dat = data.table::fread("gtex\_RSEM\_gene\_fpkm.gz",data.table = F)

library(tidyverse)

exp = column\_to\_rownames(dat,"sample") %>% as.matrix()

rownames(exp)=rownames(exp)%>%str\_split("\\.",simplify=T)%>%.[,1]

library(AnnoProbe)

library(tinyarray)

an=annoGene(rownames(exp),ID\_type="ENSEMBL")

exp = trans\_array(exp,ids = an,from ="ENSEMBL",to = "SYMBOL")

data\_cl=data.table::fread("GTEX\_phenotype.gz",header = T,sep ='\t',data.table = F)

data\_cl=data\_cl[,c(1,3)]

names(data\_cl)=c('Barcode','Tissue')

data\_cl=data\_cl[data\_cl$Tissue =='Brain',]

data\_cl=data\_cl[data\_cl$Barcode%in%colnames(dat),]

gtex=exp[,data\_cl$Barcode]

rm(an,dat,data\_cl1,exp)

y = read.table("TCGA-GBMLGG.htseq\_fpkm.tsv.gz",

header = T,

check.names = F,

row.names=1)

y[1:4,1:4]

group4 = make\_tcga\_group(y);table(group4)

y = trans\_exp(y)

y = y[rowSums(y)>0,]

y[1:4,1:4]

yz1=which(grepl("01",str\_sub(colnames(y),14,15)))

yz2=which(grepl("06",str\_sub(colnames(y),14,15)))

yz=c(yz1,yz2)

yzz=y[,yz]

yc=which(grepl("11",str\_sub(colnames(y),14,15)))

ycc=y[,yc]

y=cbind(yzz,ycc)

a=rownames(gtex)[rownames(gtex)%in%rownames(y)]

gtex=gtex[a,]

exp=cbind(y,gtex)

Figure1

library("msigdbr")

m\_df = msigdbr(species = "Homo sapiens", category = "C2")

a=which(grepl("WP\_IL18\_SIGNALING\_PATHWAY",

m\_df$gs\_name))

m1=m\_df[a,]

IL18gene=m1$gene\_symbol

IL18gene=unique(IL18gene)[table(IL18gene) == 1]

cl = rio::import("TCGA-GBMLGG.survival.tsv")

colnames(cl)[c(2,4)] = c("event","time")

colnames(cl)[2]="event"

colnames(cl)[4]="time"

rownames(cl)=cl$sample

cl=cl[colnames(exp),]

table(rownames(cl)==colnames(exp))

cl=cl[,c(1:4)]

ph = readr::read\_tsv("TCGA-GBMLGG.GDC\_phenotype.tsv.gz")

ph = ph[,c("submitter\_id.samples",

"age\_at\_initial\_pathologic\_diagnosis",)]

colnames(ph) = c("sample\_id","age")

cl= merge(cl,ph,by.x = "sample",by.y = "sample\_id")

exp=exp[,cl$sample]

library(tinyarray)

ec = match\_exp\_cl(exp[IL18gene,group == "tumor"],

cl,

id\_column = "\_PATIENT")

cl\_matched = ec$cl\_matched

exp\_matched = ec$exp\_matched

x=t(exp\_matched)

library(survival)

y = data.matrix(Surv(cl\_matched$time,cl\_matched$event))

library(glmnet)

set.seed(1000)

cvfit = cv.glmnet(x,

Surv(cl\_matched$time,cl\_matched$event),

nfold=10, family = "cox")

plot(cvfit)

cl\_matched$riskscore = as.numeric(predict(cvfit,newx = x,s = cvfit$lambda.min))

cvfit = cv.glmnet(x, yy, family="cox")

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

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

library(plotmo)

plot\_glmnet(fit)

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

actCoef = coef[coef[,1] != 0,]

lassoGene = row.names(coef)[coef[,1] != 0]

expL=exp[lassoGene,]

expL=cbind(cl,expL)

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

unicox <- coxph(Surv(time = os\_time, event = os\_status) ~ df[,i], data = df)

unisum<- summary(unicox)

pvalue <- round(unisum$coefficients[,5],3)

if(pvalue<pfilter){

uniresult <- rbind(uniresult,

cbind(gene=i,

HR=unisum$coefficients[,2],

L95CI=unisum$conf.int[,3],

H95CI=unisum$conf.int[,4],

pvalue=unisum$coefficients[,5]))}}

Figure2

library(corrplot)

M<-cor(t(expL))

library(paletteer)

my\_color = rev(paletteer\_d("RColorBrewer::RdYlBu"))

my\_color = colorRampPalette(my\_color)(10)

corrplot(M, type="upper",

method="pie",

order="hclust",

col=my\_color,

tl.col="black",

#tl.pos = "d",

tl.srt=45)

library(patchwork)

exp\_matched=t(expL)

cl\_matched=expL[,c(1,2)]

library(tinyarray)

km\_plots = exp\_surv(exp\_matched,cl\_matched)

wrap\_plots(km\_plots,nrow = 2)

Figure4

exp1=data.frame(t(y))

expn=exp1[order(exp1$"PTX3"),]

PTX3=expn[,"PTX3"]

group = c(rep("low",times=351),

rep("high",times=351))

group = factor(group,

levels = c("low","high"))

expn=t(expn)

library(limma)

design=model.matrix(~group)

fit=lmFit(expn,design)

fit=eBayes(fit)

deg=topTable(fit,coef=2,number = Inf)

library(dplyr)

deg <- mutate(deg,probe\_id=rownames(deg))

head(deg)

logFC\_t=2

P.Value\_t = 0.05

k1 = (deg$P.Value < P.Value\_t)&(deg$logFC < -logFC\_t)

k2 = (deg$P.Value < P.Value\_t)&(deg$logFC > logFC\_t)

deg <- mutate(deg,change = ifelse(k1,"down",ifelse(k2,"up","stable")))

table(deg$change)

rownames(deg)=deg$probe\_id

deg1=which(grepl("down",deg$change))

deg2=which(grepl("up",deg$change))

deg11=deg$probe\_id[deg1]

deg22=deg$probe\_id[deg2]

deggenes=c(deg11,deg22)

write.table(deggenes,

file="diffgene.txt",

row.names = F,

col.names = F,

quote = F)

p = deg[deg$change != "stable",

c("symbol","logFC")]

write.table(p,

file = "deg.txt",

sep = "\t",

quote = F,

row.names = F)

Figure5

library(clusterProfiler)

library(org.Hs.eg.db)

ego <- enrichGO(gene = deggenes,

OrgDb = org.Hs.eg.db,

pvalueCutoff =0.05,

qvalueCutoff =0.05,

readable = TRUE)

head(ego)

write.table(ego,file="GO\_result.txt",sep="\t",

quote=F,row.names = F)

kk <- enrichKEGG(gene = deggenes,

organism = "hsa",

pvalueCutoff =0.05,

qvalueCutoff =1)

head(kk)

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

Figure6

library(org.Hs.eg.db)

library(clusterProfiler)

library(enrichplot)

genelist\_input <-deg[,c(1,6)]

genename <- as.character(genelist\_input[,1])

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

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

write.csv(as.data.frame(gene\_map)

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

aaa<-aaa[,-1]

aaa<-na.omit(aaa)

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

geneList = aaa[,2]

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

geneList

Go\_gseresult <- gseGO(geneList, 'org.Hs.eg.db', keyType = "ENTREZID", ont="all", nPerm = 1000, minGSSize = 10, maxGSSize = 1000, pvalueCutoff=1)

KEGG\_gseresult <- gseKEGG(geneList, nPerm = 1000, minGSSize = 10, maxGSSize = 1000, pvalueCutoff=1)

Go\_Reactomeresult <- gsePathway(geneList, nPerm = 1000, minGSSize = 10, maxGSSize = 1000, pvalueCutoff=1)

gseaplot2(Go\_Reactomeresult,212,pvalue\_table = TRUE)

Figure7

library(GSVA)

gene\_set<-read.csv("mmc3.csv",header=T,stringsAsFactors = FALSE)

list<- split(as.matrix(gene\_set)[,1], gene\_set[,2])

gsva\_matrix<-gsva(as.matrix(expn), list,method='ssgsea',kcdf='Gaussian',abs.ranking=TRUE)

row.names(group)=group$sample

group=group[colnames(expn),]

expn=as.data.frame(t(expn))

expn$cluster=group$Group

expn=expn[,c(29,1:28)]

test=as.list(expn[,-1])

height<-stack(test)

Group=rep(expn$cluster,ncol(expn)-1)

data1=as.data.frame(cbind(Group,height))

library(ggplot2)

library(ggpubr)

library(ggsignif)

colnames(data1)=c("group","Cell\_infiltration\_level","cell")

df=data1

p=ggplot(data=df,aes(x=cell,y=Cell\_infiltration\_level,fill=group))+

geom\_boxplot(width=0.3,

position = position\_dodge(0.5),outlier.colour =NA)+

theme\_bw()+

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

scale\_fill\_manual(values = c("#4DBBD5FF","#E64B35FF","#E18727FF"))+

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

label="p.signif",

hide.ns = TRUE,

label.y = c(0.6))+

theme(axis.text.x = element\_text(size=12,colour="black",angle=60,hjust = 0.9),axis.text.y = element\_text(size=12, angle=0),

axis.title.x = element\_text(size=15),axis.title.y = element\_text(size=15))

Figure8

exp<-read.table("GSE50161\_series\_matrix.txt.gz",sep="\t",quote = "",

fill=T,comment.char = "!",header=T)

rownames(exp)<-exp[,1]

exp<-exp[,-1]

expn=exp[,103:115]

expt=exp[,47:80]

exp=cbind(expt,expn)

Group = c(rep("Tumour",times=34),

rep("Normal",times=13))

Group = factor(Group,

levels = c("Normal","Tumour"))

table(Group)

library(tinyarray)

find\_anno('GPL570')

ids <- AnnoProbe::idmap('GPL570')

library(dplyr)

library(stringr)

exp=as.data.frame(exp)

rownames(exp)=substr(rownames(exp),2,nchar(rownames(exp)))

rownames(exp)=str\_sub(rownames(exp),1,nchar(rownames(exp))-1)

exp$probe\_id=rownames(exp)

table(exp$probe\_id%in%ids$probe\_id)

exp2= merge(exp,ids,by.x="probe\_id", by.y="probe\_id")

exp2=exp2[!duplicated(exp2$symbol),]

rownames(exp2)=exp2$symbol

exp=exp2

which(grepl("PTX3",exp$symbol))

expPTX3=exp[5342,]

path<-file.path("D:","ff.xls")

library(xlsx)

write.xlsx(expPTX3,file = path)

exp<-read.table("GSE7696\_series\_matrix.txt.gz",sep="\t",quote = "",

fill=T,comment.char = "!",header=T)

rownames(exp)<-exp[,1]

exp<-exp[,-1]

which(grepl("206157\_at",rownames(exp)))

expPTX3=exp[15604,]

path<-file.path("D:","ff.xls")

library(xlsx)

write.xlsx(expPTX3,file = path)

library(dplyr)

library(ggplot2)

dat = deg[!duplicated(deg$probe\_id),]

p <- ggplot(data = dat,

aes(x = logFC,

y = -log10(P.Value))) +

geom\_point(alpha=0.4, size=3.5,

aes(color=change)) +

ylab("-log10(Pvalue)")+

scale\_color\_manual(values=c("blue", "grey","red"))+

geom\_vline(xintercept=c(-logFC\_t,logFC\_t),lty=4,col="black",lwd=0.8) +

geom\_hline(yintercept = -log10(P.Value\_t),lty=4,col="black",lwd=0.8) +

theme\_bw()

p

for\_label <- dat%>%

filter(probe\_id %in% c(“PTX3"))

volcano\_plot <- p +

geom\_point(size = 3, shape = 1, data = for\_label) +

ggrepel::geom\_label\_repel(

aes(label = probe\_id),

data = for\_label,

color="black"

)

volcano\_plot

Figure11

library(regplot)

library(survival)

mod <- coxph(formula = s,data=cl\_matched)

regplot(mod,

observation=cl\_matched[1,],

obscol = "#326db1",

failtime = c(1095,1825),

plots = c("no plot","no plot"),

points = T,

prfail = T)

library(rms)

options(datadist=datadist(dat))

f3 <- cph(formula = s,

data=dat,x=T,y=T,surv = T, time.inc=1095)

cal3 <- calibrate(f3, cmethod="KM", method="boot", u=1095, m=50, B=1000)

f5 <- cph(formula = s,

data=dat,x=T,y=T,surv = T, time.inc=1825)

cal5 <- calibrate(f5, cmethod="KM", method="boot", u=1825, m=50, B=1000)

png("plot/cal.png")

plot(cal3,lwd = 2,lty = 0,errbar.col = c("#2166AC"),

bty = "l",

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

xlab = "Nomogram-prediced OS (%)",ylab = "Observed OS (%)",

col = c("#2166AC"),

cex.lab=1.2,cex.axis=1, cex.main=1.2, cex.sub=0.6)

lines(cal3[,c('mean.predicted',"KM")],

type = 'b', lwd = 1, col = c("#2166AC"), pch = 16)

mtext("")

plot(cal5,lwd = 2,lty = 0,errbar.col = c("#B2182B"),

xlim = c(0,1),ylim= c(0,1),col = c("#B2182B"),add = T)

lines(cal5[,c('mean.predicted',"KM")],

type = 'b', lwd = 1, col = c("#B2182B"), pch = 16)

abline(0,1, lwd = 2, lty = 3, col = c("#224444"))

legend("topleft",

legend = c("1-year","3-year","5-year"),

col =c("#2166AC","#B2182B"),

lwd = 2,

cex = 1.2,

bty = "n")