1. ssGSEA评估免疫细胞比例

library(genefilter)

library(GSVA)

library(Biobase)

library(stringr)

rm(list=ls())

library(tidyverse)

options(stringsAsFactors = F)

geneSet <- read.csv("reference.txt",header = F,sep = "\t",)

class(geneSet)

geneSet <- geneSet %>%

column\_to\_rownames("V1")%>%t()

a <- geneSet

a <- a[1:nrow(a),]

set <- colnames(a)

l <- list()

#i <- "Activated CD8 T cell"

for (i in set) {

x <- as.character(a[,i])

x <- x[nchar(x)!=0]

x <- as.character(x)

l[[i]] <-x

}

save(l,file = "d:/gene\_set.Rdata")

logTPM=read.table("input.txt",header=T)

save(logTPM,file = "logTPM.Rdata")

load(file = "d:/logTPM.Rdata")

load(file = "d:/gene\_set.Rdata")

dat <- as.matrix(logTPM[,-1])

ssgsea<- gsva(dat, l,method='ssgsea',kcdf='Gaussian',abs.ranking=TRUE)

ssgsea.1 <- ssgsea

for (i in colnames(ssgsea)) {

#i <- colnames(ssgsea)[1]

ssgsea.1[,i] <- (ssgsea[,i] -min(ssgsea[,i]))/(max(ssgsea[,i] )-min(ssgsea[,i] ))

}

apply(ssgsea.1[,1:6], 2, range)

library(pheatmap)

pheatmap(ssgsea.1,show\_colnames = T,cluster\_rows = T, cluster\_cols = T,fontsize=8)

write.table(ssgsea.1,file="output.txt",sep="\t")

2. Estimate评估

library(estimate)

filterCommonGenes(input.f="input.txt", output.f="d:/output.gct", id="GeneSymbol")

estimateScore(input.ds = "d:/output.gct",output.ds="d:/output\_estimate\_score.gct", platform="affymetrix")

plotPurity(scores="d:/output\_estimate\_score.gct", platform="affymetrix")

scores=read.table("d:/output\_estimate\_score.gct",skip = 2,header = T)

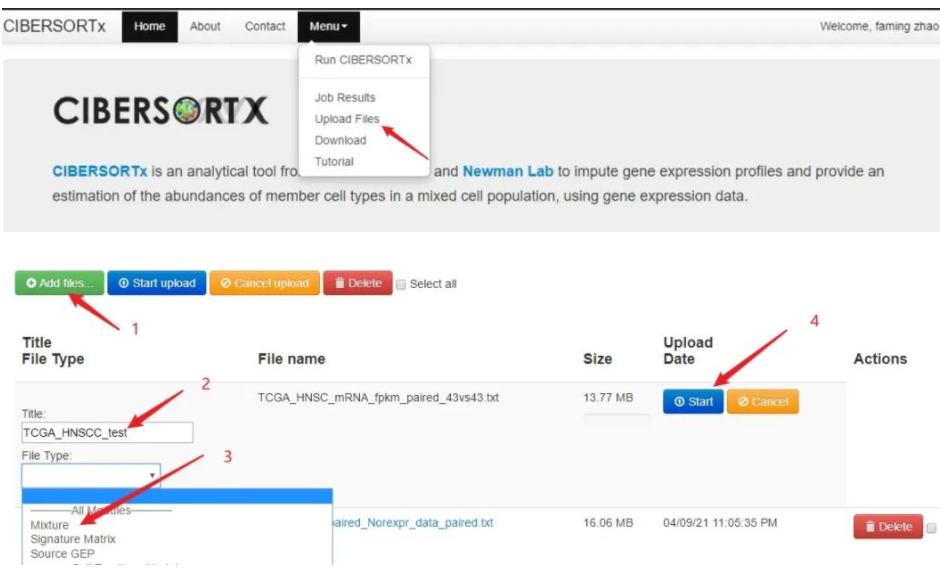
rownames(scores)=scores[,1]

scores=t(scores[,3:ncol(scores)])

scores

3.CIBERSORT评估

CIBERSORT（https://cibersort.stanford.edu/index.php）在线网站



4. Limma算法显著差异表达RNAs的筛选

library(Biobase)

library(GEOquery)

library(limma)

data=read.table(“input.txt”,header=T,sep=”\t”)

gset=as.matrix(data[,-1])

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) ||

(qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)

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

exprs(gset) <- log2(ex) }

# set up the data and proceed with analysis

fl <- as.factor(sml)

gset$description <- fl

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

colnames(design) <- levels(fl)

fit <- lmFit(gset, design)

cont.matrix <- makeContrasts(G1-G0, levels=design)

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

fit2 <- eBayes(fit2, 0.01)

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

# load NCBI platform annotation

gpl <- annotation(gset)

platf <- getGEO(gpl, AnnotGPL=TRUE)

ncbifd <- data.frame(attr(dataTable(platf), “table”))

# replace original platform annotation

tT <- tT[setdiff(colnames(tT), setdiff(fvarLabels(gset), “ID”))]

tT <- merge(tT, ncbifd, by=“ID”)

tT <- tT[order(tT$P.Value), ] # restore correct order

tT <- subset(tT, select=c(“ID”,”adj.P.Val”,”P.Value”,”t”,”B”,”logFC”,”Gene.symbol”,”Gene.title”))

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

5. 功能分析：DAVID在线软件，如下图：



6. 独立预后lncRNAs的筛选和模型构建

单因素cox回归分析：

Library(survival)

A=read.table("input.txt",header=T,sep="\t")

sur<-Surv(time,death)

fit=coxph(sur~A[,N],data=A) #N=1,2,3,….

多因素cox回归分析：

A=read.table("input.txt",header=T)

mydata=A[,N:M]

fit=coxph(Surv(time,death)~.,data=mydata)

KM曲线：

library(survival)

library(survminer)

A=read.table("input.txt",header=T,sep="\t")

fit<- survfit(Surv(time,OS) ~status,data = A)

ggsurvplot(fit,pval=TRUE,conf.int=TRUE,xlab="Overall survival time(months)",ylab="Survival ratio",risk.table=TRUE,ggtheme =theme\_light(), ncensor.plot = T,palette = c("darkgreen","navy"))