("enrichplot")

Load the samples, run DESeq and normalise the counts

##Set up your directory  
directory = setwd('/Users/u2093090/Desktop/Drosophila experiments/CTRL vs HTT Ex1-120q flies/NOVAGENE ANALYSIS/RNAseq/RNAseq CtrlvsHTT guts 4d/JS\_s analysis results/4\_counts')  
  
##Read the file about samples,make table, read in the results from the LibiNorm analysis (the counts files),  
#perform the analysis and check your data:  
Guts\_data= read.table("Guts\_data.txt",header=TRUE)  
row.names(Guts\_data) <- Guts\_data$sampleName   
head(Guts\_data)  
View(Guts\_data)  
  
#Store your results in a different type/format of table  
ddsHTSeq.E.gut <- DESeqDataSetFromHTSeqCount(sampleTable = Guts\_data, directory = '//Users//u2093090//Desktop//Drosophila experiments//CTRL vs HTT Ex1-120q flies//NOVAGENE ANALYSIS//RNAseq//RNAseq CtrlvsHTT guts 4d//JS\_s analysis results//4\_counts', design = ~ conditions)  
#View(ddsHTSeq.E.gut)  
  
#Filter out the low read counts, the NA results and 0 results. Store them into the "keep" vector.   
keep.gut <- rowSums(counts(ddsHTSeq.E.gut)) >=10  
#Results you are keeping are located in the new vector call dds  
dds.gut <- ddsHTSeq.E.gut[keep.gut,]  
  
#Run the differential expression genes analysis and save the data in a vector (dds.E)  
dds.E.gut<-DESeq(dds.gut)  
  
#count and normalize  
head(counts(dds.E.gut))  
head(counts(dds.E.gut, normalized=TRUE))   
countsnorm.E.gut<- counts(dds.E.gut, normalized=TRUE)  
View(countsnorm.E.gut)

PCA and hierarchical clustering (Work on the dds.e file not normilased)

rld.E.gut <- rlog(dds.E.gut) #reduce the outling  
pca.E.gut <- prcomp(t(assay(rld.E.gut)))  
  
### Summary PCA  
summary(pca.E.gut)  
#bar plot with the percentage of variances  
fviz\_screeplot(pca.E.gut, addlabels=T)+ ggtitle('')  
#Put your samples in the dimension of PCs choosing the dimension thanks to the plot run before. Plot the dimensions where you see the slope so the 2 dimensions that have the biggest difference  
# Exploring axes explanation (asp is to set up the scale with the same units and be sure to interpretate thje data in the right way)  
fviz\_pca\_ind(pca.E.gut, asp=1, pointshape=20, habillage = Guts\_data$condition,axis= c (1,2)) + scale\_color\_discrete() + ggtitle('PCA transcriptomes of Drosophila gut samples')  
  
# Do dataframe to visualise all axis at once. [,1;4] IT IS THE number of axis (pca dimensions) you are taking in account  
pca\_df.gut <- pca.E.gut$x[,1:4] %>%   
 as\_tibble() %>%  
 add\_column(sample = Guts\_data$sampleName,  
 group = Guts\_data$condition)  
# Pivot the dataframe for plotting (flipping the dataframe in a specific kind of dataframe)  
pca\_pivot.gut <- pivot\_longer(pca\_df.gut, # dataframe to be pivoted  
 cols = PC1:PC4, # column names to be stored as a SINGLE variable  
 names\_to = "PC", # name of that new variable (column)  
 values\_to = "loadings") # name of new variable (column) storing all the values (data)  
# Plot one graph by PCA axis  
ggplot(pca\_pivot.gut) +  
 aes(x=sample, y=loadings, fill=group) +   
 geom\_bar(stat="identity") +  
 facet\_wrap(~PC) +  
 labs(title="PCA plot for each axis for Drosophila gut samples") +  
 theme\_bw() +  
 coord\_flip()  
  
  
### Hierarchical clustering   
# get distance  
distance.gut = dist(t(assay(rld.E.gut)), method = "maximum")  
# get cluster  
cluster.gut = hclust(distance.gut, method = "ward.D2")  
# plot  
plot(cluster.gut, labels=Guts\_data$conditions, main = 'Cluster of Drosophila gut samples')

Set the conditions (wtHTT vs mHTT flies)

# Extracting differentially expressed genes  
# The DEseq() function has already calculated p-values and log fold changes for every comparison and every gene.   
# Extract the results for a specific comparison:  
  
factor='condition' ## Specifies the column in the sample table that contains the condition designations  
## 'condition'.  
  
#wtHTT Vs mHTT  
factor= 'conditions'  
reference\_C.gut = 'wtHTT'  
treatment\_H.gut = 'mHTT'   
res\_C\_vs\_H.gut= results(dds.E.gut, contrast = c(factor, treatment\_H.gut, reference\_C.gut))  
View(res\_C\_vs\_H.gut)  
summary(res\_C\_vs\_H.gut)

Volcano Plots

# Set up adjusted p value and logfold change  
log10.pval\_C\_H.gut <- -log10(res\_C\_vs\_H.gut$padj) #the adjusted p value  
log2.fc\_C\_H.gut<- res\_C\_vs\_H.gut$log2FoldChange #the Logfold  
  
#Enhanced Volcano plot   
p1.gut=EnhancedVolcano(res\_C\_vs\_H.gut,  
 lab = rownames(res\_C\_vs\_H.gut),  
 selectLab = c('chinmo','D', 'Notum','tok','Rbp6','elfless', #cell diff/pro  
 'Fad2', 'FASN3', 'Cyp4g1', ' CG17560', 'CG31089', 'CG16904', #lipid metabolism   
 'LManIV', 'LManIII', 'LManVI', 'LManV','Mal-B1', #Carbo metabolism   
 'Cht5','Cht9','e', 'CG6788', 'CG10725', 'CG7298', #chitin metabolism   
 'Nepl18',' Try29F', ' Lsp2','CG18179', 'CG30371', 'CG30371', 'CG30090', #protein metabolism   
 'TotA','LysP','LysS','CG2736', 'asRNA:CR11538', #defense metabolism   
 'Fie','Gfrl', #nervous system dev  
 'CG3397','CG9360','GstD8', #oxidation stress  
 'I(2)03659', 'CG9864' #transport  
 ),  
 x = 'log2FoldChange',  
 y = 'padj',  
 xlab = bquote(~Log[2]~ 'fold change'),  
 FCcutoff = 1.5,  
 pCutoff = 0.05,  
 pointSize = 2.0,  
 title = 'RNAseq of Drosophila gut samples - wtHTT vs mHTT',  
 labSize = 3.0,  
 boxedLabels = TRUE,  
 drawConnectors = TRUE,  
 legendLabSize = 10,  
 legendIconSize = 2.0,  
 labCol = 'black',  
 labFace = 'bold',  
 col=c('black', 'green', 'blue', 'red'),  
 colAlpha = 5,  
 colConnectors = 'black')  
p1.gut + ggplot2::coord\_cartesian(ylim=c(0,30), xlim = c(-6,6))

Heatmaps and clustering

#wtHTT VS mHTT  
#just to show the genes filtered basing on the pvalue and logFC chosen  
topGenes\_C\_H\_gut.up <-res\_C\_vs\_H.gut$padj < 0.05 & (res\_C\_vs\_H.gut$log2FoldChange>1.5)   
genes.gut.u=assay(rld.E.gut)[which(topGenes\_C\_H\_gut.up),] #to select just a specific number of genes that you want to show in the heatmap  
pheatmap(genes.gut.u[1:51,] , fontsize\_row=7,   
 annotation\_col=Guts\_data["conditions"], scale="row", main = "wtHTT vs mHTT Drosophila head samples - Genes upregulated", treeheight\_row = 0, cutree\_cols=3, cutree\_rows=3)  
  
topGenes\_C\_H\_gut.down <-res\_C\_vs\_H.gut$padj < 0.05 & (res\_C\_vs\_H.gut$log2FoldChange<(-1.5))   
genes.gut.d=assay(rld.E.gut)[which(topGenes\_C\_H\_gut.down),] #to select just a specific number of genes that you want to show in the heatmap  
pheatmap(genes.gut.d[1:71,] , fontsize\_row=7,   
 annotation\_col=Guts\_data["conditions"], scale="row", main = "wtHTT vs mHTT Drosophila Head samples - Genes downregulated", treeheight\_row = 0, cutree\_cols=3, cutree\_rows=3)

Table of results

# Order by adjusted p value   
topGenes\_C\_H.gut <-res\_C\_vs\_H.gut$padj < 0.05 & (res\_C\_vs\_H.gut$log2FoldChange>1.5|res\_C\_vs\_H.gut$log2FoldChange<(-1.5))  
topGenes\_C\_H.gut<-topGenes\_C\_H.gut[order(res\_C\_vs\_H.gut$padj)]  
  
res\_C\_H\_gut.ad<-res\_C\_vs\_H.gut[order(res\_C\_vs\_H.gut$padj) , ]  
  
# and save results for this comparison into a text file  
write.table(res\_C\_H\_gut.ad,file=paste('/Users/u2093090/Desktop/Drosophila experiments/CTRL vs HTT Ex1-120q flies/NOVAGENE ANALYSIS/RNAseq/RNAseq CtrlvsHTT guts 4d/JS\_s analysis results/4\_counts', treatment\_H.gut, "-", reference\_C.gut, ".txt", sep=""), sep="\t", quote=FALSE)  
  
  
# and again for significant differential expressed genes only   
# (per our specified p-value cutoff)  
write.table(res\_C\_H\_gut.ad[which(topGenes\_C\_H.gut),],file=paste('/Users/u2093090/Desktop/Drosophila experiments/CTRL vs HTT Ex1-120q flies/NOVAGENE ANALYSIS/RNAseq/RNAseq CtrlvsHTT guts 4d/JS\_s analysis results/4\_counts',treatment\_H.gut,"-",reference\_C.gut,".txt",sep=""), sep="\t", quote=FALSE)  
  
#all your data  
write.table(countsnorm.E.gut, file= '/Users/u2093090/Desktop/Drosophila experiments/CTRL vs HTT Ex1-120q flies/NOVAGENE ANALYSIS/RNAseq/RNAseq CtrlvsHTT guts 4d/JS\_s analysis results/4\_counts/Results\_GUTUpdated.txt', sep="\t")  
  
# Finally, revert to the unordered table for other analyses (topGenes is defined on the unordered table):  
res\_c\_C\_H.gut = results(dds.E.gut, contrast=c(factor,treatment\_H.gut,reference\_C.gut))  
res\_ord.c\_C\_H.gut = res\_c\_C\_H.gut[order(res\_c\_C\_H.gut$padj) , ]

GOterms analysis

#Create a dataframe where you will put the DEGs of interested basing on the padj and logFC desidered. On this data you will run the gene ontology  
res\_C\_H\_gut.ad <-as.data.frame(res\_C\_vs\_H.gut[which(res\_C\_vs\_H.gut$padj < 0.05 & (res\_C\_vs\_H.gut$log2FoldChange>1.5|res\_C\_vs\_H.gut$log2FoldChange<(-1.5))), ])   
#Convert in a table   
write.table(row.names(res\_C\_H\_gut.ad), file='/Users/u2093090/Desktop/Drosophila experiments/CTRL vs HTT Ex1-120q flies/NOVAGENE ANALYSIS/RNAseq/RNAseq CtrlvsHTT guts 4d/JS\_s analysis results/4\_counts/genes\_gutflybase.txt', sep="\t", row.names = F, quote = F)  
topGenes\_C\_H.gut <-res\_C\_vs\_H.gut$padj < 0.05 & (res\_C\_vs\_H.gut$log2FoldChange>1.5|res\_C\_vs\_H.gut$log2FoldChange<(-1.5)) # filter pval and logFC  
  
###Goterms analysis:Molecular functions  
ego <- enrichGO(gene = rownames(res\_C\_H\_gut.ad),#DEGs with the adj p value  
 OrgDb = org.Dm.eg.db,  
 keyType = 'SYMBOL',  
 universe = rownames(ddsHTSeq.E.gut), #the universe will be the all the genes I got from my analysis  
 ont = "MF",  
 pAdjustMethod = "fdr",  
 pvalueCutoff = 0.05,  
 qvalueCutoff = 0.2,  
 readable = T)  
  
options(enrichplot.colours = c("red","blue")) #specific scale colour  
  
#Goplot  
MF\_Gut\_goplot<- goplot(ego,  
 showCategory=30) + ggtitle("GOterms analysis of Drosophila gut samples - Molecular Functions")  
MF\_Gut\_goplot  
  
#Bar plot  
MF\_Gut\_barplot<- barplot(ego,  
 x= 'Count',  
 showCategory=30,   
 color = "p.adjust",  
 title = "GOterms analysis of Drosophila gut samples - Molecular Functions",  
 font.size = 12,  
 label\_format = 25)  
MF\_Gut\_barplot  
  
#Dot plot  
MF\_Gut\_dotplot <- dotplot(ego,  
 x= 'GeneRatio',  
 showCategory=30,   
 color = "p.adjust",  
 title = "GOterms analysis of Drosophila gut samples - Molecular Functions",  
 font.size = 12,  
 label\_format = 25)  
MF\_Gut\_dotplot #Gene ratio equals to the number of DEGs against the number of genes associated with a GO term in whole genome.  
  
#Emap plot  
edo.MF.Gut=pairwise\_termsim(ego) #This function add similarity matrix to the termsim slot of enrichment result.   
MF\_Gut\_emapplot<- emapplot(edo.MF.Gut,  
 cluster.params = list(cluster = TRUE, method = stats::kmeans, n = NULL, legend =  
 TRUE, label\_words\_n = 5, label\_format = 20)) + ggtitle('GOterms analysis of Drosophila gut samples - Molecular Functions')  
MF\_Gut\_emapplot  
  
#Heatplot  
MF\_Gut\_heatplot<- heatplot(ego,showCategory=30, label\_format = 35, symbol = "rect")  
MF\_Gut\_heatplot  
  
#Tree plot  
treeplot(edo.MF.Gut) + ggtitle("GOterms analysis of Drosophila gut samples - Molecular Functions")  
  
head(summary(ego))  
  
###Goterms analysis: Biological Processes  
ego.BP.gut <- enrichGO(gene = rownames(res\_C\_H\_gut.ad),#DEGs with the adj p value  
 OrgDb = org.Dm.eg.db,  
 keyType = 'SYMBOL',  
 universe = rownames(ddsHTSeq.E.gut), #the universe will be the all the genes I got from my analysis  
 ont = "BP",  
 pAdjustMethod = "fdr",  
 pvalueCutoff = 0.05,  
 qvalueCutoff = 0.2,  
 readable = T)  
  
options(enrichplot.colours = c("red","blue")) #for putting the specific scale colour  
  
#Go plot  
BP\_Gut\_goplot<- goplot(ego.BP.gut,  
 showCategory=30) + ggtitle("GOterms analysis of Drosophila gut samples\_Biological Processes")  
BP\_Gut\_goplot  
  
#Bar plot  
BP\_Gut\_barplot<- barplot(ego.BP.gut,  
 x= 'Count',  
 showCategory=30,   
 color = "p.adjust",  
 title = 'GOterms analysis of Drosophila gut samples\_Biological Processes',  
 font.size = 12,  
 label\_format = 25)  
BP\_Gut\_barplot  
  
#Dot plot  
BP\_Gut\_dotplot <- dotplot(ego.BP.gut,  
 x= 'GeneRatio',  
 showCategory=30,   
 color = "p.adjust",  
 title = 'GOterms analysis of Drosophila gut samples\_Biological Processes',  
 font.size = 12,  
 label\_format = 25)  
BP\_Gut\_dotplot  
  
#Emap plot  
edo.BP.Gut=pairwise\_termsim(ego.BP.gut) #This function add similarity matrix to the termsim slot of enrichment result.   
BP\_Gut\_emapplot<- emapplot(edo.BP.Gut,  
 cluster.params = list(cluster = TRUE, method = stats::kmeans, n = NULL, legend =  
 TRUE, label\_words\_n = 2, label\_format = 15)) + ggtitle('GOterms analysis of Drosophila gut samples\_Biological Processes')  
BP\_Gut\_emapplot#not too many dots, so no reason to use it  
  
#Heat plot  
BP\_Gut\_heatplot<- heatplot(ego.BP.gut,showCategory=30)  
BP\_Gut\_heatplot  
  
#Tree plot  
#treeplot(edo.BP.Gut) + ggtitle('GOterms analysis of Drosophila gut samples\_Biological Processes')  
  
head(summary(ego.BP.gut))

Check unique and shared DEGs between gut and head samples\_Venn diagram

# Set directory  
setwd('/Users/u2093090/Desktop')  
  
# Read the gene lists from files  
genes\_head <- unique(scan('/Users/u2093090/Desktop/genes\_flybase\_head.txt', what = character(), skip = 1))  
genes\_gut <- unique(scan('/Users/u2093090/Desktop/genes\_gutflybase.txt', what = character(), skip = 1))  
  
view(genes\_gut)  
view(genes\_head)  
  
# Create a list of gene sets  
venn\_data <- list(Head = genes\_head, Gut = genes\_gut)  
  
# Create the Venn diagram  
ggvenn(venn\_data, fill\_color = c('#FF012C', '#0096FF'))  
  
# Find unique genes in each set  
unique\_in\_head <- setdiff(genes\_head, genes\_gut)  
unique\_in\_gut <- setdiff(genes\_gut, genes\_head)  
  
# Find shared genes  
shared\_genes <- intersect(genes\_head, genes\_gut)  
  
# Display the results  
print(unique\_in\_head)  
print(unique\_in\_gut)  
print(shared\_genes)