TCGA-SARC Project: RNA-seq dataset

# Clinical information

library(stringr)

library(survminer)

library(openxlsx)

#Obtain clinical information

clin <- read.xlsx('TCGA-SARC\_clinical.xlsx')

#Survival outcome

clin$status=as.numeric(ifelse(clin$vital\_status=='Alive','0','1'))

clin$time=as.numeric(ifelse(clin$vital\_status=='Alive',clin$"days\_to\_last\_followup",clin$days\_to\_death))

clin=clin[which(clin$time>0),] #Keep the patients whose survival time > 0 days

#Gender

clin$Gender=str\_to\_title(clin$gender)

table(clin$Gender)

clin$Gender=factor(clin$Gender,levels=c('Female','Male'))

#Age

clin$Age\_con=as.numeric(clin$age\_at\_initial\_pathologic\_diagnosis)

res.cut <- surv\_cutpoint(

clin,

time = "time",

event = "status",

variables='Age\_con')

summary(res.cut)#obtain cutoff reference

#cutpoint statistic

#Age\_con 68 4.442007

clin$Age=ifelse(clin$Age\_con < res.cut[["cutpoint"]][["cutpoint"]],'< 65','≥ 65')

clin$Age=factor(clin$Age, levels=c('< 65','≥ 65'))

#Race

clin$Race=ifelse(clin$race=='WHITE','White','Black, Asian or unknown')

clin$Race =factor(clin$Race,levels=c('White','Black, Asian or unknown'))

#Metastatic status

clin$Metastatic=ifelse(clin$metastatic\_diagnosis=='YES','Yes','No or unknown')

clin$Metastatic=factor(clin$Metastatic,levels=c('No or unknown','Yes'))

#Histological type

table(clin$histological\_type,useNA = 'always')

clin$Histological\_type[grepl('liposarcoma',clin$histological\_type)]<- c('DLP')

clin$Histological\_type[grepl('LMS',clin$histological\_type)]<- c('LMS')

clin$Histological\_type[grepl('Undifferentiated pleomorphic sarcoma',clin$histological\_type,ignore.case=T)]<- c('UPS')

clin$Histological\_type[grepl('Myxofibrosarcoma',clin$histological\_type)] <- c('MYX')

clin$Histological\_type=ifelse(is.na(clin$Histological\_type)=='TRUE', 'Other', clin$Histological\_type)

clin$Histological\_type=factor(clin$Histological\_type,levels=c('LMS','DLP','UPS', 'MYX', 'Other'))

table(clin$Histological\_type,useNA = 'always')

#Tumour site

table(clin$icd\_o\_3\_site)

clin$Tumour\_site[grepl('48', clin$icd\_o\_3\_site)]<- c('C48')

clin$Tumour\_site[grepl('49', clin$icd\_o\_3\_site)]<- c('C49')

clin$Tumour\_site=ifelse(is.na(clin$Tumour\_site)=='TRUE', 'Other', clin$Tumour\_site)

clin$Tumour\_site=factor(clin$Tumour\_site,levels=c('C48','C49', 'Other'))

table(clin$Tumour\_site,useNA = 'always')

#WHO guideline p 43

# C48 retroperitoneum and peritoneum # C49 connective, subcutaneous and other soft tissues (includes adipose tissue, aponeuroses, artery, blood vessel, bursa, connective tissue, fascia, fatty tissue, fibrous tissue, ligament, lymphatic, muscle, skeletal muscle, subcutaneous tissue, synovia, tendon, tendon sheath, vein, vessel)

#Radiation therapy

table(clin$radiation\_therapy, useNA='always')

clin$Radiation\_therapy=ifelse(clin$radiation\_therapy=='YES','Yes', 'No or unknown')

clin$Radiation\_therapy=factor(clin$Radiation\_therapy , levels=c('No or unknown', 'Yes'))

#Residual tumour

table(clin$residual\_tumor,useNA = 'always')

clin$R\_classification=ifelse(clin$residual\_tumor=='R1'| clin$residual\_tumor=='R2', 'R1/R2', 'R0/RX or unknown')

table(clin$R\_classification,useNA = 'always')

clin$R\_classification=factor(clin$R\_classification,levels=c('R0/RX or unknown', 'R1/R2'))

#Tumour depth

clin$Tumour\_depth=ifelse(clin$tumor\_depth=='Deep','Deep','Superficial or unknown')

clin$Tumour\_depth=factor(clin$Tumour\_depth,levels=c('Superficial or unknown','Deep'))

#Tumour\_multifocal

clin$Tumour\_multifocal=ifelse(clin$tumor\_multifocal=='YES','Yes','No or unknown')

table(clin$Tumour\_multifocal)

clin$Tumour\_multifocal=factor(clin$Tumour\_multifocal,levels=c('No or unknown','Yes'))

#clinical information dataset

clinical=dplyr::select(clin, bcr\_patient\_barcode,time, status, Age\_con, Age, Gender, Race,

Metastatic, Histological\_type, Tumour\_site,Tumour\_depth,

Tumour\_multifocal, Radiation\_therapy, R\_classification)

clinical=na.omit(clinical)

save(clin, clinical, file='clinical.Rdata')

# Expression matrix

## Count read matrix

library(dplyr)

load("exprSet\_by\_group.Rdata")

load("clinical.Rdata")

exp=as.data.frame(t(Tumour\_sample))

sample= substring(rownames(exp),1,15)

exp=cbind(sample,exp)

#Onlly keep 01A if one patient have multiple samples

exp=exp[order(exp$sample,decreasing = F),]

table(duplicated(exp$sample)) #No duplicated samples

index <- duplicated(exp$sample)

exp= exp[!index,]

table(duplicated(exp$sample)) #No duplicated samples

exp$sample= substring(exp$sample,1,12)

table(duplicated(exp$sample)) #4 duplicated samples

#FALSE TRUE

# 259 4

index <- duplicated(exp$sample)

exp= exp[!index,]

table(duplicated(exp$sample)) #No duplicated samples

clin\_expr <- clinical %>%inner\_join(exp,by = c("bcr\_patient\_barcode" = "sample")) #match clinical information with expression matrix

rownames(clin\_expr)= clin\_expr$bcr\_patient\_barcode

clin\_expr= clin\_expr[,-1]

clin\_expr$status=as.numeric(clin\_expr$status)

clin\_expr$time=as.numeric(clin\_expr$time)

clin\_expr$time= clin\_expr$time/365.25

status= clin\_expr[,1:2]#survival time and event

expr= clin\_expr[,-1:-(ncol(clinical)-1)]

clinical= clin\_expr[,1:(ncol(clinical)-1)]

expr[,1:ncol(expr)] <- lapply(expr[,1:ncol(expr)], function(x) as.numeric(as.character(x)))

str(expr)

Tumour=as.data.frame(t(expr))

#counts = Tumour[apply(Tumour, 1, function(x) sum(x > 1) > 24), ]#Keep gene which expressed in 25 or more patients

counts=Tumour

names=rownames(counts)#The TPM matrix will keep the same gene

clin=clin[match(rownames(clinical),clin$bcr\_patient\_barcode),]

write.xlsx(clin, 'clinical\_origin.xlsx')

write.xlsx(clinical, 'clinical.xlsx',rowNames=TRUE)

library(gtsummary)

table=clinical[,-1:-2] %>% tbl\_summary( statistic = list(all\_continuous() ~ "{mean} ± {sd}",

all\_categorical() ~ "{n} ({p}%)"),

# 控制小数点

digits =list(all\_continuous() ~ 2,

all\_categorical() ~ c(0,2)))

table

library(flextable)

# 转换成flextable

mytable <- as\_flex\_table(table)

# 导出word

save\_as\_docx(mytable, path = 'mytable.docx')

save(clinical, clin, status, counts,expr,names, file='counts.Rdata')

## TPM matrix

library(dplyr)

load('counts.Rdata')

load("exprSet\_by\_group\_tpm.Rdata")

load("clinical.Rdata")

exp=as.data.frame(t(Tumour\_sample))

sample= substring(rownames(exp),1,15)

exp=cbind(sample,exp)

exp=as.data.frame(t(Tumour\_sample))

sample= substring(rownames(exp),1,15)

exp=cbind(sample,exp)

#Onlly keep 01A if one patient have multiple samples

exp=exp[order(exp$sample,decreasing = F),]

table(duplicated(exp$sample)) #No duplicated samples

index <- duplicated(exp$sample)

exp= exp[!index,]

table(duplicated(exp$sample)) #No duplicated samples

exp$sample= substring(exp$sample,1,12)

table(duplicated(exp$sample)) #4 duplicated samples

#FALSE TRUE

# 259 4

index <- duplicated(exp$sample)

exp= exp[!index,]

table(duplicated(exp$sample)) #No duplicated samples

clin\_expr <- clinical %>%inner\_join(exp,by = c("bcr\_patient\_barcode" = "sample")) #match clinical information with expression matrix

rownames(clin\_expr)= clin\_expr$bcr\_patient\_barcode

clin\_expr= clin\_expr[,-1]

clin\_expr$status=as.numeric(clin\_expr$status)

clin\_expr$time=as.numeric(clin\_expr$time)

clin\_expr$time= clin\_expr$time/365.25

status= clin\_expr[,1:2]#survival time and event

expr= clin\_expr[,-1:-(ncol(clinical)-1)]

clinical= clin\_expr[,1:(ncol(clinical)-1)]

expr[,1:ncol(expr)] <- lapply(expr[,1:ncol(expr)], function(x) as.numeric(as.character(x)))

str(expr)

tpm=expr[,names]

#TPM matrix for CIBERSORT

dat=as.data.frame(t(tpm))

symbol=rownames(dat)

out=cbind(symbol, dat)

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

log2tpm= as.data.frame(t(log2(expr +1) ))

write.csv(log2tpm, file = "log2tpm.csv", row.names = TRUE)

#log transformation

norm\_log<-matrix(data = NA, nrow =nrow(dat), ncol = ncol(dat), byrow = TRUE, dimnames = NULL)

for (i in 1: nrow(dat)){

for (j in 1: ncol(dat)){

norm\_log[i,j]<-log(dat[i,j]+1)/log(2)

}

}

expr= as.data.frame(t(limma::normalizeBetweenArrays(norm\_log,method= "quantile")))

colnames(expr)=colnames(tpm)

rownames(expr)=rownames(expr)

clin=clin[match(rownames(clinical),clin$bcr\_patient\_barcode),]

library(openxlsx)

write.xlsx(clin, 'clinical\_origin.xlsx')

write.xlsx(clinical, 'clinical.xlsx',rowNames=TRUE)

save(clinical, clin, status, tpm, log2tpm,expr,names, file='tpm.Rdata')

# Selection of Hub Gene

library(openxlsx)

load("tpm.Rdata")

gene=read.xlsx('gene1.xlsx') #symbol of 105 OS-related IRGs

sameGene=intersect(as.vector(gene[,1]), colnames(expr))

setdiff(as.vector(gene[,1]), colnames(expr))

exp=expr[,sameGene] #expression matrix of 105 OS-related IRGs

colnames(exp)=gsub('-','\_',colnames(exp))

unicox=cbind(status,exp)

covariates <- colnames(unicox)[3:ncol(unicox)]

univ\_formulas <- sapply(covariates,

function(x) as.formula(paste('Surv(time, status)~', x)))

univ\_models <- lapply( univ\_formulas, function(x){coxph(x, data = unicox)})

univ\_results <- lapply(univ\_models,

function(x){

cox <- summary(x)

#获取p值

p.value<- cox$coefficients[,5]

#获取HR

HR <- sprintf("%.03f", cox$coefficients[,2])

#获取95%置信区间

CI\_L<- sprintf("%.03f", cox$conf.int[,3])

CI\_U <- sprintf("%.03f", cox$conf.int[,4])

HR\_CI <- paste0(HR, " [",

CI\_L, "–", CI\_U, "]")

res<-c(HR,HR\_CI,CI\_L, CI\_U, p.value)

names(res)<-c("HR", "HR [95% CI for HR]", "CI\_L", "CI\_U","p")

return(res)

})

Uni\_cox <- as.data.frame(t(as.data.frame(univ\_results, check.names = FALSE)))

Characteristics =rownames(Uni\_cox)

Uni\_cox=cbind(Characteristics, Uni\_cox)

Uni\_cox$p=as.numeric(Uni\_cox$p)

View(Uni\_cox)

write.xlsx(Uni\_cox, "uni\_gene\_0.1.xlsx", rowNames=FALSE) #save as xlsx file

unicox\_result= Uni\_cox[which(Uni\_cox$p<0.05), ]

unicox\_result = unicox\_result[order(unicox\_result[,"p"], decreasing=F),]#order in accordance with p-value

unicox\_gene= unicox\_result$Characteristics

#Adverse/favourable

unicox\_result$Type=as.factor(ifelse(unicox\_result$HR<1,'Favourable gene', 'Adverse gene'))

save(status, exp, unicox\_gene, unicox\_result, file="unicox\_gene\_normtpm.Rdata")

unicox\_result = unicox\_result[order(unicox\_result[,"p"], decreasing=T),]#按显著性排序

unicox\_result$p=ifelse(unicox\_result$p <0.001,'p < 0.001', sprintf("%.04f",unicox\_result$p)) #p<0.001显示p<0.001，否则保留4位小数

View(unicox\_result)

write.xlsx(unicox\_result, "uni\_gene\_result\_0.1.xlsx", rowNames=FALSE) #save as xlsx file

## LASSO-Cox

library(openxlsx)

library(survival)

library(glmnet)

load("tpm.Rdata")

gene=read.xlsx('gene1.xlsx')

sameGene=intersect(as.vector(gene[,1]), colnames(expr))

colnames(exp)=gsub('-','\_',colnames(exp))

data = expr[,sameGene] #keep IRG

x <- as.matrix(data.frame(data))

y <- data.matrix(Surv(status$time, status$status))

set.seed(123456) #set seed

cvfit = cv.glmnet(x, y,family = "cox", alpha=1,

nfolds=5)

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

library(svglite)

svglite("LASSO.svg",width=12,height=6)

layout(t(c(1, 2)),widths=c(1,1))

layout.show(2)

plot(cvfit)

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

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

lty = 3,lwd=1)

dev.off()

lambdamin= cvfit$lambda.min

#lambdamin=cvfit$lambda.1se

lambdamin

fit.coef.lambda.min<-coef(cvfit$glmnet.fit,s= lambdamin,exact = F)

fit.min.out = fit.coef.lambda.min[which(fit.coef.lambda.min!= 0),]

fit.min.out = format(fit.min.out, digits=4)

fit.min.out2 <- matrix(as.numeric(fit.min.out), length(fit.min.out), 1)

rownames(fit.min.out2)<-names(fit.min.out)

colnames(fit.min.out2)<- c("coef")

fit.min.out3=as.matrix((fit.min.out2[order(abs(fit.min.out2), decreasing=TRUE),]))

colnames(fit.min.out3) <- c("coef")

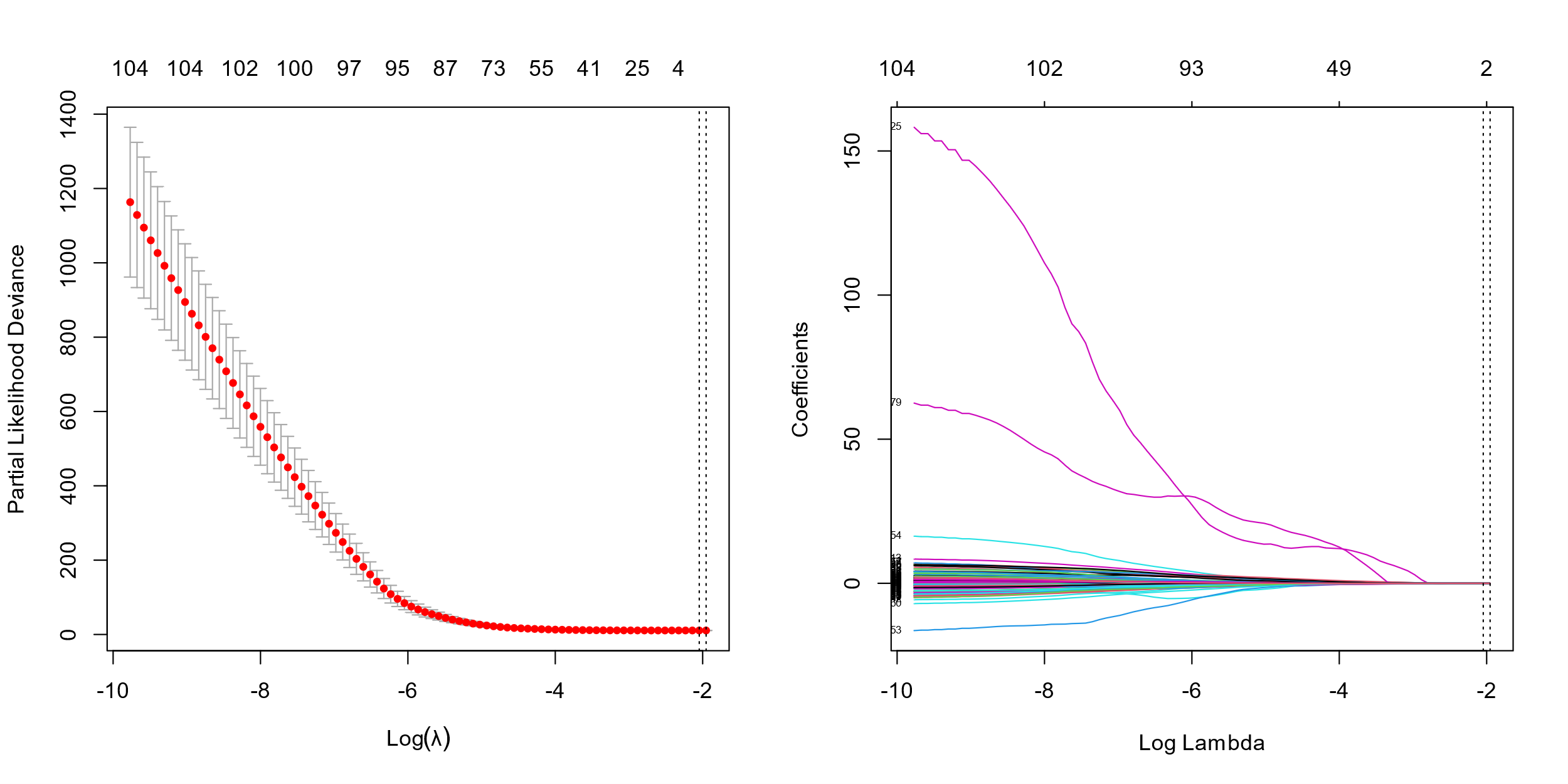
lasso\_gene=rownames(fit.min.out3)

lasso\_gene

# [1] "DHX58" "CTSG"

write.csv(fit.min.out3,'lasso\_gene.csv') #LASSO-Cox selecte gene

save(lasso\_gene, fit, file='lasso\_gene.Rdata')



## multivariate Cox

expr=cbind(status, expr[,sameGene])

#MCHR2+NFYA+CNTFR+NPR3+VEGFA+GDF7+IL1RL1+ACTG1+PSMD10+TNFSF4+MR1+ERAP2+NR2F2+RARG+IL22RA1+CYR61+RHOA+NFKBIB+TRAJ61

#CYR61= CCN1

tstep <- coxph(Surv(time,status) ~ MCHR2+NFYA+CNTFR+NPR3+VEGFA+GDF7+IL1RL1+ACTG1+PSMD10+TNFSF4+MR1+ERAP2+NR2F2+RARG+IL22RA1+ RHOA+NFKBIB+TRAJ61+ CCN1,

data=expr) #多因素cox

cox=summary(tstep)

cox

p.value=as.matrix(cox$coefficients)[,5]

coefficients= sprintf("%.05f", as.matrix(cox$coefficients)[,1])

HR= sprintf("%.03f", as.matrix(cox$coefficients)[,2])

CI\_L= sprintf("%.03f", cox$conf.int[,3])

CI\_U= sprintf("%.03f", cox$conf.int[,4])

HR\_CI <- paste0(HR, " [",

CI\_L, "–", CI\_U, "]")

multi\_res=data.frame(

HR=HR,

Coefficients= coefficients,

HR\_CI= HR\_CI,

CI\_L= CI\_L,

CI\_U= CI\_U,

p=p.value)

colnames(multi\_res)<-c("HR","Coefficients", "HR [95% CI for HR]", "CI\_L", "CI\_U","p")

Characteristics=rownames(multi\_res)

mul\_cox=cbind(Characteristics, multi\_res)

mul\_cox$Coefficients=as.numeric(mul\_cox$Coefficients)

mul\_cox$p=as.numeric(mul\_cox$p)

#mul\_result= mul\_cox[which(mul\_cox$p<0.99), ]

mul\_result= mul\_cox

goi=rownames(mul\_result)

mul\_result = mul\_result[order(mul\_result[,"p"], decreasing=F),]#按显著性排序

mul\_result$Type=as.factor(ifelse(mul\_result$HR<1,'Favourable gene', 'Adverse gene'))

coef=mul\_result$Coefficients

save(goi,coef, mul\_result,file="goi.Rdata")

mul\_result = mul\_result[order(mul\_result[,"p"], decreasing=T),]#按显著性排序

mul\_result$p=ifelse(mul\_result$p <0.001,'p < 0.001', sprintf("%.04f", mul\_result$p)) #p<0.001显示p<0.001，否则保留4位小数

openxlsx::write.xlsx(mul\_result, "mul\_gene\_0.1.xlsx",rowNames=TRUE)

# Risk score validation in subgroups of histological type

data=openxlsx::read.xlsx('mul\_coef.xlsx')

goi=data[,1]

coef=data[,ncol(data)]

load('tpm.Rdata')

expr\_goi=as.data.frame(expr[,goi]) #免疫相关基因表达矩阵

expr=cbind(status,expr\_goi)

risk=function(x){crossprod(as.numeric(x),coef)} #构建Risk\_score的公式

Risk\_score=apply(expr\_goi,1,risk)

expr=cbind(expr, Risk\_score)

#如果验证集也为counts数据，则使用同样的数据做cutoff

cut= median(Risk\_score) #中位数求cutoff

cut

#expr$Risk\_level = ifelse(expr$Risk\_score < res.cut[["cutpoint"]][["cutpoint"]], 'Low','High')

#table (expr$Risk\_level)

expr$Risk\_level = ifelse(expr$Risk\_score < cut, 'Low','High')

table (expr$Risk\_level)

Riskscoreall=data.frame(expr[order(expr[,'Risk\_score'], decreasing=F),]) #按Risk\_score排序

rank=1:nrow(Riskscoreall)

Riskscoreall =cbind(rank, Riskscoreall)

Riskscoreall$Risk\_level =factor(Riskscoreall$Risk\_level,levels = c("Low","High"))

Risk\_score= Riskscoreall[,c(1:3, ncol(Riskscoreall)-1, ncol(Riskscoreall))]

Risk\_score$status=ifelse(Risk\_score$status ==0,'Alive','Dead')

Risk\_score$status =factor(Risk\_score$status,levels = c("Dead","Alive"))

save(Risk\_score, Riskscoreall, Risk\_score, cut, file="Risk\_score.Rdata") #保存风险文件

load("Risk\_score.Rdata") #风险得分结果

load('goi.Rdata')

load('tpm.Rdata')

expr\_goi= log2tpm[goi,]

expr\_goi=as.data.frame(expr\_goi[,match(rownames(Risk\_score),colnames(expr\_goi))])

identical(rownames(Risk\_score),colnames(expr\_goi))#表达矩阵和Risk\_score保持对齐

clinical=clinical[match(rownames(Risk\_score), rownames(clinical)),]

ca=cbind(dplyr::select(clinical,time, status, Age, Gender, Race,

Metastatic, Histological\_type, Tumour\_site,Tumour\_depth,

Tumour\_multifocal, Radiation\_therapy, R\_classification), dplyr::select(Risk\_score, Risk\_level))

cb= cbind(dplyr::select(clinical,time, status, Age, Gender, Race,

Metastatic, Histological\_type, Tumour\_site,Tumour\_depth,

Tumour\_multifocal, Radiation\_therapy, R\_classification), dplyr::select(Risk\_score, Risk\_level, Risk\_score))

save(ca, cb, file='signature.Rdata')

load("signature.Rdata")

library(timeROC)

library(survival)

library(survminer)

Riskscoreall =cb[which(cb$Histological\_type==levels(cb$Histological\_type)[1]),]

fit <- survfit(Surv(time, status==1) ~ Risk\_level, data = Riskscoreall)

print(fit)

p=surv\_pvalue(fit)

pValue=p$pval

pValue=ifelse(pValue<0.05,

ifelse(pValue<0.01,

ifelse(pValue<0.001, paste0("p ＜ 0.001","\*\*\*"),

paste0("p = ",sprintf("%.04f",pValue),"\*\*")),

paste0("p = ",sprintf("%.04f",pValue),"\*")),

paste0("p = ",sprintf("%.04f",pValue)))

ggsurv=ggsurvplot(fit,

pval = pValue,

pval.size=3,

conf.int = TRUE,

risk.table = "abs\_pct", # Add risk table

fontsize=2,

risk.table.col = "strata", # Change risk table color by groups

linetype = "strata", # Change line type by groups

surv.median.line = "none", # Specify median survival

# title = "Dead (TCGA cohort)",

palette= c('#00BEC3', '#F6766D'),

xlab = "Survival time (y)",

ylab = "Survival probability",

ggtheme = theme\_bw(),

ncensor.plot = TRUE,

legend.title='Risk score level',

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

surv\_LMS= ggarrange(ggsurv$plot+rremove('x.title') + theme(plot.title = element\_text(hjust = 0.5, face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$table+rremove('x.title') +rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$ncensor.plot+rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black",

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8), #居中对齐，ticks长度，数字到标签的距离

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"))+

scale\_y\_continuous(limits = c(0, 2), breaks=seq(0, 2, by=1)),

nrow=3, widths = c(1,1,1),heights=c(1.35,0.525,0.7), align = "v")

surv\_LMS

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_LMS=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.text.x = element\_text(vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, size = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

)

ROC\_LMS

# DLP

load("signature.Rdata")

library(ggplot2)

library(timeROC)

Riskscoreall =cb[which(cb$Histological\_type== levels(cb$Histological\_type)[2]),]

fit <- survfit(Surv(time, status==1) ~ Risk\_level, data = Riskscoreall)

print(fit)

p=surv\_pvalue(fit)

pValue=p$pval

pValue=ifelse(pValue<0.05,

ifelse(pValue<0.01,

ifelse(pValue<0.001, paste0("p ＜ 0.001","\*\*\*"),

paste0("p = ",sprintf("%.04f",pValue),"\*\*")),

paste0("p = ",sprintf("%.04f",pValue),"\*")),

paste0("p = ",sprintf("%.04f",pValue)))

ggsurv=ggsurvplot(fit,

pval = pValue,

pval.size=3,

conf.int = TRUE,

risk.table = "abs\_pct", # Add risk table

fontsize=2,

risk.table.col = "strata", # Change risk table color by groups

linetype = "strata", # Change line type by groups

surv.median.line = "none", # Specify median survival

# title = "Dead (TCGA cohort)",

palette= c('#00BEC3', '#F6766D'),

xlab = "Survival time (y)",

ylab = "Survival probability",

ggtheme = theme\_bw(),

ncensor.plot = TRUE,

legend.title='Risk score level',

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

surv\_DLP= ggarrange(ggsurv$plot+rremove('x.title') + theme(plot.title = element\_text(hjust = 0.5, face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$table+rremove('x.title') +rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$ncensor.plot+rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black",

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8), #居中对齐，ticks长度，数字到标签的距离

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"))+

scale\_y\_continuous(limits = c(0, 2), breaks=seq(0, 2, by=1)),

nrow=3, widths = c(1,1,1),heights=c(1.35,0.525,0.7), align = "v")

surv\_DLP

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_DLP=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.text.x = element\_text(vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, size = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

)

ROC\_DLP

# UPS

load("signature.Rdata")

library(ggplot2)

library(timeROC)

Riskscoreall =cb[which(cb$Histological\_type== levels(cb$Histological\_type)[3]),]

fit <- survfit(Surv(time, status==1) ~ Risk\_level, data = Riskscoreall)

print(fit)

p=surv\_pvalue(fit)

pValue=p$pval

pValue=ifelse(pValue<0.05,

ifelse(pValue<0.01,

ifelse(pValue<0.001, paste0("p ＜ 0.001","\*\*\*"),

paste0("p = ",sprintf("%.04f",pValue),"\*\*")),

paste0("p = ",sprintf("%.04f",pValue),"\*")),

paste0("p = ",sprintf("%.04f",pValue)))

ggsurv=ggsurvplot(fit,

pval = pValue,

pval.size=3,

conf.int = TRUE,

risk.table = "abs\_pct", # Add risk table

fontsize=2,

risk.table.col = "strata", # Change risk table color by groups

linetype = "strata", # Change line type by groups

surv.median.line = "none", # Specify median survival

# title = "Dead (TCGA cohort)",

palette= c('#00BEC3', '#F6766D'),

xlab = "Survival time (y)",

ylab = "Survival probability",

ggtheme = theme\_bw(),

ncensor.plot = TRUE,

legend.title='Risk score level',

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

surv\_UPS= ggarrange(ggsurv$plot+rremove('x.title') + theme(plot.title = element\_text(hjust = 0.5, face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$table+rremove('x.title') +rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$ncensor.plot+rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black",

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8), #居中对齐，ticks长度，数字到标签的距离

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"))+

scale\_y\_continuous(limits = c(0, 2), breaks=seq(0, 2, by=1)),

nrow=3, widths = c(1,1,1),heights=c(1.35,0.525,0.7), align = "v")

surv\_UPS

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_UPS=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.text.x = element\_text(vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, size = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

)

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_UPS=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.text.x = element\_text(vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, size = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

)

ROC\_UPS

#MYX

load("signature.Rdata")

library(ggplot2)

library(timeROC)

Riskscoreall =cb[which(cb$Histological\_type=='MYX'),]

fit <- survfit(Surv(time, status==1) ~ Risk\_level, data = Riskscoreall)

print(fit)

p=surv\_pvalue(fit)

pValue=p$pval

pValue=ifelse(pValue<0.05,

ifelse(pValue<0.01,

ifelse(pValue<0.001, paste0("p ＜ 0.001","\*\*\*"),

paste0("p = ",sprintf("%.04f",pValue),"\*\*")),

paste0("p = ",sprintf("%.04f",pValue),"\*")),

paste0("p = ",sprintf("%.04f",pValue)))

ggsurv=ggsurvplot(fit,

pval = pValue,

pval.size=3,

conf.int = TRUE,

risk.table = "abs\_pct", # Add risk table

fontsize=2,

risk.table.col = "strata", # Change risk table color by groMYX

linetype = "strata", # Change line type by groMYX

surv.median.line = "none", # Specify median survival

# title = "Dead (TCGA cohort)",

palette= c('#00BEC3', '#F6766D'),

xlab = "Survival time (y)",

ylab = "Survival probability",

ggtheme = theme\_bw(),

ncensor.plot = TRUE,

legend.title='Risk score level',

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

surv\_MYX= ggarrange(ggsurv$plot+rremove('x.title') + theme(plot.title = element\_text(hjust = 0.5, face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$table+rremove('x.title') +rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black", size=8),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm")),

ggsurv$ncensor.plot+rremove("legend")+theme(plot.title = element\_text(face = "bold",size=8),

axis.text.x = element\_text(vjust = 0.5, color="black",

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8), #居中对齐，ticks长度，数字到标签的距离

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"))+

scale\_y\_continuous(limits = c(0, 2), breaks=seq(0, 2, by=1)),

nrow=3, widths = c(1,1,1),heights=c(1.35,0.525,0.7), align = "v")

surv\_MYX

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_MYX=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.text.x = element\_text(vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.text.y = element\_text(angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm"), size=8),

axis.title.x=element\_text(size=8,color="black"),

axis.title.y=element\_text(size=8,color="black"),

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, size = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

)

time\_ROC<-timeROC(T= Riskscoreall$time, #生存时间(Dead和Survival的生存时间).

delta= Riskscoreall$status, #生存结局，Censored的样本必须用0表示

marker= Riskscoreall$Risk\_score, #预测的变量，这里是风险评分，在没有特殊情况下，值越大，越容易发生危险事件

cause=1, #阳性结局的赋值（必须是1或2），也就是Dead的赋值，这里Dead是1表示的

weighting="marginal",# [选用K-M]分析

times = c(1,2,3), #计算3、5、10年的ROC曲线

ROC=TRUE,

iid=TRUE #计算AUC

)

round(time\_ROC[["AUC"]],3)

label=paste0("AUC of ",c(1,2,3),"-y OS = ", sprintf("%.03f",time\_ROC$AUC), ' [', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,1]/100), '–', sprintf("%.03f",confint(time\_ROC , level = 0.95)$CI\_AUC[,2]/100), ']')

label

#[1] "AUC of 1-y OS = 0.798 [0.675–0.920]"

#[2] "AUC of 3-y OS = 0.746 [0.658–0.835]"

#[3] "AUC of 5-y OS = 0.726 [0.618–0.835]"

time\_ROC.res1<-data.frame(TPR=time\_ROC$TP[,1], #获取1年的ROC的TP

FPR=time\_ROC$FP[,1])

time\_ROC.res1$times=rep('1', nrow(time\_ROC.res1))

time\_ROC.res3<-data.frame(TPR=time\_ROC$TP[,2], #获取3年的ROC的TP

FPR=time\_ROC$FP[,2])

time\_ROC.res3$times=rep('3', nrow(time\_ROC.res3))

time\_ROC.res5<-data.frame(TPR=time\_ROC$TP[,3], #获取5年的ROC的TP

FPR=time\_ROC$FP[,3])

time\_ROC.res5$times=rep('5', nrow(time\_ROC.res5))

res=rbind(time\_ROC.res1, time\_ROC.res3, time\_ROC.res5)

ROC\_MYX=ggplot(res, aes(x=FPR, y=TPR, color=factor(times, levels=c('1', '3', '5') ))) +

geom\_line(size=1)+

labs(x="1-Specificity", y="Sensitivity")+ #横纵坐标的名字

labs(color=NULL,fill=NULL)+

scale\_color\_manual(values=c('#4DBBD5FF', 'gold', '#00A087FF'),

labels= label)+

geom\_ribbon(aes(ymin = 0, ymax = TPR, fill=factor(times, levels=c('1', '3', '5'))), show.legend = FALSE, alpha=0.1,outline.type="upper")+ #填充线下面积，不要图例、下方边界

scale\_fill\_manual(values=c('#00BEC3', '#F9F932', '#00BFFF'))+ #填充面积设置透明度

# scale\_x\_continuous(labels = scales::percent\_format(accuracy = 1))+

# scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))+ #百分比形式的横纵坐标

geom\_abline(slope = 1, size=1, intercept = 0, linetype=5,col="grey70")+ #增加一个对角线，如果ROC曲线在对角线的左上边，就说明是还可以的

theme\_bw()+

theme(

panel.border = element\_rect(colour = "black", fill=NA),

legend.position = c(0.9,0.1),

legend.justification = c(0.9,0.1),

axis.title.x=element\_text(size=12,color="black"), #x轴标题修改

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

axis.text.x = element\_text(size=12,vjust = 0.5, color="black", #居中对齐，ticks长度，数字到标签的距离

margin=unit(c(0.2,0.2,0.2,0.2), "cm")),

axis.text.y = element\_text(size=12,angle = 90, hjust = 0.5, color="black", #纵坐标的数字旋转90°

margin=unit(c(0.2,0.2,0.2,0.2), "cm")),

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

axis.ticks.length=unit(0.2, "cm"),

legend.background = element\_rect(linetype = 1, linewidth = 0.5, colour = 1, fill=NA),

#legend.key = element\_blank() #图例线条后面的背景如果拿掉看不清，所以我注释掉了

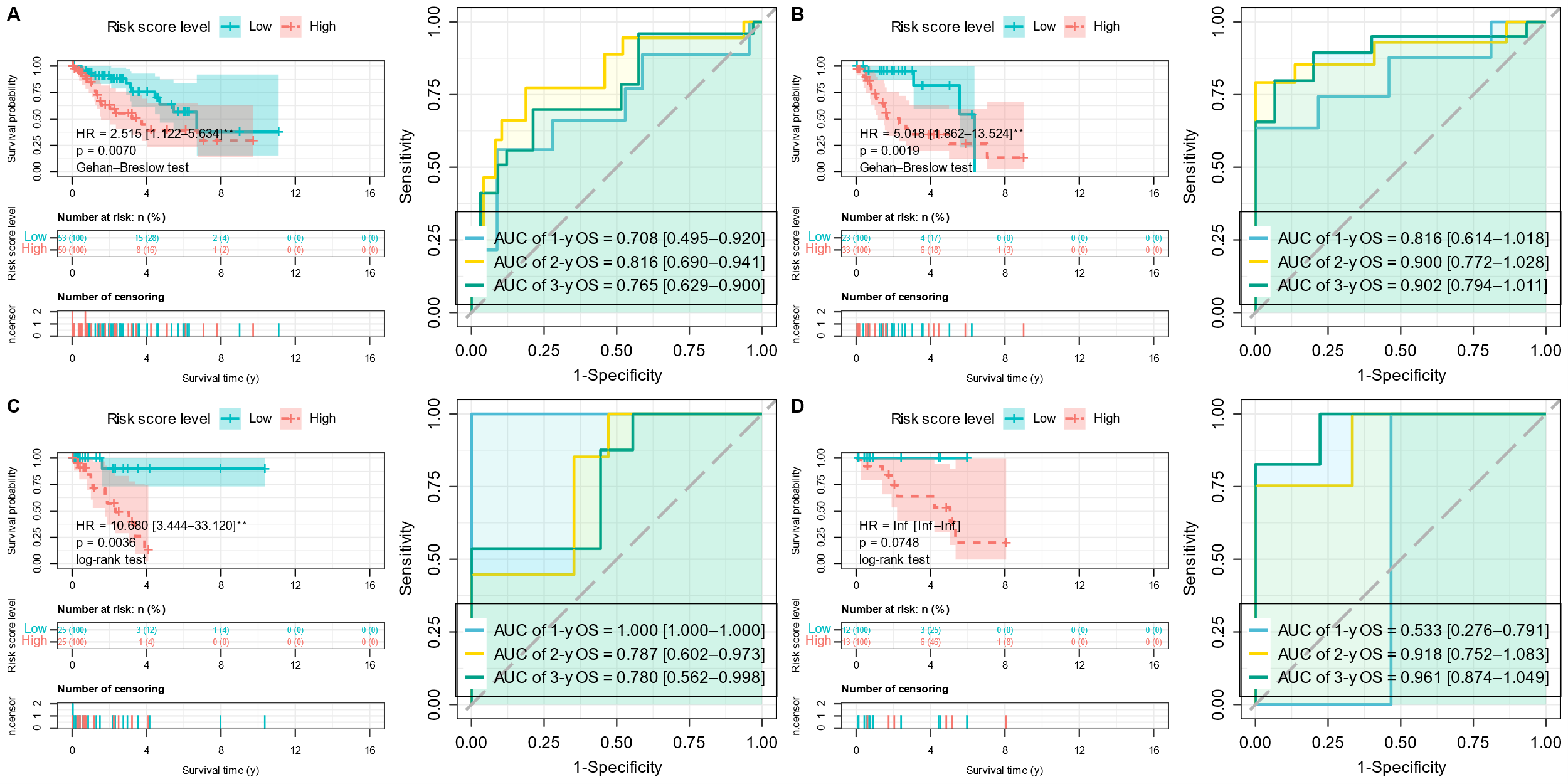
)

ROC\_MYX

cairo\_pdf(filename='HT.pdf',height=24,width=16)

ggarrange(surv\_LMS,ROC\_LMS,surv\_DLP,ROC\_DLP, surv\_UPS,ROC\_UPS,surv\_MYX,ROC\_MYX,ncol=4,nrow=2,labels = c('A','','B','','C','','D',''))

dev.off()



# Tumour immune microenvironment

## CIBERSORT

source("CIBERSORT.R")

set.seed(123456)

results=CIBERSORT("LM22.txt", "tpm.txt", perm=1000, QN=FALSE)

results=as.data.frame(results)

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

## barplot

load("results.Rdata")

results=results[match(rownames(ca),rownames(results)),]

p1=as.data.frame(t(results[, 1:22]))

phylum\_color <- rainbow(22)

names(phylum\_color) <- rownames(p1)

svglite::svglite(filename='CIBERSORT\_barplot.svg',height=5,width=12)

layout(t(c(1, 2, 3)),widths=c(0.1,3,1))

#layout.show(3)

par(mar = c(0.01, 0.01, 0.01, 0.01))

plot(0, type = 'n', xaxt = 'n', yaxt = 'n', bty = 'n', xlab = '', ylab = '')

par(mai = c(0.5, 0.5, 0.5, 0))

barplot(as.matrix(p1), las=2,

col=phylum\_color, xaxt="n", xlab="", yaxt = 'n', ylab= "CIBERSORT fraction", space = 0)

axis(2,at=c(0, 0.2, 0.4, 0.6, 0.8, 1),labels=c('0%', '20%', '40%', '60%', '80%', '100%'))

par(mar = c(5, 1, 5, 0))

plot(0, type = 'n', xaxt = 'n', yaxt = 'n', bty = 'n', xlab = '', ylab = '')

legend('left', pch = 15, col = phylum\_color, legend = names(phylum\_color), bty = 'n', cex = 1, ncol=1)

dev.off()

