**#####################R script for Figure 2 and Figure 9################**

rm(list = ls())

#setting work directory

mainDir <-"C:/Bioinformatics/000 EPC rt/radiosensitivity"

subDir <-"Step07.diff"

outputDir <- file.path(mainDir, subDir)

setwd(outputDir)

getwd()

suppressMessages(library(ggpubr));suppressMessages(library(ggthemes))

load("Step07.02.GSE45670.DEG.limma.rda")

#write.csv(DEG.limma,file = "dataset for Figure 2.csv")

deg.data <- DEG.limma

head(deg.data)

deg.data$logP <- -log10(deg.data$P.Value)

symbol<-rownames(deg.data)

deg.data<-cbind(symbol,deg.data)

deg.data$Label=""

deg.data <- deg.data[order(deg.data$P.Value),]

up.genes <- head(deg.data$symbol[which(deg.data$change == "UP")],10)

down.genes <- head(deg.data$symbol[which(deg.data$change =="DOWN")],10)

deg.top10.genes <- c(as.character(up.genes),as.character(down.genes))

deg.data$Label[match(deg.top10.genes,deg.data$symbol)] <- deg.top10.genes

#plotting

suppressMessages(library(ggrepel))

suppressMessages(library(ggplot2))

df = data.frame('logFC'=deg.data$logFC,

'FDR'=deg.data$logP,

'Gene'=rownames(deg.data),

'change'=deg.data$change,

'label'=deg.data$Label,

stringsAsFactors = F)

colnames(df)

wn\_volcano = function(df,logFC\_cutoff=1,P.Value\_cutoff=0.05){

require(RColorBrewer,quietly = T,warn.conflicts =F)

require(ggplot2,quietly = T,warn.conflicts =F)

volcano1=ggplot(data = df, aes(x = logFC, y = FDR))

volcano1=volcano1+geom\_point(alpha=0.8, size=2, aes(color=change)) +

scale\_color\_manual(values=c('#00AFBB', '#999999', '#FC4E07')) +

geom\_vline(xintercept=c(-logFC\_cutoff,logFC\_cutoff),lty=3,col="azure4",lwd=1)+

geom\_hline(yintercept = -log10(P.Value\_cutoff),lty=3,col="azure4",lwd=1)+

ylab('-log10(FDR)')+

xlab('log2(FoldChange)')+

theme\_bw()

#p1

volcano<-volcano1 + geom\_text\_repel(data=df, aes(label= label),

color="black", family="Times New Roman",

size=2.5, fontface="italic",

arrow = arrow(ends="first", length = unit(0.01, "npc")),

box.padding = 0.2,

point.padding = 0.3,

segment.color = 'black',

segment.size = 0.3,

max.overlaps = getOption("ggrepel.max.overlaps", default = 10),

segment.alpha=0.8,

force = 1,

max.iter = 3e3)

return(volcano)

}

##call the function

volcano = wn\_volcano(df=df,logFC\_cutoff=logFC\_cutoff,P.Value\_cutoff=P.Value\_cutoff)

volcano

class(volcano)

##################heatmap#################

load("C:/Bioinformatics/000 EPC rt/radiosensitivity/Step07.diff/Step07.04.diffLncRNAExp.rda")

write.csv(diffLncRNAExp,file = "heatmapdata for Figure 2.csv")

#install.packages("pheatmap")

#install.packages("ggplot2")

wn\_pheatmap = function(heat.df,Group){

suppressMessages(library(pheatmap))

suppressMessages(library(ggplot2))

annotation\_col = data.frame(Group)

rownames(annotation\_col)

colnames(heat.df)

rownames(annotation\_col) <- colnames(heat.df)

pheatmap <- pheatmap(heat.df,scale="row",

annotation\_col = annotation\_col,

border="white",

cluster\_cols = F,treeheight\_col = 25,

cluster\_rows = T,treeheight\_row = 25,

show\_rownames = T,

show\_colnames = T,

legend = T,

fontsize\_row = 5,fontsize\_col = 5,

clustering\_distance\_rows = "correlation",

clustering\_method="single",

angle\_col = 45,

cellwidth = 8,cellheight = 8,

cutree\_cols = 6, cutree\_rows =5

)

return(pheatmap)

}

pheatmap = wn\_pheatmap(heat.df=diffLncRNAExp,Group=group\_list)

print(pheatmap)

if (!require("ggplotify", character.only=T, quietly=T)) {

install.packages("ggplotify")

library("ggplotify", character.only=T)

}

if (!require("patchwork", character.only=T, quietly=T)) {

install.packages("patchwork")

library("patchwork", character.only=T)

}

if (!require("cowplot", character.only=T, quietly=T)) {

install.packages("cowplot")

library("cowplot", character.only=T)

}

pheatmap.grid <- grid2grob(print(pheatmap))

class(pheatmap)

save(volcano,pheatmap,pheatmap.grid,file="Step07.05.picdata.rda")

p3<-plot\_grid(pheatmap.grid,volcano,

labels = "AUTO",

rel\_widths = c(1.5, 1),

rel\_heights=c(1.5,1),

ncol=2)

print(p3)

#save plotting

phname<-"FIGURE 2 Differential expression results of immune-related lncRNAs in GSE45670. (A) Heatmap, (B) Volcano map"

phtype<-"tiff"

ggsave(filename = paste0(phname,".",phtype),p3,width =10,

height = 5.5, dpi = 300, units = "in", device=phtype)

**#####################R script for Figure 3 and Figure 7################**

rm(list = ls())

memory.limit(102400)

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step10.uniCox"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

setwd(outputDir)

getwd()

library(plyr)

library(dplyr)

suppressMessages(library(forestplot))

data.prepare= function(dataset=data.read){

names(dataset)[1]<-"Variable"

input.df<-dataset %>% plyr::rename(c(HR="estimate",HR.95L="lowerCI",HR.95H="upperCI"))%>%

dplyr::select(Variable, estimate, lowerCI,upperCI,pvalue)

effectindex="HR"

if(effectindex=="HR") {

effectname<-"HR with 95% CI"

} else if(effectindex=="OR") {

effectname<-"OR with 95% CI"

} else if(effectindex=="RR" ) {

effectname<-"RR with 95% CI"

}

Variable <- input.df[,"Variable"]

hr <- sprintf("%.3f",input.df$"estimate")

hrLow <- sprintf("%.3f",input.df$"lowerCI")

hrHigh <- sprintf("%.3f",input.df$"upperCI")

effect.indicator <- paste0(hr,"(",hrLow,"-",hrHigh,")")

NA.check<-grepl("NA",effect.indicator,ignore.case=T)

effect.indicator[NA.check==TRUE]<-NA

Pvalue <- ifelse(input.df$pvalue<0.001, "<0.001", sprintf("%.3f", input.df$pvalue))

labelmatrix<-cbind(c("Variable",Variable),

c(effectname,effect.indicator),

c("P value",Pvalue))

return(list(input.df,labelmatrix))

}

#

my\_forest= function(input.df=input.df,labelmatrix=labelmatrix,effectindex="HR"){

forst<-forestplot(labeltext=labelmatrix,

mean=c(NA,signif(input.df$"estimate",4)),

lower=c(NA,signif(input.df$"lowerCI",4)),

upper=c(NA,signif(input.df$"upperCI",4)),

#title = "Forestplot",

clip = c(-4, 4),

hrzl\_lines = list("1" = gpar(lty=1, lwd=2,columns=1:4, col="black"),

"2" = gpar(lty=2, lwd=2,columns=1:4, col="black"),

"5" = gpar(lty=1, lwd=2,columns=1:4, col="black")),

graphwidth = unit(0.4,"npc"),

align=c("l","c","c","c"),

col=fpColors(box='#458B00',

summary='#8B008B',

lines = 'black',

zero = '#7AC5CD'),

zero=1,

boxsize=0.5,

mar=unit(rep(1.25, times = 4), "cm"),

colgap=unit(5,'mm'),

lineheight=unit(9,'mm'),line.margin = 0.08,

graph.pos=2,

txt\_gp=fpTxtGp(label=gpar(cex=0.8), ticks=gpar(cex=0.8), xlab=gpar(cex = 0.8), title=gpar(cex = 0.8)),

xlab=effectindex,

xticks = c(-1.0,0, 1,2, 4.0),

lty.ci = "solid",

ci.vertices=T,

lwd.zero=1.5, lwd.ci=2,lwd.xaxis =1)

return(forst)

}

#load data

data.unicox.clin <- read.csv("uniCox.clin.csv", header = T)

#data processing

data.prepare.clin<-data.prepare(dataset=data.unicox.clin)

#get last line

linelast <-nrow(data.prepare.clin[[2]])+1

print(linelast)

#

forst.uniCOX.clin<-my\_forest(input.df=data.prepare.clin[[1]],

labelmatrix=data.prepare.clin[[2]],

effectindex="HR")

print(forst.uniCOX.clin)

save(data.unicox.clin,data.prepare.clin,forst.uniCOX.clin,

file = "unicox.clin.Rda")

#save plot

library(ggplotify)

library(cowplot)

library(ggplot2)

forst.uniCOX.clin.grid <- grid2grob(print(forst.uniCOX.clin))

phname<-"forst.uniCOX.clin"

phtype<-"tiff"

savefile<-forst.uniCOX.clin.grid

ggsave(filename = paste0(phname,".",phtype),savefile,width =8,

height = 6, dpi = 300, units = "in", device=phtype)

data.multicox.clin <- read.csv("multiCox.clin.csv", header = T)

data.prepare.clin<-data.prepare(dataset=data.multicox.clin)

linelast <-nrow(data.prepare.clin[[2]])+1

print(linelast)

forst.multicox.clin<-my\_forest(input.df=data.prepare.clin[[1]],

labelmatrix=data.prepare.clin[[2]],

effectindex="HR")

print(forst.multicox.clin)

save(data.multicox.clin,data.prepare.clin,forst.multicox.clin,

file = "multicox.clin.Rda")

forst.multicox.clin.grid <- grid2grob(print(forst.multicox.clin))

phname<-"forst.multicox.clin"

phtype<-"tiff"

savefile<-forst.multicox.clin.grid

ggsave(filename = paste0(phname,".",phtype),savefile,width =8,

height = 5, dpi = 300, units = "in", device=phtype)

#merge plotting

pmerge.clin<-plot\_grid(forst.uniCOX.clin.grid,

forst.multicox.clin.grid,

rel\_heights=c(1.8,1),

labels = "AUTO",ncol = 1)

phname<-"forst.merge.clin"

phtype<-"pdf"

savefile<-pmerge.clin

ggsave(filename = paste0(phname,".",phtype),savefile,width =6,

height = 9, dpi = 300, units = "in", device=phtype)

#

data.unicox.lncRNAs <- read.csv("unicox.lncRNAs.csv", header = T)

data.prepare.lncRNAs<-data.prepare(dataset=data.unicox.lncRNAs)

linelast <-nrow(data.prepare.lncRNAs[[2]])+1

print(linelast)

forst.uniCOX.lncRNAs<-my\_forest(input.df=data.prepare.lncRNAs[[1]],

labelmatrix=data.prepare.lncRNAs[[2]],

effectindex="HR")

print(forst.uniCOX.lncRNAs)

save(data.unicox.lncRNAs,data.prepare.lncRNAs,forst.uniCOX.lncRNAs,

file = "unicox.lncRNAs.Rda")

forst.uniCOX.lncRNAs.grid <- grid2grob(print(forst.uniCOX.lncRNAs))

phname<-"forst.uniCOX.lncRNAs"

phtype<-"tiff"

savefile<-forst.uniCOX.lncRNAs.grid

ggsave(filename = paste0(phname,".",phtype),savefile,width =8,

height = 5, dpi = 300, units = "in", device=phtype)

data.multicox.lncRNAs <- read.csv("multicox.lncrna.csv", header = T)

data.prepare.lncRNAs<-data.prepare(dataset=data.multicox.lncRNAs)

linelast <-nrow(data.prepare.lncRNAs[[2]])+1

print(linelast)

forst.multicox.lncRNAs<-my\_forest(input.df=data.prepare.lncRNAs[[1]],

labelmatrix=data.prepare.lncRNAs[[2]],

effectindex="HR")

print(forst.multicox.lncRNAs)

save(data.multicox.lncRNAs,data.prepare.lncRNAs,forst.multicox.lncRNAs,

file = "multicox.lncRNAs.Rda")

forst.multicox.lncRNAs.grid <- grid2grob(print(forst.multicox.lncRNAs))

phname<-"forst.multicox.lncRNAs"

phtype<-"tiff"

savefile<-forst.multicox.lncRNAs.grid

ggsave(filename = paste0(phname,".",phtype),savefile,width =8,

height = 5, dpi = 300, units = "in", device=phtype)

#merge plotting

pmerge.lncRNAs<-plot\_grid(forst.uniCOX.lncRNAs.grid,

forst.multicox.lncRNAs.grid,

rel\_heights=c(1.8,1),

labels = "AUTO",ncol = 1)

phname<-"forst.merge.lncRNAs"

phtype<-"pdf"

savefile<-pmerge.lncRNAs

ggsave(filename = paste0(phname,".",phtype),savefile,width =8,

height = 6, dpi = 300, units = "in", device=phtype)

save(pmerge.lncRNAs,pmerge.clin,

file = "pmerge.Rda")

**#####################R script for Figure 4################**

###############ROC#####################

rm(list = ls())

memory.limit(102400)

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step12.ROC"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

file.copy("Step11.model/Step11.07.riskOut.rda", "Step12.ROC/Step11.07.riskOut.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

setwd(outputDir)

getwd()

library(survivalROC)

load("Step11.07.riskOut.rda")

rt<-riskOut

class(rt$futime)

class(rt$fustat)

class(rt$riskScore)

#ROC曲线绘制

predictTime=1\*12

roc=survivalROC(Stime=rt$futime, status=rt$fustat, marker=rt$riskScore, predict.time =predictTime, method="KM")

pdf(file="Step12.01.ROC.pdf", width=5.5, height=5.5)

plot(roc$FP, roc$TP, type="l", xlim=c(0,1), ylim=c(0,1),col="black",

xlab="False positive rate", ylab="True positive rate",

lwd = 2, cex.main=1.2, cex.lab=1.2, cex.axis=1.2, font=1.2)

polygon(x=c(0,roc$FP,1,0),y=c(0,roc$TP,1,0),col="#24B35D",border=NA)

text(0.85, 0.1, paste0("AUC=",sprintf("%.3f",roc$AUC)), cex=1.2)

segments(0,0,1,1,lty=2)

dev.off()

#Get the best cutoff

predictTime=1\*12

roc=survivalROC(Stime=rt$futime, status=rt$fustat, marker=rt$riskScore, predict.time =predictTime, method="KM")

sum=roc$TP-roc$FP

cutOp=roc$cut.values[which.max(sum)]

cutTP=roc$TP[which.max(sum)]

cutFP=roc$FP[which.max(sum)]

pdf(file="Step12.02.ROC.cutoff.pdf",width=5.5,height=5.5)

plot(roc$FP, roc$TP, type="l", xlim=c(0,1), ylim=c(0,1),col="black",

xlab="False positive rate", ylab="True positive rate",

lwd = 2, cex.main=1.2, cex.lab=1.2, cex.axis=1.2, font=1.2)

polygon(x=c(0,roc$FP,1,0),y=c(0,roc$TP,1,0),col="#24B35D",border=NA)

segments(0,0,1,1,lty=2)

points(cutFP,cutTP, pch=20, col="red",cex=1.5)

text(cutFP+0.15,cutTP-0.05,paste0("Cutoff:",sprintf("%0.3f",cutOp)))

text(0.85, 0.1, paste0("AUC=",sprintf("%.3f",roc$AUC)), cex=1.2)

dev.off()

#Grouped by best cutoff

risk=as.vector(ifelse(rt$riskScore>cutOp,"high","low"))

risk.df=cbind(rt, risk)

class(risk.df)

save(risk.df,file = "Step12.03.risk.df.rda")

######ROC curves for multiple times######

rocCol=c("red", "green", "blue")

aucText=c()

pdf(file="Step12.04.ROC.multiTime.pdf",width=6,height=6)

#1 year ROC

predictTime=1\*12

par(oma=c(0.5,1,0,1),font.lab=1.5,font.axis=1.5)

roc=survivalROC(Stime=rt$futime,

status=rt$fustat,

marker=rt$riskScore,

predict.time=predictTime,

method="KM")

plot(roc$FP, roc$TP, type="l", xlim=c(0,1), ylim=c(0,1),col=rocCol[1],

xlab="False positive rate", ylab="True positive rate",

lwd = 2, cex.main=1.3, cex.lab=1.2, cex.axis=1.2, font=1.2)

aucText=c(aucText,paste0("one year"," (AUC=",sprintf("%.3f",roc$AUC),")"))

abline(0,1)

#2 year ROC

predictTime=2\*12

roc=survivalROC(Stime=rt$futime, status=rt$fustat, marker=rt$riskScore, predict.time =predictTime, method="KM")

lines(roc$FP, roc$TP, type="l", xlim=c(0,1), ylim=c(0,1),col=rocCol[2],lwd = 2)

aucText=c(aucText,paste0("two year"," (AUC=",sprintf("%.3f",roc$AUC),")"))

#3 year ROC

predictTime=3\*12

roc=survivalROC(Stime=rt$futime, status=rt$fustat, marker=rt$riskScore, predict.time =predictTime, method="KM")

lines(roc$FP, roc$TP, type="l", xlim=c(0,1), ylim=c(0,1),col=rocCol[3],lwd = 2)

aucText=c(aucText,paste0("three year"," (AUC=",sprintf("%.3f",roc$AUC),")"))

#draw legend

legend("bottomright", aucText,lwd=2,bty="n",col=rocCol)

dev.off()

**#####################R script for Figure 5################**

#survival analysis

rm(list = ls())

memory.limit(102400)

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step13.survival"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

file.copy("Step12.ROC/Step12.03.risk.df.rda",

"Step13.survival/Step12.03.risk.df.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE) # 文件复制

setwd(outputDir)

getwd()

#install.packages("survival")

#install.packages("survminer")

library(survival)

library(survminer)

load("Step12.03.risk.df.rda")

inputdata<-risk.df

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

rt<-inputFile

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("%0.3f",pValue))

}

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

surPlot=ggsurvplot(fit,

data=rt,

conf.int=TRUE,

pval=pValue,

pval.size=6,

palette=c("red", "blue"),

legend.title="Risk",

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

xlab="Time(years)",

break.time.by = 12,

risk.table=TRUE,

risk.table.title="",

risk.table.height=.25)

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

print(surPlot)

dev.off()

}

bioSurvival(inputFile=inputdata,outFile="Step13.01.survival.pdf")

**#####################R script for Figure 6 and Figure 11################**

#clear data

rm(list = ls())

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step14.difflncRNA.plot"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

file=paste0("immuneLncRNAexp.rda")

filefrom<-paste0("Step05.irlncRNAselect/",file)

fileto<-paste0("Step14.difflncRNA.plot/",file)

file.copy(filefrom,

fileto,

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE) # 文件复制

setwd(outputDir)

getwd()

load(file)

load("Step14.01.grp.Rda")

load("Step12.03.risk.df.rda")

lncRNAlist<-c("LINC01121","FAM167A-AS1","ADAMTS9-AS2",

"MGC12916","MIR124-2HG","FAM167A-AS1")

group\_risk<-risk.df$risk

sample\_list<-rownames(risk.df)

save(lncRNAlist,group\_risk,sample\_list, file = "Step14.01.grp.Rda")

#lncRNAs selection

lncRNAexp.diff<-lncRNAexp.select[which(rownames(lncRNAexp.select) %in% lncRNAlist),which(colnames(lncRNAexp.select) %in% sample\_list)]

#Transpose Expression Matrix

lncRNAexp.trans<-t(lncRNAexp.diff)

lncRNAexp.sig<-data.frame(group\_risk,lncRNAexp.trans)

colnames(lncRNAexp.sig)<-gsub("\\.","-",colnames(lncRNAexp.sig))

#Organize data to fit in a panel boxplot

#Grouped by risk

col<-colnames(lncRNAexp.sig)

nrow.lncRNAexp.sig<-nrow(lncRNAexp.sig)

rownames(lncRNAexp.sig)<-NULL

length(col)

exp.lncRNA<-data.frame()

for (i in (2:length(col))){

exp.temp<-NULL

gene<-rep(col[i],nrow.lncRNAexp.sig)

exp.temp<-subset(lncRNAexp.sig,select = c(1,i))

colnames(exp.temp)<-c("Group","exp")

exp.temp<-data.frame(exp.temp,gene)

exp.lncRNA<-rbind(exp.lncRNA,exp.temp)

}

save(exp.lncRNA,file = "Step14.02.01.exp.lncRNA.Rda")

library("dplyr")

library("ggplot2")

library("ggpubr")

p <- ggplot(data=exp.lncRNA,aes(x=Group,y=exp))

print(p)

p+geom\_boxplot(aes(fill=Group))+facet\_wrap(~gene)+

labs(x="Different group of riskscore", y = "Gene expression")+

stat\_compare\_means(method = "t.test",aes(label = paste0("p =",as.character(sprintf("%0.3f", as.numeric(..p.format..))))))

ggsave(paste0("Fig6 different expression of ir-lncRNAs",".tiff"),width = 10,height = 10)

**#####################R script for Figure 8################**

setwd("C:/Bioinformatics/000 EPC rt/GSE53625/Step19.immuneCor/CIBERSORT")#设置三个文件所在的文件夹

rm(list = ls())

#data preparation

load("Figure 8 data.Rdata")

rowname<-rownames(exprSet.annotated.radiosensity)

data<-data.frame(rowname,exprSet.annotated.radiosensity)

library(tidyverse)

colnames(data)[1] <- 'Gene symbol'

write.table(data,file ="data.txt" ,sep ='\t',row.names=F,col.names=T)

library(ggplot2)

library(reshape2)

library(ggpubr)

library(dplyr)

source('Cibersort.R')#Download from Cibersort official website

LM22.file <- "LM22.txt"#Download from Cibersort official website

TCGA\_exp.file <- "data.txt"

TCGA\_TME.results <- CIBERSORT(LM22.file ,TCGA\_exp.file, perm =1000, QN = F)

save(TCGA\_TME.results,group\_list, file = "CIBERSORT.Rda")

load("CIBERSORT.Rda")

group\_list

Group<-factor(as.character(group\_list),levels=c('2','1'),labels=c("responder","non\_responder"))

table(Group) # Normal 43 Tumor 43

## 3. ploting

# 3.1 Coarse processing of data

TME\_data <- as.data.frame(TCGA\_TME.results[,1:22])

TME\_data$group <- Group

TME\_data$sample <- row.names(TME\_data)

# 3.2 data melt

TME\_New = melt(TME\_data)

colnames(TME\_New)=c("Group","Sample","Celltype","Composition") #设置行名

head(TME\_New)

# 3.3 Plot by median proportion of immune cells (optional)

plot\_order = TME\_New[TME\_New$Group=="non\_responder",] %>%

group\_by(Celltype) %>%

summarise(m = median(Composition)) %>%

arrange(desc(m)) %>%

pull(Celltype)

TME\_New$Celltype = factor(TME\_New$Celltype,levels = plot\_order)

# 3.3 plotting

if(T){

mytheme <- theme(plot.title = element\_text(size = 12,color="black",hjust = 0.5),

axis.title = element\_text(size = 12,color ="black"),

axis.text = element\_text(size= 12,color = "black"),

panel.grid.minor.y = element\_blank(),

panel.grid.minor.x = element\_blank(),

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

panel.grid=element\_blank(),

legend.position = "top",

legend.text = element\_text(size= 12),

legend.title= element\_text(size= 12)

) }

box\_TME <- ggplot(TME\_New, aes(x = Celltype, y = Composition))+

labs(y="Cell composition",x= NULL,title = "TME Cell composition")+

geom\_boxplot(aes(fill = Group),position=position\_dodge(0.5),width=0.5,outlier.alpha = 0)+

scale\_fill\_manual(values = c("#1CB4B8", "#EB7369"))+

theme\_classic() + mytheme +

stat\_compare\_means(aes(group = Group),

label = "p.signif",

method = "wilcox.test",

hide.ns = T)

box\_TME;ggsave("EPC\_TME.pdf",box\_TME,height=15,width=25,unit="cm")

##4.1. Extract the top 20 immune cells used in the literature

# 4.1 Extract the top 20 immune cells

TCGA\_TME\_four = as.data.frame(TCGA\_TME.results[,1:20])

head(TCGA\_TME\_four,3)

# 4.2 Classification of immune cells based on literature

immCell\_four\_type <- read.table("Cibersort\_four\_types.txt", header = T, row.names = NULL, sep = "\t")

colname<- colnames(TCGA\_TME\_four)

colname1<-gsub('\\.', ' ', colname)

colnames(TCGA\_TME\_four)<-colname1

####

colnames(TCGA\_TME\_four) == immCell\_four\_type$Immune.cells #T

## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

head(immCell\_four\_type)

# 4.3

TCGA\_TME\_four$group = Group

TCGA\_TME\_four$sample <- row.names(TCGA\_TME\_four)

TME\_four\_new = melt(TCGA\_TME\_four)

## Using group, sample as id variables

colnames(TME\_four\_new) = c("Group","Sample","Immune.cells","Composition")

TCGA\_TME\_four\_new2 = left\_join(TME\_four\_new, immCell\_four\_type, by = "Immune.cells") %>%

group\_by(Sample,Group,Types) %>%

drop\_na(Types)%>%

summarize(Sum = sum(Composition))

## `summarise()` regrouping output by 'Sample', 'Group' (override with `.groups` argument)

# plotting

box\_four\_immtypes <- ggplot(TCGA\_TME\_four\_new2, aes(x = Group, y = Sum))+

labs(y="Cell composition",x= NULL)+ #,title = "TCGA"

geom\_boxplot(aes(fill = Group),position=position\_dodge(0.5),width=0.5,size=0.4,

outlier.alpha = 1, outlier.size = 0.5)+

theme\_bw() + mytheme +

scale\_fill\_manual(values = c("#1CB4B8","#EB7369"))+

scale\_y\_continuous(labels = scales::percent)+

facet\_wrap(~ Types,scales = "free",ncol = 4) +

stat\_compare\_means(aes(group = Group),

label = "p.format",

method = "wilcox.test",

size = 3.5,

hide.ns = T)

box\_four\_immtypes;ggsave("Cibersort\_four\_immune\_cell\_types.pdf",

box\_four\_immtypes ,height= 12,width=25,unit="cm")

#Merge the two graphs

class(box\_four\_immtypes)

library(ggplotify)

library(patchwork)

library(cowplot)

p1 <- grid2grob(print(forst1))

p2 <- grid2grob(print(forst2))

p3<-plot\_grid(box\_TME,box\_four\_immtypes,labels = "AUTO",ncol = 1)

class(p3)

ggsave(filename = " infiltrating immune cell types.tiff",p3,width =12,

height = 15, dpi = 300, units = "in", device='tiff')

**#####################R script for Figure 10################**

rm(list = ls())

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step20 GSE45670 irgenes DEA"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

setwd(outputDir)

getwd()

#

load("Step03.02.GSE45670.exprSet.annotated.radiosensity.Rdata")

lncRNAlist<-c("LINC01121","FAM167A-AS1","ADAMTS9-AS2",

"MGC12916","MIR124-2HG","FAM167A-AS1")

lncRNAexp.diff<-exprSet.annotated.radiosensity[which(rownames(exprSet.annotated.radiosensity) %in% lncRNAlist),]

#Screening tumor samples

GSE45670.lncRNAexp.diff<-lncRNAexp.diff

#Load immune-related genes

load("0irgene.rda")

save(exprSet.annotated.radiosensity,lncRNAlist,

group\_list,irgene,

file = "Figure 10 data.rda")

GSE45670.irgene.mRNAexp<-exprSet.annotated.radiosensity[which(rownames(exprSet.annotated.radiosensity) %in% irgene$irgene),]

#GSE45670.irgene.mRNAexp<-GSE45670.irgene.mRNAexp[,GSE45670.Group=="non\_responder"]

library(limma)

#载入lncRNA表达文件,并对数据进行处理

lncRNAExpMarix<-GSE45670.lncRNAexp.diff

class(lncRNAExpMarix)

exp <-lncRNAExpMarix;dim(exp)

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

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

data=avereps(data)

dim(data)

data=data[rowMeans(data)>0.5,];dim(data);class(data)

save(data, file = "Step20.01.GSE45670.data.rda")

#

#

#Load immune gene expression files and process data

immGeneExp<-GSE45670.irgene.mRNAexp

class(immGeneExp)

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

immuneGene=matrix(as.numeric(as.matrix(immGeneExp)),

nrow=nrow(immGeneExp),dimnames=dimnames)

immuneGene=avereps(immuneGene)

immuneGene=immuneGene[rowMeans(immuneGene)>0.5,]#提取行均值大于0.5的条目

dim(immuneGene)

save(data,immuneGene, file = "Step20.02.GSE45670.data.immuneGene.rda")

#Correlation test

memory.limit(102400)

dim(immuneGene)

dim(data)

immuneGene<-immuneGene

lncRNA<-data

library(dplyr)

samplenum<-ceiling(dim(lncRNA)[1]/100)

samplenum

grp<-as.factor(rep(1:samplenum, each = 100,len =nrow(lncRNA)))

data <- cbind(grp,as.data.frame(lncRNA))

data\_split <- split(data, data$grp)

class(data\_split)

save(data\_split,file = "Step20.03.data\_split.rda")

load("Step20.03.data\_split.rda")

outTablist<-list()

for (k in 1:length(data\_split)){

tmp <- as.matrix(data\_split[[k]][,-1])

corX = pvalueX =NULL

lncRNAi = immuneGenej =NULL

outTab = NULL

for(i in row.names(tmp)){

if(sd(tmp[i,])>0.5){

for(j in row.names(immuneGene)){

x=as.numeric(tmp[i,])

y=as.numeric(immuneGene[j,])

corT=cor.test(x,y)

cor=corT$estimate

pvalue=corT$p.value

lncRNAi = c(lncRNAi,i)

immuneGenej = c(immuneGenej,j)

corX = c(corX,cor)

pvalueX= c(pvalueX,pvalue)

}

}

}

outTab = data.frame(immuneGenej,lncRNAi,corX,pvalueX)

colnames(outTab)<-c("immuneGene","lncRNA","cor","pvalue")

outTablist[[k]]<-outTab

nameoutput=paste0("Step20.04.outTab.",k,".rda", sep = "")

save(outTab, file = nameoutput)

save(outTablist, file = "Step20.05.outTablist.rda")

}

load("Step20.05.outTablist.rda")

length(outTablist)

library("plyr")

corResults.all<-do.call(rbind.fill,outTablist)

corResults.all$Regulation[corResults.all$cor>0]<-"postive"

corResults.all$Regulation[corResults.all$cor<0]<-"negative"

table(corResults.all$Regulation)

save(corResults.all,file = "Step20.06.GSE45670.corResults.all.rda")

load("Step20.06.GSE45670.corResults.all.rda")

library(limma)

corFilter.negtive=-0.4

pvalueFilter.negtive=0.01

corFilter.positive=0.4

pvalueFilter.positive=0.01

negtivedf<-subset(corResults.all, cor <= corFilter.negtive & pvalue<=pvalueFilter.negtive)

dim(negtivedf)

positivedf<-subset(corResults.all, cor >= (corFilter.positive) & pvalue<=pvalueFilter.positive)

dim(positivedf)

corResults<-rbind(negtivedf,positivedf)

save(negtivedf,positivedf,corResults,file = "Step20.07.GSE45670.corResults.rda")

GSE45670.imugene=unique(as.vector(corResults[,"immuneGene"]))

GSE45670.irGene.exp=exprSet.annotated.radiosensity[GSE45670.imugene,]

save(GSE45670.irGene.exp,GSE45670.imugene,group\_list,file = "Step20.08.GSE45670.irGene.exp.rda")

expSet<-as.data.frame(GSE45670.irGene.exp)

expSet <- expSet[rowMeans(expSet)>0,]

dim(expSet)

suppressMessages(library("stringr"))

class(group\_list)

group\_list

Group<-as.factor(ifelse(group\_list==2,0,1))#注意1才是对照组

Group <- factor(Group,labels = c("responder","non\_responder"))

table(Group)

###################=====limma DEA=====######################

suppressMessages(library(reshape2))

names.expSet<-rownames(expSet)

expSet.tmp<-cbind(names.expSet,expSet)

exp\_L = melt(expSet.tmp)

head(exp\_L)

colnames(exp\_L)=c('symbol','sample','value')

head(exp\_L)

# get group info

exp\_L$group = rep(group\_list,each=nrow(expSet))

head(exp\_L)

# ggplot2 plotting

library(ggplot2)

p = ggplot(exp\_L,

aes(x=sample,y=value,fill=group))+geom\_boxplot()

print(p)

p=ggplot(exp\_L,aes(x=sample,y=value,fill=group))+geom\_boxplot()

p=p+stat\_summary(fun="mean",geom="point",shape=23,size=3,fill="red")

p=p+theme\_set(theme\_set(theme\_bw(base\_size=20)))

p=p+theme(text=element\_text(face='bold'),axis.text.x=element\_text(angle=30,hjust=1),axis.title=element\_blank())

print(p)

#batch infect

library(limma)

expSet = normalizeBetweenArrays(expSet)

#############Check sample grouping information#################

#hclust plotting

head(expSet)

group\_list0<-group\_list[group\_list=="non\_responder"]

group\_list1<-group\_list[group\_list=="responder"]

clname<-c(paste(group\_list0,1:17,sep=''),paste(group\_list1,1:11,sep=''))

colnames(expSet) = clname

head(expSet)

#nodePar

nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),

cex = 0.7, col = "blue")

# clust

hc=hclust(dist(t(expSet)))

par(mar=c(5,5,5,10))

# plotting

plot(as.dendrogram(hc), nodePar = nodePar, horiz = TRUE)

###############PCA##################

options("repos" = c(CRAN="https://mirrors.tuna.tsinghua.edu.cn/CRAN/"))

if(! require("ggfortify")) install.packages("ggfortify")

#install.packages("ggfortify")

df=as.data.frame(t(expSet))

dim(df)

dim(expSet)

exp[1:6,1:6]

df[1:6,1:6]

df$group=group\_list

autoplot(prcomp(df[,1:(ncol(df)-1)] ), data=df,colour = 'group')

save(expSet,group\_list,file = "step20.09.changecolname.Rdata")

##################数据准备阶段###############################

suppressMessages(library(limma))

suppressMessages(library("stringr"))

####log2 #####

log2expSet=log2(expSet)

exp<-normalizeBetweenArrays(log2expSet)

par(mfrow=c(1,2))

n.sample<-ncol(log2expSet)

cols <- rainbow(n.sample\*1.2)

boxplot(data.frame(exp),col=cols,main="expression value",las=2) ## 画箱式图，比较数据分布情况

####grouping matrix

#group\_list <-as.data.frame(group\_list)

class(group\_list)

group\_list

group\_list<-as.factor(ifelse(group\_list==2,0,1))#注意1才是对照组

group\_list <- factor(group\_list,labels = c("responder","non\_responder"))

table(group\_list)

group\_list

design <- model.matrix(~0+factor(group\_list))

colnames(design)=levels(factor(group\_list))

rownames(design)=colnames(exp)

design

####3.0Difference Comparison Matrix

contrast.matrix<-makeContrasts(paste0(unique(group\_list),collapse = "-"),levels = design)

contrast.matrix

#####4.0 fit

##step1

fit <- lmFit(exp,design)

##step2

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

fit2 <- eBayes(fit2)

tempOutput = topTable(fit2, coef=1, n=Inf)

DEG.limma = na.omit(tempOutput)

head(DEG.limma)

##Select differentially expressed genes

#或者logFC\_cutoff <- with(DEG.limma,mean(abs(logFC)) + 2\*sd(abs(logFC)) )

logFC\_cutoff<-1

logFC\_cutoff

P.Value\_cutoff=0.05

k1 = (DEG.limma$P.Value < 0.05)&(DEG.limma$logFC < -logFC\_cutoff)

k2 = (DEG.limma$P.Value < 0.05)&(DEG.limma$logFC > logFC\_cutoff)

DEG.limma$change = as.factor(ifelse(k1,"DOWN",ifelse(k2,"UP","NOT")))

table(DEG.limma$change)

head(DEG.limma)

save(DEG.limma,logFC\_cutoff,P.Value\_cutoff,file = paste0("Step20.10.irgene.DEG.limma.rda"))

library("tidyverse")

DEG.limma.diff<-DEG.limma%>% filter(change %in% c("UP","DOWN"))

save(DEG.limma.diff, file = "Step20.11.DEG.limma.diff.rda")

####Output differentially expressed genes

diffirgenelist=as.vector(rownames(DEG.limma.diff))

log2.exp.normlize<-as.data.frame(exp)

diffirgene=log2.exp.normlize[diffirgenelist,]

save(diffirgene,group\_list,diffirgenelist, file = "Step20.11.diffirgene.rda")

#####################Plot histogram of differentially expressed genes########################

lncRNAlist<-diffirgenelist

group\_list<-group\_list

sample\_list<-colnames(diffirgene)

save(lncRNAlist,group\_list,sample\_list, file = "Step20.12.grp.Rda")

#选择差异的lncRNA

lncRNAexp.diff<-diffirgene

#转置表达矩阵

lncRNAexp.trans<-t(lncRNAexp.diff)

lncRNAexp.sig<-data.frame(group\_list,lncRNAexp.trans)

colnames(lncRNAexp.sig)<-gsub("\\.","-",colnames(lncRNAexp.sig))

#整理数据，使之适合面板箱型图

col<-colnames(lncRNAexp.sig)

nrow.lncRNAexp.sig<-nrow(lncRNAexp.sig)

rownames(lncRNAexp.sig)<-NULL

length(col)

exp.lncRNA<-data.frame()

#函数形式：rep(x, time = , length = , each = ,)

for (i in (2:length(col))){#

exp.temp<-NULL

gene<-rep(col[i],nrow.lncRNAexp.sig)

exp.temp<-subset(lncRNAexp.sig,select = c(1,i))

colnames(exp.temp)<-c("Group","exp")

exp.temp<-data.frame(exp.temp,gene)

exp.lncRNA<-rbind(exp.lncRNA,exp.temp)

}

save(exp.lncRNA,file = "Step20.13.exp.lncRNA.Rda")

library("dplyr")

library("ggplot2")

library("ggpubr")

dev.off()

p <- ggplot(data=exp.lncRNA,aes(x=Group,y=exp))

p+geom\_boxplot(aes(fill=Group))+facet\_wrap(~gene)+

labs(x="Group", y = "Gene expression")+

stat\_compare\_means(method = "t.test",aes(label = paste0("p =",as.character(sprintf("%0.3f", as.numeric(..p.format..))))))

ggsave(paste0("Differential expression analysis of immune-related genes",".tiff"),width = 15,height = 10)

**#####################R script for Figure 12################**

############# GO KEGG ###################

rm(list = ls())

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step21 GSE45670 irgenes GO.KEGG"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

file=paste0("Step20.11.DEG.limma.diff.rda")

filefrom<-paste0("Step20 GSE45670 irgenes DEA/",file)

fileto<-paste0("Step21 GSE45670 irgenes GO.KEGG/",file)

file.copy(filefrom,

fileto,

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE) # 文件复制

setwd(outputDir)

getwd()

#BiocManager::install("ReactomePA")

library(ReactomePA)

#if (!require("BiocManager", quietly = TRUE))

# install.packages("BiocManager")

#BiocManager::install("reactome.db")

library(tidyverse)

library(data.table)

library(org.Hs.eg.db)

library(clusterProfiler)

library(biomaRt)

library(enrichplot)

#install.packages("ggridges")

library("ggridges")

library("DO.db")

load(file)

genelist\_input <- DEG.limma.diff

genelist\_input$Gene<-rownames(genelist\_input)

class(genelist\_input$Gene)

genename <- rownames(genelist\_input)

gene\_map <- select(org.Hs.eg.db, keys=genename, keytype="SYMBOL", columns=c("ENTREZID"))

colnames(gene\_map)[1]<-"Gene"

rt<-inner\_join(gene\_map,genelist\_input,by = "Gene")

rt<-rt[,-1]

rt<-na.omit(rt)

rt$logFC<-sort(rt$logFC,decreasing = T)

geneList = rt[,2]

names(geneList) = as.character(rt[,1])

geneList

save(geneList,rt,file="Step21.01.GO.KEGG.geneList.Rda")

geneFC=rt$logFC

gene=rt$ENTREZID

names(geneFC)=gene

class(gene)

# BP enrichGO

BPenrich <- enrichGO(gene = gene,

OrgDb = org.Hs.eg.db,

ont = "BP",

pAdjustMethod = "BH",

pvalueCutoff = 1,

qvalueCutoff = 1,

readable = TRUE)

head(BPenrich,2)

summary.BP<-summary(BPenrich)

write.csv(summary(BPenrich),"Step21.02.GO.BPenrich.csv",row.names =FALSE)

save(summary.BP,file = "Step21.02.GO.BPenrich.Rda")

tiff(file="Step21.02.GO.BPenrich.dotplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

dotplot(BPenrich, showCategory = 47)

dev.off()

tiff(file="Step21.02.GO.BPenrich.barplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

barplot(BPenrich, showCategory = 20)

dev.off()

tiff(file="Step21.02.GO.BPenrich.plotGOgraph.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

plotGOgraph(BPenrich)

dev.off()

# CC enrichGO

CCenrich <- enrichGO(gene = gene,

OrgDb = org.Hs.eg.db,

ont = "CC",

pAdjustMethod = "BH",

pvalueCutoff = 1,

qvalueCutoff = 1,

readable = TRUE)

head(CCenrich,2)

summary.CC<-summary(CCenrich)

write.csv(summary(CCenrich),"Step21.03.GO.CCenrich.csv",row.names =FALSE)

save(summary.CC,file = "Step21.03.GO.CCenrich.Rda")

tiff(file="Step21.03.GO.CCenrich.dotplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

dotplot(CCenrich, showCategory = 47)

dev.off()

tiff(file="Step21.03.GO.CCenrich.barplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

barplot(CCenrich, showCategory = 20)

dev.off()

tiff(file="Step21.03.GO.CCenrich.plotGOgraph.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

plotGOgraph(CCenrich)

dev.off()

# MF enrichGO

MFenrich <- enrichGO(gene = gene,

OrgDb = org.Hs.eg.db,

ont = "MF",

pAdjustMethod = "BH",

pvalueCutoff = 1,

qvalueCutoff = 1,

readable = TRUE)

head(MFenrich,2)

summary.MF<-summary(MFenrich)

write.csv(summary(MFenrich),"Step21.04.GO.MFenrich.csv",row.names =FALSE)

save(summary.MF,file = "Step21.04.GO.MFenrich.Rda")

tiff(file="Step21.04.GO.MFenrich.dotplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

dotplot(MFenrich, showCategory = 20)

dev.off()

tiff(file="Step21.04.GO.MFenrich.barplot.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

barplot(MFenrich, showCategory = 20)

dev.off()

tiff(file="Step21.04.GO.MFenrich.plotGOgraph.tiff",width =20,height =30,units ="cm",compression="lzw",bg="white",res=300)

plotGOgraph(MFenrich)

dev.off()

gene

genename

KEGGenrich <- enrichKEGG(gene = gene,

organism = "hsa",

pvalueCutoff =1,

qvalueCutoff =1,

use\_internal\_data = T)

save(KEGGenrich, file ="Step21.05.KEGGenrich.Rda")

write.csv(summary(KEGGenrich),"Step21.05.KEGG.enrich.csv",row.names =FALSE)

#KEGG bar

tiff(file="Step21.05.KEGGenrich.barplot.tiff",width = 20,height = 20,units ="cm",compression="lzw",bg="white",res=300)

barplot(KEGGenrich, drop = TRUE, showCategory = 12)

dev.off()

#KEGG dot

tiff(file="Step21.05.KEGGenrich.dotplot.tiff",width = 20,height = 20,units ="cm",compression="lzw",bg="white",res=300)

dotplot(KEGGenrich)

dev.off()

#Merge plotting

BP.plotdata<- BPenrich@result

BP.plotdata <- BP.plotdata[BP.plotdata$p.adjust<0.05,]

library(ggplot2)

BP.bar<-ggplot(BP.plotdata, aes(x=Description, y=Count,fill=p.adjust)) +

geom\_bar(stat="identity") +

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

scale\_x\_discrete(name ="pathway names") +

scale\_y\_continuous(name ="Count") +

coord\_flip() +

theme\_bw()+

theme(plot.title = element\_text(hjust = 0.5))+

ggtitle("GO biological process enrichment ")

print(BP.bar)

CC.plotdata<- CCenrich@result

CC.plotdata <- CC.plotdata[CC.plotdata$p.adjust<0.3,]

library(ggplot2)

CC.bar<-ggplot(CC.plotdata, aes(x=Description, y=Count,fill=p.adjust)) +

geom\_bar(stat="identity") +

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

scale\_x\_discrete(name ="pathway names") +

scale\_y\_continuous(name ="Count") +

coord\_flip() +

theme\_bw()+

theme(plot.title = element\_text(hjust = 0.5))+

ggtitle("GO cellular component enrichment ")

print(CC.bar)

MF.plotdata<- MFenrich@result

MF.plotdata <- MF.plotdata[MF.plotdata$p.adjust<0.05,]

library(ggplot2)

MF.bar<-ggplot(MF.plotdata, aes(x=Description, y=Count,fill=p.adjust)) +

geom\_bar(stat="identity") +

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

scale\_x\_discrete(name ="pathway names") +

scale\_y\_continuous(name ="Count") +

coord\_flip() +

theme\_bw()+

theme(plot.title = element\_text(hjust = 0.5))+

ggtitle("GO molecular function enrichment ")

print(MF.bar)

KEGG.plotdata<- KEGGenrich@result

KEGG.plotdata <- KEGG.plotdata[KEGG.plotdata$pvalue<0.05,]

library(ggplot2)

KEGG.bar<-ggplot(KEGG.plotdata, aes(x=Description, y=Count,fill=pvalue)) +

geom\_bar(stat="identity") +

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

scale\_x\_discrete(name ="pathway names") +

scale\_y\_continuous(name ="Count") +

coord\_flip() +

theme\_bw()+

theme(plot.title = element\_text(hjust = 0.5))+

ggtitle("KEGG Pathway Enrichment")

print(KEGG.bar)

library(cowplot)

p3<-plot\_grid(BP.bar, CC.bar, MF.bar, KEGG.bar,labels = "AUTO")

class(p3)

ggsave(filename = "Step21.06.Pathway Enrichment.png",p3,width =30,

height = 10, dpi = 300, units = "in", device='png')

ggsave(filename = "Step21.06.Pathway Enrichment.tiff",p3,width =30,

height = 10, dpi = 300, units = "in", device='tiff')

save(BPenrich,CCenrich,MFenrich,KEGGenrich,file = "Step21.06.Pathway Enrichment.Rda")

#pathview

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("pathview")

library("pathview")

keggxls=KEGG\_gseresult

for(i in keggxls$ID){

pv.out <- pathview(gene.data = geneFC, pathway.id = i, species = "hsa", out.suffix = "pathview")

}

**#####################R script for Figure S1 and S2################**

rm(list = ls())

memory.limit(102400)

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step11.model"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

#copy files

file.copy("Step10.uniCox/Step10.04.uniCoxdf.select.rda", "Step11.model/Step10.04.uniCoxdf.select.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

file.copy("Step10.uniCox/Step10.05.surSigExp.rda", "Step11.model/Step10.05.surSigExp.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

file.copy("Step09.mergeTime/Step09.01.survdata.expselect.cliselect.rda",

"Step11.model/Step09.01.survdata.expselect.cliselect.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

setwd(outputDir)

getwd()

library(survival)

library(survminer)

library(glmnet)

#load files

load("Step10.05.surSigExp.rda")

class(surSigExp)

class(surSigExp$fustat)

surSigExp$fustat<-as.numeric(surSigExp$fustat)-1

library(glmnet)

library(rms)

library(VIM)

library(survival)

#load dataset

dt <-surSigExp

str(dt)

aggr(dt,prop=T,numbers=T)

dt <- na.omit(dt)

###################step 01 Data curation######################

for(i in names(dt)[c(3:ncol(dt))]) {dt[,i] <- as.factor(dt[,i])}

str(dt)

x <- data.matrix(dt[,c(3:ncol(dt))])

y <- data.matrix(Surv(dt$futime,dt$fustat))

#################step 02 Lasso Regression Analysis##########################

fit.cox <-glmnet(x,y,family = "cox",alpha = 1, nlambda = 50)

dev.new()

tiff("Step11.01.lasso.lambda.tiff")

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

dev.off()

print(fit.cox)

fit.cv <- cv.glmnet(x,y,family="cox", alpha=1,nfolds=10)

#save picture of LASSO lamda

tiff("Step11.02.lasso.cvfit.tiff")

plot(fit.cv)

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

dev.off()

print(fit.cv)

coef <-coef(fit.cox, s=c(fit.cox$lambda[6],0.1))

index <- which(coef != 0)

actCoef <- coef[index]

lassoGene=row.names(coef)[index]

lassoGene <- na.omit(lassoGene)

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

lassoSigExp=surSigExp[,lassoGene]

#save data

save(lassoSigExp,file = "Step11.03.lassoSigExp.rda")

##############Step 3: Built a Cox model#################

load("Step11.03.lassoSigExp.rda")

dim(lassoSigExp)

datainput<-lassoSigExp[,c(1:ncol(lassoSigExp))]

multiCox=coxph(Surv(futime, fustat) ~ ., data = datainput,x=T,y=T)

multiCox=step(multiCox, direction="both")

multiCoxSum=summary(multiCox)

save(multiCox,multiCoxSum,file = "Step11.04.multiCox.rda")

#output pm for the model

resultdf.cox=data.frame()

resultdf.cox=cbind(b=multiCoxSum$coefficients[,"coef"],

se=multiCoxSum$coefficients[,"se(coef)"],

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|)"])

row.names(resultdf.cox)=gsub("`","",row.names(resultdf.cox))

#savedata

#save(resultdf.cox,file = "Step11.04.resultdf.cox.rda")

saveRDS(resultdf.cox,file = "Step11.05.resultdf.cox.rds")

write.csv(resultdf.cox,file = "Step11.05.resultdf.cox.csv")

#rm(list = ls())

load("Step11.03.lassoSigExp.rda")

datainput<-lassoSigExp

load(file = "Step11.04.multiCox.rda")

riskScore=predict(multiCox, type="risk", newdata=datainput)

multiCoxSum=summary(multiCox)

coxGene=rownames(multiCoxSum$coefficients)

coxGene=gsub("`", "", coxGene)

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

riskOut=cbind(datainput[,outCol], riskScore)

#save data

save(riskOut,file = "Step11.07.riskOut.rda")

load("Step10.04.uniCoxdf.select.rda")

uniCoxdata<-uniCoxdf.select

rownames(uniCoxdata) <-uniCoxdata[,1]

uniRT<-uniCoxdata[,-1]

class(uniRT[,1])

df.unicox<-as.data.frame(lapply(uniRT,as.numeric))

rownames(df.unicox)<-rownames(uniCoxdata)

resultdf.unicox=df.unicox[coxGene,]

class(resultdf.unicox)

saveRDS(resultdf.unicox,file = "Step11.08.resultdf.unicox.rds")

**#####################R script for Figure S3################**

#Clinical characteristics and riskscore correlation analysis

rm(list = ls())

memory.limit(102400)

#Create a folder, set the working directory

mainDir <-"C:/Bioinformatics/000 EPC rt/GSE53625"

subDir <-"Step17.cliCor"

outputDir <- file.path(mainDir, subDir)

if (!dir.exists(outputDir)){

dir.create(outputDir)

} else {

print("Dir already exists!")

}

setwd(mainDir)

#copy files

file.copy("Step12.ROC/Step12.03.risk.df.rda",

"Step17.cliCor/Step12.03.risk.df.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

file.copy("Step14.indep/Step14.01.cli.factors.rda",

"Step17.cliCor/Step14.01.cli.factors.rda",

overwrite = TRUE, recursive = FALSE,

copy.mode = TRUE, copy.date = FALSE)

setwd(outputDir)

getwd()

#if (!requireNamespace("BiocManager", quietly = TRUE))

# install.packages("BiocManager")

#BiocManager::install("limma")

#install.packages("ggpubr")

#引用包

library(limma)

library(ggpubr)

#load risk files

load("Step12.03.risk.df.rda")

risk=risk.df[,c("futime","fustat","riskScore")]

#load clinical data files

load("Step14.01.cli.factors.rda")

cli<-cli.factors

#Consolidate data

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

risk=risk[samSample,"riskScore",drop=F]

cli=cli[samSample,,drop=F]

rt=cbind(risk, cli)

library(tidyverse)

boxplot<-list()

m<-0

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

m=m+1

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

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

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

data<-data %>% drop\_na()

#set comparison group

group=levels(factor(data$clinical))

data$clinical=factor(data$clinical, levels=group)

comp=combn(group,2)

my\_comparisons=list()

for(i in 1:ncol(comp)){my\_comparisons[[i]]<-comp[,i]}

#draw a boxplot

boxplot[[m]]=ggboxplot(data, x="clinical", y="riskScore", color="clinical",

xlab=clinical,

ylab="Risk score",

legend.title=clinical,

add = "jitter")+

stat\_compare\_means(comparisons = my\_comparisons)

}

p3<-plot\_grid(boxplot[[1]], boxplot[[2]],

boxplot[[3]], boxplot[[4]],

boxplot[[5]], boxplot[[6]],

boxplot[[7]], boxplot[[8]],

boxplot[[9]],

labels = "AUTO")

class(p3)

ggsave(filename = "clinical.riskScore.png",p3,width =20,

height = 15, dpi = 300, units = "in", device='png')