TF_master with DoRothEA regulons in HTN

Loranda_Calderon

2025-06-01

```
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 HTN
HTNdata <- read.csv("/Users/lorandacalderonzamora/Downloads/datExpr_integrative_HTN.csv", row.names = NU
colnames(HTNdata)[1] <- "ensembl_gene_id"
rownames(HTNdata) <- HTNdata$ensembl_gene_id
HTNdata <- HTNdata[, -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(HTNdata),
                 mart = ensembl)
```

```r
# Prepare Expression Matrix with Unique Gene Names
HTNdata$ensembl_gene_id <- rownames(HTNdata)
expr_matrix <- merge(HTNdata, geneDat, by = "ensembl_gene_id", all.x = TRUE)
expr_matrix <- expr_matrix[!duplicated(expr_matrix$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
##   316  6092
```

```
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(
  "GSM700797", "GSM700798", "GSM609528", "GSM609529",
                                 "GSM609530", "GSM701161", "GSM701162", "GSM701163", "GSM701164", "GSM7

group_labels[names(group_labels) %in% c(
  "GSM700799", "GSM700800", "GSM700801", "GSM700802", "GSM700803",
                                 "GSM609526", "GSM609527", "GSM701166", "GSM701167", "GSM701168", "GSM7

table(group_labels)
```

```
## group_labels
##     CTL Disease
##      10      15
```

```r
ctl_HTN <- which(group_labels == "CTL")
disease_HTN <- which(group_labels == "Disease")
```

```r
# Differential TF Activity Between Disease and Control Groups
tf_means <- data.frame(
  TF = rownames(tf_activity),
  Disease = rowMeans(tf_activity[, disease_HTN]),
  CTL = rowMeans(tf_activity[, ctl_HTN]),
  stringsAsFactors = FALSE
)
tf_means$NES_Diff <- tf_means$Disease - tf_means$CTL
p_values <- apply(tf_activity, 1, function(x) {
  t.test(x[disease_HTN], x[ctl_HTN])$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
## MYC       MYC  1.029742389 -1.14035336  2.1700957438 0.002821204
## IRF1     IRF1 -0.928736717  0.72438174 -1.6531184526 0.009246005
## STAT1   STAT1 -0.896676487  1.04482428 -1.9415007653 0.022539370
## BACH1   BACH1 -0.248729768  0.21865482 -0.4673845889 0.029985910
## AR         AR  0.900256432 -0.46222860  1.3624850281 0.030254039
## STAT2   STAT2 -1.490591239  1.01993589 -2.5105271324 0.034874604
## NFIC     NFIC  0.744533939 -0.43181826  1.1763521960 0.037941489
## GATA2   GATA2 -0.461924247  0.48767944 -0.9496036842 0.040654444
## HNF4A   HNF4A  1.323430160 -1.15445149  2.4778816532 0.041778702
## FOXO1   FOXO1  0.438166463 -0.36166742  0.7998338790 0.056233137
## PPARG   PPARG  0.359368297 -0.38772707  0.7470953640 0.060037088
## GATA3   GATA3 -0.332275092  0.60364116 -0.9359162562 0.066342239
## TP53     TP53 -0.691140901  0.50805048 -1.1991913843 0.069121015
## WT1       WT1  0.192267951 -0.27997871  0.4722466590 0.073211175
## SPI1     SPI1 -1.841172627  0.98363748 -2.8248101104 0.090743376
## HIF1A   HIF1A  0.356348628 -0.34023163  0.6965802560 0.102636419
## CEBPA   CEBPA  0.529175788 -0.24023987  0.7694156603 0.106596103
## PPARA   PPARA  0.487527213 -0.39163912  0.8791663353 0.113476100
## SP1       SP1  0.535161708 -0.55748601  1.0926477157 0.123958356
## SP3       SP3  0.715294515 -0.40872436  1.1240188785 0.125272098
## MYB       MYB -0.221979900  0.35824747 -0.5802273693 0.143838781
## EGR1     EGR1  0.474368572 -0.24987015  0.7242387227 0.175151216
## FOS       FOS -0.332814065  0.21245048 -0.5452645403 0.221360254
## ELK1     ELK1  0.520370522 -0.31876189  0.8391324135 0.222818440
## YY1       YY1 -0.190711039  0.21810612 -0.4088171613 0.231385040
## FOXM1   FOXM1 -0.416488278  0.07880612 -0.4952943968 0.237578888
## RARA     RARA  0.142396729 -0.30784804  0.4502447717 0.270798437
## ETS1     ETS1 -0.366266858  0.30905933 -0.6753261903 0.279268538
## E2F1     E2F1 -0.392937899  0.41602611 -0.8089640072 0.293166454
## FOXA1   FOXA1  0.124729884 -0.47864930  0.6033791843 0.313853545
## USF2     USF2  0.263275895 -0.18494540  0.4482212935 0.346051962
## SMAD3   SMAD3  0.298984226 -0.26612197  0.5651061979 0.349145161
## FOXP1   FOXP1  0.452709239 -1.00362895  1.4563381867 0.353837871
## E2F4     E2F4 -0.453794782  0.43408083 -0.8878756096 0.357646234
## USF1     USF1  0.319455325 -0.16497559  0.4844309163 0.379465963
## RELA     RELA -0.616430116  0.19151953 -0.8079496511 0.380283449
## NFKB1   NFKB1 -0.703150942  0.19164807 -0.8947990108 0.417006723
## ESR1     ESR1  0.151609778 -0.14858913  0.3001989051 0.462089307
## SMAD4   SMAD4  0.145044060 -0.27546887  0.4205129323 0.475480130
## ZNF263 ZNF263 -0.209024929  0.66420583 -0.8732307623 0.616637061
## TFAP2A TFAP2A -0.044418168 -0.25914268  0.2147245125 0.630972607
## CTCF     CTCF -0.122130992  0.05903300 -0.1811639903 0.689433730
## JUN       JUN  0.017447363 -0.14068631  0.1581336699 0.710214390
## MITF     MITF  0.215980868  0.01702302  0.1989578445 0.721071007
## CREB1   CREB1  0.189319781  0.04037621  0.1489435751 0.742895671
## STAT3   STAT3 -0.293572629 -0.08957053 -0.2040020946 0.768523283
## CEBPB   CEBPB -0.089378855 -0.19850562  0.1091267640 0.881257257
## PRDM14 PRDM14  0.002986301  0.13270540 -0.1297191036 0.900912393
## ETS2     ETS2 -0.077839774 -0.03185588 -0.0459838918 0.911361887
## POU2F1 POU2F1  0.038870443  0.05188478 -0.0130143393 0.964805236
## FOXO3   FOXO3 -0.141510266 -0.14114323 -0.0003670319 0.999296673
```