R codes

\*All the file names that are specific to the user are highlighted in Yellow

library(GEOquery)

library(limma)

# load series and platform data from GEO

gset <- getGEO("GSE15932", GSEMatrix =TRUE, AnnotGPL=TRUE)

if (length(gset) > 1) idx <- grep("GPL570", attr(gset, "names")) else idx <- 1

gset <- gset[[idx]]

# make proper column names to match toptable

fvarLabels(gset) <- make.names(fvarLabels(gset))

# group membership for all samples

gsms <- "XXXXXXXX00000000XXXXXXXX11111111"

sml <- strsplit(gsms, split="")[[1]]

# filter out excluded samples (marked as "X")

sel <- which(sml != "X")

sml <- sml[sel]

gset <- gset[ ,sel]

# log2 transformation

ex <- exprs(gset)

qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T))

LogC <- (qx[5] > 100) ||

(qx[6]-qx[1] > 50 && qx[2] > 0)

if (LogC) { ex[which(ex <= 0)] <- NaN

exprs(gset) <- log2(ex) }

# assign samples to groups and set up design matrix

gs <- factor(sml)

groups <- make.names(c("T2DM","Healthy"))

levels(gs) <- groups

gset$group <- gs

design <- model.matrix(~group + 0, gset)

colnames(design) <- levels(gs)

gset <- gset[complete.cases(exprs(gset)), ] # skip missing values

fit <- lmFit(gset, design) # fit linear model

# set up contrasts of interest and recalculate model coefficients

cts <- c(paste(groups[1],"-",groups[2],sep=""))

cont.matrix <- makeContrasts(contrasts=cts, levels=design)

fit2 <- contrasts.fit(fit, cont.matrix)

# compute statistics and table of top significant genes

fit2 <- eBayes(fit2, 0.01)

tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)

tT <- subset(tT, select=c("ID","adj.P.Val","P.Value","t","B","logFC","GB\_ACC","SPOT\_ID","Gene.Symbol","Gene.symbol","Gene.title"))

write.table(tT, file=stdout(), row.names=F, sep="\t")

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

# General expression data analysis

ex <- exprs(gset)

# box-and-whisker plot

ord <- order(gs) # order samples by group

palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02",

"#66A61E", "#A6761D", "#B32424", "#B324B3", "#666666"))

par(mar=c(7,4,2,1))

title <- paste ("GSE15932", "/", annotation(gset), sep ="")

boxplot(ex[,ord], boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs[ord])

legend("topleft", groups, fill=palette(), bty="n")

### R code for filteration

##################GSE15932

######## log transformation\_auto-detect\_Force normalization = 0, limma precision = 0, significance level = 0

###### gene ID match

rm(list = ls())

PC<- read.delim("~/PC\_H\_GSE15932.top.table.tsv")

T2D<- read.delim("~/T2D\_H\_GSE15932.top.table.tsv")

library(dplyr)

## X = T2D

## Y = PC

T2D\_PC=merge(T2D, PC, by.x = "ID", by.y = "ID")

Dist=((abs((T2D$logFC)-(PC$logFC)))/(max((abs(T2D$logFC)),(abs(PC$logFC)))))

T2D\_PC\_Dist=data.frame(T2D\_PC,"Dist"=Dist)

Z=unique(T2D\_PC\_Dist$Gene.symbol.y)

getwd()

setwd("D:/PhD/T2D\_Cancer\_Project/Geo/filter")

write.csv(T2D\_PC\_Dist, "GEO2R\_T2D\_PC\_GSE1932.csv", quote = FALSE, row.names = FALSE)

#d = |x\_i - y\_i| divided by max(|x\_i|, |y\_i|)

#2) filter the genes with d<=10%

#where:

#x\_i: expression values for ith gene in T2D

#y\_i: expression values for ith gene in PC

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

filter\_pValue\_T2D=T2D\_PC\_Dist%>%

filter(adj.P.Val.x<=0.05)

filter\_pValue\_PC=filter\_pValue\_T2D%>%

filter(adj.P.Val.y<=0.05)

A=unique(filter\_pValue\_PC$Gene.symbol.x)

write.csv(filter\_pValue\_PC, "GEO2R\_T2D\_PC\_GSE1932\_filter1.csv", quote = FALSE, row.names = FALSE)

write.csv(A, "GEO2R\_T2D\_PC\_GSE1932\_filter1\_unique.csv", quote = FALSE, row.names = FALSE)

filter\_T2D\_FC\_A=filter\_pValue\_PC%>%

filter(logFC.x>=0.5)

filter\_T2D\_FC\_B=filter\_pValue\_PC%>%

filter(logFC.x<=-0.5)

filter\_T2D=rbind(filter\_T2D\_FC\_A, filter\_T2D\_FC\_B)

AA=unique(filter\_T2D$Gene.symbol.x)

filter\_PC\_FC\_A=filter\_T2D%>%

filter(logFC.y>=0.5)

filter\_PC\_FC\_B=filter\_T2D%>%

filter(logFC.y<=-0.5)

filter\_PC=rbind(filter\_PC\_FC\_A, filter\_PC\_FC\_B)

AB=unique(filter\_PC$Gene.symbol.y)

write.csv(filter\_PC, "GEO2R\_T2D\_PC\_GSE1932\_filter2.csv", quote = FALSE, row.names = FALSE)

write.csv(AB, "GEO2R\_T2D\_PC\_GSE1932\_filter2\_unique.csv", quote = FALSE, row.names = FALSE)

filter\_dist=filter\_PC%>%

filter(Dist<=0.1)

AAx=unique(filter\_dist$Gene.symbol.x)

AAy=unique(filter\_dist$Gene.symbol.y)

write.csv(filter\_dist, "GEO2R\_T2D\_PC\_GSE1932\_filter3.csv", quote = FALSE, row.names = FALSE)

write.csv(AAx, "GEO2R\_T2D\_PC\_GSE1932\_filter3\_unique.csv", quote = FALSE, row.names = FALSE)

########################GSE15932 \_T2D\_PC both

######## log transformation\_auto-detect\_Force normalization = 0, limma precision = 0, significance level = 0

###### gene ID match

rm(list = ls())

T2D\_PC<- read.delim("~/T2D\_PC\_H\_GSE15932.top.table.tsv")

Z=unique(T2D\_PC$Gene.symbol)

getwd()

setwd("D:/PhD/T2D\_Cancer\_Project/Geo/filter")

#########

filter\_pValue\_T2D=T2D\_PC%>%

filter(adj.P.Val<=0.05)

A=unique(filter\_pValue\_T2D$Gene.symbol)

write.csv(filter\_pValue\_T2D, "GEO2R\_T2D&PC\_0.05\_GSE1932\_filter1.csv", quote = FALSE, row.names = FALSE)

write.csv(A, "GEO2R\_T2D&PC\_0.05\_GSE1932\_filter1\_unique.csv", quote = FALSE, row.names = FALSE)

filter\_T2D\_FC\_A=filter\_pValue\_T2D%>%

filter(logFC>=0.5)

filter\_T2D\_FC\_B=filter\_pValue\_T2D%>%

filter(logFC<=-0.5)

filter\_T2D=rbind(filter\_T2D\_FC\_A, filter\_T2D\_FC\_B)

AA=unique(filter\_T2D$Gene.symbol)

write.csv(filter\_T2D, "GEO2R\_T2D&PC\_0.05\_GSE1932\_filter2.csv", quote = FALSE, row.names = FALSE)

write.csv(AA, "GEO2R\_T2D&PC\_0.05\_GSE1932\_filter2\_unique.csv", quote = FALSE, row.names = FALSE)

#########Scatterplot

plot(T2D\_PC\_Dist$logFC.x, T2D\_PC\_Dist$logFC.y,

xlab = expression("log"[2]\*" FC (T2DM)"), ylab =expression("log"[2]\*" FC (PC)"), xlim = c(-6,6), ylim = c(-6,6),

main = expression("Common genes"))+

abline(h=0, v=0)+

abline(0,1)

#########Barplots

C=ggplot(T2DvsPC\_DOWN)+

geom\_bar(aes(fill=Disease, x= reorder(Gene, +FC), y=FC),

position = "dodge", stat = "identity")

C+ylab(expression("log"[2]\* " FC")) + xlab("Genes")+theme\_bw()+

theme(element\_blank(), axis.line=element\_line(colour = "black"), panel.border = element\_blank())+

geom\_hline(yintercept = 0)+

geom\_vline(xintercept = 0)

#########Correlation analysis

library(dplyr)

T2D\_PC\_FC=T2D\_PC$logFC[match(T2D$ID, T2D\_PC$ID)]

A=cbind.data.frame(T2D, T2D\_PC\_FC)

LC\_FC=LC$logFC[match(A$ID, LC$ID)]

B=cbind.data.frame(A, LC\_FC)

BC\_FC=BC$logFC[match(B$ID, BC$ID)]

C=cbind.data.frame(B, BC\_FC)

C$adj.P.Val=NULL

C$P.Value=NULL

C$t=NULL

C$B=NULL

C$Gene.title=NULL

cor.test(C$logFC, C$LC\_FC, method = c("pearson"))