TF_master with DoRothEA regulons in T2DM

Loranda_Calderon

2025-06-01

```r
library(biomaRt)
library(ensembldb)
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
## Loading required package: GenomeInfoDb

## Loading required package: GenomicFeatures

## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: AnnotationFilter

##
## Attaching package: 'ensembldb'

## The following object is masked from 'package:stats':
##
##     filter
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:ensembldb':
##
##     filter, select

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

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

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

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from ’package:S4Vectors’:
##
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from ’package:BiocGenerics’:
##
##     combine, intersect, setdiff, union

## The following object is masked from ’package:biomaRt’:
##
##     select

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

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

```r
library(viper)
library(dorothea)
```

```r
# Load the gene expression dataset for T2DM
T2DMdata <- read.csv("/Users/lorandacalderonzamora/Downloads/datExpr_integrative_T2DM.csv", row.names =
colnames(T2DMdata)[1] <- "ensembl_gene_id"
rownames(T2DMdata) <- T2DMdata$ensembl_gene_id
T2DMdata <- T2DMdata[, -1]
```

```r
# Retrieving Gene Annotation from Ensembl with biomaRt
ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
getinfo <- c("ensembl_gene_id", "entrezgene_id", "external_gene_name")

geneDat <- getBM(attributes = getinfo,
                 filters = "ensembl_gene_id",
                 values = rownames(T2DMdata),
                 mart = ensembl)
```

```r
# Prepare Expression Matrix with Unique Gene Names
T2DMdata$ensembl_gene_id <- rownames(T2DMdata)
expr_matrix <- merge(T2DMdata, geneDat, by = "ensembl_gene_id", all.x = TRUE)
expr_matrix <- expr_matrix[!duplicated(expr_matrix$external_gene_name), ]

expr_matrix_numeric <- expr_matrix[, grep("GSM", colnames(expr_matrix))]

expr_matrix_unique <- aggregate(expr_matrix_numeric,
                                by = list(expr_matrix$external_gene_name),
                                FUN = mean)

colnames(expr_matrix_unique)[1] <- "external_gene_name"
rownames(expr_matrix_unique) <- expr_matrix_unique$external_gene_name
expr_matrix_unique <- expr_matrix_unique[, -1]
```

```
# Infer Transcription Factor Activity Using DoRothEA and VIPER
data(dorothea_hs, package = "dorothea")

dorothea_regulons <- dorothea_hs %>%
  filter(confidence %in% c("A", "B"))

table(dorothea_regulons$target %in% rownames(expr_matrix_unique))
```

```
##
## FALSE  TRUE
##   665  5743
```

```
my_regulon <- dorothea_regulons %>%
  group_by(tf) %>%
  summarise(
    targets = list(setNames(mor, target)),
    .groups = "drop"
  )
my_regulon <- setNames(my_regulon$targets, my_regulon$tf)

my_regulon_viper <- lapply(my_regulon, function(targets) {
  list(tfmode = targets, likelihood = rep(1, length(targets)))
})

tf_activity <- viper(eset = as.matrix(expr_matrix_unique),
                     regulon = my_regulon_viper,
                     method = "scale",
                     nes = TRUE)
```

```
##
## Computing the association scores

## Computing regulons enrichment with aREA

##   |                                                                              |
```

```
# Assign Sample Group Labels
group_labels <- rep(NA, ncol(tf_activity))
names(group_labels) <- colnames(tf_activity)

group_labels[names(group_labels) %in% c(
  "GSM631755", "GSM631756", "GSM631757", "GSM631758", "GSM631759", "GSM631760", "GSM631761",
  "GSM524151", "GSM524152", "GSM524153", "GSM524154", "GSM524155", "GSM524156", "GSM524157",
  "GSM524158", "GSM524159", "GSM524160"
)] <- "CTL"

group_labels[names(group_labels) %in% c(
  "GSM631762", "GSM631763", "GSM631764", "GSM631765", "GSM631766", "GSM631767",
  "GSM524161", "GSM524162", "GSM524163", "GSM524164", "GSM524165", "GSM524166",
  "GSM524167", "GSM524168", "GSM524169", "GSM524170"
)] <- "Disease"

table(group_labels)
```

```
## group_labels
##     CTL Disease
##      17      16
```

```
ctl_T2DM <- which(group_labels == "CTL")
disease_T2DM <- which(group_labels == "Disease")
```

```
# Differential TF Activity Between Disease and Control Groups
tf_means <- data.frame(
  TF = rownames(tf_activity),
  Disease = rowMeans(tf_activity[, disease_T2DM]),
  CTL = rowMeans(tf_activity[, ctl_T2DM]),
  stringsAsFactors = FALSE
)
tf_means$NES_Diff <- tf_means$Disease - tf_means$CTL

p_values <- apply(tf_activity, 1, function(x) {
  t.test(x[disease_T2DM], x[ctl_T2DM])$p.value
})

tf_means$p_value <- p_values
tf_means <- tf_means[order(tf_means$p_value), ]
print(tf_means)
```

```
##            TF      Disease         CTL    NES_Diff      p_value
## CEBPA   CEBPA  0.698850736 -0.70142588  1.40027661 1.301703e-05
## E2F4     E2F4 -1.187091583  1.45538579 -2.64247737 2.022536e-04
## PRDM14 PRDM14  1.215797325 -1.44446500  2.66026232 5.003614e-04
## FOXP1   FOXP1  1.470917480 -1.67232364  3.14324112 1.925676e-03
## FOXO3   FOXO3 -0.934824225  0.87468059 -1.80950482 2.148529e-03
## E2F1     E2F1 -0.811292307  0.92477009 -1.73606240 2.250983e-03
## SP1       SP1  0.722834000 -0.77225565  1.49508965 2.588342e-03
## ZNF263 ZNF263 -1.714646306  2.01208813 -3.72673444 4.638144e-03
## FOXO1   FOXO1 -0.520806004  0.51319863 -1.03400464 4.823330e-03
## USF1     USF1  0.541532097 -0.64678039  1.18831248 6.104274e-03
## ESR1     ESR1  0.579991594 -0.65968011  1.23967170 1.035338e-02
## SMAD4   SMAD4  0.725200792 -0.72571077  1.45091156 1.193440e-02
## SPI1     SPI1  0.883302108 -0.95543100  1.83873311 1.570274e-02
## SP3       SP3  0.496521926 -0.61759743  1.11411936 2.186484e-02
## HNF4A   HNF4A  0.229460610 -0.33426699  0.56372760 2.753137e-02
## CEBPB   CEBPB  0.691579158 -0.52626101  1.21784017 3.853148e-02
## EGR1     EGR1  0.420582097 -0.38435672  0.80493882 4.587560e-02
## STAT2   STAT2  0.668103800 -0.59336775  1.26147155 4.890044e-02
## USF2     USF2  0.275138070 -0.28345258  0.55859065 5.392394e-02
## NFKB1   NFKB1  1.000481301 -0.90827417  1.90875548 5.514195e-02
## RELA     RELA  0.839829462 -0.80487637  1.64470583 5.764470e-02
## MYC       MYC -0.548489739  0.72965959 -1.27814933 6.143308e-02
## POU2F1 POU2F1  0.248564021 -0.17529209  0.42385611 6.342470e-02
## CREB1   CREB1  0.291543068 -0.23842166  0.52996472 9.574464e-02
## FOXA1   FOXA1  0.404320732 -0.28102497  0.68534570 1.066292e-01
## RARA     RARA  0.307959426 -0.36115339  0.66911281 1.101666e-01
## HIF1A   HIF1A -0.395263528  0.54208084 -0.93734437 1.614610e-01
## JUN       JUN  0.572433417 -0.48648973  1.05892315 1.928151e-01
## MITF     MITF -0.174366871  0.22371904 -0.39808591 1.942280e-01
## STAT3   STAT3  0.426223277 -0.26600680  0.69223008 2.017725e-01
## WT1       WT1 -0.192045041  0.20279342 -0.39483846 2.267090e-01
## YY1       YY1  0.157720689 -0.19639103  0.35411172 2.419266e-01
## FOS       FOS  0.333550145 -0.20980971  0.54335985 2.938933e-01
## IRF1     IRF1  0.294117844 -0.11282402  0.40694186 3.104308e-01
## GATA2   GATA2  0.165607310 -0.31099269  0.47660000 3.357450e-01
## TFAP2A TFAP2A  0.138567194 -0.12440554  0.26297273 3.695835e-01
## ETS2     ETS2  0.154522286 -0.15102815  0.30555044 3.909946e-01
## ETS1     ETS1  0.311731933 -0.15718867  0.46892060 4.305963e-01
## SMAD3   SMAD3  0.273632256 -0.12745483  0.40108708 4.523862e-01
## PPARA   PPARA -0.043789965  0.11157770 -0.15536767 5.204151e-01
## GATA3   GATA3 -0.088361260  0.07587451 -0.16423577 5.891638e-01
## AR         AR  0.143489046 -0.07548272  0.21897177 6.280410e-01
## PPARG   PPARG -0.077238438  0.06370762 -0.14094606 6.985112e-01
## STAT1   STAT1  0.151063758 -0.04814787  0.19921163 7.095594e-01
## ELK1     ELK1  0.071563697 -0.01550449  0.08706819 8.071984e-01
## CTCF     CTCF -0.024629716  0.02533019 -0.04995991 8.782196e-01
## NFIC     NFIC  0.023764280  0.04991605 -0.02615177 8.821280e-01
## FOXM1   FOXM1 -0.008024378 -0.05414852  0.04612414 9.165742e-01
## TP53     TP53  0.007245159  0.02658336 -0.01933820 9.427124e-01
```