**Final used R code for WGCNA:**

#Loading required libraries

library(tidyverse)

library(magrittr)

library(WGCNA)

library(ggplot2)

library(DESeq2)

#Data Loading and transposition

normalized <- read.delim("E:/1semester/Applied data science in biology/Final RNA differential analysis project/counts/normalized.tabular")

View(normalized)

rownames(normalized) <- normalized$X

normalized <- normalized[,-c(1)]

normalized <- t(normalized)

matrixdata <- normalized

#Significant genes selection

matrixdata <- matrixdata[, colnames(matrixdata) %in% significantgenes$GeneID]

#Soft threshold selection

powers <- c(c(1:10), seq(from = 10, to = 30, by = 5), seq(from = 30, to = 44, by = 2))

sft <- pickSoftThreshold(matrixdata, powerVector = powers)

#Plot for soft threshold

sizeGrWindow(9, 5)

par(mfrow = c(1,2));

cex1 = 0.9;

plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])\*sft$fitIndices[,2],

xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit,signed R^2",type="n",

main = paste("Scale independence"));

text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])\*sft$fitIndices[,2],

labels=powers,cex=cex1,col="red");

abline(h = 0.90, col = "blue")

plot(sft$fitIndices[,1], sft$fitIndices[,5],

xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n",

main = paste("Mean connectivity"))

text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers, cex=cex1,col="red")

abline(h = 0.0, col = "blue")

#Gene Co-expression Network Construction

picked\_power = 20

temp\_cor <- cor

cor <- WGCNA::cor

netwk <- blockwiseModules(matrixdata,

#Adjacency Function

power = picked\_power,

networkType = "signed",

#Tree and Block Optionn

deepSplit = 2,

pamRespectsDendro = F,

# detectCutHeight = 0.75,

minModuleSize = 30,

maxBlockSize = 4000,

#Module Adjustments

reassignThreshold = 0,

mergeCutHeight = 0.25,

#TOM

saveTOMs = T,

saveTOMFileBase = "ER",

#Output Options

numericLabels = T,

verbose = 3)

cor <- temp\_cor

#Dendrogram and colors of module

mergedColors = labels2colors(netwk$colors)

plotDendroAndColors(

netwk$dendrograms[[1]],

mergedColors[netwk$blockGenes[[1]]],

"Module colors",

dendroLabels = FALSE,

hang = 0.03,

addGuide = TRUE,

guideHang = 0.05 )

module\_df <- data.frame(

gene\_id = names(netwk$colors),

colors = labels2colors(netwk$colors)

)

#Output of modules information

write\_delim(module\_df,

file = "gene\_modules.txt",

delim = "\t")

#Module epigenes determination

MEs0 <- moduleEigengenes(matrixdata, mergedColors)$eigengenes

MEs0 <- orderMEs(MEs0)

module\_order = names(MEs0) %>% gsub("ME","", .)

MEs0$treatment = row.names(MEs0)

ordersampl <- c("DGRP.332.Control", "DGRP.303.Control" , "DGRP.280.Control",

"DGRP.287.Control" , "DGRP.360.Control" , "DGRP.379.Control",

"DGRP.332.Infected" ,"DGRP.303.Infected" ,"DGRP.280.Infected",

"DGRP.287.Infected" ,"DGRP.360.Infected" ,"DGRP.379.Infected")

MEs0$treatment <- factor(MEs0$treatment, levels = ordersampl)

#Graph of Module-trait relationship

mME = MEs0 %>%

pivot\_longer(-treatment) %>%

mutate(

name = gsub("ME", "", name),

name = factor(name, levels = module\_order)

)

mME %>% ggplot(., aes(x=treatment, y=name, fill=value)) +

geom\_tile() +

theme\_bw() +

scale\_fill\_gradient2(

low = "blue",

high = "red",

mid = "white",

midpoint = 0,

limit = c(-1,1)) +

theme(axis.text.x = element\_text(angle=90)) +

labs(title = "Module-trait Relationships", y = "Modules", fill="corr")

expr\_normalized = t(matrixdata)

#Modules of interest expression data study

modules\_of\_interest = c("green", "turquoise", "brown", "blue", "yellow","red")

submod = module\_df %>%

subset(colors %in% modules\_of\_interest)

row.names(module\_df) = module\_df$gene\_id

subexpr = expr\_normalized[submod$gene\_id,]

#Data preparation for module expression plot

submod\_df = data.frame(subexpr) %>%

mutate(

gene\_id = row.names(.)

) %>%

pivot\_longer(-gene\_id) %>%

mutate(

module = module\_df[gene\_id,]$colors

)

submod\_df$name <- factor(submod\_df$name, levels = ordersampl)

#Module Expression plot

submod\_df %>%

ggplot(aes(x = name, y = value, group = gene\_id)) +

geom\_line(aes(color = module), alpha = 0.2) +

scale\_color\_manual(values = c("yellow" = "yellow", "turquoise" = "turquoise","brown" = "brown", "blue" = "blue","green","green","red","red")) +

theme\_bw() +

theme(axis.text.x = element\_text(angle = 90)) +

facet\_grid(rows = vars(module)) +

labs(x = "Treatment",

y = "Normalised expression")

#Genes of interest selection

genes\_of\_interest = module\_df %>%

subset(colors %in% modules\_of\_interest)

expr\_of\_interest = expr\_normalized[genes\_of\_interest$gene\_id,]

#Loading reuired library

library(gprofiler2)

#Gene Enrichment Analysis

results <- gconvert(query= significantgenes$GeneID, organism = "dmelanogaster", target = "ENSG")

View(results)

#GEA- Blue module

genes\_blu <- module\_df$gene\_id[module\_df$colors == "blue"]

ENSG\_blu <- results[results$target %in% genes\_blu, ]

write.table(ENSG\_blu, file = "ENSG\_blue.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_blu <- gost(

query = ENSG\_blu$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#GEA- Turquoise module

genes\_turq <- module\_df$gene\_id[module\_df$colors == "turquoise"]

ENSG\_turq <- results[results$target %in% genes\_turq, ]

write.table(ENSG\_turq, file = "ENSG\_turq.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_turq <- gost(

query = ENSG\_turq$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#GEA Visualization for Turquoise module

p1 <- gostplot(gost\_turq, capped = TRUE, interactive = TRUE)

pf1 <- gostplot(gost\_turq, capped = TRUE, interactive = FALSE)

publish\_gosttable(gost\_turq,

show\_columns = c("source", "term\_name", "term\_size", "intersection\_size"),

filename = "turquiose gene table.pdf" , limitsize = FALSE)

publish\_gostplot(pf1, highlight\_terms = c("GO:0006468",

"GO:0016310",

"GO:0005886",

"GO:0071944",

"GO:0004672",

"GO:0003824",

"GO:0016773",

"GO:0020037",

"GO:0016301",

"GO:0016491",

"GO:0046906",

"GO:0004674",

"GO:0030246",

"GO:0043167",

"KEGG:00940",

"KEGG:01110"),

width = 15, height = 10, filename = "gostplot+talbe\_turq\_toponly.pdf")

#Gene Ontology Semantic Similarity Analysis for Turquoise Module

gostgem\_turq <- gost(

query = ENSG\_turq$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_turq <- gostgem\_turq$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_turq) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_turq$FDR <- gem\_turq$p.Val

gem\_turq$Phenotype = "+1"

gem\_turq <- gem\_turq[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_turq, file = "gProfiler\_gem\_turq.txt", sep = "\t", quote = F, row.names = F)

#Gene Ontology Semantic Similarity Analysis for Blue Module

gostgem\_blu <- gost(

query = ENSG\_blu$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_blu <- gostgem\_blu$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_blu) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_blu$FDR <- gem\_blu$p.Val

gem\_blu$Phenotype = "+1"

gem\_blu <- gem\_blu[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_blu, file = "gProfiler\_gem\_blu.txt", sep = "\t", quote = F, row.names = F)

#GEA- Yellow module

genes\_yel <- module\_df$gene\_id[module\_df$colors == "yellow"]

ENSG\_yel <- results[results$target %in% genes\_yel, ]

write.table(ENSG\_yel, file = "ENSG\_yele.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_yel <- gost(

query = ENSG\_yel$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#Gene Ontology Semantic Similarity Analysis for Yellow Module

gostgem\_yel <- gost(

query = ENSG\_yel$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_yel <- gostgem\_yel$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_yel) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_yel$FDR <- gem\_yel$p.Val

gem\_yel$Phenotype = "+1"

gem\_yel <- gem\_yel[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_yel, file = "gProfiler\_gem\_yel.txt", sep = "\t", quote = F, row.names = F)

#GEA- Brown module

genes\_bro <- module\_df$gene\_id[module\_df$colors == "brown"]

ENSG\_bro <- results[results$target %in% genes\_bro, ]

write.table(ENSG\_bro, file = "ENSG\_broe.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_bro <- gost(

query = ENSG\_bro$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#Gene Ontology Semantic Similarity Analysis for Brown Module

gostgem\_bro <- gost(

query = ENSG\_bro$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_bro <- gostgem\_bro$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_bro) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_bro$FDR <- gem\_bro$p.Val

gem\_bro$Phenotype = "+1"

gem\_bro <- gem\_bro[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_bro, file = "gProfiler\_gem\_bro.txt", sep = "\t", quote = F, row.names = F)

#GEA- Green module

genes\_gree <- module\_df$gene\_id[module\_df$colors == "green"]

ENSG\_gree <- results[results$target %in% genes\_gree, ]

write.table(ENSG\_gree, file = "ENSG\_greee.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_gree <- gost(

query = ENSG\_gree$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#Gene Ontology Semantic Similarity Analysis for Green Module

gostgem\_gree <- gost(

query = ENSG\_gree$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_gree <- gostgem\_gree$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_gree) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_gree$FDR <- gem\_gree$p.Val

gem\_gree$Phenotype = "+1"

gem\_gree <- gem\_gree[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_gree, file = "gProfiler\_gem\_gre.txt", sep = "\t", quote = F, row.names = F)

#GEA- Red module

genes\_red <- module\_df$gene\_id[module\_df$colors == "red"]

ENSG\_red <- results[results$target %in% genes\_red, ]

write.table(ENSG\_red, file = "ENSG\_rede.txt", sep = "\t",

row.names = FALSE, col.names = FALSE, quote = FALSE)

gost\_red <- gost(

query = ENSG\_red$target,

organism = "dmelanogaster", # Organism under study

sources = c("GO:MF", "GO:BP", "GO:CC", "KEGG"),

user\_threshold = 0.05, # P-value threshold for significance

ordered\_query = FALSE

)

#Gene Ontology Semantic Similarity Analysis for Red Module

gostgem\_red <- gost(

query = ENSG\_red$target,

organism = "dmelanogaster",

evcodes = TRUE,

multi\_query = FALSE,

sources = c("GO", "REAC", "MIRNA", "CORUM", "HP", "HPA", "WP"),

user\_threshold = 0.05,

ordered\_query = FALSE)

gem\_red <- gostgem\_red$result[,c("term\_id", "term\_name", "p\_value", "intersection")]

colnames(gem\_red) <- c("GO.ID", "Description", "p.Val", "Genes")

gem\_red$FDR <- gem\_red$p.Val

gem\_red$Phenotype = "+1"

gem\_red <- gem\_red[,c("GO.ID", "Description", "p.Val", "FDR", "Phenotype", "Genes")]

write.table(gem\_red, file = "gProfiler\_gem\_red.txt", sep = "\t", quote = F, row.names = F)

#Using DESeq2 results

#count of upregulated and downregulated genes

deseq\_results <- read.delim("E:/1semester/Applied data science in biology/Final RNA differential analysis project/counts/deseq2 result.tabular")

# Upregulated genes

upregulated\_genes <- subset(deseq\_results, Padj < 0.05 & log2FC > 0)$GeneID

# Downregulated genes

downregulated\_genes <- subset(deseq\_results, Padj < 0.05 & log2FC < 0)$GeneID

# Write the lists to files

write.table(upregulated\_genes, file = "upregulated\_genes.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)

write.table(downregulated\_genes, file = "downregulated\_genes.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)