**Feeding disruptions lead to a significant increase in disease modules in adult mice**

Xiaoqin Mou, Pengxing Nie, Renrui Chen, Yang Cheng, Guang-Zhong Wang\*

\*Corresponding author: Tel: (0086) 021-54920578;

Email: [guangzhong.wang@picb.ac.cn](mailto:guangzhong.wang@picb.ac.cn) (G.W.)

**Supplemental table**

Table S1. Detailed information on disease genes across three disease sets

Table S2. Significant disease modules under six feeding conditions

Table S3. Significant disease modules under ALF and TRF conditions

Table S4. Significant disease modules under TB and WT conditions

Table S5. Enrichment Analysis of WGCNA modules and disease genes

Table S6. GO pathway enrichment for WGCNA modules significantly associated with disease genes

Table S7. Comparison of co-expression levels in disease modules across all experiments

**Supplemental code**

############codes for the WGCNA Co-expression Network ############

for (con in conditions) {

cor <- WGCNA::cor

if(T){

net <- blockwiseModules(

datExpr,

power = 16,

maxBlockSize = ncol(datExpr),

corType = "pearson",

networkType = "signed",

TOMType = "signed",

minModuleSize = 30,

mergeCutHeight = 0.25,

numericLabels = TRUE,

saveTOMs = F,

verbose = 3

)

table(net$colors)

}

cor<-stats::cor

## Module visualization

if(T){

# Convert labels to colors for plotting

moduleColors <- labels2colors(net$colors)

table(moduleColors)

# Plot the dendrogram and the module colors underneath

pdf("step3\_genes-modules\_ClusterDendrogram.pdf",width = 16,height = 12)

plotDendroAndColors(net$dendrograms[[1]], moduleColors[net$blockGenes[[1]]],

"Module colors",

dendroLabels = FALSE, hang = 0.03,

addGuide = TRUE, guideHang = 0.05)

dev.off()

}

}

###### Calculation of significant disease modules #########

ismodule <- function(disease\_gene,dataexpr){

disease\_gene <- dic$ensembl\_gene\_id[which(dic$HGNC.symbol %in% disease\_gene)]

genes <- intersect(row.names(dataexpr),disease\_gene)

if(length(genes) > 2){

subexpr <- dataexpr[genes,]

cormat <- cor(t(subexpr))

obvalue <- mean(cormat[lower.tri(cormat)])

randomvalue <- sapply(1:1000,function(x){randomgenes <- sample(rownames(dataexpr),length(genes))

randomcormat <- cor(t(dataexpr[randomgenes,]))

return(mean(randomcormat[lower.tri(randomcormat)]))})

pvalue <- sum(randomvalue > obvalue,na.rm = T)/length(randomvalue)

return(c(obvalue = obvalue,pvalue=pvalue,gene\_num=length(genes)))

}else{

return(c(obvalue = NA,pvalue=NA,gene\_num=NA))

}

}

cor\_module <- function(disease\_set,exp){

res\_ismodule <- data.frame(obvalue=NA,pvalue=NA,gene\_num=NA,disease = names(disease\_set))

for (m in names(disease\_set)) {

disease\_gene <- disease\_set[[m]]

a=ismodule(disease\_gene,exp)

res\_ismodule[which(res\_ismodule$disease==m),1:3] <- a

}

return(res\_ismodule)

}

library(stringr)

WT <- U2OS\_TPM[,1:18]

TB <- U2OS\_TPM[,19:36]

# res\_mycOFFrep1 <- cor\_module(disease\_set22,mycOFFdata1)

# substitute(test)

list\_file <- c(list(WT),list(TB))

names(list\_file) <- c("WT","TB")

result\_myc <- c()

for(i in 1:length(list\_file)){

res\_myc <- cor\_module(disease\_set22,list\_file[[i]])

#write.csv(res\_myc,paste0("./ismodules/", names(list\_file)[i],"\_ismodule\_disease22\_modules\_mean\_cor.csv"))

res\_myc$group <- names(list\_file)[i]

result\_myc <- rbind(result\_myc,res\_myc)

}

result\_myc22 <- result\_myc

table(result\_myc22$group[which(result\_myc22$pvalue<0.05)])

result\_myc <- c()

for(i in 1:length(list\_file)){

res\_myc <- cor\_module(disease\_geenset3,list\_file[[i]])

#write.csv(res\_myc,paste0("./ismodules/", names(list\_file)[i],"\_ismodule\_disease22\_modules\_mean\_cor.csv"))

res\_myc$group <- names(list\_file)[i]

result\_myc <- rbind(result\_myc,res\_myc)

}

result\_myc232 <- result\_myc

table(result\_myc232$group[which(result\_myc232$pvalue<0.05)])

result\_myc <- c()

for(i in 1:length(list\_file)){

res\_myc <- cor\_module(diseSETcirc40,list\_file[[i]])

res\_myc$group <- names(list\_file)[i]

result\_myc <- rbind(result\_myc,res\_myc)

}

result\_diseSETcirc40 <- result\_myc

table(result\_diseSETcirc40$group[which(result\_diseSETcirc40$pvalue<0.05)])

#######################