#LASSO LEARNING

VEIN=read.table("VEIN.csv",sep=",",check.names=F,header=T,row.names = 1)

colnames(VEIN)

rownames(VEIN)

exp=read.table("bindgeo\_exp.csv",sep=",",check.names=F,header=T,row.names = 1)

DEG\_gene\_exprvein1<-exp[rownames(VEIN),]

DEG\_gene\_exprvein1<-t(DEG\_gene\_exprvein1)

write.table(DEG\_gene\_exprvein1,file="DEG\_gene\_exprvein1.csv",sep=",",quote=T,col.names=T)

data<- read.csv("DEG\_gene\_exprvein1.csv", header = T, sep=",",row.names =1)

library(tidyverse)

library(glmnet)

library(sigFeature)

library(e1071)

library(caret)

library(randomForest)

train <- read.csv("DEG\_gene\_exprvein1.csv",row.names = 1,

as.is = F)

dim(train)

set.seed(123)

#train[1:4,1:4]

x <- as.matrix(train[,-15])

(y <- ifelse(train$group == "NC", 0,1)) #把分组信息换成01

fit = glmnet(x, y, family = "binomial", alpha = 1, lambda = NULL)

# PLOT LASSO

pdf("1A\_lasso.pdf", width = 30, height = 15)

plot(fit, xvar = "dev", label = TRUE)

dev.off()

#10 validation

cvfit = cv.glmnet(x, y,

nfold=10, #10-fold cross-validation

family = "binomial", type.measure = "class")

pdf("2cvfit.pdf")

plot(cvfit)

dev.off()

#best lambda

cvfit$lambda.min

#

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

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

(lasso\_fea <- lasso\_fea[-1])

#

write.csv(lasso\_fea,"3feature\_lasso.csv")

rm(list=ls())

#SVM-REF

predcm<-read.csv("DEG\_gene\_exprvein1.csv",row.names = 1,

as.is = F)

control <- rfeControl(functions = caretFuncs, method = "cv", number = 5)

# SVM-RFE analysis

results <- rfe(predcm[,1:14],

predcm[,15],

sizes = c(1:14),

rfeControl = control,

method = "svmRadial") # method = "svmRadial" specifies that the SVM model should use a radial kernel

#

print(results)

#

predictors(results)

#

pdf("6B\_svm-accuracy.pdf",width = 5,height = 5)

plot(results, type=c("g", "o"))

dev.off()

p<-predictors(results)

write.csv(p,"feature\_svm.csv")

#randomforest

data<- read.csv("DEG\_gene\_exprvein1.csv", header = T, sep=",",row.names =1)

data$group<- ifelse(data$group == "NC", 0,1)

set.seed(12345)

library(randomForest)

y<-factor(data$group)

x<-data[,-15]

rf<-randomForest(x, y,data=data,importance=TRUE)

plot(margin(rf, data$group), main = '观测值被判断正确的概率图')

importance\_otu <- data.frame(importance(rf), check.names = FALSE)

head(importance\_otu)

importance\_otu <- importance\_otu[order(importance\_otu$MeanDecreaseAccuracy, decreasing = TRUE), ]

#根据表格输出前30变量

rf

varImpPlot(rf, n.var = min(30, nrow(rf$importance)), main = 'Top 30 - variable importance')

write.table(importance\_otu, 'importance\_otu.txt', sep = '\t', col.names = NA, quote = FALSE)

otu.cv <- replicate(5, rfcv(data[-ncol(data)], data$group, cv.fold = 10,step = 1.5), simplify = FALSE)

otu.cv

otu.cv <- data.frame(sapply(otu.cv, '[[', 'error.cv'))

otu.cv$otus <- rownames(otu.cv)

otu.cv <- reshape2::melt(otu.cv, id = 'otus')

otu.cv$otus <- as.numeric(as.character(otu.cv$otus))

otu.cv.mean <- aggregate(otu.cv$value, by = list(otu.cv$otus), FUN = mean)

head(otu.cv.mean, 14)

library(ggplot2)

library(splines) #用于在 geom\_smooth() 中添加拟合线，或者使用 geom\_line() 替代 geom\_smooth() 绘制普通折线

p <- ggplot(otu.cv, aes(otus, value)) +

geom\_smooth(se = FALSE, method = 'glm', formula = y~ns(x, 6)) +

theme(panel.grid = element\_blank(), panel.background = element\_rect(color = 'black', fill = 'transparent')) +

labs(title = '',x = 'Number of OTUs', y = 'Cross-validation error')

p

p + geom\_vline(xintercept =6)

importance\_otu <- importance\_otu[order(importance\_otu$MeanDecreaseAccuracy, decreasing = TRUE), ]

importance\_otu.select <- importance\_otu[1:6, ]

importance\_otu.select

write.table(importance\_otu.select, 'importance\_otu.select.csv', sep = ',', col.names = NA, quote = FALSE)