################启动##############

library(tidyverse)

library(limma)

library(GEOquery)

library(ggplot2)

library(ggsci)

if (!file.exists("EGSE171272\_eSet.Rdata")) {

 GEO\_file <- getGEO('GSE171272',#需要下载的series

 # destdir = 'E:/GEOdoc/GSE64634',#设置文件保存路径

 getGPL = T #下载平台文件

 )

 save(GEO\_file, file = "EGSE171272\_eSet.Rdata")#将下载下来的文件保存为我们可以处理的格式}

load("EGSE171272\_eSet.Rdata")#将更改好格式并保存好的的数据加载到这里

#提取下载数据中我们需要的部分http://127.0.0.1:12591/graphics/plot\_zoom\_png?width=1200&height=900

GEO\_file[[1]]#提取GEO\_file中第一个数据http://127.0.0.1:8587/graphics/plot\_zoom\_png?width=1200&height=900

exp <- exprs(GEO\_file[[1]])#提取数据中的样本基因表达矩阵http://127.0.0.1:8587/graphics/plot\_zoom\_png?width=1200&height=900

#exp2 <- exprs(GEO\_file[[1]])#提取数据中的样本基因表达矩阵

plate <- fData(GEO\_file[[1]])#提取数据中的平台信息

clinical <- pData(GEO\_file[[1]])#提取数据中的样本临床信息（比如：年龄、性别、是否存活等等）

#看一下数据分布

exp <- as.matrix(exp)

boxplot(exp, col=c("#FFE4C4","#F0FFFF","#696969","#6495ED","#00CED1","#006400",

"#BDB76B","#CD853F","#E9967A","#FF00FF","#8B7765","#836FFF","#8B008B","#BF3EFF","#FF7F00"))

#将每一个基因都对应出来

ID <- data.frame(ID\_REF = plate$ID, Gene\_Symbol = plate$`Human\_miRNA`)#将平台文件的ID列和SYMBOL列取出

x <- tibble(unlist(apply(ID,1,function(x){paste(x[1],str\_split(x[2],'/',simplify=T),sep = "...")})))

colnames(x) <- "ABC"

file <- separate(x,ABC,c("ID\_REF","Gene\_Symbol"),sep = "\\...")#处理一个探针对应多个基因\

exp <- as.data.frame(exp)#将表达矩阵转换为数据框

#exp <- exp %>% rownames\_to\_column("ID\_REF")

exp$ID\_REF <- rownames(exp)

#file[file == ""]<-NA#将空白负值NA

#file<-na.omit(file)#删除GENE\_SYBOL缺失的数据2201

#install.packages("stringr")

#library(stringr)

file <- as.data.frame(file)

file[, grep("ID\_REF", colnames(file))] <- trimws(file[, grep("ID\_REF", colnames(file))])#去除数据头尾空格

exp <- merge(exp, file, by = "ID\_REF")#以ID为参照值，对表达矩阵和GPL进行合并1394

exp[, grep("Gene\_Symbol", colnames(exp))] <- trimws(exp[, grep("Gene\_Symbol", colnames(exp))])#去除数据头尾空格

exp[exp == ""]<-NA#将空白负值NA

#exp<-na.omit(exp)#删除GENE\_SYBOL缺失的数据

exp <- exp %>% drop\_na(Gene\_Symbol)#删除Human\_miRNA列缺失的数据

exp <- as.data.frame(exp)

write.csv(exp, "exp.csv")

exp <- read.csv("exp.csv", row.names = 1)

#############几种处理重复基因的方式，下面的方式选择一种就行了##################

table(duplicated(exp$Gene\_Symbol))#看一下有多少重复

#将重复基因取平均值

exp1 <- avereps(exp, ID = exp$Gene\_Symbo)

exp1 <- as.data.frame(exp1)

rownames(exp1) <- exp1$Gene\_Symbol#加上行名

exp1 <- exp1[,-c(1,ncol(exp1))]#将多余列的数据剔除

write.csv(exp1, "exp\_average.csv")#保存为csv格式

#提取分组信息

clinical <- as.data.frame(clinical)

sample\_names <- clinical[,1:2]

rownames(sample\_names) <- NULL

sample\_names <- sample\_names %>% column\_to\_rownames("title")

sample\_names <- sample\_names %>% rownames\_to\_column("sample")

sample\_names <- sample\_names %>% rownames\_to\_column("Type")

sample\_names$Type <- c(1:15)

normal <- 10

sample\_names$Type <- ifelse(sample\_names$Type > normal,"normal","tumor")

sample\_names <- as.data.frame(sample\_names)

sample\_names <- sample\_names[,-c(2,4)]

sample\_names <- column\_to\_rownames(sample\_names,var = 'geo\_accession')

sample\_names <- as.data.frame(sample\_names)

write.csv(sample\_names, "Type.csv")#保存为csv格式

#sample\_names <- sample\_names %>% column\_to\_rownames("geo\_accession")

#sample\_names <- as.data.frame(sample\_names)

#column(sample\_names) <- substring(column(sample\_names),1,7)

#sample\_names <- t(sample\_names)

#sample\_names <- as.data.frame(sample\_names)

#######EXO\_lima操作##########

setwd("exo\_mirna")

#差异分析——limma

rm(list = ls()) ## 魔幻操作，一键清空~

options(stringsAsFactors = F)

#RE=rownames(pd[grepl('diagnosis:ch1: DLBCL',as.character(pd$`diagnosis:ch1`)),]) #复发组

#PRI=rownames(pd[grepl('diagnosis:ch1: healthy',as.character(pd$`diagnosis:ch1`)),]) #未复发组

#dat=dat[,c(RE,PRI)]

#group\_list=c(rep('RE',length(RE)),

#rep('PRI',length(PRI))) #分组信息

group\_list=Type

Type=group\_list

group\_list <- group\_list %>% rownames\_to\_column("sample")

table(group\_list)

dim(exp\_average)

library(limma)

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

# 做分组矩阵

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

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

rownames(design)=colnames(exp\_average)

design #分组矩阵

# 做比较矩阵

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

# contrast.matrix ##这个矩阵声明，我们要把treat组和contorl组进行差异分析比较

# -1和1的意思是contorl是用来被比的，treat是来比的

contrast.matrix<-makeContrasts(paste0(c("tumor","normal"),collapse = "-"),levels = design)

contrast.matrix

##step1

fit <- lmFit(exp\_average,design,na.rm=TURE) #线性模型拟合

##step2

fit2 <- contrasts.fit(fit, contrast.matrix) #根据对比模型进行差值计算

fit2 <- eBayes(fit2) #贝叶斯检验

##eBayes() with trend=TRUE

##step3

tempOutput = topTable(fit2, coef=1, n=Inf) #生成所有基因的检验结果报表

write.csv(tempOutput, "DEG.csv")#保存为csv格式

#write.csv(nrDEG2,"limma\_notrend.results.csv",quote = F)

save(exp\_average,group\_list,tempOutput,file = "DEGoutput.Rdata")

#用limma包得到差异分析表达矩阵后作图检查差异基因是否真的很差异

###########绘制EXOSOME热图####

DEG <- as.data.frame(tempOutput)

#读取表达谱

exp <- exp\_average

#添加上下调信息

logFC\_cutoff <- 1.5

type1 = (DEG$adj.P.Val < 0.05)&(DEG$logFC < -logFC\_cutoff)

type2 = (DEG$adj.P.Val < 0.05)&(DEG$logFC > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

write.csv(c, "exo\_up\_down\_set.csv")#保存为csv格式

d <- rownames(c)

exp\_diff <- exp[d,]

write.csv(exp\_diff, "exp\_diff.csv")#保存为csv格式

#设置分组信息

annotation\_col <- Type

#class<- t(class)

#class <- as.data.frame(class)

#identical(colnames(exp\_average),colnames(class))

#exp\_diff <- rbind(exp\_diff,class)

#class<- t(class)

#annotation\_col <- as.data.frame(annotation\_col)

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

a <- a[,-1]

a <- column\_to\_rownames(a,"X")

a <- as.data.frame(a)

b <- b[,-1]

b <- column\_to\_rownames(b,"X")

b <- as.data.frame(b)

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

annotation\_col <- annotation\_col[,-1]

annotation\_col <- column\_to\_rownames(annotation\_col,"X")

a <- as.data.frame(a)

#exp\_diff <- exp\_diff[-nrow(exp\_diff),]

#exp\_diff = na.omit(exp\_diff)

#exp\_diff <- as.data.frame(exp\_diff)

#exp\_diff <- as.numeric(exp\_diff)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames = T,

 color = colorRampPalette(c("navy", "white", "red"))(50),

 cluster\_cols = F,

 cluster\_rows = T,

 fontsize = 10,

 fontsize\_row=3.5,

 fontsize\_col=7,)

#保存图片 调整大小

dev.off()#关闭画板

######绘制火山图

getwd()

df<-read.csv("DEG.csv",header=T,

 stringsAsFactors = F)

df <- tempOutput

df = na.omit(df)

head(df)

dim(df)

df$group<-ifelse(df$logFC > 1.5&df$adj.P.Val < 0.05,"Up",

 ifelse(df$logFC < -1.5&df$adj.P.Val<0.05,

 "Down","Not sig"))

table(df$group)

library(ggplot2)

#install.packages("ggrepel")

library(ggrepel)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))

df$pvalue\_log10<-(-log10(df$adj.P.Val))

df1<-df[df$pvalue\_log10>=7.5,]

df2<-df1[abs(df1$logFC)>=1.5,]

dim(df2)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))+

 geom\_label\_repel(data=df2,aes(x=logFC,y=-log10(adj.P.Val),

 label=rownames(df2)))+

 theme\_bw()+

 labs(y="-log10(adj.P.Val)",x="log(Fold Change)")+

 xlim(c(-10,10))

write.csv(df, "DEG-DLBCL-exo.csv")

###########DLBCL-miRNA-limma数据########

library("tidyverse")

library("limma")

library("GEOquery")

library("ggplot2")

setwd("dlbcl\_mirna")

#以GSE173080为例，进行下载

#if (!file.exists("GSE173080\_eSet.Rdata")) {

GEO\_file <- getGEO('GSE173080',#需要下载的series

 # destdir = 'E:/GEOdoc/GSE64634',#设置文件保存路径

 getGPL = T #下载平台文件

)

save(GEO\_file, file = "GSE173080\_eSet.Rdata")#将下载下来的文件保存为我们可以处理的格式

}

load("GSE173080\_eSet.Rdata")#将更改好格式并保存好的的数据加载到这里

#提取下载数据中我们需要的部分http://127.0.0.1:41527/graphics/plot\_zoom\_png?width=1200&height=900

GEO\_file[[1]]#提取GEO\_file中第一个数据

exp <- exprs(GEO\_file[[1]])#提取数据中的样本基因表达矩阵

#exp2 <- exprs(GEO\_file[[1]])#提取数据中的样本基因表达矩阵

plate <- fData(GEO\_file[[1]])#提取数据中的平台信息

clinical <- pData(GEO\_file[[1]])#提取数据中的样本临床信息（比如：年龄、性别、是否存活等等）

#看一下数据分布

exp <- as.matrix(exp)

exp <- as.data.frame(exp)

str(exp)

data(exp)

boxplot(exp,col=pal\_d3()(10))

exp <- as.data.frame(exp)

write.csv(exp, "exp10.csv")

#exp1<- exp1[,-c(2,5)]

#write.csv(exp1, "exp1.csv")

#table(duplicated(exp$X))

#exp2 <- 2^exp

#exp3 <- log2(exp2+1)

#提取分组信息

clinical <- as.data.frame(clinical)

sample\_names <- clinical[,1:2]

rownames(sample\_names) <- NULL

sample\_names <- sample\_names %>% column\_to\_rownames("title")

sample\_names <- sample\_names %>% rownames\_to\_column("sample")

sample\_names <- sample\_names %>% rownames\_to\_column("Type")

sample\_names$Type <- c(1:10)

normal <- 5

sample\_names$Type <- ifelse(sample\_names$Type > normal,"tumor","normal")

sample\_names<- sample\_names[-c(2,5),]

sample\_names <- as.data.frame(sample\_names)

sample\_names <- sample\_names[,-c(2,4)]

rownames(sample\_names) <- NULL

sample\_names <- column\_to\_rownames(sample\_names,var = 'geo\_accession')

sample\_names <- as.data.frame(sample\_names)

a <- filter(sample\_names,Type == 'tumor')

b <- filter(sample\_names,Type == 'normal')

c <- rbind(a,b)

Type <- c

write.csv(Type, "Type.csv")#保存为csv格式

a<- rownames(Type)

exp <- exp[,a]

write.csv(exp, "exp.csv")

#sample\_names <- sample\_names %>% column\_to\_rownames("geo\_accession")

#sample\_names <- as.data.frame(sample\_names)

#column(sample\_names) <- substring(column(sample\_names),1,7)

#sample\_names <- t(sample\_names)

#sample\_names <- as.data.frame(sample\_names)

#######lima操作##########

library(tidyverse)

library(limma)

library(GEOquery)

library("ggplot2")

library("ggsci")

setwd("dlbcl\_mirna")

#差异分析——limma

rm(list = ls()) ## 魔幻操作，一键清空~

options(stringsAsFactors = F)

group\_list=Type

#group\_list <- group\_list %>% rownames\_to\_column("Type")

table(group\_list)

dim(exp)

library(limma)

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

# 做分组矩阵

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

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

rownames(design)=colnames(exp)

design #分组矩阵

# 做比较矩阵

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

# contrast.matrix ##这个矩阵声明，我们要把treat组和contorl组进行差异分析比较

# -1和1的意思是contorl是用来被比的，treat是来比的

contrast.matrix<-makeContrasts(paste0(c("tumor","normal"),collapse = "-"),levels = design)

contrast.matrix

#到此，做差异分析所需要的三个矩阵就做好了：表达矩阵、分组矩阵、差异比较矩阵

#我们已经制作好了必要的输入数据，下面开始讲如何使用limma这个包来进行差异分析了！

##step1

fit <- lmFit(exp,design,na.rm=TURE) #线性模型拟合

##step2

fit2 <- contrasts.fit(fit, contrast.matrix) #根据对比模型进行差值计算

fit2 <- eBayes(fit2) #贝叶斯检验

##eBayes() with trend=TRUE

##step3

tempOutput = topTable(fit2, coef=1, n=Inf) #生成所有基因的检验结果报表

write.csv(tempOutput, "DEG.csv")#保存为csv格式

#write.csv(nrDEG2,"limma\_notrend.results.csv",quote = F)

head(tempOutput)

save(exp,group\_list,tempOutput,file = "DEGoutput.Rdata")

#用limma包得到差异分析表达矩阵后作图检查差异基因是否真的很差异

###########绘制热图####

#a<- rownames(Type2)

#exp1 <- exp1[,a]

DEG <- tempOutput

#读取表达谱

exp <- exp

#添加上下调信息

logFC\_cutoff <- 1.5

type1 = (DEG$adj.P.Val < 0.05)&(DEG$logFC < -logFC\_cutoff)

type2 = (DEG$adj.P.Val < 0.05)&(DEG$logFC > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

write.csv(c, "exp\_diff\_d.csv")#保存为csv格式

d <- rownames(c)

exp\_diff <- exp[d,]

#write.csv(exp\_diff, "exp\_diff.csv")#保存为csv格式

#设置分组信息

#e <- filter(Type1,Type == 'tumor')

#f <- filter(Type1,Type == 'normal')

#g <- rbind(e,f)

#write.csv(g, "Type2")#保存为csv格式

annotation\_col <- group\_list

annotation\_col <- annotation\_col %>% column\_to\_rownames("X")

#class<- t(class)

#class <- as.data.frame(class)

#identical(colnames(exp\_average),colnames(class))

#exp\_diff <- rbind(exp\_diff,class)

#class<- t(class)

#annotation\_col <- as.data.frame(annotation\_col)

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

#exp\_diff <- exp\_diff[-nrow(exp\_diff),]

#exp\_diff = na.omit(exp\_diff)

#exp\_diff <- as.data.frame(exp\_diff)

#exp\_diff <- as.numeric(exp\_diff)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames = T,

 color = colorRampPalette(c("navy", "white", "red"))(50),

 cluster\_cols = F,

 cluster\_rows = T,

 fontsize = 10,

 fontsize\_row=4,

 fontsize\_col=4,)

#保存图片 调整大小

dev.off()#关闭画板

######绘制火山图

getwd()

df<-read.csv("DEG.CSV",header=T,

 stringsAsFactors = F)

df<-DEG

df = na.omit(df)

head(df)

dim(df)

df$group<-ifelse(df$logFC > 1.5&df$adj.P.Val<0.05,"Up",

 ifelse(df$logFC < -1.5&df$adj.P.Val<=0.05,

 "Down","Not sig"))

table(df$group)

library(ggplot2)

#install.packages("ggrepel")

library(ggrepel)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))

df$adj.P.Val\_log10<-(-log10(df$adj.P.Val))

df <- df %>% rownames\_to\_column("X")

df1<-df[df$adj.P.Val\_log10 > 1.25,]

df2<-df1[abs(df1$logFC) > 1.5,]

dim(df2)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))+

 geom\_label\_repel(data=df2,aes(x=logFC,y=-log10(adj.P.Val),

 label=X))+

 theme\_bw()+

 labs(y="-log10(adj.P.Val)",x="log(Fold Change)")+

 xlim(c(-7.5,7.5))

write.csv(df, "DEG-DLBCL-mirna.csv")

#######miRNA取交集#########

#exo共2182个miRNA

#dlbcl共2549个miRNA

exo\_dlbcl <- intersect(rownames(DEG),rownames(exp\_average))

exo\_dlbcl <- as.data.frame(exo\_dlbcl)

write.csv(exo\_dlbcl, "exo\_dlbcl\_miRNA.csv")

exo <- exp\_average

dlbcl <- DEG

a <- rownames(exo)

b <- rownames(dlbcl)

a <- as.data.frame(a)

b <- as.data.frame(b)

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- a #输入第一个基因集

gene\_set2 <- b

a <- list(`Exosome miRNA` = gene\_set1$a,

 `DLBCL miRNA` = gene\_set2$b) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(2),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

###########venn-all图制作########

setwd("dlbcl\_mirna")

exo <- exo\_up\_down\_set

dlbcl <- exp\_diff\_d

a <- filter(exo,change == 'DOWN')

b <- filter(dlbcl,change == 'DOWN')

c <- filter(exo,change == 'UP')

d <- filter(dlbcl,change == 'UP')

e <- row.names(a)

f <- row.names(b)

g <- row.names(c)

h <- row.names(d)

e <- as.data.frame(e)

f <- as.data.frame(f)

g <- as.data.frame(g)

h <- as.data.frame(h)

#x<-list(c=c$c,

#d=d$d)

#fill.col<-c("red","blue")

#install.packages("VennDiagram")

#library(VennDiagram)

#library("ggplot2")

#venn.diagram(x=x,

#filename = "vennplot.tiff",

#fill=fill.col,

#set\_names\_size = 10,text\_size = 10)

###########venn图制作########

#install.packages("ggvenn")

#install.packages("ggsci")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- e #输入第一个基因集

gene\_set2 <- f

gene\_set3 <- g

gene\_set4 <- h

a <- list(`GSM171272 down-regulated set` = gene\_set1$e,

 `GSM173080 down-regulated set` = gene\_set2$f,

 `GSM171272 up-regulated set` = gene\_set3$g,

 `GSM173080 up-regulated set` = gene\_set4$h) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 8,

 text\_size = 6)

p2

exo\_dlbcl\_up <- intersect(rownames(c),rownames(d))

exo\_dlbcl\_up <- as.data.frame(exo\_dlbcl\_up)

write.csv(exo\_dlbcl\_up, "exo\_dlbcl\_up.csv")

exo\_dlbcl\_down <- intersect(rownames(a),rownames(b))

exo\_dlbcl\_down <- as.data.frame(exo\_dlbcl\_down)

write.csv(exo\_dlbcl\_down, "exo\_dlbcl\_down.csv")

dup\_edown <- intersect(rownames(a),rownames(d))

dup\_edown <- as.data.frame(dup\_edown)

write.csv(dup\_edown, "dlbclup\_exodown.csv")

ddown\_eup <- intersect(rownames(b),rownames(c))

ddown\_eup <- as.data.frame(ddown\_eup)

write.csv(ddown\_eup, "dlbcldown\_exoup.csv")

###########venn-up图制作########

exo <- exp\_diff\_e

dlbcl <- exp\_diff\_d

a <- filter(exo,change == 'UP')

b <- filter(dlbcl,change == 'UP')

c <- row.names(a)

c <- as.data.frame(c)

#c <- rownames\_to\_column(c,"n")

#c <- column\_to\_rownames(c,"c")

d <- row.names(b)

d <- as.data.frame(d)

#d <- rownames\_to\_column(d,"n")

#d <- column\_to\_rownames(d,"d")

#e <- intersect(rownames(c),rownames(d))

#e <- as.data.frame(e)

#write.csv(e, "exo\_dlblc\_up.csv")

#f <- intersect(c$c,d$d)

#f <- as.data.frame(f)

#x<-list(c=c$c,

 #d=d$d)

#fill.col<-c("red","blue")

#install.packages("VennDiagram")

#library(VennDiagram)

#library("ggplot2")

#venn.diagram(x=x,

 #filename = "vennplot.tiff",

 #fill=fill.col,

 #set\_names\_size = 10,text\_size = 10)

#install.packages("ggvenn")

#install.packages("ggsci")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library(ggvenn) #加载ggvenn包

gene\_set1 <- c #输入第一个基因集

gene\_set2 <- d

a <- list(`EXOSOME-UP` = gene\_set1$c,

 `DLBCL-UP` = gene\_set2$d) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

p2 <- ggvenn(a, show\_percentage = T,fill\_color = c("#008B45","#FFD700"),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

###########venn-down图制作########

exo <- exp\_diff\_e

dlbcl <- exp\_diff\_d

a <- filter(exo,change == 'DOWN')

b <- filter(dlbcl,change == 'DOWN')

c <- row.names(a)

c <- as.data.frame(c)

c <- rownames\_to\_column(c,"n")

c <- column\_to\_rownames(c,"c")

d <- row.names(b)

d <- as.data.frame(d)

d <- rownames\_to\_column(d,"n")

d <- column\_to\_rownames(d,"d")

e <- intersect(rownames(c),rownames(d))

e <- as.data.frame(e)

write.csv(e, "exo\_dlblc\_down.csv")

#x<-list(c=c$c,

#d=d$d)

#fill.col<-c("red","blue")

#install.packages("VennDiagram")

#library(VennDiagram)

#library("ggplot2")

#venn.diagram(x=x,

#filename = "vennplot.tiff",

#fill=fill.col,

#set\_names\_size = 10,text\_size = 10)

###########venn图制作########

#install.packages("ggvenn")

#install.packages("ggsci")

#library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library(ggvenn) #加载ggvenn包

gene\_set1 <- c #输入第一个基因集

gene\_set2 <- d

a <- list(`EXOSOME-DOWN` = gene\_set1$c,

 `DLBCL-DOWN` = gene\_set2$d) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

p2 <- ggvenn(a, show\_percentage = T,fill\_color = c("#00BFFF","#8A2BE2"),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

###########第一个up and down图制作########

exo <- exp\_diff\_e

dlbcl <- exp\_diff\_d

a <- filter(exo,change == 'DOWN')

b <- filter(dlbcl,change == 'UP')

c <- row.names(a)

c <- as.data.frame(c)

d <- row.names(b)

d <- as.data.frame(d)

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library(ggvenn) #加载ggvenn包

gene\_set1 <- c #输入第一个基因集

gene\_set2 <- d

a <- list(`EXOSOME-DOWN` = gene\_set1$c,

 `DLBCL-UP` = gene\_set2$d) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

p2 <- ggvenn(a, show\_percentage = T,fill\_color = c("#00BFFF","#FFD700"),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

###########第二个up+down#######

exo <- exp\_diff\_e

dlbcl <- exp\_diff\_d

a <- filter(exo,change == 'UP')

b <- filter(dlbcl,change == 'DOWN')

c <- row.names(a)

c <- as.data.frame(c)

d <- row.names(b)

d <- as.data.frame(d)

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library(ggvenn) #加载ggvenn包

gene\_set1 <- c #输入第一个基因集

gene\_set2 <- d

a <- list(`EXOSOME-UP` = gene\_set1$c,

 `DLBCL-DOWN` = gene\_set2$d) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

p2 <- ggvenn(a, show\_percentage = T,fill\_color = c("#008b45","#8a2be2"),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

#################GO#########

########hsa\_miR\_23a\_5p##########

setwd("GO")

setwd("hsa\_miR\_23a\_5p")

library(tidyverse)

library(BiocManager)

#安装加载包

#BiocManager::install('clusterProfiler')

#BiocManager::install('org.Hs.eg.db')

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

#导入DEG\_final.txt

#导入immune或stromal差异分析结果 均可

#(intersect\_1,intersect\_2,intersect\_3,intersect\_4,intersect\_5,

 intersect\_6,intersect\_7,intersect\_8,intersect\_9,intersect\_10,

 intersect\_11,intersect\_12,intersect\_13,intersect\_14,intersect\_15)

#SET <- rbind(intersect\_1,intersect\_2,intersect\_3)

#write.csv(SET, "SET.csv")

library(dplyr)

DEG <- as.data.frame(hsa\_miR\_23a\_5p)

duplicated(DEG$RNA)

DEG <- distinct(DEG)

write.csv(DEG, "hsa\_miR\_23a\_5p\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_23a\_5p\_ENTREZID")

DEG <- hsa\_miR\_23a\_5p\_ENTREZID

####GO画图####

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.2,

 qvalueCutoff =0.2,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_23a\_5p\_GO")

library(ggnewscale)

#install.packages("ggnewscale")

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =10,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

####KEGG-hsa\_miR\_23a\_5p####

#install.packages("GOplot")

library(clusterProfiler)#GO富集分析、KEGG通路富集分析

library(org.Hs.eg.db)#基因注释数据库

library(enrichplot)

library(ggplot2)

library(GOplot)

library(KEGG.db)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.2,

 qvalueCutoff =0.2)

kk\_res <- kk@result

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

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

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_23a\_5p.Rda")

##############GO-hsa\_miR\_3147##########

setwd("GO")

setwd("hsa\_miR\_3147")

library(tidyverse)

library(BiocManager)

#安装加载包

#BiocManager::install('clusterProfiler')

#BiocManager::install('org.Hs.eg.db')

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

library(dplyr)

DEG <- as.data.frame(hsa\_miR\_3147\_ENTREZID)

duplicated(DEG$RNA)

DEG <- distinct(DEG)

write.csv(DEG, "hsa\_miR\_3147\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

count(DEG)

count(SET)

DEG <- rownames\_to\_column(DEG,var = 'No')

DEG <- column\_to\_rownames(DEG,var = 'No')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

DEG <- rownames\_to\_column(DEG,"SYMBOL")

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_3147\_ENTREZID")

####GO画图####

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.10,

 qvalueCutoff =0.10,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_3147.Rda")

library(ggnewscale)

#install.packages("ggnewscale")

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =9,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

######KEGG-hsa\_miR\_3147####

DEG <- as.data.frame(hsa\_miR\_4458)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.2,

 qvalueCutoff =0.2)

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

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

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_3147.Rda")

#############GO-hsa\_miR\_4458########

setwd("GO")

setwd("hsa\_miR\_4458")

library(dplyr)

library(tidyverse)

library(BiocManager)

library(org.Hs.eg.db)

library(clusterProfiler)

DEG <- as.data.frame(hsa\_miR\_4458)

DEG <- as.data.frame(DEG)

duplicated(DEG$DEG)

DEG <- distinct(DEG)

write.csv(DEG, "hsa\_miR\_4458\_test\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_4458\_ENTREZID")

DEG <- as.data.frame(hsa\_miR\_4458\_ENTREZID)

#GO画图

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.2,

 qvalueCutoff =0.2,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_4458.Rda")

#GO、KEGG作图

library(ggnewscale)

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =8,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

#DEG <- as.data.frame(hsa\_miR\_4458)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.2,

 qvalueCutoff =0.2)

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_4458\_KEGG.Rda")

##########新的思路##########

##########hub基因############

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

#install.packages("ggvenn")

library("venn") #加载ggvenn包

#remove.packages("ggvenn")

gene\_set1 <- a#输入第一个基因集

gene\_set2 <- b

gene\_set3 <- c

gene\_set4 <- d

gene\_set5 <- e

f <- list(`ClusteringCoefficient` = gene\_set1$node\_name,

 `Degree` = gene\_set2$node\_name,

 `DMNC` = gene\_set3$node\_name,

 `MCC` = gene\_set4$node\_name,

 `MNC` = gene\_set5$node\_name) #将基因集变成列表变量

p1 <- ggvenn(f,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(f, show\_percentage = T,fill\_color = pal\_npg()(5),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 5,

 text\_size = 3)

p2

##########test-GO#########

########hsa\_miR\_23a\_5p##########

setwd("GO")

setwd("test")

setwd("hsa\_miR\_23a\_5p")

library(tidyverse)

library(BiocManager)

#安装加载包

#BiocManager::install('clusterProfiler')

#BiocManager::install('org.Hs.eg.db')

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

library(dplyr)

hsa\_miR\_23a\_5p <- distinct(hsa\_miR\_23a\_5p, RNA, .keep\_all = T)#去重

DEG <- as.data.frame(hsa\_miR\_23a\_5p)

write.csv(DEG, "hsa\_miR\_23a\_5p\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_23a\_5p\_ENTREZID")

DEG <- hsa\_miR\_23a\_5p\_ENTREZID

####GO画图####

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.2,

 qvalueCutoff =0.2,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_23a\_5p\_GO")

library(ggnewscale)

#install.packages("ggnewscale")

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =10,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

####KEGG-hsa\_miR\_23a\_5p####

#install.packages("GOplot")

library(clusterProfiler)#GO富集分析、KEGG通路富集分析

library(org.Hs.eg.db)#基因注释数据库

library(enrichplot)

library(ggplot2)

library(GOplot)

library(KEGG.db)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.9,

 qvalueCutoff =0.9)

kk\_res <- kk@result

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

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

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_23a\_5p.Rda")

##############GO-hsa\_miR\_3147##########

setwd("GO")

setwd("test")

setwd("hsa\_miR\_3147")

library(tidyverse)

library(BiocManager)

#安装加载包

#BiocManager::install('clusterProfiler')

#BiocManager::install('org.Hs.eg.db')

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

library(dplyr)

hsa\_miR\_3147 <- distinct(hsa\_miR\_3147, RNA, .keep\_all = T)#去重

DEG <- as.data.frame(hsa\_miR\_3147)

write.csv(DEG, "hsa\_miR\_3147\_distinct.csv")

DEG <- as.data.frame(hsa\_miR\_3147\_ENTREZID)

DEG <- as.data.frame(hsa\_miR\_3147)

duplicated(DEG$RNA)

write.csv(DEG, "hsa\_miR\_3147\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

count(DEG)

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_3147\_ENTREZID")

####GO画图####

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.9,

 qvalueCutoff =0.9,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_3147.Rda")

library(ggnewscale)

#install.packages("ggnewscale")

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =9,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

######KEGG-hsa\_miR\_3147####

DEG <- as.data.frame(hsa\_miR\_4458)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.9,

 qvalueCutoff =0.9)

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

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

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_3147.Rda")

#############GO-hsa\_miR\_4458########

setwd("GO")

setwd("test")

setwd("hsa\_miR\_4458")

library(dplyr)

library(tidyverse)

library(BiocManager)

library(org.Hs.eg.db)

library(clusterProfiler)

DEG <- as.data.frame(hsa\_miR\_4458)

DEG <- as.data.frame(DEG)

DEG <- distinct(DEG, RNA, .keep\_all = T)#去重

write.csv(DEG, "hsa\_miR\_4458\_test\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_4458\_ENTREZID")

DEG <- as.data.frame(hsa\_miR\_4458\_ENTREZID)

#GO画图

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.1,

 qvalueCutoff =0.1,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hsa\_miR\_4458.Rda")

#GO、KEGG作图

library(ggnewscale)

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =8,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

#DEG <- as.data.frame(hsa\_miR\_4458)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.2,

 qvalueCutoff =0.2)

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_4458\_KEGG.Rda")

########筛选靶基因的hub gene#########

####213a#####

setwd("hsa\_miR\_23a\_5p")

setwd("hub")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

#install.packages("ggVennDiagram")

library("ggvenn") #加载ggvenn包

gene\_set1 <- Degree #输入第一个基因集

gene\_set2 <- MCC

gene\_set3 <- MNC

gene\_set4 <- DMNC#输入第一个基因集

#gene\_set6 <- BottleNeck

#gene\_set7 <- Closeness

#gene\_set8 <- Stress

#gene\_set9 <- EcCentricity #输入第一个基因集

#gene\_set10 <- Radiality

#gene\_set11 <- Betweenness

#gene\_set12 <- ClusteringCoefficient

a <- list(`Degree` = gene\_set1$RNA,

 `MCC` = gene\_set2$RNA,

 `MNC` = gene\_set3$RNA,

 `DMNC` = gene\_set4$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

d <- gene\_set4$RNA

g <- intersect(d,intersect(c,intersect(a,b)))

write.table(g ,"213A",sep = "\t",row.names = T,col.names = NA,quote = F)

g <- as.data.frame(g)

jiaoji <- merge(g, fanyi, by.y ="V1", by.x = "g" )

write.table(jiaoji ,"213A\_gene",sep = "\t",row.names = T,col.names = NA,quote = F)

####3147#####

setwd("hsa\_miR\_3147")

setwd("hub")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

#install.packages("ggVennDiagram")

library("ggvenn") #加载ggvenn包

gene\_set1 <- Degree #输入第一个基因集

gene\_set2 <- MCC

gene\_set3 <- MNC

gene\_set4 <- DMNC#输入第一个基因集

#gene\_set6 <- BottleNeck

#gene\_set7 <- Closeness

#gene\_set8 <- Stress

#gene\_set9 <- EcCentricity #输入第一个基因集

#gene\_set10 <- Radiality

#gene\_set11 <- Betweenness

#gene\_set12 <- ClusteringCoefficient

a <- list(`Degree` = gene\_set1$RNA,

 `MCC` = gene\_set2$RNA,

 `MNC` = gene\_set3$RNA,

 `DMNC` = gene\_set4$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

d <- gene\_set4$RNA

g <- intersect(d,intersect(c,intersect(a,b)))

write.table(g ,"3147",sep = "\t",row.names = T,col.names = NA,quote = F)

g <- as.data.frame(g)

jiaoji <- merge(g, fanyi, by.y ="V1", by.x = "g" )

write.table(jiaoji ,"213A\_gene",sep = "\t",row.names = T,col.names = NA,quote = F)

ggVennDiagram(a,category.names = c("Degree","MCC","MNC","DMNC","EPC"),

 label = "count",

 label\_color = "black",

 label\_alpha = 0,

 edge\_lty = "dashed",

 edge\_size = 1) +

 scale\_fill\_gradient(low="white",high = "#b9292b",name = "gene count")

####4458####

setwd("hsa\_miR\_4458")

setwd("hub")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

#install.packages("ggVennDiagram")

library("ggvenn") #加载ggvenn包

gene\_set1 <- Degree #输入第一个基因集

gene\_set2 <- MCC

gene\_set3 <- MNC

gene\_set4 <- DMNC#输入第一个基因集

#gene\_set6 <- BottleNeck

#gene\_set7 <- Closeness

#gene\_set8 <- Stress

#gene\_set9 <- EcCentricity #输入第一个基因集

#gene\_set10 <- Radiality

#gene\_set11 <- Betweenness

#gene\_set12 <- ClusteringCoefficient

a <- list(`Degree` = gene\_set1$RNA,

 `MCC` = gene\_set2$RNA,

 `MNC` = gene\_set3$RNA,

 `DMNC` = gene\_set4$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

d <- gene\_set4$RNA

g <- intersect(d,intersect(c,intersect(a,b)))

write.table(g ,"4458",sep = "\t",row.names = T,col.names = NA,quote = F)

g <- as.data.frame(g)

jiaoji <- merge(g, fanyi, by.y ="V1", by.x = "g" )

write.table(jiaoji ,"4458\_gene",sep = "\t",row.names = T,col.names = NA,quote = F)

########

library("VennDiagram")

library(dplyr)

a <- a$node\_name

b <- b$node\_name

c <- c$node\_name

d <- d$node\_name

e <- e$node\_name

#作图

venn.diagram(list(ClusteringCoefficient = a, Degree = b, DMNC = c, MCC = d,

 MNC = e), fill = pal\_npg()(5), cex = 1.5,filename = "Venn5.png")

g <- intersect(e,intersect(d,intersect(c,intersect(a,b))))

write.csv(g, "g.csv")#保存为csv格式

############TCGA-GETX数据##########

library(tidyverse)#加载 Error

setwd("TCGA\_GETX")

setwd("TCGA\_GTEX")

setwd("chayifenxi\_heatmap")

#安装TCGAbiolinks包

#install.packages("BiocManager")

#library(BiocManager)

#BiocManager::install("BioinformaticsFMRP/TCGAbiolinksGUI.data")

#BiocManager::install("remotes")

#BiocManager::install("ExperimentHub")

#BiocManager::install("BioinformaticsFMRP/TCGAbiolinks")

#chooseBioCmirror()

#BiocManager::install("SparseArray")

#BiocManager::install("TCGAbiolinks")

###########整理TCGA数据库counts和tpm数据#############

library(TCGAbiolinks)#加载包

cancer\_type = "TCGA-DLBC" #肿瘤类型，这里可修改癌症类型

#TCGA 肿瘤缩写：https://www.jianshu.com/p/3c0f74e85825

expquery <- GDCquery(project = cancer\_type,

 data.category = "Transcriptome Profiling",

 data.type = "Gene Expression Quantification",

 workflow.type = "STAR - Counts"

)

GDCdownload(expquery,directory = "GDCdata")

expquery2 <- GDCprepare(expquery,directory = "GDCdata",summarizedExperiment = T)

save(expquery2,file = "DLBC.gdc\_2024.rda") # 保存 rda格式

load("DLBC.gdc\_2024.rda")#导入文件，rda格式文件也可直接从文件夹双击导入

load("gene\_annotation\_2022.rda")#导入gene注释文件

table(gene\_annotation\_2022$type)#table 分组计数

#基因名称symbol ENSEMBL

#提取counts 以下三句无需掌握 tpms

counts <- expquery2@assays@data@listData[["unstranded"]]

colnames(counts) <- expquery2@colData@rownames

rownames(counts) <- expquery2@rowRanges@ranges@NAMES

#基因ID转换

counts <- counts %>%

 as.data.frame() %>%

 rownames\_to\_column("ENSEMBL") %>%

 inner\_join(gene\_annotation\_2022,"ENSEMBL") %>%

 .[!duplicated(.$symbol),]

rownames(counts) <- NULL

counts <- counts %>% column\_to\_rownames("symbol")

#拆解

#继续跑

# 保留miRNA

#table

table(counts$type)#（注：可通过table(counts$type)查看基因类型）#lncRNA

counts <- counts[counts$type == "protein\_coding",]

#counts <- counts[counts$type == "lncRNA",]

counts <- counts[,-c(1,ncol(counts))]

#counts <- counts[,-c(1,602)]

#ncol是什么

ncol(counts)

nrow(counts)

# 把TCGA barcode切割为16位字符,并去除重复样本

colnames(counts) <- substring(colnames(counts),1,16)

counts <- counts[,!duplicated(colnames(counts))]

table(substring(colnames(counts),14,16))

write.table(counts,"tcga\_dlbcl\_counts.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

# 保留01A （注：可通过table(substring(colnames(counts),14,16))查看样本类型）

counts01A <- counts[,substring(colnames(counts),14,16) == c("01A")]

counts01B <- counts[,substring(colnames(counts),14,16) == c("01")]

# 保留11A

counts11A <- counts[,substring(colnames(counts),14,16) == c("11A")]

table(substring(colnames(counts01A),14,16))

table(substring(colnames(counts11A),14,16))

####tpms####

#和counts基本一模一样

tpms <- expquery2@assays@data@listData[["tpm\_unstrand"]]

colnames(tpms) <- expquery2@colData@rownames

rownames(tpms) <- expquery2@rowRanges@ranges@NAMES

tpms <- tpms %>%

 as.data.frame() %>%

 rownames\_to\_column("ENSEMBL") %>%

 inner\_join(gene\_annotation\_2022,"ENSEMBL") %>%

 .[!duplicated(.$symbol),]

rownames(tpms) <- NULL

tpms <- tpms %>% column\_to\_rownames("symbol")

# 保留mRNA （注：可通过table(tpms$type)查看基因类型）

table(tpms$type)

tpms <- tpms[tpms$type == "protein\_coding",]

tpms <- tpms[,-c(1,ncol(tpms))]

# 把TCGA barcode切割为16位字符,并去除重复样本

colnames(tpms) <- substring(colnames(tpms),1,16)

tpms <- tpms[,!duplicated(colnames(tpms))]

# 保留01A （注：可通过table(substring(colnames(tpms),14,16))查看样本类型）

#判断counts和tpms的行列名是否一致

identical(rownames(counts),rownames(tpms))

#保存counts和tpms数据

write.table(tpms,"tcga\_dlbcl\_tpms.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

#cbind和rbind 合并 col row

#cbind之前需要确认两个数据框的行名

counts <- cbind(counts01A,counts11A)

tpms <- cbind(tpms01A,tpms11A)

write.table(counts,"counts.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(tpms,"tpms.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

####tpms\_log2####

range(tpms)#查看数据范围

tpms\_log2 <- log2(tpms+0.001)#log2转换 为什么要加1

#保存log2转换后的数据

write.table(tpms\_log2,"tcga\_dlbcl\_tpms\_log2.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

#TCGA表达谱整理完毕

####GTES数据库整理####

library(tidyverse)#加载 Error

setwd("TCGA\_GETX\_set")

setwd("GTEx\_BLOOD")

setwd("transcript")

setwd("jiaoji")

library(data.table) # 用于高效处理大数据集的库

library(dplyr) # 数据操作和转换的库

gtex.exp <- fread("gtex\_expected\_count", header = T, sep = '\t', data.table = F)

gtex.exp <- gtex\_expected\_count

dim(gtex.exp)

saveRDS(gtex.exp, file = 'gtex.exp.rds')

gtex.pro <- fread("probeMap\_gencode.v23.annotation.transcript", header = T, sep = '\t', data.table = F)

gtex.pro <- probeMap\_gencode.v23.annotation.transcript

head(gtex.pro)

dim(gtex.exp)

dim(gtex.pro)

gtex.pro <- gtex.pro[, c(1,2)]

gtex.counts <- merge(gtex.pro, gtex.exp, by.y ="sample", by.x = "id" )

#a <- rownames(gtex.pro)

#gtex.exp <- gtex.exp[a,]

#gtex.exp <- cbind(gtex.exp,gtex.pro)

#gtex.exp <- avereps(gtex.exp, ID = exp$gene)

#gtex.exp <- as.data.frame(gtex.exp)

#gtex.exp <- distinct(gtex.exp, gene, .keep\_all = T)#去重

#rownames(gtex.exp) <- NULL

#gtex.exp <- gtex.exp %>% column\_to\_rownames("gene")

#gtex.exp <- gtex.exp[ , -7846]

#gtex.counts <-gtex.counts[,-1]

#rownames(gtex.counts) <- NULL

#gtex.counts <- gtex.counts %>% column\_to\_rownames("gene")

gtex.phe <- fread("GTEX\_phenotype", header = T, sep = '\t', data.table = F)

rownames(gtex.phe) <- gtex.phe$Sample#将Sample设定为行名

colnames(gtex.phe) <- c("Sample", "body\_site\_detail (SMTSD)", "primary\_site", "gender", "patient", "cohort")

table(gtex.phe$primary\_site)

Blood<- filter(gtex.phe, primary\_site == "Blood")#筛选表达谱为血液的样本名

row.names(Blood) <- gsub("-", ".", row.names(Blood))

merge\_phe\_counts\_gtex <- intersect(rownames(Blood), colnames(gtex.counts)) # 444

gtex.s <- gtex.counts[ , c("gene", merge\_phe\_counts\_gtex)]#样本为Blood的样本提取出来

gtex.s <- as.data.frame(gtex.s)

gtex.s <- distinct(gtex.s, gene, .keep\_all = T)#去重

rownames(gtex.s) <- gtex.s$gene

gtex.s <- gtex.s[ , -1]

dim(gtex.s)

gtex.s2 <- 2^gtex.s - 1

gtex.s3 <- gtex.s2-1

GTEX.Blood.counts <- gtex.s2

dim(GTEX.Blood.counts)

rounded\_value <- round(GTEX.Blood.counts)

GTEX\_Blood\_round\_counts <- rounded\_value

range(GTEX\_Blood\_round\_counts)

write.table(GTEX\_Blood\_round\_counts,"GTEX\_Blood\_round\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

###############getx-tpm数据######

library(tidyverse)#加载 Error

setwd("TCGA\_GETX\_set")

setwd("GTEx\_BLOOD")

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

a <- intersect(colnames(gtex\_RSEM\_gene\_tpm), colnames(counts))

gtex\_RSEM\_gene\_tpm <- gtex\_RSEM\_gene\_tpm[,a]

write.table(gtex\_RSEM\_gene\_tpm,"gtex\_tpm",sep = "\t",row.names = T,col.names = NA,quote = F)

res <- gtex\_tpm

DEG <- as.data.frame(res)

b <- DEG$X

b <- as.data.frame(b)

b1 = b

b1$V1 = apply(b1[1], 1,function(x){

 str\_split(x,'[.]')[[1]][1]

}) %>% unlist()############去除小数点及后面的数字

b1 <- as.data.frame(b1)

b2 <- b1

b1 <- column\_to\_rownames(b1,var = 'V1')

name <- bitr(rownames(b1),fromType = 'ENSEMBL',toType = 'SYMBOL',OrgDb = 'org.Hs.eg.db')

name <- as.data.frame(name)

name1 <- name

name2 <- name1

name <- distinct(name, ENSEMBL, .keep\_all = T)

name1 <- distinct(name1, ENSEMBL, .keep\_all = T)

name2 <- distinct(name2, ENSEMBL, .keep\_all = T)

rownames(name) <- NULL

name <- name %>% column\_to\_rownames("ENSEMBL")

a <- intersect(rownames(b1),rownames(name))

name2$ENSEMBL2 <- name2$ENSEMBL

name2 <- name2 %>% column\_to\_rownames("ENSEMBL")

name2 <- name2[a,]

name <- as.data.frame(name)

b1$ENSEMBL2 <- b1$b

b1 <- b1[a,]

tname <- cbind(b1,name2)

rownames(tname) <- NULL

tname <- tname %>% column\_to\_rownames("b")

c <- rownames(tname)

rownames(DEG) <- NULL

DEG <- DEG %>% column\_to\_rownames("X")

DEG <- DEG[c,]

gtpm <- cbind(DEG,tname)

gtpm <- distinct(gtpm, SYMBOL, .keep\_all = T)

rownames(gtpm) <- NULL

gtpm <- gtpm %>% column\_to\_rownames("SYMBOL")

gtpm <- gtpm[,-c(445,446)]

write.table(gtpm,"gtex\_tpm\_444",sep = "\t",row.names = T,col.names = NA,quote = F)

###########getx的tpm数据在这里###########

##########tcga-gtex差异分析###########

####Day4 TCGA-GTEX数据库差异分析####

#library(BiocManager)

#BiocManager::install('edgeR')

getx <- GTEX\_Blood\_round\_counts

tcga <- tcga\_dlbcl\_counts

range(getx)

range(tcga)

a <- intersect(rownames(getx),rownames(tcga))

getx <- getx[a,]

tcga <- tcga[a,]

TCGA\_GETX\_counts <- cbind(getx,tcga)

write.table(TCGA\_GETX\_counts,"TCGA\_GETX\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

df$new\_column <- rep(df$sample, each = 1)#复制别的一列

setwd("TCGA\_GETX\_set")

setwd("GTEx\_BLOOD")

setwd("transcript")

library(limma) # 差异分析三号选手

library(edgeR)

library(tidyverse)

library(data.table) # 用于高效处理大数据集

library(dplyr) # 用于数据操作和转换

library(ggplot2) # 画图图

library(pheatmap) # 绘制热图

library(DESeq2) # 差异分析一号选手

library(edgeR) # 差异分析二号选手

library(tinyarray) # 用于绘制各种图表，今天用它绘制韦恩图

library(DESeq2)

a <- h\_distinct$X

DEG\_TCGA\_GTEX\_hub <- DEG\_TCGA\_GTEX[a,]

write.table(DEG\_TCGA\_GTEX\_hub,"DEG\_TCGA\_GTEX\_hub",sep = "\t",row.names = T,col.names = NA,quote = F)

sample\_names <- colnames(TCGA\_GETX\_counts)

counts = TCGA\_GETX\_counts

counts <- counts[rowname,]

counts <- counts[,colname]

sample\_names <- colnames(counts)

sample\_names <- as.data.frame(sample\_names)

group$Database <- c(1:492)

normal <- 444

group$Database <- ifelse(group$Database > normal,"TCGA","GETx")

group <- group[,-2]

#rownames(exp1) <- exp1$Gene\_Symbol#加上行名

#sample\_names$Type <- rep(sample\_names$Type, each = 1)

#sample\_names <- rownames\_to\_column(sample\_names,var = 'Type')

sample\_names$Type <- c(1:492)

normal <- 444

sample\_names$Type <- ifelse(sample\_names$Type > normal,"tumor","normal")

table(sample\_names )

write.table(sample\_names,"sample\_names\_TCGA\_GTEX",sep = "\t",row.names = T,col.names = NA,quote = F)

rownames(sample\_names) <- NULL

sample\_names <- column\_to\_rownames(sample\_names,var = 'sample\_names')

group = sample\_names\_TCGA\_GTEX

group <- column\_to\_rownames(group,var = 'sample\_names')

#group <- column\_to\_rownames(group,var = 'Sample')

#a <- colnames(counts)

#counts <- counts[,a]

#group <- group[a,]

#counts <- as.data.frame(counts)

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

identical(rownames(group),colnames(counts))

range(counts)

group$Type <- factor(group$Type, levels = c("normal", "tumor"))

group$Database<- factor(group$Database, levels = c("TCGA","GETx"))

table(group)

ncol(counts)

length(group)#counts <- counts + 1

counts[is.na(counts)] <- 0#将NA值转换成0

suppressMessages(library(DESeq2))

dds <- DESeqDataSetFromMatrix(countData = counts, # 表达矩阵

 colData = group, # 表达矩阵列名和分组信息的对应关系

 design = ~Type) # group为colData中的group，也就是分组信息

head(dds)

#keep <- rowSums(counts(dds) > 1) >= 2

dds <- DESeq(dds)

vst(DEG\_TCGA\_GTEX, blind = FALSE)

resultsNames(dds)

save(res,file="DEG\_TCGA\_GTEX")

####热图绘制####

DEG <- as.data.frame(res)

DEG <- as.data.frame(DEG\_TCGA\_GTEX)

write.table(DEG,"DEG\_TCGA\_GTEX",sep = "\t",row.names = T,col.names = NA,quote = F)

DEG <- as.data.frame(res)

group = sample\_names\_TCGA\_GTEX

rownames(group) <- NULL

group <- column\_to\_rownames(group,var = 'sample\_names')

n <- rownames(group)

adjusted\_before\_tpm <- adjusted\_before\_tpm[,n]

#读取表达谱

counts = dlbcl\_counts\_TCGA\_GTEX

rowname <- intersect(rownames(counts),rownames(adjusted\_before\_tpm))

colname <- intersect(colnames(counts),colnames(adjusted\_before\_tpm))

adjusted\_before\_tpm <- adjusted\_before\_tpm[rowname,]

adjusted\_before\_tpm <- adjusted\_before\_tpm[,colname]

write.table(adjusted\_before\_tpm,"adjusted\_before\_tpm\_set",sep = "\t",row.names = T,col.names = NA,quote = F)

#na.omit(adjusted\_before\_tpm)

exp <- adjusted\_before\_tpm\_set

exp <- counts

#添加上下调信息

logFC\_cutoff <- 1.5

type1 = (DEG$padj < 0.05)&(DEG$log2FoldChange < -logFC\_cutoff)

type2 = (DEG$padj < 0.05)&(DEG$log2FoldChange > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

a <- rownames(a)

b <- rownames(b)

write.table(a,"UP",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(b,"DOWN",sep = "\t",row.names = T,col.names = NA,quote = F)

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

d <- rownames(c)

write.table(c ,"DEG\_gene\_up\_down\_0",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(a ,"DEG\_gene\_up\_0",sep = "\t",row.names = T,col.names = NA,quote = F)

exp\_diff <- exp[d,]

#设置分组信息

annotation\_col <- group

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

f <-

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames =F,

 color = colorRampPalette(c(rep("royalblue",5),"white",rep("firebrick3",5)))(50),

 cluster\_cols =F,

 cluster\_rows = F,

 fontsize = 10,

 fontsize\_row=0.1,

 fontsize\_col=3)

#保存图片 调整大小

dev.off()#关闭画板

#######transcript几个差异基因的热图#######

setwd("TCGA\_GETX\_set")

setwd("transcript")

library(limma) # 差异分析三号选手

library(edgeR)

library(tidyverse)

library(data.table) # 用于高效处理大数据集

library(dplyr) # 用于数据操作和转换

library(ggplot2) # 画图图

library(pheatmap) # 绘制热图

library(DESeq2) # 差异分析一号选手

library(edgeR) # 差异分析二号选手

library(tinyarray) # 用于绘制各种图表，今天用它绘制韦恩图

library(DESeq2)

DEG <- as.data.frame(DEG\_TCGA\_GTEX)

group = sample\_names\_TCGA\_GTEX

rownames(group) <- NULL

group <- column\_to\_rownames(group,var = 'sample\_names')

exp <- TCGA\_GETX\_counts

h <- hub$RNA

h <- as.data.frame(h)

h <- distinct(h,h)

write.table(h ,"h\_distinct",sep = "\t",row.names = T,col.names = NA,quote = F)

h <- h$h

DEG <- DEG[h,]

write.table(DEG ,"deg23a",sep = "\t",row.names = T,col.names = NA,quote = F)

logFC\_cutoff <- 1.5

type1 = (DEG$padj < 0.05)&(DEG$log2FoldChange < -logFC\_cutoff)

type2 = (DEG$padj < 0.05)&(DEG$log2FoldChange > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

write.table(c ,"DEG\_gene\_up\_down\_1",sep = "\t",row.names = T,col.names = NA,quote = F)

identical(colnames(exp),rownames(group))

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

d <- rownames(c)

write.table(c ,"DEG\_gene\_up\_down\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(a ,"DEG\_gene\_up\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(b ,"DEG\_gene\_down\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

exp\_diff <- exp[d,]

#设置分组信息

annotation\_col <- group

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames =T,

 color = colorRampPalette(c(rep("royalblue",5),"white",rep("firebrick3",5)))(50),

 cluster\_cols =F,

 cluster\_rows = F,

 fontsize = 18,

 fontsize\_row=16,

 fontsize\_col=3)

#a <- intersect(DEG\_gene\_up\_0$X,all$RNA)

#a <- intersect(DEG\_gene\_up\_down\_0$X,all$RNA)

###########hub与test提取交集########

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- gene #输入第一个基因集

gene\_set2 <- DIANA\_TarBase

gene\_set3 <- miRTarBase

a <- list(`hsa\_miR\_23a\_5p` = gene\_set1$RNA,

 `DIANA\_TarBase` = gene\_set2$RNA,

 `miRTarBase` = gene\_set3$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g2 ,"23ajiaoji",sep = "\t",row.names = T,col.names = NA,quote = F)

##########已验证过的test基因#########

setwd("TCGA\_GETX\_set")

setwd("transcript")

setwd("test")

library(limma) # 差异分析三号选手

library(edgeR)

library(tidyverse)

library(data.table) # 用于高效处理大数据集

library(dplyr) # 用于数据操作和转换

library(ggplot2) # 画图图

library(pheatmap) # 绘制热图

library(DESeq2) # 差异分析一号选手

library(edgeR) # 差异分析二号选手

library(tinyarray) # 用于绘制各种图表，今天用它绘制韦恩图

library(DESeq2)

DEG <- as.data.frame(DEG\_TCGA\_GTEX)

group = sample\_names\_TCGA\_GTEX

rownames(group) <- NULL

group <- column\_to\_rownames(group,var = 'sample\_names')

exp <- TCGA\_GETX\_counts

a <- test\_gene$RNA

a <- as.data.frame(a)

a <- distinct(a,a)

write.table(a ,"a\_distinct",sep = "\t",row.names = T,col.names = NA,quote = F)

a <- a$a

DEG <- DEG[a,]

exp <- exp[a,]

logFC\_cutoff <- 1

type1 = (DEG$padj < 0.05)&(DEG$log2FoldChange < -logFC\_cutoff)

type2 = (DEG$padj < 0.05)&(DEG$log2FoldChange > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

write.table(c ,"DEG\_gene\_up\_down",sep = "\t",row.names = T,col.names = NA,quote = F)

identical(colnames(exp),rownames(group))

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

d <- rownames(c)

write.table(c ,"DEG\_gene\_up\_down\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(a ,"DEG\_gene\_up\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(b ,"DEG\_gene\_down\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

exp\_diff <- exp[d,]

#设置分组信息

annotation\_col <- group

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames =T,

 color = colorRampPalette(c(rep("royalblue",5),"white",rep("firebrick3",5)))(50),

 cluster\_cols =F,

 cluster\_rows = F,

 fontsize = 16,

 fontsize\_row=2,

 fontsize\_col=3)

########最终结果汇总-venn图########

#####hsa\_miR\_23a\_5p#########

setwd("jiaoji")

setwd("test\_gene")

setwd("hsa\_miR\_23a\_5p")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- gene #输入第一个基因集

gene\_set2 <- DIANA\_TarBase

gene\_set3 <- miRTarBase

a <- list(`hsa\_miR\_23a\_5p` = gene\_set1$RNA,

 `DIANA\_TarBase` = gene\_set2$RNA,

 `miRTarBase` = gene\_set3$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g2 ,"23ajiaoji",sep = "\t",row.names = T,col.names = NA,quote = F)

#####hsa\_miR\_3147#########

setwd("jiaoji")

setwd("test\_gene")

setwd("hsa\_miR\_3147")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- gene #输入第一个基因集

gene\_set2 <- miRTarBase

gene\_set3 <- NPInter

a <- list(`hsa\_miR\_3147` = gene\_set1$RNA,

 `miRTarBase` = gene\_set2$RNA,

 `NPInter` = gene\_set3$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g ,"3147\_jiaoji\_miRTarBase",sep = "\t",row.names = T,col.names = NA,quote = F)

#####hsa\_miR\_4458#########

setwd("jiaoji")

setwd("test\_gene")

setwd("hsa\_miR\_4458")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- gene#输入第一个基因集

gene\_set2 <- miRTarBase

gene\_set3 <- NPInter

gene\_set4 <- DIANA\_TarBase

a <- list(`hsa\_miR\_4458` = gene\_set1$RNA,

 `miRTarBase` = gene\_set2$RNA,

 `NPInter` = gene\_set3$RNA,

 `DIANA\_TarBase` = gene\_set4$RNA) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 6)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$RNA

c <- gene\_set3$RNA

d <- gene\_set4$RNA

g <- intersect(a,b)

g2 <- intersect(a,c)

g3 <- intersect(a,d)

write.table(g ,"4458\_jiaoji\_miRTarBase",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(g2 ,"4458\_jiaoji\_NPInter",sep = "\t",row.names = T,col.names = NA,quote = F)

#####靶基因与TCGA库#########

#####hsa\_miR\_23a\_5p#####

setwd("jiaoji")

library("ggsci")

library("dplyr")

library("grid")

library("ggplot2")

library("futile.logger")

library("ggvenn") #加载ggvenn包

gene\_set1 <- hsa\_miR\_23a\_5p #输入第一个基因集

gene\_set2 <- UP

gene\_set3 <- DOWN

a <- list(`hsa\_miR\_23a\_5p` = gene\_set1$RNA,

 `DLBCL up-regulate gene` = gene\_set2$x,

 `DLBCL down-regulate gene` = gene\_set3$x) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 5)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$x

c <- gene\_set3$x

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g,"hsa\_miR\_23a\_5p\_tcga\_up",sep = "\t",row.names = T,col.names = NA,quote = F)

#####hsa\_miR\_3147#####

#输入第一个基因集

gene\_set1 <- hsa\_miR\_3147

gene\_set2 <- UP

gene\_set3 <- DOWN

a <- list(`hsa\_miR\_3147` = gene\_set1$RNA,

 `DLBCL up-regulate gene` = gene\_set2$x,

 `DLBCL down-regulate gene` = gene\_set3$x) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 5)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$x

c <- gene\_set3$x

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g,"hsa\_miR\_3147\_tcga\_up",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(g2,"hsa\_miR\_3147\_tcga\_down",sep = "\t",row.names = T,col.names = NA,quote = F)

#####hsa\_miR\_4458#####

gene\_set1 <- hsa\_miR\_4458

gene\_set2 <- UP

gene\_set3 <- DOWN

a <- list(`hsa\_miR\_4458` = gene\_set1$RNA,

 `DLBCL up-regulate gene` = gene\_set2$x,

 `DLBCL down-regulate gene` = gene\_set3$x) #将基因集变成列表变量

p1 <- ggvenn(a,show\_percentage = T)

p1

#c("#00BFFF","#8A2BE2","#008B45","#FFD700")

p2 <- ggvenn(a, show\_percentage = T,fill\_color = pal\_npg()(4),

 label\_sep = "\n", stroke\_size = 1.5,set\_name\_size = 7,

 text\_size = 5)

p2

a <- gene\_set1$RNA #输入第一个基因集

b <- gene\_set2$x

c <- gene\_set3$x

g <- intersect(a,b)

g2 <- intersect(a,c)

write.table(g,"hsa\_miR\_4458\_tcga\_up",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(g2,"hsa\_miR\_4458\_tcga\_down",sep = "\t",row.names = T,col.names = NA,quote = F)

######绘制火山图########

getwd()

df<-read.csv("DEG.CSV",header=T,

 stringsAsFactors = F)

df<-DEG

df = na.omit(df)

head(df)

dim(df)

df$group<-ifelse(df$logFC > 1.5&df$adj.P.Val<0.05,"Up",

 ifelse(df$logFC < -1.5&df$adj.P.Val<=0.05,

 "Down","Not sig"))

table(df$group)

library(ggplot2)

#install.packages("ggrepel")

library(ggrepel)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))

df$adj.P.Val\_log10<-(-log10(df$adj.P.Val))

df <- df %>% rownames\_to\_column("X")

df1<-df[df$adj.P.Val\_log10 > 1.25,]

df2<-df1[abs(df1$logFC) > 1.5,]

dim(df2)

ggplot(df,aes(x=logFC,y=-log10(adj.P.Val)))+

 geom\_point(aes(color=group))+

 scale\_color\_manual(values=c("dodgerblue","gray","firebrick"))+

 geom\_label\_repel(data=df2,aes(x=logFC,y=-log10(adj.P.Val),

 label=X))+

 theme\_bw()+

 labs(y="-log10(adj.P.Val)",x="log(Fold Change)")+

 xlim(c(-7.5,7.5))

write.csv(df, "DEG-DLBCL-mirna.csv")

#install.packages("sva")

#library(nlme)

#library(mgcv)

#library(genefilter)

#library(BiocParallel)

#library(sva)

###########验证miRNA的上调##########

######tcga######

setwd("miRNA")

setwd("tcga")

library(tidyverse)

library(limma)

library(GEOquery)

library(ggplot2)

library(ggsci)

library(TCGAbiolinks)#加载包

cancer\_type = "TCGA-DLBC" #肿瘤类型，这里可修改癌症类型

#TCGA 肿瘤缩写：https://www.jianshu.com/p/3c0f74e85825

expquery <- GDCquery(project = cancer\_type,

 data.category = "Transcriptome Profiling",

 data.type = "Gene Expression Quantification",

 workflow.type = "STAR - Counts"

)

GDCdownload(expquery,directory = "GDCdata")

expquery2 <- GDCprepare(expquery,directory = "GDCdata",summarizedExperiment = T)

save(expquery2,file = "DLBC.gdc\_2024.rda") # 保存 rda格式

load("DLBC.gdc\_2024.rda")#导入文件，rda格式文件也可直接从文件夹双击导入

load("gene\_annotation\_2022.rda")#导入gene注释文件

table(gene\_annotation\_2022$type)#table 分组计数

counts <- expquery2@assays@data@listData[["unstranded"]]

colnames(counts) <- expquery2@colData@rownames

rownames(counts) <- expquery2@rowRanges@ranges@NAMES

#基因ID转换

counts <- counts %>%

 as.data.frame() %>%

 rownames\_to\_column("ENSEMBL") %>%

 inner\_join(gene\_annotation\_2022,"ENSEMBL") %>%

 .[!duplicated(.$symbol),]

rownames(counts) <- NULL

counts <- counts %>% column\_to\_rownames("symbol")

#拆解

#继续跑

# 保留miRNA

#table

table(counts$type)#（注：可通过table(counts$type)查看基因类型）#lncRNA

counts <- counts[counts$type == "miRNA",]

#counts <- counts[counts$type == "lncRNA",]

counts <- counts[,-c(1,ncol(counts))]

#counts <- counts[,-c(1,602)]

#ncol是什么

ncol(counts)

nrow(counts)

# 把TCGA barcode切割为16位字符,并去除重复样本

colnames(counts) <- substring(colnames(counts),1,16)

counts <- counts[,!duplicated(colnames(counts))]

table(substring(colnames(counts),14,16))

write.table(counts,"tcga\_counts.txt",sep = "\t",row.names = T,col.names = NA,quote = F)

######getx######

setwd("miRNA")

setwd("getx")

library(tidyverse)#加载 Error

library(data.table) # 用于高效处理大数据集的库

library(dplyr) # 数据操作和转换的库

gtex.exp <- fread("gtex\_expected\_count", header = T, sep = '\t', data.table = F)

gtex.exp <- gtex\_expected\_count

dim(gtex.exp)

saveRDS(gtex.exp, file = 'gtex.exp.rds')

gtex.pro <- fread("probeMap\_gencode.v23.annotation.transcript", header = T, sep = '\t', data.table = F)

gtex.pro <- probeMap\_gencode.v23.annotation.transcript

head(gtex.pro)

dim(gtex.exp)

dim(gtex.pro)

gtex.pro <- gtex.pro[, c(1,2)]

gtex.counts <- merge(gtex.pro, gtex.exp, by.y ="sample", by.x = "id" )

#a <- rownames(gtex.pro)

#gtex.exp <- gtex.exp[a,]

#gtex.exp <- cbind(gtex.exp,gtex.pro)

#gtex.exp <- avereps(gtex.exp, ID = exp$gene)

#gtex.exp <- as.data.frame(gtex.exp)

#gtex.exp <- distinct(gtex.exp, gene, .keep\_all = T)#去重

#rownames(gtex.exp) <- NULL

#gtex.exp <- gtex.exp %>% column\_to\_rownames("gene")

#gtex.exp <- gtex.exp[ , -7846]

#gtex.counts <-gtex.counts[,-1]

#rownames(gtex.counts) <- NULL

#gtex.counts <- gtex.counts %>% column\_to\_rownames("gene")

gtex.phe <- fread("GTEX\_phenotype", header = T, sep = '\t', data.table = F)

rownames(gtex.phe) <- gtex.phe$Sample#将Sample设定为行名

colnames(gtex.phe) <- c("Sample", "body\_site\_detail (SMTSD)", "primary\_site", "gender", "patient", "cohort")

table(gtex.phe$primary\_site)

Blood<- filter(gtex.phe, primary\_site == "Blood")#筛选表达谱为血液的样本名

row.names(Blood) <- gsub("-", ".", row.names(Blood))

merge\_phe\_counts\_gtex <- intersect(rownames(Blood), colnames(gtex.counts)) # 444

gtex.s <- gtex.counts[ , c("gene", merge\_phe\_counts\_gtex)]#样本为Blood的样本提取出来

gtex.s <- as.data.frame(gtex.s)

gtex.s <- distinct(gtex.s, gene, .keep\_all = T)#去重

rownames(gtex.s) <- gtex.s$gene

gtex.s <- gtex.s[ , -1]

dim(gtex.s)

gtex.s2 <- 2^gtex.s - 1

GTEX.Blood.counts <- gtex.s2

dim(GTEX.Blood.counts)

rounded\_value <- round(GTEX.Blood.counts)

GTEX\_Blood\_round\_counts <- rounded\_value

range(GTEX\_Blood\_round\_counts)

write.table(GTEX\_Blood\_round\_counts,"GTEX\_Blood\_round\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

##########tcga-getx的miRNA交集#########

getx <- GTEX\_Blood\_round\_counts

tcga <- tcga\_counts

getx\_mirna <- intersect(rownmes(getx),rown <- mes(tcga))

getx <- as.data.frame(getx\_mirna)

getx <- getx[getx\_mirna,]

write.table(getx,"getx\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

tcga <- tcga[getx\_mirna,]

write.table(tcga,"tcga\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

TCGA\_GETX\_counts <- cbind(getx,tcga)

write.table(TCGA\_GETX\_counts,"TCGA\_GETX\_counts",sep = "\t",row.names = T,col.names = NA,quote = F)

df$new\_column <- rep(df$sample, each = 1)#复制别的一列

library(limma) # 差异分析三号选手

library(edgeR)

library(tidyverse)

library(data.table) # 用于高效处理大数据集

library(dplyr) # 用于数据操作和转换

library(ggplot2) # 画图图

library(pheatmap) # 绘制热图

library(DESeq2) # 差异分析一号选手

library(edgeR) # 差异分析二号选手

library(tinyarray) # 用于绘制各种图表，今天用它绘制韦恩图

library(DESeq2)

sample\_names <- colnames(TCGA\_GETX\_counts)

counts = TCGA\_GETX\_counts

sample\_names <- as.data.frame(sample\_names)

group <- sample\_names

group$Type <- c(1:492)

normal <- 444

group$Type <- ifelse(group$Type > normal,"tumor","normal")

group <- group[,-2]

#rownames(exp1) <- exp1$Gene\_Symbol#加上行名

#sample\_names$Type <- rep(sample\_names$Type, each = 1)

group <- column\_to\_rownames(group,var = 'sample\_names')

table(sample\_names )

write.table(group,"sample\_names\_TCGA\_GTEX",sep = "\t",row.names = T,col.names = NA,quote = F)

group = sample\_names\_TCGA\_GTEX

identical(rownames(group),colnames(counts))

range(counts)

group$Type <- factor(group$Type, levels = c("normal", "tumor"))

table(group)

ncol(counts)

length(group)

#estimateSizeFactors(dds, type = 'iterate')#牛逼！

counts <- counts + 1

counts[is.na(counts)] <- 0

suppressMessages(library(DESeq2))

dds <- DESeqDataSetFromMatrix(countData = counts, # 表达矩阵

 colData = group, # 表达矩阵列名和分组信息的对应关系

 design = ~Type) # group为colData中的group，也就是分组信息

head(dds)

#keep <- rowSums(counts(dds) > 1) >= 2

dds <- DESeq(dds)

resultsNames(dds)

res <- results(dds)

save(res,file="DEG\_TCGA\_GTEX")

DEG <- as.data.frame(res)

logFC\_cutoff <- 1.5

type1 = (DEG$padj < 0.05)&(DEG$log2FoldChange < -logFC\_cutoff)

type2 = (DEG$padj < 0.05)&(DEG$log2FoldChange > logFC\_cutoff)

DEG$change = ifelse(type1,"DOWN",ifelse(type2,"UP","NOT"))

table(DEG$change)

write.table(c ,"DEG\_gene\_up\_down",sep = "\t",row.names = T,col.names = NA,quote = F)

identical(colnames(exp),rownames(group))

#下载pheatmap包

#install.packages("pheatmap")

library(pheatmap)

#提取差异基因表达谱

a <- filter(DEG,change == 'UP')

b <- filter(DEG,change == 'DOWN')

c <- rbind(a,b)

d <- rownames(c)

write.table(c ,"DEG\_gene\_up\_down\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

write.table(a ,"DEG\_gene\_up\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

exp\_diff <- counts[d,]

#设置分组信息

annotation\_col <- group

#对exp\_diff 列的顺序进行处理

a <- filter(annotation\_col,Type == 'tumor')

b <- filter(annotation\_col,Type == 'normal')

exp\_diff\_high <- exp\_diff[,rownames(a)]

exp\_diff\_low <- exp\_diff[,rownames(b)]

exp\_diff <- cbind(exp\_diff\_high,exp\_diff\_low)

#开始画图

pheatmap(exp\_diff,

 annotation\_col=annotation\_col,

 scale = "row",

 show\_rownames = T,

 show\_colnames =T,

 color = colorRampPalette(c(rep("royalblue",5),"white",rep("firebrick3",5)))(50),

 cluster\_cols =F,

 cluster\_rows = F,

 fontsize = 16,

 fontsize\_row=10,

 fontsize\_col=3)

#####23a高低表达临床生存图####

library(tidyverse)

setwd("miRNA\_suv\_chengshumirna")

mirna <- miRNA.RPM1848

cli <- TCGA\_DLBC\_phenotype

colnames(mirna) <- substring(colnames(mirna),1,16)

colnames(mirna) <- gsub("\\.","-",colnames(mirna))

a <- intersect(rownames(cli),colnames(mirna))

cli <- cli[a,]

mirna <- mirna[,a]

identical(rownames(cli),colnames(mirna))

library(utils)

library(survival)

cli$days\_to\_last\_follow\_up.diagnoses <- cli$days\_to\_last\_follow\_up.diagnoses/365

cli <- t(cli)

cli <- as.data.frame(cli)

identical(colnames(cli),colnames(mirna))

mirna <- rbind(mirna,cli)

mirna <- t(mirna)

write.table(mirna ,"dlbcl\_miRNA",sep = "\t",row.names = T,col.names = NA,quote = F)

mirna <- as.data.frame(mirna)

mirna$Group <- ifelse(mirna$`hsa-miR-23a-5p` > median(mirna$`hsa-miR-23a-5p`),"High","Low")

class(mirna$Group)

mirna$Group <- factor(mirna$Group, levels = c("Low","High"))

table(mirna$Group)

mirna$mir23a <- rep(mirna$`hsa-miR-23a-5p`, each = 1)#复制别的一列

mirna <- mirna[,-c(1:1848)]

write.table(mirna ,"c\_miRNA\_23a",sep = "\t",row.names = T,col.names = NA,quote = F)

fitd <- survdiff(Surv(days\_to\_last\_follow\_up.diagnoses, OS) ~ Group,

 data = mirna,

 na.action = na.exclude)

pValue <- 1 - pchisq(fitd$chisq, length(fitd$n) - 1)

fit <- survfit(Surv(days\_to\_last\_follow\_up.diagnoses, OS)~ Group, data = mirna)

summary(fit)

p.lab <- paste0("P", ifelse(pValue < 0.001, " < 0.001", paste0(" = ",round(pValue, 3))))

#install.packages("survminer")

library(survminer)

ggsurvplot(fit,

 data = mirna,

 pval = p.lab,

 conf.int = TRUE, # 显示置信区间

 risk.table = TRUE, # 显示风险表

 risk.table.col = "strata",

 palette = "lancet", # 配色采用

 legend.labs = c("Low", "High"), # 图例

 size = 1,

 xlim = c(0,20), # x轴长度

 break.time.by = 5, # x轴步长为5

 legend.title = "hsa-miR-3147",

 surv.median.line = "hv", # 限制垂直和水平的中位生存

 ylab = "Survival probability (%)", # 修改y轴标签

 xlab = "Time (Years)", # 修改x轴标签

 ncensor.plot = TRUE, # 显示删失图块

 ncensor.plot.height = 0.25,

 risk.table.y.text = FALSE)

########临床数据处理#######

setwd("miRNA\_suv\_chengshumirna")

setwd("TCGA\_DLBC.GDC\_phenotype.tsv")

a <- rownames(c\_miRNA\_23a\_1)

c\_miRNA\_23a <- c\_miRNA\_23a[a,]

identical(rownames(c\_miRNA\_23a\_1),rownames(c\_miRNA\_23a))

write.table(c\_miRNA\_23a ,"c\_miRNA\_23a\_2",sep = "\t",row.names = T,col.names = NA,quote = F)

#####高低表达临床生存图####

library(tidyverse)

setwd("miRNA\_suv\_chengshumirna")

mirna <- tcga\_dlbcl\_tpms

cli <- TCGA\_DLBC\_phenotype

colnames(mirna) <- substring(colnames(mirna),1,16)

colnames(mirna) <- gsub("\\.","-",colnames(mirna))

a <- intersect(rownames(cli),colnames(mirna))

cli <- cli[a,]

mirna <- mirna[,a]

identical(rownames(cli),colnames(mirna))

library(utils)

library(survival)

cli$days\_to\_last\_follow\_up.diagnoses <- cli$days\_to\_last\_follow\_up.diagnoses/365

cli <- t(cli)

cli <- as.data.frame(cli)

identical(colnames(cli),colnames(mirna))

mirna <- rbind(mirna,cli)

mirna <- t(mirna)

write.table(mirna ,"dlbcl\_mRNA",sep = "\t",row.names = T,col.names = NA,quote = F)

mirna <- as.data.frame(mirna)

mirna$Group <- ifelse(mirna$SNRPD1 > median(mirna$SNRPD1),"High","Low")

class(mirna$Group)

mirna$Group <- factor(mirna$Group, levels = c("Low","High"))

table(mirna$Group)

mirna$snrpd1 <- rep(mirna$SNRPD1, each = 1)#复制别的一列

mirna <- mirna[,-c(1:19934)]

write.table(mirna ,"c\_mRNA\_SNRPD1",sep = "\t",row.names = T,col.names = NA,quote = F)

fitd <- survdiff(Surv(days\_to\_last\_follow\_up.diagnoses, OS) ~ Group,

 data = mirna,

 na.action = na.exclude)

pValue <- 1 - pchisq(fitd$chisq, length(fitd$n) - 1)

fit <- survfit(Surv(days\_to\_last\_follow\_up.diagnoses, OS)~ Group, data = mirna)

summary(fit)

p.lab <- paste0("P", ifelse(pValue < 0.001, " < 0.001", paste0(" = ",round(pValue, 3))))

#install.packages("survminer")

library(survminer)

ggsurvplot(fit,

 data = mirna,

 pval = p.lab,

 conf.int = TRUE, # 显示置信区间

 risk.table = TRUE, # 显示风险表

 risk.table.col = "strata",

 palette = "lancet", # 配色采用

 legend.labs = c("Low", "High"), # 图例

 size = 1,

 xlim = c(0,20), # x轴长度

 break.time.by = 5, # x轴步长为5

 legend.title = "SNRPD1",

 surv.median.line = "hv", # 限制垂直和水平的中位生存

 ylab = "Survival probability (%)", # 修改y轴标签

 xlab = "Time (Years)", # 修改x轴标签

 ncensor.plot = TRUE, # 显示删失图块

 ncensor.plot.height = 0.25,

 risk.table.y.text = FALSE)

####hub gene#####

setwd("GO")

setwd("hsa\_miR\_23a\_5p")

library(tidyverse)

library(BiocManager)

#安装加载包

#BiocManager::install('clusterProfiler')

#BiocManager::install('org.Hs.eg.db')

library(org.Hs.eg.db)

#org.Hs.eg.db包主要注释人类基因:用于不同数据库ID间的转化

library(clusterProfiler)

#导入DEG\_final.txt

#导入immune或stromal差异分析结果 均可

#(intersect\_1,intersect\_2,intersect\_3,intersect\_4,intersect\_5,

intersect\_6,intersect\_7,intersect\_8,intersect\_9,intersect\_10,

intersect\_11,intersect\_12,intersect\_13,intersect\_14,intersect\_15)

#SET <- rbind(intersect\_1,intersect\_2,intersect\_3)

#write.csv(SET, "SET.csv")

library(dplyr)

DEG <- as.data.frame(hub29)

duplicated(DEG$RNA)

DEG <- distinct(DEG)

write.csv(DEG, "hsa\_miR\_23a\_5p\_distinct.csv")

DEG <- column\_to\_rownames(DEG,var = 'RNA')

DEG <- rownames\_to\_column(DEG,var = 'SYMBOL')

genelist <- bitr(DEG$SYMBOL, fromType="SYMBOL",

 toType="ENTREZID", OrgDb='org.Hs.eg.db')

DEG <- inner\_join(DEG,genelist,by="SYMBOL")

DEG <- DEG[,-2]

write.csv(DEG, "hsa\_miR\_23a\_5p\_ENTREZID")

DEG <- hsa\_miR\_23a\_5p\_ENTREZID

####GO画图####

ego <- enrichGO(gene = DEG$ENTREZID,

 OrgDb = org.Hs.eg.db,

 ont = "all",

 pAdjustMethod = "BH",

 minGSSize = 1,

 pvalueCutoff =0.9,

 qvalueCutoff =0.9,

 readable = TRUE)

ego\_res <- ego@result

save(ego,ego\_res,file = "hub\_GO")

library(ggnewscale)

#install.packages("ggnewscale")

cnetplot(ego,circular = TRUE, colorEdge = TRUE)

barplot(ego, drop = TRUE, showCategory =10,split="ONTOLOGY") +

 facet\_grid(ONTOLOGY~., scale='free')

####KEGG-hsa\_miR\_23a\_5p####

#install.packages("GOplot")

library(clusterProfiler)#GO富集分析、KEGG通路富集分析

library(org.Hs.eg.db)#基因注释数据库

library(enrichplot)

library(ggplot2)

library(GOplot)

library(KEGG.db)

kk <- enrichKEGG(gene = DEG$ENTREZID,

 organism = 'hsa',

 pvalueCutoff = 0.1,

 qvalueCutoff =0.1)

kk\_res <- kk@result

kk=DOSE::setReadable(kk, OrgDb='org.Hs.eg.db',keyType='ENTREZID')

kk\_res <- kk@result

write.table(kk\_res,file="kk\_hub29.txt",sep="\t",quote=F,row.names = F)

cnetplot(kk, circular = TRUE, colorEdge = TRUE)

barplot(kk, drop = TRUE, showCategory =10)

save(kk,kk\_res,file = "hsa\_miR\_23a\_5p.Rda")