TFBSfindR: Variant analysis of Transcription Factor Binding Sites

Douwe J. Spaanderman[1]

[1]Division of Gene Regulation, the Netherlands Cancer Institute

30 April 2019

Abstract

Contents

1 Introduction

2 Using TFBSfindR: A quick overview

Here TFBSfindR is run in its most simplified way. Analysing the example variant dataset provided with TFBSfindR.

```
library(TFBSfindR)
data <- system.file("extdata", "variant.dataset.fasta", package = "TFBSfindR")
data <- read.input.file(input = data, ref.genome = BSgenome.Hsapiens.UCSC.hg19)
data <- TFBS.findR(data, motiflist = MotifDb)
```

3 In depth overview

3.1 Step 1| Object initialization

Here we look more into detail in TFBSfindR and it's customizable functions. First we select the example variant dataset provided with TFBSfindR.

```
library(TFBSfindR)
data <- system.file("extdata", "variant.dataset.fasta", package = "TFBSfindR")
```

Next, we select the reference genome, we want to compare the variant data to.

```
library(BSgenome.Hsapiens.UCSC.hg19)
ref.genome <- BSgenome.Hsapiens.UCSC.hg19
```

Finally, we give our sample a name, which can be anything, and read the input file and output a GRangesobject. ATAC.only can be used to filter variants in FASTA of there presence in ATAC peaks. In order to do these filter steps, ATAC.only needs to be a string with the location of a BED file, which consists of variants present in ATAC data. For now, we set ATAC.only to FALSE as it is set as default. read.input.file outputs a GRangesobject with sample.name, rs number, allel, reference and alternative nucleotide and their sequences including 20 nucleotides before and after the variant.

```
sample.name <- "example.dataset"
data <- read.input.file(input = data, ref.genome = ref.genome,
    sample.name = sample.name, ATAC.only = FALSE)
head(data)
## GRanges object with 6 ranges and 7 metadata columns:
##                 seqnames               ranges strand |          Sample
##                    <Rle>            <IRanges>  <Rle> |     <character>
##    rs60216355     chr1 [11046517, 11046558]      + | example.dataset
##    rs58092391     chr1 [11046539, 11046580]      + | example.dataset
##   rs113663169     chr1 [11046544, 11046585]      + | example.dataset
##   rs112732333     chr1 [11046576, 11046617]      + | example.dataset
##    rs72868197     chr1 [11046634, 11046675]      + | example.dataset
##    rs60216355     chr1 [11046517, 11046558]      - | example.dataset
##                       SNP       Allel            REF            ALT
##               <character> <character> <DNAStringSet> <DNAStringSet>
##    rs60216355  rs60216355         *|*              T              C
##    rs58092391  rs58092391         *|*              A              G
##   rs113663169 rs113663169         *|*              T              C
##   rs112732333 rs112732333         *|*              G              A
##    rs72868197  rs72868197         *|*              T              A
##    rs60216355  rs60216355         *|*              T              C
##                          REF.sequence            ALT.sequence
##                        <DNAStringSet>          <DNAStringSet>
##    rs60216355 CGTGTTAGCC...CCTCGTGATC CGTGTTAGCC...CCTCGTGATC
##    rs58092391 ATCTCCTGAC...CCTCCCAAAG ATCTCCTGAC...CCTCCCAAAG
##   rs113663169 CTGACCTCGT...CAAAGTGCTG CTGACCTCGT...CAAAGTGCTG
##   rs112732333 AAAGTGCTGG...CGCCCGGTCA AAAGTGCTGG...CGCCCGGTCA
##    rs72868197 ATAGTTGGAA...AGCCCCAGCA ATAGTTGGAA...AGCCCCAGCA
##    rs60216355 GCACAATCGG...GGAGCACTAG GCACAATCGG...GGAGCACTAG
##   -------
##   seqinfo: 93 sequences (1 circular) from hg19 genome
```

3.2 Step 2| Motif selection

In order to analyse the variant dataset we have to select motifs to compare our dataset to. A usefull library is MotifDb, which consists of several motif databases. Here we have selected only human motifs provided by the JASPARCORE database. This database consists of 66 well known Transcription factor motifs.

```
library(MotifDb)
JASPARCORE <- query(MotifDb, "JASPAR_CORE")
JASPARCORE <- query(JASPARCORE, "hsapiens")
JASPARCORE
## MotifDb object of length 66
## | Created from downloaded public sources: 2013-Aug-30
## | 66 position frequency matrices from 1 source:
## |        JASPAR_CORE:   66
## | 1 organism/s
## |           Hsapiens:   66
## Hsapiens-JASPAR_CORE-TFAP2A-MA0003.1
## Hsapiens-JASPAR_CORE-NR2F1-MA0017.1
## Hsapiens-JASPAR_CORE-E2F1-MA0024.1
## Hsapiens-JASPAR_CORE-NFIL3-MA0025.1
## Hsapiens-JASPAR_CORE-ELK1-MA0028.1
## ...
## Hsapiens-JASPAR_CORE-SPI1-MA0080.2
## Hsapiens-JASPAR_CORE-AP1-MA0099.2
## Hsapiens-JASPAR_CORE-SP1-MA0079.2
## Hsapiens-JASPAR_CORE-ESR2-MA0258.1
## Hsapiens-JASPAR_CORE-HIF1A::ARNT-MA0259.1
```

Additionally in our library we provide the hocomoco core position count matrix in text format. This is an example, on how to provide your own motif database (currently only possible in .txt).

```
motifs <- system.file("extdata", "hocomoco.core.txt", package = "TFBSfindR")
motifs <- read.motif.database(motifs)
motifs
## List of length 401
## names(401): >AHR_HUMAN.H11MO.0.B ... >ZSC31_HUMAN.H11MO.0.C
```

3.3 Step 3| Motif analysis

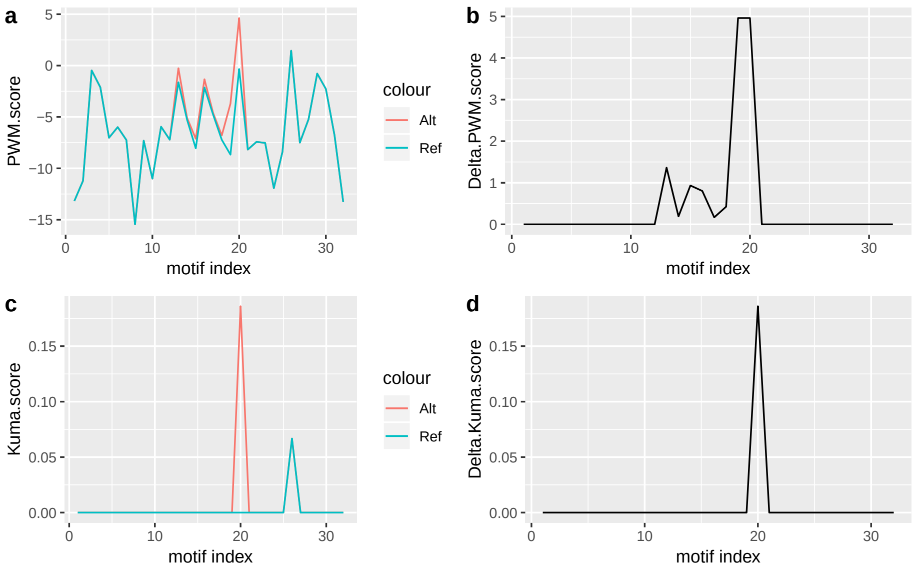
In Step 3, the motifs from JASPARCORE are compared to our variants provided earlier. Several

```
data <- TFBS.findR(data, motiflist = JASPARCORE, motif.type = "PPM",
    pseudocount = "log.of.reads", prior = 0.1, BPPARAM = bpparam)
```

4 Data analysis

some plots . . .

```
snp <- data[data$SNP %in% "rs113663169"]
snp.plot(snp, method = "both", motif = "TFAP2A", strand = "+")
```



update to data.frame

```
data <- data.update(data)
head(data)
##    seqnames     start       end width strand            Sample           SNP Allel
## 1     chr1 11046563 11046571     9      + example.dataset rs113663169   *|*
## 2     chr1 11046654 11046660     7      - example.dataset  rs72868197   *|*
## 3     chr1 11046594 11046598     5      + example.dataset rs112732333   *|*
## 4     chr1 11046590 11046596     7      + example.dataset rs112732333   *|*
## 5     chr1 11046536 11046541     6      + example.dataset  rs60216355   *|*
## 6     chr1 11046593 11046598     6      + example.dataset rs112732333   *|*
##   REF ALT Snp.loc  Sequence     MotifDB provider  Motif Ref.score Alt.score
## 1   T   C       2 GTCTCAGCC JASPAR_CORE MA0003.1 TFAP2A    -0.333     4.628
## 2   T   A       1   AACCGGT JASPAR_CORE MA0099.2    AP1    -0.102     3.098
## 3   G   A       3     GCGTG JASPAR_CORE MA0036.1  GATA2    -1.096     2.185
## 4   G   A       7   ACAGGCG JASPAR_CORE MA0081.1   SPIB    -0.998     2.900
## 5   T   C       2    TTGATC JASPAR_CORE MA0095.1    YY1    -0.193     2.088
## 6   G   A       4    GGCGTG JASPAR_CORE MA0095.1    YY1    -1.195     2.024
##   Delta.score Kuma.ref.score Kuma.alt.score Kuma.delta.score
## 1       4.961              0          0.186            4.293
## 2       3.200              0          0.164            4.124
## 3       3.281              0          0.163            4.109
## 4       3.898              0          0.155            4.049
## 5       2.280              0          0.134            3.847
## 6       3.219              0          0.130            3.811
```

4.1 Extra p-value calculations

Pvalue calculations can be done as follows

```
data <- p.value.calculation(data, motiflist = JASPARCORE, background = c(A = 0.25,
    C = 0.25, G = 0.25, T = 0.25), motif.type = "PPM", pseudocount = "log.of.reads")
head(data)
##   seqnames    start      end width strand          Sample         SNP Allel
## 1     chr1 11046563 11046571     9      + example.dataset rs113663169   *|*
## 2     chr1 11046654 11046660     7      - example.dataset  rs72868197   *|*
## 3     chr1 11046594 11046598     5      + example.dataset rs112732333   *|*
## 4     chr1 11046590 11046596     7      + example.dataset rs112732333   *|*
## 5     chr1 11046536 11046541     6      + example.dataset  rs60216355   *|*
##   REF ALT Snp.loc  Sequence     MotifDB provider  Motif Ref.score Alt.score
## 1   T   C       2 GTCTCAGCC JASPAR_CORE MA0003.1 TFAP2A    -0.333     4.628
## 2   T   A       1   AACCGGT JASPAR_CORE MA0099.2    AP1    -0.102     3.098
## 3   G   A       3     GCGTG JASPAR_CORE MA0036.1  GATA2    -1.096     2.185
## 4   G   A       7   ACAGGCG JASPAR_CORE MA0081.1   SPIB    -0.998     2.900
## 5   T   C       2    TTGATC JASPAR_CORE MA0095.1    YY1    -0.193     2.088
##   Delta.score Kuma.ref.score Kuma.alt.score Kuma.delta.score pvalue.REF
## 1       4.961              0          0.186            4.293 0.03458023
## 2       3.200              0          0.164            4.124 0.06378174
## 3       3.281              0          0.163            4.109 0.15429688
## 4       3.898              0          0.155            4.049 0.06262207
## 5       2.280              0          0.134            3.847 0.05737305
##    pvalue.ALT
## 1 0.002155304
## 2 0.006958008
## 3 0.018554688
## 4 0.006774902
## 5 0.012939453
```

5 SessionInfo

```
sessionInfo()
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.6 LTS
##
## Matrix products: default
## BLAS: /usr/lib/openblas-base/libblas.so.3
## LAPACK: /usr/lib/libopenblasp-r0.2.18.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] MotifDb_1.20.0                     BSgenome.Hsapiens.UCSC.hg19_1.4.0
##  [3] BSgenome_1.46.0                    rtracklayer_1.38.3
##  [5] Biostrings_2.46.0                  XVector_0.18.0
##  [7] GenomicRanges_1.30.3               GenomeInfoDb_1.14.0
##  [9] IRanges_2.12.0                     S4Vectors_0.16.0
## [11] BiocGenerics_0.24.0                TFBSfindR_0.1.0
## [13] BiocParallel_1.12.0                knitr_1.22
## [15] BiocStyle_2.6.1
##
## loaded via a namespace (and not attached):
##  [1] Biobase_2.38.0            httr_1.4.0
##  [3] RMySQL_0.10.16            bit64_0.9-7
##  [5] assertthat_0.2.1          blob_1.1.1
##  [7] GenomeInfoDbData_1.0.0    Rsamtools_1.30.0
##  [9] yaml_2.2.0                progress_1.2.0
## [11] pillar_1.3.1              RSQLite_2.1.1
## [13] lattice_0.20-38           glue_1.3.1
## [15] digest_0.6.18             colorspace_1.4-1
## [17] cowplot_0.9.4             htmltools_0.3.6
## [19] Matrix_1.2-15             plyr_1.8.4
## [21] XML_3.98-1.17              pkgconfig_2.0.2
## [23] biomaRt_2.34.2             bookdown_0.9
## [25] zlibbioc_1.24.0            purrr_0.3.2
## [27] scales_1.0.0               tibble_2.1.1
## [29] ggplot2_3.1.1              SummarizedExperiment_1.8.1
## [31] GenomicFeatures_1.30.3     TFMPvalue_0.0.8
## [33] lazyeval_0.2.2             splitstackshape_1.4.6
## [35] magrittr_1.5               crayon_1.3.4
## [37] memoise_1.1.0              evaluate_0.13
## [39] data.table_1.12.0          tools_3.4.4
## [41] prettyunits_1.0.2          hms_0.4.2
## [43] formatR_1.5                matrixStats_0.54.0
## [45] stringr_1.4.0              munsell_0.5.0
## [47] DelayedArray_0.4.1         AnnotationDbi_1.40.0
## [49] compiler_3.4.4             rlang_0.3.4
## [51] grid_3.4.4                 RCurl_1.95-4.11
## [53] VariantAnnotation_1.24.5   labeling_0.3
## [55] bitops_1.0-6               rmarkdown_1.11
## [57] gtable_0.3.0               codetools_0.2-16
## [59] DBI_1.0.0                  R6_2.4.0
## [61] GenomicAlignments_1.14.2   dplyr_0.8.0.1
## [63] bit_1.1-14                 stringi_1.4.3
## [65] Rcpp_1.0.1                 tidyselect_0.2.5
## [67] xfun_0.6
```