library(igraph)

library(psych)

library(reshape2)

library(RColorBrewer)

GeneExpfile="cuproGeneExp.txt"

Genefile="gene.txt"

Coxfile="uniCox.txt"

setwd("C:\\Users\\DELL\\Desktop\\20.network")

gene.group <- read.table(Genefile,header=T,sep="\t")

gene.exp <- read.table(GeneExpfile,header=T,sep="\t",row.names=1)

gene.cox <- read.table(Coxfile,header=T,sep="\t")

colnames(gene.group) <- c('id','group')

genelist <- intersect(gene.group$id, gene.cox$id)

genelist <- intersect(genelist, rownames(gene.exp))

gene.group <- gene.group[match(genelist,gene.group$id),]

gene.group <- gene.group[order(gene.group$group),]

gene.exp <- gene.exp[match(gene.group$id,rownames(gene.exp)),]

gene.cox <- gene.cox[match(gene.group$id,gene.cox$id),]

gene.cor <- corr.test(t(gene.exp))

gene.cor.cor <- gene.cor$r

gene.cor.pvalue <- gene.cor$p

gene.cor.cor[upper.tri(gene.cor.cor)] = NA

gene.cor.pvalue[upper.tri(gene.cor.pvalue)] = NA

gene.cor.cor.melt <- melt(gene.cor.cor) #gene1 \t gene2 \t cor

gene.cor.pvalue.melt <- melt(gene.cor.pvalue)

gene.melt <- data.frame(from = gene.cor.cor.melt$Var2,to=gene.cor.cor.melt$Var1,cor=gene.cor.cor.melt$value,pvalue=gene.cor.pvalue.melt$value)

gene.melt <- gene.melt[gene.melt$from!=gene.melt$to&!is.na(gene.melt$pvalue),,drop=F]

gene.edge <- gene.melt[gene.melt$pvalue<0.0001,,drop=F]

gene.edge$color <- ifelse(gene.edge$cor>0,'antiquewhite','skyblue')

gene.edge$weight <- abs(gene.edge$cor)\*6

gene.node <- gene.group

group.color <- colorRampPalette(brewer.pal(9, "Set1"))(length(unique(gene.node$group)))

gene.node$color <- group.color[as.numeric(as.factor(gene.node$group))]

gene.node$shape <- "circle"

gene.node$frame <- ifelse(gene.cox$HR>1,'mediumpurple','green4')

gene.node$pvalue <- gene.cox$pvalue

pvalue.breaks <- c(0,0.0001,0.001,0.01,0.05,1)

pvalue.size <- c(16,14,12,10,8)

cutpvalue <- cut(gene.node$pvalue,breaks=pvalue.breaks)

gene.node$size <- pvalue.size[as.numeric(cutpvalue)]

nodefile <- "network.node.txt"

edgefile <- "network.edge.txt"

write.table(gene.node, nodefile, sep="\t", col.names=T, row.names=F, quote=F)

write.table(gene.edge, edgefile, sep="\t", col.names=T, row.names=F, quote=F)

node = read.table(nodefile, header=T, sep="\t", comment.char="")

edge = read.table(edgefile, header=T, sep="\t", comment.char="")

g = graph.data.frame(edge,directed = FALSE)

node = node[match(names(components(g)$membership),node$id),]

if(!is.na(match('color',colnames(node)))) V(g)$color = node$color

if(!is.na(match('size',colnames(node)))) V(g)$size = node$size

if(!is.na(match('shape',colnames(node)))) V(g)$shape = node$shape

if(!is.na(match('frame',colnames(node)))) V(g)$frame = node$frame

pdf(file="network.pdf", width=10, height=8)

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

layout(matrix(c(1,1,4,2,3,4),nc=2),height=c(4,4,2),width=c(8,3))

coord = layout\_in\_circle(g)

degree.x = acos(coord[,1])

degree.y = asin(coord[,2])

degree.alpha = c()

for(i in 1:length(degree.x)){

if(degree.y[i]<0) degree.alpha=c(degree.alpha,2\*pi-degree.x[i]) else degree.alpha=c(degree.alpha,degree.x[i])

}

degree.cut.group = (0:8)/4\*pi

degree.cut.group[1] = -0.0001

degree.cut = cut(degree.alpha,degree.cut.group)

degree.degree = c(-pi/4,-pi/4,-pi/2,-pi/2,pi/2,pi/2,pi/2,pi/4)

degree = degree.degree[as.numeric(degree.cut)]

values <- lapply(node$id,function(x)c(1,1))

V(g)$pie.color = lapply(1:nrow(node),function(x)c(node$color[x],node$frame[x]))

V(g)$frame = NA

plot(g,layout=layout\_in\_circle,vertex.shape="pie",vertex.pie=values,

vertex.label.cex=V(g)$lable.cex,edge.width = E(g)$weight,edge.arrow.size=0,

vertex.label.color=V(g)$color,vertex.frame.color=V(g)$frame,edge.color=E(g)$color,

vertex.label.cex=2.5,vertex.label.font=2.5,vertex.size=V(g)$size,edge.curved=0.4,

vertex.color=V(g)$color,vertex.label.dist=1.35,vertex.label.degree=degree)

# label.degree : zero means to the right; and pi means to the left; up is -pi/2 and down is pi/2; The default value is -pi/4

# label.dist If it is 0 then the label is centered on the vertex; If it is 1 then the label is displayed beside the vertex.

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

plot(1,type="n",xlab="",ylab="",axes=F)

groupinfo = unique(data.frame(group=node$group,color=node$color))

legend("left",legend=groupinfo$group,col=groupinfo$color,pch=16,bty="n",cex=3)

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

plot(1,type="n",xlab="",ylab="",axes=F)

legend("left",legend=c('Risk factors','Favorable factors'),col=c('mediumpurple','green4'),pch=16,bty="n",cex=2.5)

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

plot(1,type="n",xlab="",axes=F,ylab="")

legend("top",legend=c('Postive correlation with P<0.0001','Negative correlation with P<0.0001'),lty=1,lwd=4,col=c('antiquewhite','skyblue'),bty="n",cex=2.2)

legend('bottom',legend=c(0.0001,0.001,0.01,0.05,1),pch=16,pt.cex=c(1.6,1.4,1.2,1,0.8)\*6,bty="n",ncol=5,cex=2.2,col="black",title="Cox test, pvalue")

dev.off()

library(ggpubr)

inputFile="input.txt"

outFile="Deviation.pdf"

setwd("C:\\Users\\DELL\\Desktop\\Figure2A")

rt=read.table(inputFile,sep="\t",header=T,check.names=F)

x=colnames(rt)[1]

y=colnames(rt)[2]

colnames(rt)=c("Name","Value")

rt$Regulate=factor(ifelse(rt$Value<0, "Down", "Up"), levels = c("Up", "Down"))

pdf(file=outFile,width=6,height=8)

ggbarplot(rt, x="Name", y="Value", fill = "Regulate", color = "black",

palette = "jco",

sort.val = "desc", sort.by.groups = FALSE,

x.text.angle=90,

xlab = x, ylab = y,

legend.title="Regulate", rotate=TRUE, ggtheme = theme\_minimal()) #rotatex璁剧疆x/y瀵硅皟

dev.off()

library(limma)

library(ComplexHeatmap)

expFile="geneExp.txt"

cliFile="clinical.txt"

setwd("C:\\Users\\DELL\\Desktop\\14.cliHeatmGSE53625")

rt=read.table(expFile, header=T, sep="\t", check.names=F, row.names=1)

gene=colnames(rt)[1]

#tumorData=rt[rt$Type=="Tumor",1,drop=F]

#tumorData=as.matrix(tumorData)

#rownames(tumorData)=gsub("(.\*?)\\-(.\*?)\\-(.\*?)\\-(.\*?)\\-.\*", "\\1\\-\\2\\-\\3", rownames(tumorData))

#data=avereps(tumorData)

data=rt

data=as.data.frame(data)

Type2=ifelse(data$`Risk score`>median(data$`Risk score`), "High", "Low")

Type2=factor(Type2, levels=c("Low","High"))

data=cbind(as.data.frame(data), Type2)

data=data[order(data[,gene]),]

cli=read.table(cliFile, header=T, sep="\t", check.names=F, row.names=1)

cli[,"Age"]=ifelse(cli[,"Age"]=="unknow", "unknow", ifelse(cli[,"Age"]>65,">65","<=65"))

samSample=intersect(row.names(data), row.names(cli))

data=data[samSample,"Type2",drop=F]

cli=cli[samSample,,drop=F]

rt=cbind(data, cli)

sigVec=c(gene)

for(clinical in colnames(rt[,2:ncol(rt)])){

data=rt[c("Type2", clinical)]

colnames(data)=c("Type2", "clinical")

data=data[(data[,"clinical"]!="unknow"),]

tableStat=table(data)

stat=chisq.test(tableStat)

pvalue=stat$p.value

Sig=ifelse(pvalue<0.001,"\*\*\*",ifelse(pvalue<0.01,"\*\*",ifelse(pvalue<0.05,"\*","")))

sigVec=c(sigVec, paste0(clinical, Sig))

}

colnames(rt)=sigVec

#rt=rt[apply(rt,1,function(x)any(is.na(match('unknow',x)))),,drop=F]

bioCol=c("#6ad157", "#373bbf", "#a1ce4c", "#ef3bb6", "#d66551", "#1a918f", "#7149af", "#ff66fc", "#2927c4", "#57e559" ,"#8e3af4" ,"#f9a270" ,"#22547f", "#db5e92",

"#4aef7b", "#e86502", "#99db27", "#e07233", "#8249aa","#cebb10", "#03827f", "#931635", "#ff523f",

"#edd05e", "#6f25e8", "#0dbc21", "#167275", "#280f7a", "#6373ed", "#5b910f" ,"#7b34c1" ,"#0cf29a" ,"#d80fc1",

"#dd27ce", "#07a301", "#ddd53e", "#391c82", "#2baeb5","#925bea", "#09f9f5", "#63ff4f")

colorList=list()

colorList[[gene]]=c("Low"="#0080FF", "High"="#CE0000")

j=0

for(cli in colnames(rt[,2:ncol(rt)])){

cliLength=length(levels(factor(rt[,cli])))

cliCol=bioCol[(j+1):(j+cliLength)]

j=j+cliLength

names(cliCol)=levels(factor(rt[,cli]))

cliCol["unknow"]="grey75"

colorList[[cli]]=cliCol

}

ha=HeatmapAnnotation(df=rt, col=colorList)

zero\_row\_mat=matrix(nrow=0, ncol=nrow(rt))

Hm=Heatmap(zero\_row\_mat, top\_annotation=ha)

pdf(file="heatmap.pdf", width=7, height=5)

draw(Hm, merge\_legend = TRUE, heatmap\_legend\_side = "bottom", annotation\_legend\_side = "bottom")

dev.off()

library(survival)

library(survminer)

setwd("C:\\Users\\DELL\\Desktop\\20.survivalrisk")

bioSurvival=function(inputFile=null, outFile=null){

rt=read.table(inputFile, header=T, sep="\t", check.names=F)

pvalue

diff=survdiff(Surv(futime, fustat) ~ Risk,data = rt)

pValue=1-pchisq(diff$chisq,df=1)

if(pValue<0.001){

pValue="p<0.001"

}else{

pValue=paste0("p=",sprintf("%.03f",pValue))

}

fit <- survfit(Surv(futime, fustat) ~ Risk, data = rt)

surPlot=ggsurvplot(fit,

data=rt,

conf.int=T,

pval=pValue,

pval.size=6,

legend.title="Risk",

legend.labs=c("High risk", "Low risk"),

xlab="Time(years)",

ylab="Overall survival",

break.time.by = 1,

palette=c("#CE0000", "#0080FF"),

risk.table=TRUE,

risk.table.title="",

risk.table.height=.25)

pdf(file=outFile, width=6.5, height=5.5, onefile=FALSE)

print(surPlot)

dev.off()

}

bioSurvival(inputFile="risk.TCGA.txt", outFile="survival.TCGA.pdf")

bioSurvival(inputFile="risk.GEO.txt", outFile="survival.GEO.pdf")

library(pheatmap)

setwd("C:\\Users\\DELL\\Desktop\\28.riskPlotdGSE53625")

bioRiskPlot=function(inputFile=null, riskScoreFile=null, survStatFile=null, heatmapFile=null){

rt=read.table(inputFile, header=T, sep="\t", check.names=F, row.names=1)

rt=rt[order(rt$riskScore),]

riskClass=rt[,"risk"]

lowLength=length(riskClass[riskClass=="low"])

highLength=length(riskClass[riskClass=="high"])

lowMax=max(rt$riskScore[riskClass=="low"])

line=rt[,"riskScore"]

line[line>10]=10

pdf(file=riskScoreFile, width=7, height=4)

plot(line, type="p", pch=20,

xlab="Patients (increasing risk socre)", ylab="Risk score",

col=c(rep("#0080FF",lowLength),rep("#CE0000",highLength)) )

abline(h=lowMax,v=lowLength,lty=2)

legend("topleft", c("High risk", "Low risk"),bty="n",pch=19,col=c("#CE0000","#0080FF"),cex=1.2)

dev.off()

color=as.vector(rt$fustat)

color[color==1]="#CE0000"

color[color==0]="#0080FF"

pdf(file=survStatFile, width=7, height=4)

plot(rt$futime, pch=19,

xlab="Patients (increasing risk socre)", ylab="Survival time (years)",

col=color)

legend("topleft", c("Dead", "Alive"),bty="n",pch=19,col=c("#CE0000","#0080FF"),cex=1.2)

abline(v=lowLength,lty=2)

dev.off()

rt1=rt[c(3:(ncol(rt)-2))]

rt1=t(rt1)

annotation=data.frame(type=rt[,ncol(rt)])

rownames(annotation)=rownames(rt)

pdf(file=heatmapFile, width=7, height=4)

pheatmap(rt1,

annotation=annotation,

cluster\_cols = FALSE,

cluster\_rows = FALSE,

show\_colnames = F,

scale="row",

color = colorRampPalette(c(rep("#0080FF",3), "white", rep("#CE0000",3)))(50),

fontsize\_col=3,

fontsize=7,

fontsize\_row=8)

dev.off()

}

bioRiskPlot(inputFile="risk.Geo.txt",

riskScoreFile="riskScore.pdf",

survStatFile="survStat.pdf",

heatmapFile="heatmap.pdf")

library(survival)

library(survminer)

library(timeROC)

setwd("C:\\Users\\DELL\\Desktop\\39.ROC")

bioROC=function(inputFile=null, rocFile=null){

rt=read.table(inputFile, header=T, sep="\t", check.names=F)

ROC\_rt=timeROC(T=rt$futime,delta=rt$fustat,

marker=rt$riskScore,cause=1,

weighting='aalen',

times=c(1,2,3),ROC=TRUE)

pdf(file=rocFile, width=5, height=5)

plot(ROC\_rt,time=1,col='#0080FF',title=FALSE,lwd=2)

plot(ROC\_rt,time=2,col='#DCB5FF',add=TRUE,title=FALSE,lwd=2)

plot(ROC\_rt,time=3,col='#CE0000',add=TRUE,title=FALSE,lwd=2)

legend('bottomright',

c(paste0('AUC at 1 years: ',sprintf("%.03f",ROC\_rt$AUC[1])),

paste0('AUC at 2 years: ',sprintf("%.03f",ROC\_rt$AUC[2])),

paste0('AUC at 3 years: ',sprintf("%.03f",ROC\_rt$AUC[3]))),

col=c("#0080FF",'#DCB5FF','#CE0000'),lwd=2,bty = 'n')

dev.off()

}

bioROC(inputFile="risk.GEO.txt", rocFile="ROC.GEO.pdf")

bioROC(inputFile="risk.TCGA.txt", rocFile="ROC.TCGA.pdf")

library(survival)

setwd("C:\\Users\\DELL\\Desktop\\22.indepGSE53625")

bioForest=function(coxFile=null, forestFile=null, forestCol=null){

rt <- read.table(coxFile, header=T, sep="\t", check.names=F, row.names=1)

gene <- rownames(rt)

hr <- sprintf("%.3f",rt$"HR")

hrLow <- sprintf("%.3f",rt$"HR.95L")

hrHigh <- sprintf("%.3f",rt$"HR.95H")

Hazard.ratio <- paste0(hr,"(",hrLow,"-",hrHigh,")")

pVal <- ifelse(rt$pvalue<0.001, "<0.001", sprintf("%.3f", rt$pvalue))

pdf(file=forestFile, width=6.5, height=4.5)

n <- nrow(rt)

nRow <- n+1

ylim <- c(1,nRow)

layout(matrix(c(1,2),nc=2),width=c(3,2.5))

xlim = c(0,3)

par(mar=c(4,2.5,2,1))

plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,xlab="",ylab="")

text.cex=0.8

text(0,n:1,gene,adj=0,cex=text.cex)

text(1.5-0.5\*0.2,n:1,pVal,adj=1,cex=text.cex);text(1.5-0.5\*0.2,n+1,'pvalue',cex=text.cex,font=2,adj=1)

text(3.1,n:1,Hazard.ratio,adj=1,cex=text.cex);text(3.1,n+1,'Hazard ratio',cex=text.cex,font=2,adj=1)

par(mar=c(4,1,2,1),mgp=c(2,0.5,0))

xlim = c(0,max(as.numeric(hrLow),as.numeric(hrHigh)))

plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,ylab="",xaxs="i",xlab="Hazard ratio")

arrows(as.numeric(hrLow),n:1,as.numeric(hrHigh),n:1,angle=90,code=3,length=0.05,col="darkblue",lwd=3)

abline(v=1, col="black", lty=2, lwd=2)

boxcolor = ifelse(as.numeric(hr) > 1, forestCol, forestCol)

points(as.numeric(hr), n:1, pch = 15, col = boxcolor, cex=2)

axis(1)

dev.off()

}

indep=function(riskFile=null,cliFile=null,uniOutFile=null,multiOutFile=null,uniForest=null,multiForest=null){

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

cli=read.table(cliFile, header=T, sep="\t", check.names=F, row.names=1)

sameSample=intersect(row.names(cli),row.names(risk))

risk=risk[sameSample,]

cli=cli[sameSample,]

rt=cbind(futime=risk[,1], fustat=risk[,2], cli, riskScore=risk[,(ncol(risk)-1)])

uniTab=data.frame()

for(i in colnames(rt[,3:ncol(rt)])){

cox <- coxph(Surv(futime, fustat) ~ rt[,i], data = rt)

coxSummary = summary(cox)

uniTab=rbind(uniTab,

cbind(id=i,

HR=coxSummary$conf.int[,"exp(coef)"],

HR.95L=coxSummary$conf.int[,"lower .95"],

HR.95H=coxSummary$conf.int[,"upper .95"],

pvalue=coxSummary$coefficients[,"Pr(>|z|)"])

)

}

write.table(uniTab,file=uniOutFile,sep="\t",row.names=F,quote=F)

bioForest(coxFile=uniOutFile, forestFile=uniForest, forestCol="#019858")

uniTab=uniTab[as.numeric(uniTab[,"pvalue"])<1,]

rt1=rt[,c("futime", "fustat", as.vector(uniTab[,"id"]))]

multiCox=coxph(Surv(futime, fustat) ~ ., data = rt1)

multiCoxSum=summary(multiCox)

multiTab=data.frame()

multiTab=cbind(

HR=multiCoxSum$conf.int[,"exp(coef)"],

HR.95L=multiCoxSum$conf.int[,"lower .95"],

HR.95H=multiCoxSum$conf.int[,"upper .95"],

pvalue=multiCoxSum$coefficients[,"Pr(>|z|)"])

multiTab=cbind(id=row.names(multiTab),multiTab)

write.table(multiTab,file=multiOutFile,sep="\t",row.names=F,quote=F)

bioForest(coxFile=multiOutFile, forestFile=multiForest, forestCol="#CE0000")

}

indep(riskFile="risk.GEO.txt",

cliFile="clinical.txt",

uniOutFile="uniCox.txt",

multiOutFile="multiCox.txt",

uniForest="uniForest.pdf",

multiForest="multiForest.pdf")

riskFile="risk.all.txt"

cliFile="clinical.txt"

setwd("C:\\Users\\DELL\\Desktop\\41.Nomo")

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

cli=read.table(cliFile, header=T, sep="\t", check.names=F, row.names=1)

cli=cli[apply(cli,1,function(x)any(is.na(match('unknow',x)))),,drop=F]

cli$Age=as.numeric(cli$Age)

samSample=intersect(row.names(risk), row.names(cli))

risk1=risk[samSample,,drop=F]

cli=cli[samSample,,drop=F]

rt=cbind(risk1[,c("futime", "fustat", "risk")], cli)

res.cox=coxph(Surv(futime, fustat) ~ . , data = rt)

nom1=regplot(res.cox,

plots = c("density", "boxes"),

clickable=F,

title="",

points=TRUE,

droplines=TRUE,

observation=rt[2,],

rank="sd",

failtime = c(1,3,5),

prfail = F)

nomoRisk=predict(res.cox, data=rt, type="risk")

rt=cbind(risk1, Nomogram=nomoRisk)

outTab=rbind(ID=colnames(rt), rt)

write.table(outTab, file="nomoRisk.txt", sep="\t", col.names=F, quote=F)

pdf(file="calibration.pdf", width=5, height=5)

f <- cph(Surv(futime, fustat) ~ Nomogram, x=T, y=T, surv=T, data=rt, time.inc=1)

cal <- calibrate(f, cmethod="KM", method="boot", u=1, m=(nrow(rt)/3), B=1000)

plot(cal, xlim=c(0,1), ylim=c(0,1),

xlab="Nomogram-predicted OS (%)", ylab="Observed OS (%)", lwd=1.5, col="#019858", sub=F)

f <- cph(Surv(futime, fustat) ~ Nomogram, x=T, y=T, surv=T, data=rt, time.inc=3)

cal <- calibrate(f, cmethod="KM", method="boot", u=3, m=(nrow(rt)/3), B=1000)

plot(cal, xlim=c(0,1), ylim=c(0,1), xlab="", ylab="", lwd=1.5, col="#0080FF", sub=F, add=T)

f <- cph(Surv(futime, fustat) ~ Nomogram, x=T, y=T, surv=T, data=rt, time.inc=5)

cal <- calibrate(f, cmethod="KM", method="boot", u=5, m=(nrow(rt)/3), B=1000)

plot(cal, xlim=c(0,1), ylim=c(0,1), xlab="", ylab="", lwd=1.5, col="#CE0000", sub=F, add=T)

legend('bottomright', c('1-year', '3-year', '5-year'),

col=c("#019858","#0080FF","#CE0000"), lwd=1.5, bty = 'n')

dev.off()

riskFile="nomoRisk.txt" #列线图的打分文件

cliFile="clinical.txt" #临床数据文件

setwd("C:\\Users\\DELL\\Desktop\\1") #设置工作目录

#读取风险输入文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

#读取临床数据文件

cli=read.table(cliFile, header=T, sep="\t", check.names=F, row.names=1)

cli=cli[apply(cli,1,function(x)any(is.na(match('unknow',x)))),,drop=F]

cli$Age=as.numeric(cli$Age)

#合并数据

samSample=intersect(row.names(risk), row.names(cli))

risk1=risk[samSample,,drop=F]

cli1=cli[samSample,,drop=F]

data=cbind(risk1, cli1)

rt=cbind(risk1[,c("futime","fustat","riskScore","Nomogram")], cli1)

Nomogram<-coxph(Surv(futime,fustat)~Nomogram,rt)

Risk<-coxph(Surv(futime,fustat)~riskScore,rt)

Age<-coxph(Surv(futime,fustat)~Age, rt)

Gender<-coxph(Surv(futime,fustat)~Gender, rt)

Grade<-coxph(Surv(futime,fustat)~Grade, rt)

TNM\_stage<-coxph(Surv(futime,fustat)~TNM\_stage, rt)

#????1????????????

d\_train1=dca(Nomogram, Risk, Age, Gender, Grade, TNM\_stage, times=1)

pdf(file="DCA1.pdf", width=6, height=5)

ggplot(d\_train1, linetype=1)

dev.off()

#????3????????????

d\_train3=dca(Nomogram, Risk, Age, Gender, Grade, TNM\_stage, times=3)

pdf(file="DCA3.pdf", width=6, height=5)

ggplot(d\_train3, linetype=1)

dev.off()

#????5????????????

d\_train5=dca(Nomogram, Risk, Age, Gender, Grade, TNM\_stage, times=5)

pdf(file="DCA5.pdf", width=6, height=5)

ggplot(d\_train5, linetype=1)

dev.off()

immFile="CIBERSORT-Results.txt" #免疫细胞浸润的结果文件

riskFile="risk.GEO.txt" #风险文件

setwd("C:\\Users\\DELL\\Desktop\\43.immuneCorhecell") #设置工作目录

#读取免疫细胞浸润的结果文件，并对数据进行整理

immune=read.table(immFile, header=T, sep="\t", check.names=F, row.names=1)

immune=immune[immune[,"P-value"]<0.05,]

data=as.matrix(immune[,1:(ncol(immune)-3)])

rownames(data)=gsub("(.\*?)\\\_(.\*?)", "\\2", rownames(data))

#读取风险文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

sameSample=intersect(row.names(data), row.names(risk))

data=data[sameSample,,drop=F]

risk=risk[sameSample,,drop=F]

#对所有免疫细胞进行循环，得到风险打分与免疫细胞的相关性

for(i in colnames(data)[1:ncol(data)]){

x=as.numeric(risk[,"riskScore"])

x[x>quantile(x,0.99)]=quantile(x,0.99)

y=as.numeric(data[,i])

if(sd(y)<0.01){next}

cor=cor.test(x, y, method="spearman")

#对pvalue小于0.05的免疫细胞进行相关性散点图的绘制

if(cor$p.value<0.05){

outFile=paste0("cor.", i, ".pdf")

df1=as.data.frame(cbind(x,y))

p1=ggplot(df1, aes(x, y)) +

xlab("Risk score") + ylab(i)+

geom\_point() + geom\_smooth(method="lm",formula = y ~ x) + theme\_bw()+

stat\_cor(method = 'spearman', aes(x =x, y =y))

p2=ggMarginal(p1, type="density", xparams=list(fill = "#CE0000"), yparams=list(fill = "#0080FF"))

#相关性图形

pdf(file=outFile, width=5.2, height=5)

print(p2)

dev.off()

}

}

#基因与免疫细胞相关性分析

outTab=data.frame()

risk=risk[,3:(ncol(risk)-1),drop=F]

for(immune in colnames(data)){

if(sd(data[,immune])<0.01){next}

for(gene in colnames(risk)){

x=as.numeric(data[,immune])

y=as.numeric(risk[,gene])

y[y>quantile(y,0.99)]=quantile(y,0.99)

corT=cor.test(x,y,method="spearman")

cor=corT$estimate

pvalue=corT$p.value

text=ifelse(pvalue<0.001,"\*\*\*",ifelse(pvalue<0.01,"\*\*",ifelse(pvalue<0.05,"\*","")))

outTab=rbind(outTab,cbind(Gene=gene, Immune=immune, cor, text, pvalue))

}

}

#绘制相关性热图

outTab$cor=as.numeric(outTab$cor)

outTab[,"Gene"]=factor(outTab[,"Gene"], levels=colnames(risk))

pdf(file="geneImmuneCor.pdf", width=7, height=5.5)

ggplot(outTab, aes(Gene, Immune)) +

geom\_tile(aes(fill = cor), colour = "grey", size = 1)+

scale\_fill\_gradient2(low = "#CE0000", mid = "white", high = "#0080FF") +

geom\_text(aes(label=text),col ="black",size = 3) +

theme\_minimal() +

theme(axis.title.x=element\_blank(), axis.ticks.x=element\_blank(), axis.title.y=element\_blank(),

axis.text.x = element\_text(angle = 45, hjust = 1, size = 9, face = "bold"), #x轴字体

axis.text.y = element\_text(size = 9, face = "bold")) + #y轴字体

labs(fill =paste0("\*\*\* p<0.001","\n", "\*\* p<0.01","\n", " \* p<0.05","\n", "\n","Correlation")) + #设置图例

scale\_x\_discrete(position = "bottom") #X轴基因名字显示在图形的下方

dev.off()

riskFile="risk.GEO.txt" #风险文件

TMEfile="TMEscores.txt" #肿瘤微环境打分文件

setwd("C:\\Users\\DELL\\Desktop\\45.estimateVioplot") #设置工作目录

#读取风险文件

Risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

Risk$risk=factor(Risk$risk, levels=c("low","high"))

#读取肿瘤微环境打分文件

score=read.table(TMEfile, header=T, sep="\t", check.names=F, row.names=1)

score=score[,1:3]

rownames(score)=gsub("(.\*?)\\\_(.\*?)", "\\2", rownames(score))

score=score[row.names(Risk),,drop=F]

#数据合并

rt=cbind(Risk[,"risk",drop=F], score)

#将合并后的数据转换为ggplot2的输入文件

data=melt(rt, id.vars=c("risk"))

colnames(data)=c("Risk", "scoreType", "Score")

#绘制小提琴图

p=ggviolin(data, x="scoreType", y="Score", fill = "Risk",

xlab="",

ylab="TME score",

legend.title="Risk",

add = "boxplot", add.params = list(color="white"),

palette = c("#0080FF", "#CE0000"), width=1)

p=p+rotate\_x\_text(45)

p1=p+stat\_compare\_means(aes(group=Risk),

method="wilcox.test",

symnum.args=list(cutpoints = c(0, 0.001, 0.01, 0.05, 1), symbols = c("\*\*\*", "\*\*", "\*", " ")),

label = "p.signif")

#输出图形

pdf(file="vioplot.pdf", width=6, height=5)

print(p1)

dev.off()

#引用包

library(limma)

library(estimate)

inputFile="merge.txt" #表达数据文件

setwd("C:\\Users\\lexb\\Desktop\\cuproOmics\\44.estimate") #设置工作目录

#读取文件,并对输入文件进行整理

rt=read.table(inputFile, header=T, sep="\t", check.names=F)

rt=as.matrix(rt)

rownames(rt)=rt[,1]

exp=rt[,2:ncol(rt)]

dimnames=list(rownames(exp),colnames(exp))

data=matrix(as.numeric(as.matrix(exp)),nrow=nrow(exp),dimnames=dimnames)

data=avereps(data)

#输出整理后的矩阵文件

out=rbind(ID=colnames(data), data)

write.table(out,file="uniq.symbol.txt",sep="\t",quote=F,col.names=F)

#运行estimate包

filterCommonGenes(input.f="uniq.symbol.txt",

output.f="commonGenes.gct",

id="GeneSymbol")

estimateScore(input.ds = "commonGenes.gct",

output.ds="estimateScore.gct")

#对肿瘤微环境的打分进行整理, 输出每个样品的打分

scores=read.table("estimateScore.gct", skip=2, header=T)

rownames(scores)=scores[,1]

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

rownames(scores)=gsub("\\.", "\\-", rownames(scores))

out=rbind(ID=colnames(scores), scores)

write.table(out, file="TMEscores.txt", sep="\t", quote=F, col.names=F)

library(ggplot2)

library(reshape2)

inputFile="input.txt"

outFile="boxplot.pdf"

setwd("C:\\Users\\DELL\\Desktop\\10.boxplotFacet")

#璇诲彇杈撳叆鏂囦欢

rt=read.table(inputFile, header=T,sep="\t",check.names=F,row.names=1)

x=colnames(rt)[1]

colnames(rt)[1]="Type"

#宸紓鍒嗘瀽

geneSig=c("")

for(gene in colnames(rt)[2:ncol(rt)]){

rt1=rt[,c(gene,"Type")]

colnames(rt1)=c("expression","Type")

p=1

if(length(levels(factor(rt1$Type)))>2){

test=kruskal.test(expression ~ Type, data = rt1)

p=test$p.value

}else{

test=wilcox.test(expression ~ Type, data = rt1)

p=test$p.value

}

Sig=ifelse(p<0.001,"\*\*\*",ifelse(p<0.01,"\*\*",ifelse(p<0.05,"\*","")))

geneSig=c(geneSig,Sig)

}

colnames(rt)=paste0(colnames(rt),geneSig)

#鎶婃暟鎹浆鎹㈡垚ggplot2杈撳叆鏂囦欢

data=melt(rt,id.vars=c("Type"))

colnames(data)=c("Type","Gene","Expression")

#缁樺埗

p1=ggplot(data,aes(x=Type,y=Expression,fill=Type))+

guides(fill=guide\_legend(title=x))+

labs(x = x, y = "Immunosuppressive genes")+

geom\_boxplot()+ facet\_wrap(~Gene,nrow =1)+ theme\_bw()+

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

#杈撳嚭

pdf(file=outFile, width=9, height=5)

print(p1)

dev.off()

immFile="CIBERSORT-Results.txt" #免疫细胞浸润的结果文件

riskFile="risk.GEO.txt" #风险文件

setwd("C:\\Users\\DELL\\Desktop\\43.immuneCoryiHEmianyijianchadian") #设置工作目录

#读取免疫细胞浸润的结果文件，并对数据进行整理

immune=read.table(immFile, header=T, sep="\t", check.names=F, row.names=1)

immune=immune[immune[,"P-value"]<0.05,]

data=as.matrix(immune[,1:(ncol(immune)-3)])

rownames(data)=gsub("(.\*?)\\\_(.\*?)", "\\2", rownames(data))

#读取风险文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

sameSample=intersect(row.names(data), row.names(risk))

data=data[sameSample,,drop=F]

risk=risk[sameSample,,drop=F]

#对所有免疫细胞进行循环，得到风险打分与免疫细胞的相关性

for(i in colnames(data)[1:ncol(data)]){

x=as.numeric(risk[,"riskScore"])

x[x>quantile(x,0.99)]=quantile(x,0.99)

y=as.numeric(data[,i])

if(sd(y)<0.01){next}

cor=cor.test(x, y, method="spearman")

#对pvalue小于0.05的免疫细胞进行相关性散点图的绘制

if(cor$p.value<0.05){

outFile=paste0("cor.", i, ".pdf")

df1=as.data.frame(cbind(x,y))

p1=ggplot(df1, aes(x, y)) +

xlab("Risk score") + ylab(i)+

geom\_point() + geom\_smooth(method="lm",formula = y ~ x) + theme\_bw()+

stat\_cor(method = 'spearman', aes(x =x, y =y))

p2=ggMarginal(p1, type="density", xparams=list(fill = "#CE0000"), yparams=list(fill = "#0080FF"))

#相关性图形

pdf(file=outFile, width=5.2, height=5)

print(p2)

dev.off()

}

}

#基因与免疫细胞相关性分析

outTab=data.frame()

risk=risk[,3:(ncol(risk)-1),drop=F]

for(immune in colnames(data)){

if(sd(data[,immune])<0.01){next}

for(gene in colnames(risk)){

x=as.numeric(data[,immune])

y=as.numeric(risk[,gene])

y[y>quantile(y,0.99)]=quantile(y,0.99)

corT=cor.test(x,y,method="spearman")

cor=corT$estimate

pvalue=corT$p.value

text=ifelse(pvalue<0.001,"\*\*\*",ifelse(pvalue<0.01,"\*\*",ifelse(pvalue<0.05,"\*","")))

outTab=rbind(outTab,cbind(Gene=gene, Immune=immune, cor, text, pvalue))

}

}

#绘制相关性热图

outTab$cor=as.numeric(outTab$cor)

outTab[,"Gene"]=factor(outTab[,"Gene"], levels=colnames(risk))

pdf(file="geneImmuneCor.pdf", width=7, height=5.5)

ggplot(outTab, aes(Gene, Immune)) +

geom\_tile(aes(fill = cor), colour = "grey", size = 1)+

scale\_fill\_gradient2(low = "#CE0000", mid = "white", high = "#0080FF") +

geom\_text(aes(label=text),col ="black",size = 3) +

theme\_minimal() +

theme(axis.title.x=element\_blank(), axis.ticks.x=element\_blank(), axis.title.y=element\_blank(),

axis.text.x = element\_text(angle = 45, hjust = 1, size = 9, face = "bold"), #x轴字体

axis.text.y = element\_text(size = 9, face = "bold")) + #y轴字体

labs(fill =paste0("\*\*\* p<0.001","\n", "\*\* p<0.01","\n", " \* p<0.05","\n", "\n","Correlation")) + #设置图例

scale\_x\_discrete(position = "bottom") #X轴基因名字显示在图形的下方

dev.off()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ERCC5 |  |  | POLK |  |  | PPP2R2A |  |  |
| HEEC | ECA109 | TE1 | HEEC | ECA109 | TE1 | HEEC | ECA109 | TE1 |
| 1.412 | 1.737 | 2.373 | 0.877 | 0.318 | 0.227 | 0.876 | 1.435 | 2.109 |
| 1.388 | 1.687 | 1.798 | 0.647 | 0.324 | 0.330 | 0.733 | 1.786 | 1.989 |
| 1.498 | 1.775 | 1.698 | 0.420 | 0.412 | 0.271 | 0.644 | 2.134 | 2.432 |
| TNP1 |  |  | ZNF350 |  |  |  |  |  |
| HEEC | ECA109 | TE1 | HEEC | ECA109 | TE1 |  |  |  |
| 0.539 | 0.345 | 0.678 | 0.852 | 0.634 | 0.745 |  |  |  |
| 0.678 | 0.547 | 0.458 | 0.731 | 0.823 | 0.519 |  |  |  |
| 0.544 | 0.712 | 0.529 | 0.943 | 0.652 | 0.698 |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ECA109 |  |  |  |  |  |  | TE1 |  |  |  |  |  |  |
|  | si-NC |  |  | si-PPP2R2A | |  |  | si-NC |  |  | si-PPP2R2A | |  |
| 1 | 0.476 | 0.4256 | 0.417 | 0.424 | 0.451 | 0.401 | 1 | 0.278 | 0.309 | 0.316 | 0.245 | 0.311 | 0.276 |
| 2 | 0.823 | 0.786 | 0.769 | 0.529 | 0.589 | 0.578 | 2 | 0.698 | 0.725 | 0.699 | 0.498 | 0.501 | 0.524 |
| 3 | 0.969 | 1.212 | 1.275 | 0.901 | 0.856 | 0.979 | 3 | 0.908 | 1.112 | 1.219 | 0.764 | 0.799 | 0.839 |
| 4 | 1.452 | 1.652 | 1.496 | 1.125 | 1.196 | 1.084 | 4 | 1.339 | 1.423 | 1.501 | 0.945 | 0.967 | 1.076 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | si-NC |  |  | si-PPP2R2A | |  |
| ECA109 | 353 | 345 | 329 | 137 | 149 | 150 |
| TE1 | 424 | 430 | 419 | 127 | 109 | 116 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | si-NC |  |  | si-PPP2R2A |  |  |
| ECA109 | 1 | 1 | 1 | 0.215 | 0.239 | 0.304 |
| TE1 | 1 | 1 | 1 | 0.342 | 0.321 | 0.395 |

coxPfilter=0.05 #显著性的过滤标准

inputFile="GEO.expTime.txt" #输入文件

setwd("C:\\Users\\DELL\\Desktop\\16.uniCoxGSE53625") #设置工作目录

#读取输入文件

rt=read.table(inputFile, header=T, sep="\t", check.names=F, row.names=1)

rt$futime=rt$futime/365

#对基因进行循环，查找预后相关的基因

outTab=data.frame()

sigGenes=c("futime","fustat")

for(i in colnames(rt[,3:ncol(rt)])){

#cox分析

cox <- coxph(Surv(futime, fustat) ~ rt[,i], data = rt)

coxSummary = summary(cox)

coxP=coxSummary$coefficients[,"Pr(>|z|)"]

#保留预后相关的基因

if(coxP<coxPfilter){

sigGenes=c(sigGenes,i)

outTab=rbind(outTab,

cbind(id=i,

HR=coxSummary$conf.int[,"exp(coef)"],

HR.95L=coxSummary$conf.int[,"lower .95"],

HR.95H=coxSummary$conf.int[,"upper .95"],

pvalue=coxSummary$coefficients[,"Pr(>|z|)"])

)

}

}

#输出单因素的结果

write.table(outTab,file="GEO.uniCox.txt",sep="\t",row.names=F,quote=F)

#输出单因素显著基因的表达量

uniSigExp=rt[,sigGenes]

uniSigExp=cbind(id=row.names(uniSigExp),uniSigExp)

write.table(uniSigExp,file="GEO.uniSigExp.txt",sep="\t",row.names=F,quote=F)

############定义森林图函数############

bioForest=function(coxFile=null,forestFile=null,forestCol=null){

#读取输入文件

rt <- read.table(coxFile,header=T,sep="\t",row.names=1,check.names=F)

gene <- rownames(rt)

hr <- sprintf("%.3f",rt$"HR")

hrLow <- sprintf("%.3f",rt$"HR.95L")

hrHigh <- sprintf("%.3f",rt$"HR.95H")

Hazard.ratio <- paste0(hr,"(",hrLow,"-",hrHigh,")")

pVal <- ifelse(rt$pvalue<0.001, "<0.001", sprintf("%.3f", rt$pvalue))

#输出图形

height=nrow(rt)/12.5+5

pdf(file=forestFile, width = 7,height = height)

n <- nrow(rt)

nRow <- n+1

ylim <- c(1,nRow)

layout(matrix(c(1,2),nc=2),width=c(3,2.5))

#绘制森林图左边的基因信息

xlim = c(0,3)

par(mar=c(4,2.5,2,1))

plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,xlab="",ylab="")

text.cex=0.8

text(0,n:1,gene,adj=0,cex=text.cex)

text(1.5-0.5\*0.2,n:1,pVal,adj=1,cex=text.cex);text(1.5-0.5\*0.2,n+1,'pvalue',cex=text.cex,font=2,adj=1)

text(3,n:1,Hazard.ratio,adj=1,cex=text.cex);text(3,n+1,'Hazard ratio',cex=text.cex,font=2,adj=1,)

#绘制森林图

par(mar=c(4,1,2,1),mgp=c(2,0.5,0))

xlim = c(0,max(as.numeric(hrLow),as.numeric(hrHigh)))

plot(1,xlim=xlim,ylim=ylim,type="n",axes=F,ylab="",xaxs="i",xlab="Hazard ratio")

arrows(as.numeric(hrLow),n:1,as.numeric(hrHigh),n:1,angle=90,code=3,length=0.05,col="darkblue",lwd=2.5)

abline(v=1,col="black",lty=2,lwd=2)

boxcolor = ifelse(as.numeric(hr) > 1, forestCol[1], forestCol[2])

points(as.numeric(hr), n:1, pch = 15, col = boxcolor, cex=1.6)

axis(1)

dev.off()

}

#调用函数，绘制森林图

bioForest(coxFile="GEO.uniCox.txt",forestFile="forest.pdf",forestCol=c("#CE0000","#019858"))

#引用包

library("glmnet")

library("survival")

set.seed(12345)

trainFile="TCGA.uniSigExp.txt" #TCGA数据库输入文件

testFile="GEO.expTime.txt" #GEO数据库输入文件

setwd("C:\\Users\\DELL\\Desktop\\ZHIFANGSUDAO296gehoutianjiadejige\\18.modelriskscror") #设置工作目录

#读取train组数据文件

rt=read.table(trainFile, header=T, sep="\t", row.names=1)

rt$futime[rt$futime<=0]=0.003

#构建lasso回归模型

x=as.matrix(rt[,c(3:ncol(rt))])

y=data.matrix(Surv(rt$futime,rt$fustat))

fit=glmnet(x, y, family = "cox", maxit = 1000)

#绘制lasso回归图形

pdf("lasso.lambda.pdf")

plot(fit, xvar="lambda", label=TRUE)

dev.off()

#绘制交叉验证图形

cvfit=cv.glmnet(x, y, family="cox", maxit=1000)

pdf("lasso.cvfit.pdf")

plot(cvfit)

abline(v=log(c(cvfit$lambda.min,cvfit$lambda.1se)), lty="dashed")

dev.off()

#找到交叉验证误差最小的点，并且输出模型公式

coef=coef(fit, s = cvfit$lambda.min)

index=which(coef != 0)

actCoef=coef[index]

lassoGene=row.names(coef)[index]

geneCoef=cbind(Gene=lassoGene,Coef=actCoef)

write.table(geneCoef,file="geneCoef.txt",sep="\t",quote=F,row.names=F)

#输出TCGA数据库的风险文件

trainFinalGeneExp=rt[,lassoGene]

myFun=function(x){crossprod(as.numeric(x),actCoef)}

trainScore=apply(trainFinalGeneExp,1,myFun)

outCol=c("futime","fustat",lassoGene)

Risk=as.vector(ifelse(trainScore>median(trainScore),"high","low"))

outTab=cbind(rt[,outCol],riskScore=as.vector(trainScore),Risk)

write.table(cbind(id=rownames(outTab),outTab),file="risk.TCGA.txt",sep="\t",quote=F,row.names=F)

#输出GEO数据库的风险文件

rt=read.table(testFile, header=T, sep="\t", row.names=1)

rt$futime=rt$futime/365

testFinalGeneExp=rt[,lassoGene]

testScore=apply(testFinalGeneExp,1,myFun)

outCol=c("futime","fustat",lassoGene)

Risk=as.vector(ifelse(testScore>median(trainScore),"high","low"))

outTab=cbind(rt[,outCol],riskScore=as.vector(testScore),Risk)

write.table(cbind(id=rownames(outTab),outTab),file="risk.GEO.txt",sep="\t",quote=F,row.names=F)

#引用包

library(ggalluvial)

library(ggplot2)

library(dplyr)

expFile="impute.txt" #表达数据文件

riskFile="risk.GEO.txt" #风险文件

corFilter=0.5 #相关系数过滤条件

pFilter=0.001 #相关性检验pvalue过滤条件

setwd("C:\\Users\\DELL\\Desktop\\24.ggalluvial") #设置工作目录

#读取输入文件

rt=read.table(expFile, header=T, sep="\t", check.names=F, row.names=1)

rt=na.omit(rt)

#读取风险文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

#数据取交集

sameSample=intersect(colnames(rt), row.names(risk))

rt=rt[,sameSample]

risk=risk[sameSample,]

#相关性分析

outTab=data.frame()

for(protein1 in colnames(risk)[3:(ncol(risk)-2)]){

x=as.numeric(rt[protein1,])

for(protein2 in rownames(rt)){

if(protein1==protein2){next}

y=as.numeric(rt[protein2,])

corT=cor.test(x,y)

cor=corT$estimate

cor=round(cor,3)

pvalue=corT$p.value

if((abs(cor)>corFilter) & (pvalue<pFilter)){

outTab=rbind(outTab, cbind(protein1, protein2, cor, pvalue))

}

}

}

outTab=outTab[order(outTab$protein1, as.numeric(outTab[,"pvalue"])),]

write.table(file="cor.result.txt", outTab, sep="\t", quote=F, row.names=F)

#绘制桑基图

outTab=outTab %>% group\_by(protein1) %>% slice\_head(n=10)

corLodes=to\_lodes\_form(outTab[,c(1,2)], axes = 1:2, id = "Cohort")

mycol=rep(c("#0080FF","#CE0000","#FF0000","#019858","#6E568C","#E0367A","#D8D155","#223D6C","#D20A13","#431A3D","#91612D","#FFD121","#088247","#11AA4D","#58CDD9","#7A142C","#5D90BA","#64495D","#7CC767"),15)

pdf(file="ggalluvial.pdf", width=8, height=8)

ggplot(corLodes, aes(x = x, stratum = stratum, alluvium = Cohort,fill = stratum, label = stratum)) +

scale\_x\_discrete(expand = c(0, 0)) +

#设置线条颜色，forward说明线条颜色与前面的柱状图一致，backward说明线条颜色与后面的柱状图一致。

geom\_flow(width = 2/10,aes.flow = "forward") +

geom\_stratum(alpha = .9, width = 2.5/10) +

scale\_fill\_manual(values = mycol) +

#size=3代表字体大小

geom\_text(stat = "stratum", size = 3,color="black") +

xlab("") + ylab("") + theme\_bw() +

theme(axis.line = element\_blank(),axis.ticks = element\_blank(),axis.text.y = element\_blank()) + #去掉坐标轴

theme(panel.grid =element\_blank()) +

theme(panel.border = element\_blank()) +

ggtitle("") + guides(fill = FALSE)

dev.off()

#引用包

library(limma)

library(org.Hs.eg.db)

library(clusterProfiler)

library(enrichplot)

expFile="symbol.txt" #表达数据文件

riskFile="risk.GEO.txt" #风险文件

gmtFile="c5.go.symbols.gmt" #基因集文件

setwd("C:\\Users\\DELL\\Desktop\\gsea") #设置工作目录

#读取文件,并对输入文件进行整理

rt=read.table(expFile, header=T, sep="\t", check.names=F)

rt=as.matrix(rt)

rownames(rt)=rt[,1]

exp=rt[,2:ncol(rt)]

dimnames=list(rownames(exp),colnames(exp))

data=matrix(as.numeric(as.matrix(exp)),nrow=nrow(exp),dimnames=dimnames)

data=avereps(data)

data=data[rowMeans(data)>0.5,]

#读取风险文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

sameSample=intersect(row.names(risk), colnames(data))

risk=risk[sameSample,]

data=data[,sameSample]

#高低风险比较，得到logFC

dataL=data[,row.names(risk[risk[,"risk"]=="low",])]

dataH=data[,row.names(risk[risk[,"risk"]=="high",])]

meanL=rowMeans(dataL)

meanH=rowMeans(dataH)

meanL[meanL<0.00001]=0.00001

meanH[meanH<0.00001]=0.00001

logFC=log2(meanH)-log2(meanL)

logFC=sort(logFC,decreasing=T)

genes=names(logFC)

#读取基因集文件

gmt=read.gmt(gmtFile)

#GSEA富集分析

kk=GSEA(logFC, TERM2GENE=gmt, pvalueCutoff=1, minGSSize=15, maxGSSize=500)

kkTab=as.data.frame(kk)

kkTab=kkTab[kkTab$pvalue<0.05,]

write.table(kkTab,file="GSEA.result.txt",sep="\t",quote=F,row.names = F)

#输出高风险富集的图形

termNum=5 #设置展示通路的数目，展示前5个通路

kkUp=kkTab[kkTab$NES>0,]

if(nrow(kkUp)>=termNum){

showTerm=c("GOBP\_B\_CELL\_RECEPTOR\_SIGNALING\_PATHWAY",

"GOBP\_POSITIVE\_REGULATION\_OF\_DNA\_BINDING",

"GOBP\_HUMORAL\_IMMUNE\_RESPONSE\_MEDIATED\_BY\_CIRCULATING\_IMMUNOGLOBULIN",

"GOBP\_T\_CELL\_CYTOKINE\_PRODUCTION",

"GOBP\_IMMUNE\_RESPONSE\_REGULATING\_CELL\_SURFACE\_RECEPTOR\_SIGNALING\_PATHWAY")

gseaplot=gseaplot2(kk, showTerm, base\_size=8, title="Enriched in high risk group",color = c("#019858","black","#CE0000","#0080FF","lightpink"))

pdf(file="GSEA.highrisk.pdf", width=15, height=5.5)

print(gseaplot)

dev.off()

}

#输出低风险富集的图形

termNum=5 #设置展示通路的数目，展示前5个通路

kkDown=kkTab[kkTab$NES<0,]

if(nrow(kkDown)>=termNum){

showTerm=c("GOBP\_NEUTROPHIL\_MEDIATED\_IMMUNITY",

"GOBP\_INFLAMMATORY\_RESPONSE\_TO\_ANTIGENIC\_STIMULUS",

"GOBP\_POSITIVE\_REGULATION\_OF\_INTRINSIC\_APOPTOTIC\_SIGNALING\_PATHWAY",

"GOBP\_HUMORAL\_IMMUNE\_RESPONSE",

"GOBP\_REGULATION\_OF\_EPIDERMAL\_GROWTH\_FACTOR\_ACTIVATED\_RECEPTOR\_ACTIVITY")

gseaplot=gseaplot2(kk, showTerm, base\_size=8, title="Enriched in low risk group",color = c("#019858","black","#CE0000","#0080FF","lightpink"))

pdf(file="GSEA.lowrisk.pdf", width=10, height=5.5)

print(gseaplot)

dev.off()

}

expFile="symbol.txt" #表达数据文件

riskFile="risk.GEO.txt" #风险文件

gmtFile="c2.cp.kegg.symbols.gmt" #基因集文件

setwd("C:\\Users\\DELL\\Desktop\\gsea") #设置工作目录

#读取文件,并对输入文件进行整理

rt=read.table(expFile, header=T, sep="\t", check.names=F)

rt=as.matrix(rt)

rownames(rt)=rt[,1]

exp=rt[,2:ncol(rt)]

dimnames=list(rownames(exp),colnames(exp))

data=matrix(as.numeric(as.matrix(exp)),nrow=nrow(exp),dimnames=dimnames)

data=avereps(data)

data=data[rowMeans(data)>0.5,]

#读取风险文件

risk=read.table(riskFile, header=T, sep="\t", check.names=F, row.names=1)

sameSample=intersect(row.names(risk), colnames(data))

risk=risk[sameSample,]

data=data[,sameSample]

#高低风险比较，得到logFC

dataL=data[,row.names(risk[risk[,"risk"]=="low",])]

dataH=data[,row.names(risk[risk[,"risk"]=="high",])]

meanL=rowMeans(dataL)

meanH=rowMeans(dataH)

meanL[meanL<0.00001]=0.00001

meanH[meanH<0.00001]=0.00001

logFC=log2(meanH)-log2(meanL)

logFC=sort(logFC,decreasing=T)

genes=names(logFC)

#读取基因集文件

gmt=read.gmt(gmtFile)

#GSEA富集分析

kk=GSEA(logFC, TERM2GENE=gmt, pvalueCutoff=1, minGSSize=15, maxGSSize=500)

kkTab=as.data.frame(kk)

kkTab=kkTab[kkTab$pvalue<0.05,]

write.table(kkTab,file="GSEA.result.txt",sep="\t",quote=F,row.names = F)

#输出高风险富集的图形

termNum=5 #设置展示通路的数目，展示前5个通路

kkUp=kkTab[kkTab$NES>0,]

if(nrow(kkUp)>=termNum){

showTerm=c("KEGG\_TGF\_BETA\_SIGNALING\_PATHWAY",

"KEGG\_ECM\_RECEPTOR\_INTERACTION",

"KEGG\_INTESTINAL\_IMMUNE\_NETWORK\_FOR\_IGA\_PRODUCTION",

"KEGG\_CELL\_ADHESION\_MOLECULES\_CAMS",

"KEGG\_ANTIGEN\_PROCESSING\_AND\_PRESENTATION")

gseaplot=gseaplot2(kk, showTerm, base\_size=8, title="Enriched in high risk group",color = c("#019858","black","#CE0000","#0080FF","lightpink"))

pdf(file="GSEA.highrisk.KEGG.pdf", width=8, height=5.5)

print(gseaplot)

dev.off()

}

#输出低风险富集的图形

termNum=5 #设置展示通路的数目，展示前5个通路

kkDown=kkTab[kkTab$NES<0,]

if(nrow(kkDown)>=termNum){

showTerm=c("KEGG\_LINOLEIC\_ACID\_METABOLISM",

"KEGG\_GALACTOSE\_METABOLISM",

"KEGG\_GLUTATHIONE\_METABOLISM",

"KEGG\_RNA\_POLYMERASE",

"KEGG\_RIBOSOME")

gseaplot=gseaplot2(kk, showTerm, base\_size=8, title="Enriched in low risk group",color = c("#019858","black","#CE0000","#0080FF","lightpink"))

pdf(file="GSEA.lowrisk.KEGG.pdf", width=8, height=5.5)

print(gseaplot)

dev.off()

}