CE TCR V5

Gabrielle Salamanca

Oct 19, 2025

## Libraries

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Warning: package 'factoextra' was built under R version 4.2.3

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

## Warning: package 'ggfortify' was built under R version 4.2.3

## Warning: package 'gridExtra' was built under R version 4.2.3

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

## Warning: package 'psych' was built under R version 4.2.3

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'readxl' was built under R version 4.2.3

## Warning: package 'SKAT' was built under R version 4.2.3

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.2.3

## Loading required package: SPAtest

## Loading required package: RSpectra

## Warning: package 'RSpectra' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

##   
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':  
##   
## expand, pack, unpack

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: lattice

## Warning: package 'MASS' was built under R version 4.2.3

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

## Warning: package 'class' was built under R version 4.2.3

## Dataset

We read in the dataset, and replace a handful of NAs. The HD patients’ Y and Y1 were filled in with “healthy”, because Professor Tao confirmed they were healthy patients. Row 22 was removed, because there was confirmation that it didn’t have any results, even in its own ID.

## The dimensions of the dataset is:

## [1] 109 679

##   
## Are there any NAs in the Y column?

##   
## FALSE TRUE   
## 92 17

##   
## Are there NAs in the Y1 column?

##   
## FALSE TRUE   
## 92 17

##   
## We remove row 22, to then have the dimensions:

## [1] 108 679

## After replacing NAs with the healthy tag, are there any lingering NAs in the Y column?

##   
## FALSE   
## 108

##   
## in the Y1 column?

##   
## FALSE   
## 108

## Training and Test Set

## The dimensions of the training set is: 86 679

## The dimensions of the test is: 22 679

## Finding the Significant Genes

Prepping for the for loop:

For loop to find the significant genes

## Warning: The `value` argument of `names<-()` must have the same length as `x` as of  
## tibble 3.0.0.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

Create the results table and sort by significance

## Gene P\_value  
## 237 TRBV23-1\_TRBJ2-3 0.0006753638  
## 197 TRBV2\_TRBJ2-1 0.0009850619  
## 226 TRBV21-1\_TRBJ2-4 0.0016483425  
## 125 TRBV12-5\_TRBJ2-7 0.0018210175  
## 290 TRBV28\_TRBJ2-4 0.0020444177  
## 331 TRBV4-1\_TRBJ2-4 0.0026953989

Get only the genes that have a p-value less than 0.05, and make sure the genes are in both the training and test data

## [1] 37 2

knitr::opts\_chunk$set(echo = FALSE)  
library(dplyr)  
library(factoextra)  
library(ggfortify)  
library(ggplot2)  
library(gridExtra)  
library(psych)  
library(readr)  
library(readxl)  
library(SKAT)  
library(tidyr)  
  
library(caret)  
library(stats)  
library(MASS) # QDA & LDA  
library(class) # KNN   
gene <- read\_excel("D:/Coding/R Storage/Summer TCR Project/TCR Datasets/2025/fullgenes.xlsx")  
  
cat("The dimensions of the dataset is:")  
dim(gene)  
  
cat("\n Are there any NAs in the Y column?")  
table(is.na(gene$Y))  
cat("\nAre there NAs in the Y1 column?")  
table(is.na(gene$Y1))  
  
cat("\nWe remove row 22, to then have the dimensions:")  
genedit <- gene[-22,]  
dim(genedit)  
  
cat("\n")  
  
for (lab in intersect(c("Y","Y1"), names(genedit))) {  
 genedit[[lab]][is.na(genedit[[lab]])] <- "healthy"  
}  
  
cat("After replacing NAs with the healthy tag, are there any lingering NAs in the Y column?")  
table(is.na(genedit$Y))  
cat("\nin the Y1 column?")  
table(is.na(genedit$Y1))  
set.seed(895)  
  
train <- sample(1:nrow(genedit),0.8\*nrow(genedit))  
train.data <- genedit[train,]  
test.data <- genedit[-train,]  
  
cat("The dimensions of the training set is:", dim(train.data), "\n")  
cat("The dimensions of the test is:", dim(test.data), "\n")  
  
# turn Y binary  
train.data$Y <- as.numeric(ifelse(train.data$Y == "disease", 1, 0))  
test.data$Y <- as.numeric(ifelse(test.data$Y == "disease", 1, 0))  
col <- ncol(train.data)  
ycol <- match("Y", names(train.data))  
gene\_idx <- 2:(col - 2)  
gene.name <- names(train.data)[gene\_idx]  
pvalue <- numeric(length(gene\_idx))  
for (i in seq\_along(gene\_idx))  
{  
 gene\_name <- gene.name[i]  
 Xi <- train.data[, gene\_idx[i], drop = FALSE]  
 names(Xi) <- gene.name # set column name to the gene  
   
 dat <- data.frame(Y = train.data[[ycol]], Xi, check.names = FALSE)  
 glm.fit <- glm(Y ~ ., data = dat, family = binomial())  
 pvalue[i] <- coef(summary(glm.fit))[2, 4]  
}  
# Combine results into a nice table:  
results <- data.frame(Gene = gene.name, P\_value = pvalue)  
  
# Sort by significance  
results <- results[order(results$P\_value), ]  
head(results)  
alpha <- results[results$P\_value < 0.05,]  
dim(alpha)  
  
ranked <- alpha$Gene  
ranked <- intersect(ranked, intersect(names(train.data), names(test.data)))