ExampleOfSignatureQBiC

Mo Liu, Steven G. Rozen

12/30/2020

Example of Signature-QBiC for HOXD13 and mutational signature SBS7a

This is the example shown in Figure 1b of Liu et al., Mutational Processes in Cancer Preferentially Affect Binding of Particular Transcription Factors.

Load libraries

```
library(PCAWG7)
library(tibble)
library(knitr)
```

```
PCAWG_subs_signature <- PCAWG7::signature$genome$SBS96

mut.types <- lapply(row.names(PCAWG_subs_signature),function(x){
  return(paste(substring(x,1,3),
               paste(substring(x,1,1),
                     substring(x,4,4),
                     substring(x,3,3),sep=""),
               sep="_"))
})
mut.types <- row.names(PCAWG_subs_signature) <- unlist(mut.types)

signature <- "SBS7a" ##we used SBS7a as an example
sig <- PCAWG_subs_signature[ , signature, drop = FALSE]

all.possible.twelvemers <-
  tibble(readRDS("../data-raw/all.possible.twelvemers.rds")) ##all mutations based on 11mers
```

Plotting function for Figure1

```
TruncatedHist <- function(all.QBiC.scores,original.scores,weighted.prop,mutation.type){
  cut_off <- quantile(all.QBiC.scores,seq(0,1,0.01))[100] ##pile the 1% tail up
  original.scores[original.scores>cut_off] <- cut_off
  original.scores[original.scores<(-cut_off)] <- (-cut_off)
  weighted.hist <- original.hist <- hist(original.scores, breaks = seq(-(cut_off+0.5),
                                                                       (cut_off+0.5),0.5), plot=
F)
  weighted.hist$density <-  weighted.hist$density*weighted.prop
  plot(original.hist,freq = F,ylim=c(0,max(original.hist$density)+0.05),
       main=paste0("Original ",mutation.type),xaxt="n",yaxt="n")
  plot(weighted.hist,freq=F,ylim = c(0,max(original.hist$density)+0.05),
       main=paste0("Weighted ",mutation.type),xaxt="n",yaxt="n")
}
```

The SignatureQBiC function. This function can also be found in SigQBiC package. In SigQBiC package, SignatureQBiC function returns Gain Ratio and Loss Ratio with the given QBiC scores, pvalues and signature. For tutorial purpose, we split them into several chunks to show how we calculate Gain Ratio and Loss Ratio. Therefore, the SignatureQBiC here is slightly different from the one in package.

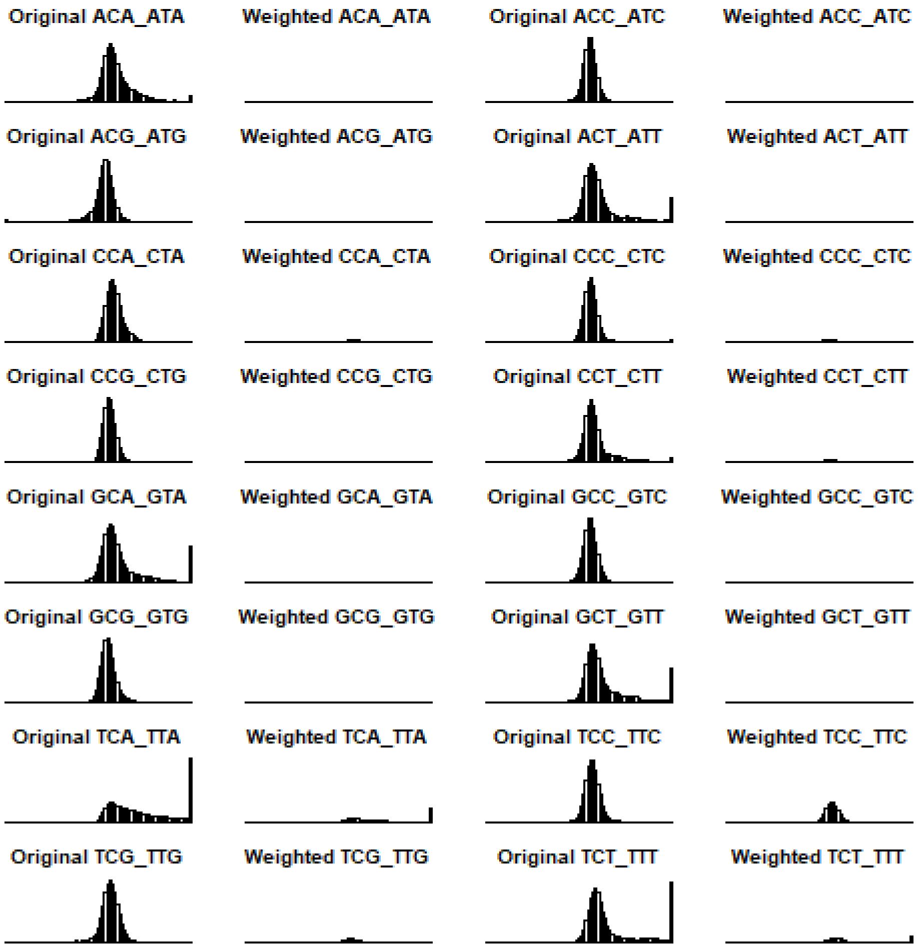
```r
SignatureQBiC <- function(QBiC_score_file_path,
                          pvalue_file_path,
                          sig,
                          plot.path = NULL) {

  QBiC_scores_table <-
    data.table::fread(QBiC_score_file_path,
                      sep=" ", header=T, stringsAsFactors = F, fill = T)
  # This gives a data frame with colums diff and z_score
  # QBiC_scores_table contains NA for non-mutations, e.g AAAAAAAAAAA -> AAAAAAAAAAA

  pvalue <-
    scan(pvalue_file_path)
  # pvalue also contains NA for non-mutations
  pvalue <- pvalue[!is.na(pvalue)]

  QBiC_scores_matrix <-
    tibble(QBiC_mut = all.possible.twelvemers$seq,
           mut_type = all.possible.twelvemers$final_signature,
           scores   = QBiC_scores_table$z_score[!is.na(QBiC_scores_table$z_score)],
           p        = pvalue,
           q        = p.adjust(pvalue, method = "BH"))

  max.score <- as.integer(max(QBiC_scores_matrix$scores)) + 2
  # Guaranteed that the QBiC scores' distribution will be symmetric

  summary <-data.frame(matrix(ncol=5,nrow=0))
  my.breaks <- seq(-max.score,max.score,0.001)

  if(!is.null(plot.path)){
    all.weighted.freq <- 0
    if (!dir.exists(plot.path)) {
      if (!dir.create(plot.path, recursive = T))
        stop("Cannot create plotting directory ", plot.path)
    }

    png(filename = paste0(plot.path, "/", "hist%03d.png"))
    par(mar = c(1,1,1,1))
    par(mfrow = c(8,4))

  }

  for (mutation.type in mut.types) {

    stopifnot(mutation.type %in% QBiC_scores_matrix$mut_type)
    # Scores for the given mutation.type
    tmp.scores <-
      QBiC_scores_matrix$scores[QBiC_scores_matrix$mut_type==mutation.type] ##the scores were pu
t into bins

    dist.hist <- hist(tmp.scores, breaks = my.breaks, plot=F)
    w.dist.hist <- dist.hist
    w.dist.hist$counts <- dist.hist$counts * sig[mutation.type, ]
    partial.summary <-
      data.frame(scores         = dist.hist$mids,
                 frequency      = dist.hist$counts,
                 mut_type       = mutation.type,
                 signature_freq = sig[mutation.type, ],
                 weighted.freq  = dist.hist$counts * sig[mutation.type, ])
    ##multiply the counts of each bin by the frequency of mutations in a signature

    if(!is.null(plot.path)){
      # Plot ...
      all.weighted.freq <- all.weighted.freq + dist.hist$counts * sig[mutation.type, ]
      # open.plot(mutation.type)
      TruncatedHist(QBiC_scores_matrix$scores,
                    original.scores = tmp.scores,
                    weighted.prop = sig[mutation.type, ],
                    mutation.type=mutation.type)
      # dev.off()
    }

    summary <- rbind(summary, partial.summary)
  }
  if(!is.null(plot.path)){
    all.scores <-  QBiC_scores_matrix$scores
    cut_off <- quantile(all.scores,seq(0,1,0.01))[100] ##pile the 1% tail up
    weighted.hist <- original.hist <- hist(all.scores, breaks = my.breaks, plot=F)
    weighted.hist$counts <-  all.weighted.freq*sum(original.hist$counts)/sum(all.weighted.freq)
    # open.plot("summary")
    plot(original.hist,main = "Original Distribution")
    plot(weighted.hist,main = "Weighted Distribution")
    dev.off()
  }

  return(list(QBiC_scores_matrix =QBiC_scores_matrix,
              summaryofscores    = summary))
}
```

Run SignatureQBiC with the input above The QBiC scores table and p value table can be downloaded from http://qbic.genome.duke.edu/downloads (http://qbic.genome.duke.edu/downloads), ‘prediction_3.zip’

```
result <-
  SignatureQBiC(QBiC_score_file_path =
                  "../data-raw/prediction6mer.Homo_sapiens!M01252_1.94d!Barrera2016!HOXD13_I322L
_R1.txt.gz",
                pvalue_file_path =
                  "../data-raw/pval6mer.Homo_sapiens!M01252_1.94d!Barrera2016!HOXD13_I322L_R1.cs
v.gz",
                sig,
                "./png.dir")
```

Here gives an example of scores for all mutations centered at C>T For each histogram, vertical axis corresponds to the density, and horizontal axis corresponds to the QBiC-scores. The ‘weighted TCA_TTA’ contributes more large QBiC scores (a higher bar on the right comparing with the rest)

```
include_graphics("./png.dir/hist003.png")
```



Select D'_{Pos} (D'_{Neg}) and D_{Pos} (D_{Neg}) to calculate GR and LR This part is included in SigQBiC::SignatureQBiC. We show this part seperately for tutorial purpose.

```
QBiC_scores_matrix <- result$QBiC_scores_matrix
summaryofscores    <- result$summaryofscores
rm(result)
pos.sig.QBiC_scores_matrix <-
  QBiC_scores_matrix[QBiC_scores_matrix$q < 0.1 & QBiC_scores_matrix$scores>0,] #select Dpos


qvalue.cutoff.score <- min(pos.sig.QBiC_scores_matrix$scores) ##get the cutoff of QBiC scores ba
sed on BH FDR

summaryofscores$weighted.freq <-
  summaryofscores$weighted.freq *
  sum(summaryofscores$frequency)/sum(summaryofscores$weighted.freq)
##Normalize the weighted freqeuencies. After multiplying with signature probability, the weighte
d frequency is 96 times less than the original frequency. sum(freq) = 96*sum(weighted.freq)

summaryofscores.Dpos <-
  summaryofscores[summaryofscores$scores>qvalue.cutoff.score, ] ##Select Dpos

Dpos <- rep(summaryofscores.Dpos$scores,
            summaryofscores.Dpos$frequency)

Dprimepos <- rep(summaryofscores.Dpos$scores,
                 round(summaryofscores.Dpos$weighted.freq, digits = 0))


summaryofscores.Dneg <-
  summaryofscores[summaryofscores$scores<(-qvalue.cutoff.score), ] ##Select Dneg

Dneg <- rep(summaryofscores.Dneg$scores,
            summaryofscores.Dneg$frequency)

Dprimeneg <- rep(summaryofscores.Dneg$scores,
                 round(summaryofscores.Dneg$weighted.freq, digits = 0))

GR = sum(Dprimepos)/sum(Dpos) ##GR = 2.952
LR = sum(Dprimeneg)/sum(Dneg) ##LR = 0.058
```

Example of generating one set of random mutations with equal frequency. We generated 1000 sets of random mutations for statistical test

```
ResampleMutationFrequency <- function(i){
  set.seed(i)
  resampling.of.mut.type <- table(sample(c(1:96),size=nrow(all.possible.twelvemers),replace=T))
##Generate mutations based on 96 trinucleotide based with equal frequency

  names(resampling.of.mut.type) <- mut.types

  resampling.of.mut.type <- resampling.of.mut.type/sum(resampling.of.mut.type) #Normalize number
of mutations to sum of 1
  return(resampling.of.mut.type)
}
```

```
random.mut.freq <- data.frame(ResampleMutationFrequency(123))

row.names(random.mut.freq) <- mut.types

resampling.result <-
  SignatureQBiC(QBiC_score_file_path =
                  "../data-raw/prediction6mer.Homo_sapiens!M01252_1.94d!Barrera2016!HOXD13_I322L
_R1.txt.gz",
                pvalue_file_path =
                  "../data-raw/pval6mer.Homo_sapiens!M01252_1.94d!Barrera2016!HOXD13_I322L_R1.cs
v.gz",
                sig)
```