**The process of screening robust lncRNA is as follows:**

library(survival)

library(glmnet)

library(pbapply)

library(survivalROC)

Sys.setenv(LANGUAGE = "en")

options(stringsAsFactors = FALSE)

display.progress = function (index, totalN, breakN=20) {

if ( index %% ceiling(totalN/breakN) ==0 ) {

cat(paste(round(index\*100/totalN), "% ", sep=""))

}

}

surv\_lasso <- function(iter.times = NULL, surv.obj = NULL, expr.obj = NULL, nfolds = 10, alpha = 1, family = "cox") {

cvfit = cv.glmnet(x = t(as.matrix(expr.obj)),

y = surv.obj,

nfolds = nfolds,

alpha = alpha,

family = family)

myCoefs <- coef(cvfit, s="lambda.min");

lasso\_fea <- myCoefs@Dimnames[[1]][which(myCoefs != 0 )]

return(lasso\_fea)

}

expr <- read.csv("easy\_input\_expr.csv", check.names = F, row.names = 1)

expr[1:3,1:3]

Sinfo <- read.csv("easy\_input\_cli.csv", check.names = F, row.names = 1)

Sinfo[1:3,1:3]

var.cutoff <- 0.2

var <- apply(expr, 1, sd)

expr.filtered <- expr[var > var.cutoff,]

cox.cutoff <- 0.5

Coxoutput.OS <- NULL

for (i in 1:nrow(expr.filtered)) {

display.progress(index = i,totalN = nrow(expr.filtered))

tmp <- data.frame(gene = as.numeric(expr.filtered[i,]),

OS.time = Sinfo[,"OS.time"],

OS = Sinfo[,"OS"],

stringsAsFactors = F)

cox <- coxph(Surv(OS.time, OS) ~ gene, data = tmp)

coxSummary = summary(cox)

Coxoutput.OS=rbind.data.frame(Coxoutput.OS,data.frame(gene=rownames(expr.filtered)[i],

HR=as.numeric(coxSummary$coefficients[,"exp(coef)"]),

z=as.numeric(coxSummary$coefficients[,"z"]),

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

lower=as.numeric(coxSummary$conf.int[,3]),

upper=as.numeric(coxSummary$conf.int[,4]),

stringsAsFactors = F),

stringsAsFactors = F)

}

if(!identical(rownames(expr.filtered),Coxoutput.OS$gene)) {stop("Error! Gene name is mismatched!\n")}

surv.expr <- expr.filtered[Coxoutput.OS$pvalue < cox.cutoff,]

dim(surv.expr)

iter.times <- 1000

lasso\_fea <- list()

surv <- Surv(Sinfo$OS.time, Sinfo$OS)

set.seed(111)

lasso\_fea <- pblapply(1:iter.times,

surv\_lasso,

surv.obj = surv,

expr.obj = surv.expr)

genes <- sort(table(unlist(lasso\_fea)), decreasing = T)

freq.cutoff <- 500

genes <- names(genes[genes > freq.cutoff])

write.table(genes, "Robustgene.txt", row.names = F, quote = F)

**Signature construction**

pred.time <- 5

roc <- list()

auc <- c()

for (i in 1:length(genes)) {

gene <- genes[i]

tmp <- data.frame(gene = as.numeric(surv.expr[gene,]),row.names = colnames(surv.expr),stringsAsFactors = F); colnames(tmp) = gene

if(i == 1) {

surv.dat <- cbind.data.frame(Sinfo[rownames(tmp),c("OS.time","OS")],tmp)

} else {

surv.dat <- cbind.data.frame(surv.dat,tmp)

}

cox <- coxph(Surv(OS.time, OS) ~ ., data = surv.dat)

riskScore <- predict(cox,type="risk",newdata=surv.dat)

roc[[i]] <- survivalROC(Stime=surv.dat$OS.time,

status=surv.dat$OS,

marker = riskScore[rownames(surv.dat)],

predict.time =pred.time\*365,

method="KM")

auc <- c(auc,roc[[i]]$AUC)

}

prog.sig <- genes[1:which.max(auc)]

write.table(prog.sig,"signature\_gene.txt", row.names = F, quote = F)