**\*\*\*程序说明\*\*\***

\*\*\*源程序中包含四个脚本，分别为：

\*\*\*CREfinder.r

\*\*\*demo\_run\_atac.sh

\*\*\*demo\_run\_rna.sh

\*\*\*demo\_run\_integration.sh

其中CREfinder.r是软件的主程序。使用时可参考示例程序demo\_run\_atac.sh、demo\_run\_rna.sh和demo\_run\_integration.sh修改对应的输入文件和参数，然后依次执行示例程序即可。

CREfinder.r

########################################################################################

# Author: xiaohan2@genomics.cn

# Date: 2023-09-19 12:00:00

# LastEditors: xiaohan2@genomics.cn

# LastEditTime: 2023-09-19 12:00:00

# Description: 单细胞类型或多细胞类型特异调控元件鉴定筛选

# Copyright 2023 by xiaohan2 All Rights Reserved.

########################################################################################

### >>> 0. 参数传递

#-------------------------------------------------------------------------------

library(optparse)

# 描述参数

option\_list <- list(

make\_option(c("--Mode"), type = "character", default = FALSE,

action = "store", help = "Mode: RNA or ATAC or Integration"

),

make\_option(c("--ExpMatrix"), type = "character", default = FALSE,

action = "store", help = "TPM matrix (for RNA-Seq)/CPM matrix (for ATAC-Seq) file path"

),

make\_option(c("--SampleMeta"), type = "character", default = FALSE,

action = "store", help = "Sample classification"

),

make\_option(c("--Features"), type = "character", default = FALSE,

action = "store", help = "DEGs (for RNA-Seq) or DORs (for ATAC-Seq)"

),

make\_option(c("--TargetCelltype"), type = "character", default = FALSE,

action = "store", help = "Interested cell types of CREs file path"

),

make\_option(c("--Outdir"), type = "character", default = FALSE,

action = "store", help = "Result directory"

),

make\_option(c("--PeakGene"), type = "character", default = FALSE,

action = "store", help = "Peak to geneId to geneSymbol"

),

make\_option(c("--AUCtbl"), type = "character", default = FALSE,

action = "store", help = "AUC file path of concerned cell types"

)

)

# 解析参数

args <- parse\_args(OptionParser(option\_list = option\_list, usage = "This script is a test for arguments!"))

library(openxlsx)

library(stringr)

library(dplyr)

library(ggplot2)

library(ComplexHeatmap)

library(grid)

library(circlize)

library(GetoptLong)

library(ggpubr)

library(ROCR)

library(caret)

library(glmnet)

library(randomForest)

library(pROC)

library(ggplot2)

### >>> Main <<<

#-------------------------------------------------------------------------------

if (args$Mode=="RNA") {

## 1. 读入数据

#-----------------------------------------------------------------------------

tpm <- read.table(args$ExpMatrix,header = T)

metadata <- read.table(args$SampleMeta,header = T)

colnames(metadata) <- c("sample","group","sample.id")

rownames(metadata) <- metadata$sample

marker <- read.table(args$Features,header = TRUE,sep="\t")

colnames(marker) <- c("id","source")

concern.celltype <- read.table(args$TargetCelltype,header = F,sep="\t")$V1

outdir <- args$Outdir

if(!dir.exists(outdir)){dir.create(outdir,recursive = T)}

setwd(outdir)

## 2. 细胞类型分为关注和背景

#-----------------------------------------------------------------------------

all.celltype <- unique(metadata$group)

backgound.celltpe <- setdiff(all.celltype,concern.celltype)

celltype <- concern.celltype

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

## 3. 随机样本扩增

#-----------------------------------------------------------------------------

cat("Processing: ",celltype[i],"\n")

set.seed(i)

celltype.df <- metadata[metadata$group %in% celltype[i],]

others.df <- metadata[metadata$group %in% backgound.celltpe,]

sample.num <- dim(others.df)[1] - dim(celltype.df)[1]

expand.srr <- sample(celltype.df$sample,size = sample.num,replace = T)

expand.celltype.df <- metadata[expand.srr,]

expand.celltype.df$sample.id <- paste(celltype[i],nrow(celltype.df):(nrow(others.df)-1),sep = ".")

all.celltype.df <- rbind(celltype.df,expand.celltype.df)

final.df <- rbind(all.celltype.df,others.df)

rownames(final.df) <- final.df$sample.id

out.tpm <- tpm[,final.df$sample]

colnames(out.tpm) <- final.df$sample.id

final.df$Type <- "CASE"

final.df$Type[!final.df$group %in% celltype[i]] <- "CTRL"

colnames(final.df)[3] <- "Sample"

sample.classification <- final.df[,c(3,4)]

out.tpm <- out.tpm %>% mutate(Gene\_ID=rownames(out.tpm),.before = 1)

write.table(out.tpm,paste0(celltype[i],"\_expand.tpm.txt"),quote = F,sep = "\t",row.names = F,col.names = T)

write.table(sample.classification,paste0(celltype[i],"\_sample.classification.txt"),quote = F,sep = "\t",row.names = F,col.names = T)

## 4. 机器学习特征筛选

#-----------------------------------------------------------------------------

df <- out.tpm

id2symbol <- df$Gene\_ID

gid <- sapply(strsplit(id2symbol,"|",fixed = T),"[[",1)

rownames(df) <- gid

df$Gene\_ID <- gid

type <- sample.classification

genes <- marker %>% filter(source %in% celltype[i]) %>% select(id)

genes$id <- as.character(genes$id)

data <- merge(y=df,x = genes,by.y = "Gene\_ID",by.x = "id",sort = F)

rownames(data) <- data$id

#anno <- data[,1:4]

data <- data[,-1]

data\_t <- as.data.frame(t(data))

data\_t$outcom <- type$Type

data\_t$outcom[data\_t$outcom=="CASE"] <- 1

data\_t$outcom[data\_t$outcom=="CTRL"] <- 0

data\_t$outcom <- as.numeric(data\_t$outcom)

#getwd()

outdf <- as.data.frame(matrix(nrow = (ncol(data\_t)-1),ncol = 2))

colnames(outdf) <- c("id","auc")

for (j in 1:(ncol(data\_t)-1)) {

df1 <- data\_t[,c(j,j,ncol(data\_t))]

nam <- colnames(data\_t)[j]

colnames(df1)[ncol(df1)]="y"

set.seeds=1

folds <- createFolds(y=df1[,"y"],k=10)

folds10=data.frame()

for(k in 1:10){

test <- df1[folds[[k]],]

train <- df1[-folds[[k]],]

train\_matrix <- as.matrix(train[, -ncol(train)])

test\_matrix <- as.matrix(test[, -ncol(test)])

#lasso

cvfit <- cv.glmnet(train\_matrix, train$y, family = "gaussian", standardize = TRUE,nlambda = 1000, nfolds = 10, alpha = 1)

lambda\_min<-cvfit$lambda.min

pred.lasso <- predict(cvfit, test\_matrix, type="response", s="lambda.min")

glm.fit <- glm(y~., data=train)

#pred.glm1 <- predict(glm.fit1,test1)

glm.step=step(glm.fit)

pred.glm <- predict(glm.step,test)

re=as.matrix(cbind(test[,ncol(test)] ,pred.lasso,pred.glm))

re=as.data.frame(re)

colnames(re)=c('event','pred\_lasso','pred\_glm')

folds10<-rbind(folds10,re)

}

predob<- prediction(folds10$pred\_glm, folds10$event)

perf.auc<- performance(predob, measure = 'auc', x.measure = 'cutoff')

perf<- performance(predob, 'tpr','fpr')

perf.auc1 <- perf.auc

auc <- round((perf.auc1@y.values[[1]]),2)

df2<- data.frame(x = attributes(perf)$x.values[[1]],y = attributes(perf)$y.values[[1]])

# p <- ggplot()+

# geom\_line(data = df2,aes(x,y),colour = "blue",size = 0.7) +

# geom\_line(aes(x=c(0,1),y=c(0,1)),color = "grey") +

# geom\_ribbon(data = df2,aes(x,ymin = 0,ymax = y),fill = alpha("yellowgreen",0.5)) +

# labs(title ="ROC Curve") +

# annotate("text",x = .75, y = .25,label = paste("AUC = ",round((perf.auc1@y.values[[1]]),2)),color = "red",size=5)+

# xlab("Specificity") +

# ylab("Sensitivity") +

# theme(plot.title = element\_text(size = 17)) +

# theme\_bw()

#

# pdf(paste(auc,"\_",nam,".pdf",sep = ""),width = 5,height = 5)

# print(p)

# dev.off()

# ggsave(file=paste(name,"ROC\_curve.pdf",sep=""),width = 6,height = 6,p)

outdf[j,] <- c(nam,auc)

}

write.table(outdf,paste0(celltype[i],"\_gene2auc.txt"),sep="\t",quote=F,col.names=T,row.names=F)

}

} else if(args$Mode=="ATAC") {

## 1. 读入数据

#-----------------------------------------------------------------------------

tpm <- read.table(args$ExpMatrix,header = T)

metadata <- read.table(args$SampleMeta,header = T)

colnames(metadata) <- c("sample","group","sample.id")

rownames(metadata) <- metadata$sample

marker <- read.table(args$Features,header = TRUE,sep="\t")

colnames(marker) <- c("id","source")

concern.celltype <- read.table(args$TargetCelltype,header = F,sep="\t")$V1

outdir <- args$Outdir

if(!dir.exists(outdir)){dir.create(outdir,recursive = T)}

setwd(outdir)

## 2. 细胞类型分为关注和背景

#-----------------------------------------------------------------------------

all.celltype <- unique(metadata$group)

backgound.celltpe <- setdiff(all.celltype,concern.celltype)

celltype <- concern.celltype

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

## 3. 随机样本扩增

#-----------------------------------------------------------------------------

cat("Processing: ",celltype[i],"\n")

set.seed(i)

celltype.df <- metadata[metadata$group %in% celltype[i],]

others.df <- metadata[metadata$group %in% backgound.celltpe,]

sample.num <- dim(others.df)[1] - dim(celltype.df)[1]

expand.srr <- sample(celltype.df$sample,size = sample.num,replace = T)

expand.celltype.df <- metadata[expand.srr,]

expand.celltype.df$sample.id <- paste(celltype[i],nrow(celltype.df):(nrow(others.df)-1),sep = ".")

all.celltype.df <- rbind(celltype.df,expand.celltype.df)

final.df <- rbind(all.celltype.df,others.df)

rownames(final.df) <- final.df$sample.id

out.tpm <- tpm[,final.df$sample]

colnames(out.tpm) <- final.df$sample.id

final.df$Type <- "CASE"

final.df$Type[!final.df$group %in% celltype[i]] <- "CTRL"

colnames(final.df)[3] <- "Sample"

sample.classification <- final.df[,c(3,4)]

out.tpm <- out.tpm %>% mutate(Gene\_ID=rownames(out.tpm),.before = 1)

write.table(out.tpm,paste0(celltype[i],"\_expand.cpm.txt"),quote = F,sep = "\t",row.names = F,col.names = T)

write.table(sample.classification,paste0(celltype[i],"\_sample.classification.txt"),quote = F,sep = "\t",row.names = F,col.names = T)

## 4. 机器学习特征筛选

#-----------------------------------------------------------------------------

df <- out.tpm

id2symbol <- df$Gene\_ID

gid <- sapply(strsplit(id2symbol,"|",fixed = T),"[[",1)

rownames(df) <- gid

df$Gene\_ID <- gid

type <- sample.classification

genes <- marker %>% filter(source %in% celltype[i]) %>% select(id)

genes$id <- as.character(genes$id)

data <- merge(y=df,x = genes,by.y = "Gene\_ID",by.x = "id",sort = F)

rownames(data) <- data$id

#anno <- data[,1:4]

data <- data[,-1]

data\_t <- as.data.frame(t(data))

data\_t$outcom <- type$Type

data\_t$outcom[data\_t$outcom=="CASE"] <- 1

data\_t$outcom[data\_t$outcom=="CTRL"] <- 0

data\_t$outcom <- as.numeric(data\_t$outcom)

# getwd()

outdf <- as.data.frame(matrix(nrow = (ncol(data\_t)-1),ncol = 2))

colnames(outdf) <- c("peakid","auc")

for (j in 1:(ncol(data\_t)-1)) {

df1 <- data\_t[,c(j,j,ncol(data\_t))]

nam <- colnames(data\_t)[j]

colnames(df1)[ncol(df1)]="y"

set.seeds=1

folds <- createFolds(y=df1[,"y"],k=10)

folds10=data.frame()

for(k in 1:10){

test <- df1[folds[[k]],]

train <- df1[-folds[[k]],]

train\_matrix <- as.matrix(train[, -ncol(train)])

test\_matrix <- as.matrix(test[, -ncol(test)])

#lasso

cvfit <- cv.glmnet(train\_matrix, train$y, family = "gaussian", standardize = TRUE,nlambda = 1000, nfolds = 10, alpha = 1)

lambda\_min<-cvfit$lambda.min

pred.lasso <- predict(cvfit, test\_matrix, type="response", s="lambda.min")

glm.fit <- glm(y~., data=train)

#pred.glm1 <- predict(glm.fit1,test1)

glm.step=step(glm.fit)

pred.glm <- predict(glm.step,test)

re=as.matrix(cbind(test[,ncol(test)] ,pred.lasso,pred.glm))

re=as.data.frame(re)

colnames(re)=c('event','pred\_lasso','pred\_glm')

folds10<-rbind(folds10,re)

}

predob<- prediction(folds10$pred\_glm, folds10$event)

perf.auc<- performance(predob, measure = 'auc', x.measure = 'cutoff')

perf<- performance(predob, 'tpr','fpr')

perf.auc1 <- perf.auc

auc <- round((perf.auc1@y.values[[1]]),2)

df2<- data.frame(x = attributes(perf)$x.values[[1]],y = attributes(perf)$y.values[[1]])

# p <- ggplot()+

# geom\_line(data = df2,aes(x,y),colour = "blue",size = 0.7) +

# geom\_line(aes(x=c(0,1),y=c(0,1)),color = "grey") +

# geom\_ribbon(data = df2,aes(x,ymin = 0,ymax = y),fill = alpha("yellowgreen",0.5)) +

# labs(title ="ROC Curve") +

# annotate("text",x = .75, y = .25,label = paste("AUC = ",round((perf.auc1@y.values[[1]]),2)),color = "red",size=5)+

# xlab("Specificity") +

# ylab("Sensitivity") +

# theme(plot.title = element\_text(size = 17)) +

# theme\_bw()

#

# pdf(paste(auc,"\_",nam,".pdf",sep = ""),width = 5,height = 5)

# print(p)

# dev.off()

# ggsave(file=paste(name,"ROC\_curve.pdf",sep=""),width = 6,height = 6,p)

outdf[j,] <- c(nam,auc)

}

write.table(outdf,paste0(celltype[i],"\_peak2auc.txt"),sep="\t",quote=F,col.names=T,row.names=F)

}

} else if(args$Mode=="Integration") {

## 1. 读入数据

#-----------------------------------------------------------------------------

peak2id2symbol <- read.table(args$PeakGene,header = T) # peak调控基因id和symbol

auc.file <- read.table(args$AUCtbl,header = T) # 细胞类型的AUC文件

outdir <- args$Outdir

if(!dir.exists(outdir)){dir.create(outdir,recursive = T)}

celltype <- auc.file[,1]

RNA.file.path <- auc.file[,2]

ATAC.file.path <- auc.file[,3]

## 2. 数据整合

#-----------------------------------------------------------------------------

out\_rna <- list()

out\_atac <- list()

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

print(paste0("Processing: ",celltype[i]))

tmp\_rna <- read.table(RNA.file.path[i],header = T) %>% filter(auc >= 0.85)

colnames(tmp\_rna)[2] <- paste0(celltype[i],"\_RNA\_AUC")

tmp\_atac <- read.table(ATAC.file.path[i],header = T) %>% filter(auc >= 0.85)

colnames(tmp\_atac)[2] <- paste0(celltype[i],"\_ATAC\_AUC")

out\_rna[[i]] <- tmp\_rna

out\_atac[[i]] <- tmp\_atac

}

rna\_auc\_fil <- Reduce(function(x,y) merge(x,y,by="id",all=FALSE),out\_rna,accumulate =FALSE)

atac\_auc\_fil <- Reduce(function(x,y) merge(x,y,by="peakid",all=FALSE),out\_atac,accumulate =FALSE)

final\_outdf1 <- merge(peak2id2symbol,rna\_auc\_fil,by.x="geneId",by.y="id",all=F)

final\_outdf2 <- merge(final\_outdf1,atac\_auc\_fil,by.x="peakId",by.y="peakid",all=F)

setwd(outdir)

write.table(final\_outdf2,"o1.IntegratedCREs.txt",sep = "\t",quote = F,col.names = T,row.names = F)

}

demo\_run\_atac.sh

#!/usr/bin/sh

## This is a demo for Software CREfinder

## Written by XiaoHan

echo "start at `date`" &&

/jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/miniconda/envs/R4.2/bin/Rscript CREfinder.r \

--Mode ATAC \

--ExpMatrix /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/ATAC/i1.ATAC\_CPMat.txt \

--SampleMeta /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/ATAC/i1.ATAC\_SampleMeta.txt \

--Features /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/ATAC/i1.ATAC\_DORs.txt \

--TargetCelltype /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/ATAC/i1.ATAC\_ConcernCelltype.txt \

--Outdir /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/02-output/ATAC &&

echo "end at `date`" &&

echo "Still water run deep" 1>&2 &&

echo "Still water run deep" > demo.sign

demo\_run\_rna.sh

#!/usr/bin/sh

## This is a demo for Software CREfinder

## Written by XiaoHan

echo "start at `date`" &&

/jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/miniconda/envs/R4.2/bin/Rscript CREfinder.r \

--Mode RNA \

--ExpMatrix /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/RNA/i1.RNA\_TPMat.txt \

--SampleMeta /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/RNA/i1.RNA\_SampleMeta.txt \

--Features /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/RNA/i1.RNA\_DEGs.txt \

--TargetCelltype /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/RNA/i1.RNA\_ConcernCelltype.txt \

--Outdir /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/02-output/RNA &&

echo "end at `date`" &&

echo "Still water run deep" 1>&2 &&

echo "Still water run deep" > demo.sign

demo\_run\_integration.sh

#!/usr/bin/sh

## This is a demo for Software CREfinder

## Written by XiaoHan

echo "start at `date`" &&

/jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/miniconda/envs/R4.2/bin/Rscript CREfinder.r \

--Mode Integration \

--PeakGene /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/Integration/i1.peak2gid2symbol.txt \

--AUCtbl /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/01-input/Integration/i2.celltype2auc.txt \

--Outdir /jdfsbjcas1/ST\_BJ/P21H28400N0232/xiaohan2/Software/CREfinder/Demo/02-output/Integration &&

echo "end at `date`" &&

echo "Still water run deep" 1>&2 &&

echo "Still water run deep" > demo.sign