cancer genomic

Ming Xu

6/4/2020

```
# load packages
library(reticulate)
library(readr)
# read data
use_python("/opt/anaconda3/python.app/Contents/MacOS/python")
source_python("pickle_reader.py")
pickle_data <- read_pickle_file("codon_mutability.pickle")
rm(r,R,read_pickle_file)
genie_data <- read.delim("~/Desktop/Dissertation/cambridge cancer/code/genie_data_mutations_extended.txt
```

```
# tidy and unlist the pickle data
library(tibble)
underlying <- tibble(codon = names(unlist(pickle_data)), muta = unlist(pickle_data))
```

```
# delete unnecessary data
genie_data <- genie_data[colSums(!is.na(genie_data)) > 0]
columns <- c(1,5,6,7,9,10,15,23,28)
genie_data <- genie_data[,columns]
genie_data <- genie_data[which(genie_data$Variant_Classification=='Missense_Mutation'  & genie_data$Vari

# turn x,y chromosome to numeric data
genie_data$Chromosome[genie_data$Chromosome == "X"] = 23
genie_data$Chromosome[genie_data$Chromosome == "Y"] = 24
genie_data$Chromosome <- as.integer(genie_data$Chromosome)

# obtain protein length of gene
len<- unlist(strsplit(genie_data$Protein_position,"/"))
genie_data['Length_protein'] <- as.integer(len[seq(2,length(len),2)])


# connect the string of codon
library(stringr)
genie_data['codon'] <- paste(genie_data$Hugo_Symbol,str_sub(genie_data$HGVSp_Short,3,str_length(genie_da

# delete the unnecessary columns
genie_data['Tumor_Sample_Barcode'] <- NULL
genie_data['Protein_position'] <- NULL

# save the data
write.csv(genie_data,'genie_data.csv')
```

```
# exploratory analysis
explor <- genie_data[, c(2,3,8)]
```

```r
cormat <- round(cor(explor),2)

# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}
upper_tri <- get_upper_tri(cormat)

# Melt the correlation matrix
library(reshape2)
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Heatmap
library(ggplot2)
explor_plot <- ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "green", high = "blue", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 9, hjust = 1), axis.title.x=element_blan
 coord_fixed()

# mutual information
library(infotheo)
dis_explor<-discretize(explor)
mutinformation(dis_explor[,1],dis_explor[,2])
mutinformation(dis_explor[,1],dis_explor[,3])
mutinformation(dis_explor[,2],dis_explor[,3])
```

```r
# calculate p value
library(dplyr)
genie_fre <- genie_data %>%
  group_by(codon, Chromosome) %>%
  summarize(n = n(), position = mean(Start_Position), Length_protein = mean(Length_protein),.groups = '
  ungroup()
# combine dataset
combine_underlying <- left_join(underlying,genie_fre,by="codon")

# do binomial test

# calculate p value
pvalue <- numeric(length = 176109)
for (i in 1:176109) {
  pvalue[i] = binom.test(combine_underlying$n[i],59815,combine_underlying$muta[i],'greater')$p.value
}

# store the p.value
combine_underlying['pvalue'] <- pvalue
write.csv(combine_underlying,'combine_underlying.csv')
```

```r
# BH method
bh_genie <- combine_underlying[p.adjust(combine_underlying$pvalue, method = "BH") <= 0.01,]
```

```r
sum(p.adjust(combine_underlying$pvalue, method = "BH") <= 0.01)
```

```r
# BY Method
by_genie <- combine_underlying[p.adjust(combine_underlying$pvalue, method = "BY") <= 0.01,]
sum(p.adjust(combine_underlying$pvalue, method = "BY") <= 0.01)
```

```r
# q-value
library(qvalue)
qvalue <- qvalue(p = as.vector(combine_underlying$pvalue), fdr.level = 0.01)

# obtain rejections
sum(qvalue$significant == TRUE)
q_genie <- combine_underlying[qvalue$significant == TRUE,]
```

```r
# IHW method
library(IHW)
ihw_length <- ihw(combine_underlying$pvalue, combine_underlying$Length_protein, 0.01)
ihw_genie <- combine_underlying[adj_pvalues(ihw_length) <= 0.01,]

# obtain rejections
rejections(ihw_length)

# plot the boundary
plot(ihw_length,what = "decisionboundary")
```

```r
# AdaFDR method
library(RadaFDR)

# change appropriate class
p <- as.array(combine_underlying$pvalue)
x <- as.array(combine_underlying$Length_protein)
x<- as.matrix(x, nrow = 176109)

# do test
res <- adafdr_test(p,x,alpha = 0.01,fast_mode = FALSE)
res$n_rej
adafdr_genie <- combine_underlying[res$decision,]

# do nest
res_1 <- adafdr_test(p,x,alpha = 0.001,fast_mode = FALSE)
res_1$n_rej
adafdr_genie_1 <- combine_underlying[res_1$decision,]
```

```r
# validation data preprocessing
validation_data <- read.delim("~/Desktop/Dissertation/cambridge cancer/code/pre_tcga_mutations_data.txt

# select useful columns
columns_validation <- c(1,3,4,5,8,9,10,12)
validation_data <- validation_data[,columns_validation]

# filter missense mutations
validation_data <- validation_data[which(validation_data$Variant_Classification=='Missense_Mutation'  &
validation_data$Chromosome[validation_data$Chromosome == "X"] = 23
validation_data$Chromosome[validation_data$Chromosome == "Y"] = 24
validation_data$Chromosome <- as.integer(validation_data$Chromosome)
```

```
# connect the string and get the name of codon
validation_data['codon'] <- paste(validation_data$Hugo_Symbol,str_sub(validation_data$HGVSp_Short,3,str
n_tcga <- length(unique(validation_data$Tumor_Sample_Barcode))
validation_data['Tumor_Sample_Barcode'] <- NULL
```

```
# calculate the frequency of the data
tcga_fre <- validation_data %>%
  group_by(codon, Chromosome) %>%
  summarize(n = n(), position = mean(Start_Position),.groups = 'drop') %>%
  ungroup()

# combine dataset
tcga_underlying <- left_join(combine_underlying[,c(1,2,6)],tcga_fre,by="codon")
tcga_underlying <- na.omit(tcga_underlying)
# do binomial test

# calculate p value
pvalue_tcga <- numeric(length = 7166)
for (i in 1:7166) {
  pvalue_tcga[i] = binom.test(tcga_underlying$n[i],n_tcga,tcga_underlying$muta[i],'greater')$p.value
}

# store the p.value
tcga_underlying['pvalue'] <- pvalue_tcga

write.csv(tcga_underlying,'tcga_underlying.csv')
```

```
# BH method
tcga_underlying[p.adjust(tcga_underlying$pvalue, method = "BH") <= 0.01,]
sum(p.adjust(tcga_underlying$pvalue, method = "BH") <= 0.01)

bh_tcga <- tcga_underlying[p.adjust(tcga_underlying$pvalue, method = "BH") <= 0.01,]
```

```
# BY Method
tcga_underlying[p.adjust(tcga_underlying$pvalue, method = "BY") <= 0.01,]
sum(p.adjust(tcga_underlying$pvalue, method = "BY") <= 0.01)

by_tcga <- tcga_underlying[p.adjust(tcga_underlying$pvalue, method = "BY") <= 0.01,]
```

```
library(IHW)

ihw_length_tcga <- ihw(tcga_underlying$pvalue, tcga_underlying$Length_protein, 0.01)
ihw_tcga <- tcga_underlying[adj_pvalues(ihw_length_tcga) <= 0.01,]
sum(adj_pvalues(ihw_length_tcga) <= 0.01)
```

```
# exploratory analysis of p-value
hist_pvalue <- hist(combine_underlying$pvalue, breaks = 30, col="royalblue", xlab="P-value", main = "",
sum(combine_underlying$pvalue < 0.001)
sum(combine_underlying$pvalue < 0.01)
sum(combine_underlying$pvalue < 0.05)
```

```
# comparing drivers from conventional method
par(mfrow=c(3,1))
hist(combine_underlying[p.adjust(combine_underlying$pvalue, method = "BH") <= 0.01,]$pvalue,breaks = 10
hist(combine_underlying[p.adjust(combine_underlying$pvalue, method = "BY") <= 0.01,]$pvalue,breaks = 10
hist(combine_underlying[qvalue$significant == TRUE,]$pvalue,breaks = 10, col="royalblue", xlab="P-value
```

```
# check the assumptions of IHW for chromosome
ihw_c1 <- combine_underlying[combine_underlying$Chromosome <= 8,]
ihw_c2 <- combine_underlying[combine_underlying$Chromosome > 8 & combine_underlying$Chromosome <= 16,]
ihw_c3 <- combine_underlying[combine_underlying$Chromosome > 16,]

par(mfrow=c(1,3))
hist(ihw_c1$pvalue,breaks = 20, col="royalblue", xlab="P-value (chromosome: 1-8) ", main = "",freq = F)
hist(ihw_c2$pvalue,breaks = 20, col="royalblue", xlab="P-value (chromosome: 9-16) ", main = "",freq = F
hist(ihw_c3$pvalue,breaks = 20, col="royalblue", xlab="P-value (chromosome: 17-24)", main = "",freq = F
```

```
# check the assumptions of IHW for start_position
ihw_p1 <- combine_underlying[combine_underlying$position <= 80000000,]
ihw_p2 <- combine_underlying[combine_underlying$position > 80000000 & combine_underlying$Chromosome <=
ihw_p3 <- combine_underlying[combine_underlying$position > 160000000,]

par(mfrow=c(1,3))
hist(ihw_p1$pvalue,breaks = 20, col="royalblue", xlab="P-value (position: 0-80 million)", main = "",fre
hist(ihw_p2$pvalue,breaks = 20, col="royalblue", xlab="P-value (position: 80-160 million)", main = "",f
hist(ihw_p3$pvalue,breaks = 20, col="royalblue", xlab="P-value (position: 160-240 million)", main = "",
```

```
# check the assumptions of IHW for protein length
ihw_l1 <- combine_underlying[log(combine_underlying$Length_protein) <= 5.5,]
ihw_l2 <- combine_underlying[log(combine_underlying$Length_protein) > 5.5 & combine_underlying$Chromoso
ihw_l3 <- combine_underlying[log(combine_underlying$Length_protein) > 8,]

par(mfrow=c(1,3))
hist(ihw_l1$pvalue,breaks = 20, col="royalblue", xlab="P-value (log(length): 0-5.5)", main = "",freq =
hist(ihw_l2$pvalue,breaks = 20, col="royalblue", xlab="P-value (log(length): 5.5-8)", main = "",freq =
hist(ihw_l3$pvalue,breaks = 20, col="royalblue", xlab="P-value (log(length): 8-10.5)", main = "",freq =
```

```
# validation outcome
driver_bh <- length(intersect(bh_genie$codon,tcga_underlying$codon))
subdriver_bh <- length(intersect(bh_genie$codon,bh_tcga$codon))
subdriver_bh/driver_bh


driver_by <- length(intersect(by_genie$codon,tcga_underlying$codon))
subdriver_by <- length(intersect(by_genie$codon,by_tcga$codon))
subdriver_by/driver_by

driver_ihw <- length(intersect(ihw_genie$codon,tcga_underlying$codon))
subdriver_ihw <- length(intersect(ihw_genie$codon,ihw_tcga$codon))
subdriver_ihw/driver_ihw

driver_adafdr <- length(intersect(adafdr_genie$codon,tcga_underlying$codon))
subdriver_adafdr <- length(intersect(adafdr_genie$codon,intersect(adafdr_tcga$codon,tcga_underlying$cod
subdriver_adafdr/driver_adafdr

driver_adafdr_1 <- length(intersect(adafdr_genie_1$codon,tcga_underlying$codon))
subdriver_adafdr_1 <- length(intersect(adafdr_genie_1$codon,intersect(adafdr_tcga_1$codon,tcga_underlyi
subdriver_adafdr_1/driver_adafdr_1
```