Codes and packages of R, version 4.0.5,

Work Platform, Rstudio

**Variance analysis using the limma package (Figure 2)**

library(limma)

library(dplyr)

df <- read.table("Counts\_data.txt", header = T, sep = "\t", row.names = 1, check.names = F)

head(df)

list <-c(rep("group 1"), rep("group 2")) %>% factor(., levels = c("group 1", "group 2"), ordered = F)

> head(list)

list <- model.matrix(~factor(list)+0)

colnames(list) <-c("group 1", "group 2")

df.fit <-lmFit(df, list)

df.matrix <-makeContrasts(tumor - normal, levels = list)

fit <- contrasts.fit(df.fit, df.matrix)

fit <- eBayes(fit)

tempOutput <- topTable(fit,n = Inf, adjust = "fdr")

**Volcano map (Figure 2AB)**

library(tidyverse)

library(ggplot2)

data <-tempOutput

data <- read.table("~/file.txt", header = T)

head(data)

data$col <- "no significant"

data$col[data$padj < 0.05 & data$logFC > 2] <- "Up"

data$col[data$padj < 0.05 & data$logFC < -2] <- "Down"

data$col <- factor (data$col, levels = c("Down", "no significant","Up"))

data$size <- 1

data$size[data$padj < 0.05 & data$logFC > 2] <- 2

data$size[data$padj < 0.05 & data$logFC < -2] <- 2

ggplot() +

 geom\_point(data = data, aes(logFC, -log10(padj), colour = col, fill = col),

 size = data$size) +

 scale\_colour\_manual(values = c("#4DBBD5", "grey", "#E64B35")) +

 geom\_vline(xintercept = c(-2, 2), color="grey40", linetype=2) +

 geom\_hline(yintercept = -log10(0.05), color="grey40", linetype=2)

**Heatmap (Figure 1FG)**

library(tidyverse)

library(ComplexHeatmap)

pheno <- read.table("~/pheno.txt", header = T)

head(pheno)

expr <- read.table("~/expr.txt", header = T, row.names = T)

head(expr[, 1:5])

col = c("#4DBBD5", "#E64B35")

names(col) <- c("group1", "group2")

top <- HeatmapAnnotation(group = pheno$group, col = list(group = col))

Heatmap(as.matrix(t(scale(t(expr)))),

 top\_annotation = top,

show\_column\_names = F)

**WGCNA (Figure 3)**

BiocManager::install("WGCNA")

library('WGCNA')

options(stringsAsFactors = FALSE)

femData=read.csv("document.csv")

dim(femData)

names(femData)

datExpr0 = as.data.frame(t(femData[, -c(1:8)]))

names(datExpr0) = femData$substanceBXH

gsg = goodSamplesGenes(datExpr0, verbose = 3);

gsg$allOK

if(!gsg$allOK)

{if(sum(!gsg$goodGenes)>0)

printFlush(paste("Removinggenes:",paste(names(datExpr0)[!gsg$goodGenes], collapse =",")));

 if(sum(!gsg$goodSamples)>0)

printFlush(paste("Removingsamples:",paste(rownames(datExpr0)[!gsg$goodSamples], collapse =",")));

 datExpr0 = datExpr0[gsg$goodSamples, gsg$goodGenes]}

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

sizeGrWindow(12,9) pdf(file="Plots/sampleClustering.pdf",width=12,height=9);

par(cex = 0.6)

par(mar =c(0,4,2,0))

plot(sampleTree, main ="Sample clustering to detectoutliers",sub="", xlab="", cex.lab = 1.5,cex.axis= 1.5, cex.main = 2

powers =c(c(1:10),seq(from = 12, to=20,by=2))

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

sizeGrWindow(9, 5)

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

cex1 = 0.9;

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

xlab="SoftThreshold(power)",ylab="ScaleFreeTopologyModelFit,signedR^2",type="n",

 main =paste("Scaleindependence"));

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

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

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

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

 xlab="SoftThreshold(power)",ylab="MeanConnectivity", type="n",

 main =paste("Meanconnectivity"))

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

net=blockwiseModules(datExpr,power=6,

TOMType ="unsigned",

minModuleSize=30,

reassignThreshold=0,

mergeCutHeight=0.25,

numericLabels =TRUE,

pamRespectsDendro=FALSE,

saveTOMs = TRUE,

saveTOMFileBase ="femaleMouseTOM", verbose = 3)

sizeGrWindow(12, 9)

mergedColors = labels2colors(net$colors)

plotDendroAndColors(net$dendrograms[[1]],mergedColors[net$blockGenes[[1]]],"Modulecolors",dendroLabels = FALSE, hang=0.03,addGuide = TRUE, guideHang = 0.05)

moduleLabels = net$colors

moduleColors = labels2colors(net$colors)

MEs = net$MEs;

geneTree = net$dendrograms[[1]];

save(MEs, moduleLabels, moduleColors, geneTree,

 file="FemaleLiver-02-networkConstruction-auto.RData")

**GO (Figure 4A-C)**

diff<-read.csv(file="C:/Users/Desktop/data.csv")

library(AnnotationDbi)

library(org.Hs.eg.db)

library(clusterProfiler)

library(dplyr)

library(ggplot2)

gene.df<-bitr(diff$SYMBOL,fromType="SYMBOL",

toType="ENTREZID", OrgDb = org.Hs.eg.db）

gene <- gene.df$ENTREZID

ego\_ALL <- enrichGO(gene = gene,

 OrgDb=org.Hs.eg.db,

 keyType = "ENTREZID",

 ont = "ALL",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff = 0.01,

 qvalueCutoff = 0.05,

 readable = TRUE)

 ego\_CC <- enrichGO(gene = gene,

 OrgDb=org.Hs.eg.db,

 keyType = "ENTREZID",

 ont = "CC",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff = 0.01,

 qvalueCutoff = 0.05,

 readable = TRUE)

ego\_BP <- enrichGO(gene = gene,

 OrgDb=org.Hs.eg.db,

 keyType = "ENTREZID",

 ont = "BP",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff = 0.01,

 qvalueCutoff = 0.05,

 readable = TRUE)

ego\_MF <- enrichGO(gene = gene,

 OrgDb=org.Hs.eg.db,

 keyType = "ENTREZID",

 ont = "MF",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff = 0.01,

 qvalueCutoff = 0.05,

 readable = TRUE)

ego\_ALL <- as.data.frame(ego\_ALL)

ego\_result\_BP <- as.data.frame(ego\_BP)

ego\_result\_CC <- as.data.frame(ego\_CC)

ego\_result\_MF <- as.data.frame(ego\_MF)

ego <- rbind(ego\_result\_BP,ego\_result\_CC,ego\_result\_MF)

write.csv(ego\_ALL,file = "ego\_ALL.csv",row.names = T)

write.csv(ego\_result\_BP,file = "ego\_result\_BP.csv",row.names = T)

write.csv(ego\_result\_CC,file = "ego\_result\_CC.csv",row.names = T)

write.csv(ego\_result\_MF,file = "ego\_result\_MF.csv",row.names = T)

write.csv(ego,file = "ego.csv",row.names = T)

ego\_result\_BP <- as.data.frame(ego\_BP)[1:display\_number[1], ]

ego\_result\_CC <- as.data.frame(ego\_CC)[1:display\_number[2], ]

ego\_result\_MF <- as.data.frame(ego\_MF)[1:display\_number[3], ]

go\_enrich\_df <- data.frame(

ID=c(ego\_result\_BP$ID,ego\_result\_CC$ID,ego\_result\_MF$ID), Description=c(ego\_result\_BP$Description,ego\_result\_CC$Description,ego\_result\_MF$Description),

GeneNumber=c(ego\_result\_BP$Count,ego\_result\_CC$Count,ego\_result\_MF$Count),

type=factor(c(rep("biological process", display\_number[1]),

 rep("cellular component", display\_number[2]),

 rep("molecular function", display\_number[3])),

levels=c("biological process","cellular component","molecular function" )))

for(i in 1:nrow(go\_enrich\_df)){

 description\_splite=strsplit(go\_enrich\_df$Description[i],split = " ")

 description\_collapse=paste(description\_splite[[1]][1:5],collapse = " ") #

 go\_enrich\_df$Description[i]=description\_collapse

go\_enrich\_df$Description=gsub(pattern="NA","",

go\_enrich\_df$Description)

go\_enrich\_df$type\_order=factor(rev(as.integer(rownames(go\_enrich\_df))),labels=rev(go\_enrich\_df$Description))

COLS <- c("#66C3A5", "#8DA1CB", "#FD8D62")

ggplot(data=go\_enrich\_df, aes(x=type\_order,

y=GeneNumber, fill=type)) +

geom\_bar(stat="identity", width=0.8) +

 scale\_fill\_manual(values = COLS) +

 coord\_flip() +

 xlab("GO term") +

 ylab("Gene\_Number") +

 labs(title = "The Most Enriched GO Terms")+

 theme\_bw()

go\_enrich\_df$type\_order=factor(rev(as.integer(rownames(go\_enrich\_df))),labels=rev(go\_enrich\_df$Description))

COLS <- c("#66C3A5", "#8DA1CB", "#FD8D62")

ggplot(data=go\_enrich\_df, aes(x=type\_order,y=GeneNumber, fill=type))+

geom\_bar(stat="identity", width=0.8) +

 scale\_fill\_manual(values = COLS) +

 theme\_bw() +

 xlab("GO term") +

 ylab("Num of Genes") +

 labs(title = "The Most Enriched GO Terms")+

theme(axis.text.x=element\_text(face = "bold", color="gray50",angle = 70,vjust = 1, hjust = 1 ))

**KEGG (Figure 4D)**

kk <- enrichKEGG(gene = gene,keyType = "kegg",organism= "human", qvalueCutoff = 0.05, pvalueCutoff= 0.05)

hh <- as.data.frame(kk)！

rownames(hh) <- 1:nrow(hh)

hh$order=factor(rev(as.integer(rownames(hh))),labels = rev(hh$Description))

ggplot(hh,aes(y=order,x=Count,fill=p.adjust))+

 geom\_bar(stat = "identity",width=0.7)+

 scale\_fill\_gradient(low = "red",high ="blue" )+

 labs(title = "KEGG Pathways Enrichment",

x="Gene numbers",

y="Pathways")+

 theme(axis.title.x = element\_text(face = "bold",size = 16),

 axis.title.y = element\_text(face = "bold",size = 16),

 legend.title = element\_text(face = "bold",size = 16))+

theme\_bw()

hh<- as.data.frame(kk)

rownames(hh) <- 1:nrow(hh)

hh$order=factor(rev(as.integer(rownames(hh)))

labels=rev(hh$Description))

ggplot(hh,aes(y=order,x=Count))+

geom\_point(aes(size=Count,color=-1\*p.adjust))+

scale\_color\_gradient(low="green",high="red")+

labs(color=expression(p.adjust,size="Count"), x="Gene Number",y="Pathways",title="KEGG Pathway Enrichment")+

theme\_bw()

**Chord diagram (Figure 5B)**

library（circlize）

chordDiagram(mat)

library(tidyverse)

library(reshape2)

df <- melt(mat) %>% rename('from' = 'Var1','to' = 'Var2')

head(df)

from to value

chordDiagram(df)

chordDiagram(mat, order = c("S2", "S1", "S3", "E4", "E1", "E5", "E2", "E6", "E3"))

circos.clear()

circos.par(gap.after = c(rep(5,nrow(mat)-1),15,rep(5,ncol(mat)-1),15))

chordDiagram(mat)

circos.clear()

circos.par(gap.after = c("S1"=5, "S2"=5, "S3"=15, "E1"=5, "E2"=5,

 "E3"=5, "E4"=5, "E5"=5, "E6"=15))

chordDiagram(mat, big.gap=30)

circos.par(start.degree=90, clock.wise=F)

chordDiagram(mat)

col=c(S1="red", S2="green", S3 ="blue",E1 ="black", E2 ="cyan", E3 = "orange", E4="green", E5="lavender", E6 = "grey", E7="light blue")

chordDiagram(mat,grid.col=col,transparency=0.5)

circos.clear()

chordDiagram(mat, grid.col = col, link.lwd = 2, link.lty = 2, link.border = "red")

circos.clear()

chordDiagram(mat, grid.col=col, transparency=0, link.zindex = rank(mat))

**Relevance Heat Map (Figure 6AB)**

library(tidyverse)

library(corrplot)

library(ggplot2)

library(ggcorrplot)

data <- read.table("~/file.txt", header = T)

rownames(data) <- data[,1]

data <- data[,-1]

corrplot(as.matrix(data))

p <- ggcorrplot(data)

**LASSO (Figure 7)**

library(readxl);

library(caret);

library(glmnet);

library(corrplot)

library(Metrics);

library(ggplot2)

data<-read.csv("C:/Users/data.csv",sep = ",")

data\_cor <- cor(data)

corrplot.mixed(data\_cor,tl.col="black",tl.pos = "d",number.cex = 0.8)

set.seed(123)

d\_index <- createDataPartition(data$Outcome,p = 0.7)

train\_d <- data[d\_index$Resample1,]

test\_d <- data[-d\_index$Resample1,]

scal <- preProcess(train\_d,method = c("center","scale"))

train\_ds <- predict(scal,train\_d)

test\_ds <- predict(scal,test\_d)

lambdas <- seq(0,2, length.out = 100)

X <- as.matrix(train\_ds[,1:8])

Y <- train\_ds[,9]

set.seed(1004)

lasso\_model <- cv.glmnet(X,Y,alpha = 1,nlambda = 200,nfolds =3)

plot(lasso\_model)

plot(lasso\_model$glmnet.fit, "lambda", label = T)

lasso\_model$lambda.min

lasso\_min <- lasso\_model$lambda.min

lasso\_best <- glmnet(X,Y,alpha = 1,lambda = lasso\_min)

coef(lasso\_best)

test\_pre <- predict(lasso\_best,as.matrix(test\_ds[,1:8]))

sprintf("The mean absolute error after standardization is: %f",mae(test\_ds$Outcome,test\_pre))

test\_pre\_o <- as.vector(test\_pre[,1] \* scal$std[9] + scal$mean[9])

sprintf("The mean absolute error before standardization is: %f",mae(test\_d$Outcome,test\_pre\_o))

**Lollipop Chart (Figure 8)**

install.packages("ggpubr")

library(ggpubr)

library(ggplot2)

df<-read.delim("Data.txt")

ggdotchart(df, x=colnames(df)[1],

 y=colnames(df)[2],

 color=colnames(df)[1],

 palette=rainbow(dim(df)[1]),

 sorting="none",

c("ascending", "descending", "none"),

 add="segments",

c("none", "segment")

 dot.size=10,

 add.params=list(color="lightgray", size=5),

 position=position\_dodge(0.1),

 label=round(df[[2]]),

 font.label=list(color="white",

 size=9,

 vjust=0.5)