#WGCNA ANALYSIS

library(WGCNA)

exprMat <-"bindgeo\_exp.csv"

dataExpr<-read.table(exprMat, sep=',', row.names=1, header=T,

quote="", comment="", check.names=F)

str(dataExpr)

my\_mad <- function(x){mad(x,na.rm = TRUE)}

m.mad <- apply(dataExpr,2,my\_mad)

datExpr0 <- dataExpr[,which(m.mad > quantile(m.mad, probs=seq(0, 1, 0.25))[2])]

write.table(datExpr0,"datExpr0.csv",row.names=T,col.names=T,sep=",")

dataExpr <- as.data.frame(t(datExpr0))

datExpr0<-dataExpr

gsg = goodSamplesGenes(datExpr0, verbose = 3)

gsg$allOK

sampleTree = hclust(dist(datExpr0), method = "average")

pdf(file = "1.sampleClustering.pdf", width = 15, height = 8)

par(cex = 0.6)

par(mar = c(0,6,6,0))

plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="", cex.lab = 2,

cex.axis = 1.5, cex.main = 2)

### Plot a line to show the cut

#abline(h = 7000, col = "red")##剪切高度不确定，故无红线

dev.off()

pdf("2\_sample clutering\_delete\_outliers.pdf", width = 6, height = 8)

par(cex = 0.6)

par(mar = c(0,6,6,0))

cutHeight = 130

plot(sampleTree, main = "Sample clustering to detect outliers", sub="", xlab="", cex.lab = 2,

cex.axis = 1.5,cex.main = 2) +

abline(h = 130, col = "red")

dev.off()

#dataExpr<-dataExpr[-(3),]

#datExpr0<-dataExpr

traitData = read.csv("sampleinfo.csv",row.names=1)

head(traitData)

allTraits = traitData

dim(allTraits)

names(allTraits)

fpkmSamples = rownames(datExpr0)

traitSamples =rownames(allTraits)

traitRows = match(fpkmSamples, traitSamples)

datTraits = allTraits[traitRows,]

datTraits=allTraits

rownames(datTraits)

#datTraits=datTraits[-(4:5),]

rownames(datTraits)

datTraits

sampleTree2 = hclust(dist(datExpr0), method = "average")

# Convert traits to a color representation: white means low, red means high, grey means missing entry

traitColors = numbers2colors(datTraits, signed = FALSE)

pdf(file="3\_Sample\_dendrogram\_and\_trait\_heatmap.pdf",width=8 ,height= 6)

plotDendroAndColors(sampleTree2, traitColors,

groupLabels = names(datTraits),

main = "Sample dendrogram and trait heatmap",cex.colorLabels = 1.5, cex.dendroLabels = 1, cex.rowText = 2)

dev.off()

enableWGCNAThreads()

powers = c(1:30)

sft = pickSoftThreshold(datExpr0, powerVector = powers, verbose = 5)

pdf(file="4\_软阈值选择.pdf",width=12, height = 8)

par(mfrow = c(1,2))

cex1 = 0.9

# Scale-free topology fit index as a function of the soft-thresholding power

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");

# this line corresponds to using an R^2 cut-off of h

abline(h=0.9,col="red")

# Mean connectivity as a function of the soft-thresholding power

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")

dev.off()

sft = pickSoftThreshold(datExpr0, powerVector = powers)

sft$powerEstimate

softPower =8

adjacency = adjacency(datExpr0, power = softPower)

TOM = TOMsimilarity(adjacency);

dissTOM = 1-TOM

geneTree = hclust(as.dist(dissTOM), method = "average");

pdf(file="4\_Gene clustering on TOM-based dissimilarity.pdf",width=6,height=4)

plot(geneTree, xlab="", sub="", main = "Gene clustering on TOM-based dissimilarity",

labels = FALSE, hang = 0.04)

dev.off()

minModuleSize =30

# Module identification using dynamic tree cut:

dynamicMods = cutreeDynamic(dendro = geneTree, distM = dissTOM,

deepSplit = 2, pamRespectsDendro = FALSE,

minClusterSize = minModuleSize);

table(dynamicMods)

# Convert numeric lables into colors

dynamicColors = labels2colors(dynamicMods)

table(dynamicColors)

# Plot the dendrogram and colors underneath

#sizeGrWindow(8,6)

pdf(file="5\_Dynamic Tree Cut.pdf",width=8,height=6)

plotDendroAndColors(geneTree, dynamicColors, "Dynamic Tree Cut",

dendroLabels = FALSE, hang = 0.03,

addGuide = TRUE, guideHang = 0.05,

main = "Gene dendrogram and module colors")

dev.off()

MEList = moduleEigengenes(datExpr0, colors = dynamicColors)

MEs = MEList$eigengenes

# Calculate dissimilarity of module eigengenes

MEDiss = 1-cor(MEs);

# Cluster module eigengenes

METree = hclust(as.dist(MEDiss), method = "average")

# Plot the result

#sizeGrWindow(7, 6)

pdf(file="6\_Clustering of module eigengenes.pdf",width=7,height=6)

plot(METree, main = "Clustering of module eigengenes",

xlab = "", sub = "")

MEDissThres = 0.25

# Plot the cut line into the dendrogram

abline(h=MEDissThres, col = "red")

dev.off()

merge = mergeCloseModules(datExpr0, dynamicColors, cutHeight = MEDissThres, verbose = 3)

# The merged module colors

mergedColors = merge$colors

# Eigengenes of the new merged modules:

mergedMEs = merge$newMEs

table(mergedColors)

#EXPORT ALL MODULES

color<-unique(mergedColors)

for (i in 1:length(color)) {

y=t(assign(paste(color[i],"expr",sep = "."),datExpr0[mergedColors==color[i]]))

write.csv(y,paste('6',color[i],"csv",sep = "."),quote = F)

}

#save.image(file = "module\_splitted.RData")

##

pdf(file="7\_merged dynamic.pdf", width = 8, height = 8)

plotDendroAndColors(geneTree, cbind(dynamicColors, mergedColors),

c("Dynamic Tree Cut", "Merged dynamic"),

dendroLabels = FALSE, hang = 0.03,

addGuide = TRUE, guideHang = 0.05)

dev.off()

moduleColors = mergedColors

# Construct numerical labels corresponding to the colors

colorOrder = c("grey", standardColors(50))

moduleLabels = match(moduleColors, colorOrder)-1

MEs = mergedMEs

nGenes = ncol(datExpr0);

nSamples = nrow(datExpr0);

#

MEs0 = moduleEigengenes(datExpr0, moduleColors)$eigengenes

MEs = orderMEs(MEs0)

moduleTraitCor = cor(MEs, datTraits, use = "p");

moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);

#

pdf(file="5.Module-trait relationships.pdf",width=6,height=6)

#

textMatrix = paste(signif(moduleTraitCor, 2), "\n(",

signif(moduleTraitPvalue, 1), ")", sep = "");

dim(textMatrix) = dim(moduleTraitCor)

par(mar = c(6, 8.5, 3, 3));

labeledHeatmap(Matrix = moduleTraitCor,

xLabels = names(datTraits),

yLabels = names(MEs),

ySymbols = names(MEs),

colorLabels = FALSE,

colors = blueWhiteRed(50),

textMatrix = textMatrix,

setStdMargins = FALSE,

cex.text = 0.5,

zlim = c(-1,1),

main = paste("Module-trait relationships"))

dev.off()

trait\_a = as.data.frame(datTraits$RA)

datTraits

names(trait\_a) = "trait\_a"

modNames = substring(names(MEs), 3)

geneModuleMembership = as.data.frame(cor(datExpr0, MEs, use = "p"));

MMPvalue = as.data.frame(corPvalueStudent(as.matrix(geneModuleMembership), nSamples));

names(geneModuleMembership) = paste("MM", modNames, sep="");

names(MMPvalue) = paste("p.MM", modNames, sep="");

geneTraitSignificance = as.data.frame(cor(datExpr0, trait\_a, use = "p"));#和性状的关联

GSPvalue = as.data.frame(corPvalueStudent(as.matrix(geneTraitSignificance), nSamples));

names(geneTraitSignificance) = paste("GS.", names(trait\_a), sep="");

names(GSPvalue) = paste("p.GS.", names(trait\_a), sep="")

module = "salmon"

column = match(module, modNames);

moduleGenes = moduleColors==module;

pdf(file="7.Module membership vs. gene significance.pdf",width=6,height=6);

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

verboseScatterplot(

abs(geneModuleMembership[moduleGenes, column]),

abs(geneTraitSignificance[moduleGenes, 1]),

xlab = paste("Module Membership in", module, "module"),

ylab = "Gene significance for proliferating",

main = paste("Module membership vs. gene significance\n"),

abline = TRUE,

pch = 21,

cex.main = 1.2,

cex.lab = 1.2,

cex.axis = 1.2,

col = "black",

bg = module

)

dev.off()

module = "salmon"

column = match(module, modNames)

moduleGenes = moduleColors==module

blue\_module<-as.data.frame(dimnames(data.frame(datExpr0))[[2]][moduleGenes])

names(blue\_module)="genename"

MM<-abs(geneModuleMembership[moduleGenes,column])

GS<-abs(geneTraitSignificance[moduleGenes, 1])

blue\_MMGS<-as.data.frame(cbind(MM,GS))

rownames(blue\_MMGS)=blue\_module$genename

hub\_b<-abs(blue\_MMGS$MM)>0.8&abs(blue\_MMGS$GS)>0.5

table(hub\_b)

blue\_hub\_b<-subset(blue\_MMGS, abs(blue\_MMGS$MM)>0.8)

write.csv(blue\_hub\_b, "hubgene\_MMGS\_123.csv")