Package 'MeinteR'

May 6, 2019

Type Package

Title MeinteR (MEthylation INTERpretation): A computational method to prioritize aberrant DNA methylation using local genomic substrate

Description MeinteR builds genomic signatures of differential methylated sites based on a set of transcriptional regulatory elements and prioritizes critical sites that more likely have strong influence on phenotype expression.

Version 0.99.0

URL https://github.com/andigoni/MeinteR

Date 2019-04-30

License GPL-3

LazyData TRUE

Depends R (>= 3.5.0), Biostrings, pqsfinder, DNAshapeR, FDb.InfiniumMethylation.hg19

Suggests testthat, knitr

VignetteBuilder knitr

biocViews DNAMethylation, AlternativeSplicing, DifferentialMethylation, GeneRegulation, Epigenetics,Sequencing,MethylationArray,MethylSeq,EpigeneticsWorkflow

Imports GenomicRanges, BSgenome.Hsapiens.UCSC.hg19, ggplot2, dplyr, plyr, rtracklayer, BSgenome, tidyverse, XVector, GenomeInfoDb, S4Vectors, stats4, stats, IRanges, BiocGenerics, parallel, reshape2, TFBSTools, JASPAR2018, graphics, utils, grDevices, TxDb.Hsapiens.UCSC.hg19.knownGene, GenomicFeatures, GEOquery, Biobase

Author Andigoni Malousi <andigoni@auth.gr>

Maintainer Andigoni Malousi <andigoni@auth.gr>

RoxygenNote 6.0.1

R topics documented:

`bed2Seq` *Fetch sequences from bed-formatted data frames*

Description

Fetch sequences from bed-formatted data frames

Usage

```
bed2Seq(bedline, offset)
```

Arguments

`bedline`	Valid bed-formatted data frame
`offset`	Number of nucleotides expanded in each direction ([1,1000])

Value

A DNAStringset containing the sequences in hg19 genome assembly

cpgIslands *CpG islands (data)*

Description

List of 27,718 human CpG islands with their corresponding GC-content and observed/expected ratio in chromosomes chr1..22,X,Y (hg19). The list is obtained from the cpgIslandExt table of UCSC Table Browser.

Format

`chrom` CpG island chromosome

`chromStart` CpG island chromosome start position

`chromEnd` CpG island chromosome end position

`perGC` GC-content i.e. Percentage of the CpG island that is C or G

`obsExp` Observed/expected ratio i.e.: Number of CpG * N / (Number of C * Number of G), N = Sequence length

cTF *Conserved transcription factors (data)*

Description

Tab-delimited data containing the 634 conserved transcription factors among mouse-rat-human alignments. The list is obtained from the tfbsConsFactors table of UCSC Table Browser.

Format

`name` Identifier of the conserved transcription factor

`factor` Name of the transcription factor

filterByCGI *Filter by CpG islands*

Description

Selects genomic coordinates included in CpG islands, using the cpgIslands dataset.

Usage

```
filterByCGI(input.data)
```

Arguments

`input.data`	A data frame containing input data in bed format

Value

A data frame with the CpG sites located in CpG islands

filterByProm *Filter by promoters*

Description

Selects genomic coordinates included in promoters based on the UCSC hg19 gene coordinates.

Usage

```
filterByProm(input.data, up.tss, down.tss)
```

Arguments

input.data	A data frame containing input data in bed format
up.tss	Number of nucleotides upstream transcription start site
down.tss	Number of nucleotides downstream transcription start site

Value

A data frame with the CpG sites located in promoter regions

findAltSplicing *Find alternative splicing events*

Description

Idetifies known alternative splicing events co-localized with input data.

Usage

```
findAltSplicing(bed.data, known.alt.splice = NULL)
```

Arguments

bed.data	A data frame containing input bed-formatted data
known.alt.splice	(optional) Full local path to the UCSC knownalt table. If the table is not available locally then the script will fetch known alternative splicing events from UCSC (needs Internet connection).

Value

1/ A data frame with the identified alternative splicing event overlaps (hg19)

2/ A summary table with the frequency of each alternative splicing event compared to the reference frequency

3/ A data frame with the number of alternative splicing events per sequence (input to `meinter` function)

4/ An overlayed bar chart object

`findConservedTFBS` *Find differentially methylated sites overlapping human/mouse/rat conserved transcription factor binding sites.*

Description

Detects transcription factor binding sites that are conserved in human/mouse/rat alignments and overlap with the input data. A binding site is considered to be conserved across the alignment if its score meets the threshold score for its binding matrix in all three species. The score and threshold are computed with the Transfac Matrix Database (v7.0) created by Biobase. The data are purely computational, and as such not all binding sites listed here are biologically functional binding sites.

Usage

```
findConservedTFBS(bed.data, known.conserved.tfbs.file = NULL)
```

Arguments

`bed.data`	A data frame containing input bed-formatted data
`known.conserved.tfbs.file`	(optional) Full local path to the UCSC conserved transcription factor binding sites. If the table is not available locally then the script will fetch it from UCSC (needs Internet connection). NOTE: It is recommended to download the compressed file (Unzipped file >290MB)

Value

1/ Data frame containing overlaps between `bed.data` and conserved transcription factor binding sites

2/ Frequency table of conserved transcription factors on human genome (input to `scatterConsTF` function)

3/ A data frame with the number of conserved transcription factor binding sites per sequence (input to `meinter` function)

`findPals` *Find palindromes in a bed-formatted dataset*

Description

Deetects whether the target cytosine overlaps with a palindromic sequences or it is located inbetween of the two arms of a palindromic sequence i.e. in the loop formed by the palindrome.

Usage

```
findPals(bed.data, offset = 10, min.arm = 5, max.loop = 5,
  max.mismatch = 1)
```

Arguments

bed.data	A data frame containing input bed-formatted data
offset	Number of nucleotides expanded in each direction (default:10, max:200)
min.arm	Minimum length of each arm (default:5)
max.loop	Maximum length of the loop between the two arms of the palindrome
max.mismatch	The maximum number of mismatching letters allowed between the two arms of the palindromes

Value

1/ DNAString subject with the identified palindromes

2/ Number of palindromes falling on/neighboring input data

3/ Number of palindromes per sequence (input to ‘meinter‘ function)

---

findQuads *Find quadruplexes in sequences centered at CpG sites*

Description

This function will detect DNA sequence patterns that likely fold into G-quadruplex structures.

Usage

```
findQuads(bed.data, offset = 100)
```

Arguments

bed.data	A data frame containing input bed-formatted data
offset	Number of nucleotides expanded in each direction (default:100, max:1000)

Value

A DNAString subject with the identified G-quadruplexes, their length and relative coordinates

Number of G-quadruplexes per sequence (input to ‘meinter‘ function)

findShapes *Find putative conformational DNA changes*

Description

Predicts conformational changes of DNA shapes, such as minor groove width (MGW), Roll, propeller twist (ProT) and helix twist (HeIT) in the unmethylated and methylated context using methylDNAshape algorithm.

Usage

```
findShapes(bed.data, offset = 50)
```

Arguments

bed.data	A data frame containing input bed-formatted data
offset	Number of nucleotides expanded in each direction (default:50, max:200)

Value

1/ p-value of the MGW in the unmethylated/methylated CpG context for each sequence

2/ p-value of the HeIT in the unmethylated/methylated CpG context for each sequence

3/ p-value of the ProT in the unmethylated/methylated CpG context for each sequence

4/ p-value of the Roll in the unmethylated/methylated CpG context for each sequence

findSpliceSites *Find splice sites*

Description

Detects potential splice sites in the proximal region of the input genomic coordinates. The function implements the prediction model proposed by Shapiro and Senapathy (Shapiro MB, Senapathy P. Nucleic Acids Research. 1987;15(17):7155-7174.)

Usage

```
findSpliceSites(bed.data, persim = 0.8, offset = 10)
```

Arguments

bed.data	A data frame containing input bed-formatted data
persim	Similarity with the splice site consensus (default:0.8, range between [0,1])
offset	Number of nucleotides expanded in each direction (default:10, min:5, max:50)

Value

1/ A detailed table with the location of the detected splice sites in each sequence and the corresponding similarity score

2/ A summary table with the number of splice sites detected in each sequence (input 'meinter' function)

findTFBS *Find putative transcription factor binding sites*

Description

Detects JASPAR's transcription factor binding sites (core collection), co-localized with input data. Both sequence strands are examined. The analysis can be restricted to promoters (use 'uptss' and 'down.tss' to define promoter length, relative to transcription start site) and CpG islands of the human genome (hg19).

Usage

```
findTFBS(bed.data, persim = 0.8, offset = 12, target = "PROMOTER",
  up.tss = 1000, down.tss = 100, mcores = NULL, tf.ID = NULL)
```

Arguments

bed.data	A data frame containing input bed-formatted data
persim	Minimum similarity with transcription factors consensus matrices (default:0.8, range in [0,1])
offset	Number of nucleotides expanded in each direction (default:12, min:5, max:100)
target	Search for transcription factor binding sites on specific regions. 'PROMOTER': selects sites located in promoter regions, 'CGI': selects sites in CpG islands, 'ALL': No filtering is applied (time-consuming for large datasets) (default: "PROMOTER")
up.tss	Number of nucleotides upstream transcription start site (Only when target="PROMOTER" is set, default: 1000)
down.tss	Number of nucleotides downstream transcription start site (Only when target="PROMOTER" is set, default: 100)
mcores	Number of cores to be used (default: maximum available)
tf.ID	A vector of JASPAR transcription factors identifiers to search for (default: all)

Value

1/ Data frame containing the transcription factors identified in each sequence, their position and binding score (input to 'plotTF' function)

2/ Data frame of the detected transcription factor binding sites per sequence (input to 'meinter' function)

importGEO *Import GEO data series in the workspace*

Description

Imports GEO data series. The function fetches data matrices corresponding to a pre-defined GSE identifier and builds valid, bed-formatted dataset with delta-beta values between two sample groups, described in a user-defined annotation file.

Usage

```
importGEO(gse.acc, annotation.file)
```

Arguments

gse.acc	A string corresponding to the accession number of the GEO data series
annotation.file	A string corresponding to the full local path to the annotation files containing sample grouping information

Value

1/ A bed-formatted data frame with the chromosomal coordinates and of each methylation probe and the corresponding delta-beta values between the two groups

2/ Beta values of each sample listed in the annotation file

3/ Annotation data frame

4/ Mean beta values of group 1

5/ Mean beta values of group 2

isEmptyDF *Check if data frame is empty*

Description

Checks if a data frame has no values

Usage

```
isEmptyDF(df)
```

Arguments

df	The input data frame

Value

TRUE/FALSE (TRUE is the data frame is empty)

loadFile *Load input data*

Description

Loads tabular files containing methylation data. The function checks the delimiter and validates the order of the columns (chr, start,end,score,strand).

Usage

```
loadFile(FH)
```

Arguments

FH	Full path of the tabular methylation data

Value

df A data frame with the tabular methylation data

loadSeqGEO *Reformat methylation sequencing data fetched from GEO*

Description

Transforms sequencing data into bed-formatted files. Valid for per sample usage.

Usage

```
loadSeqGEO(file.path, cov = 10, chroms = NULL)
```

Arguments

file.path	Local folder of the bed.gz file
cov	Minimum read coverage (default:10)
chroms	A vector of chromosome vector to be included in the analysis (default:ALL)

Value

A valid bed-formatted data frame

meinter — *Calculate the genomic index of methylation sites based on the Meinter's 'find*' functions' outputs*

Description

Calculates the genomic index given a set of features pre-analysed using MeinteR's 'find*' functions. First, the function builds the local genomic signature of each site and the it calculates the genomic index using a weighting scheme.

Usage

```
meinter(bed.data, funList, weights)
```

Arguments

bed.data	A data frame containing input bed-formatted data
funList	List of 'find*' functions outputs. At least one core function is needed to calculate the genomic index. Valid element names of the list: 'spls'-'findSpliceSites', 'altss'-'findAltSplicing', 'ctfbs'-'findConservedTFBS', 'tfbs'-'findTFBS', 'pals'-'findPals', 'quads'-'findQuads', 'shapes'-'findShapes'
weights	A list of positive values corresponding to feature weights [0,10]. Same list elements with 'funList' list

Value

A data frame with the genomic index of the input data

nameStudy — *Set a study name*

Description

Sets a name to the analysis that appears in the exported plots.

Usage

```
nameStudy(study.name)
```

Arguments

study.name	A string corresponding to the name of the study

Value

The name of the study

`plotBeta` *Plot scores of the input data Generates a density plot of the score values listed in the input dataset.*

Description

Plot scores of the input data Generates a density plot of the score values listed in the input dataset.

Usage

```
plotBeta(bed.data)
```

Arguments

`bed.data`	A data frame containing input bed-formatted data

`plotCpG` *Plot G+C-content and observed/expected ratio.*

Description

Generates density plots of the G+C content and observed/expected CpG ratio for the input dataset and the human genome CpG islands

Usage

```
plotCpG(bed.data, offset = 200)
```

Arguments

`bed.data`	A data frame containing input bed-formatted data
`offset`	Number of nucleotides expanded in each direction (default:200, min:20 max:1000)

Value

1/ A data frame containing the G+C content (percentage of island that is C or G) and ratio of observed (CpG number) to expected(Number of C* Number of G/sequence length)

2/ Density plot of the G+C-content

3/ Density plot of the observed/expected ratio

plotTF *Create barplot of the identified transcription factor binding sites*

Description

Generates an overlayed barplot of the results exported by the 'findTFBS' function. The bar plot visualises the most frequent transription factors with respect to the total number of occurrences and the number of sequences that contain these trascription factors.

Usage

```
plotTF(df, topTF = 10)
```

Arguments

df	The data frame exported by the 'findTFBS' function
topTF	Integer corresponding to the number of the most frequent trascription factors to be displayed (default:10)

Value

1/ A barplot with the 'topTF' most frequent transcription factors

2/ A barplot with the number of transcription factors per class

3/ A scatterplot comparing the observed and expected number of transcription factors per class

refFreq *Human transcription factor frequency (data)*

Description

Frequency of the human conserved transcription factors in the reference genome. The frequency of each transcription factor is calculated using the UCSC Table Browser (tables: tfbsConsFactors, tfbsConsSites).

Format

`factor` Name of the transcription factor

`freq` Frequency in the human genome

reorderBed *Reorder tabular methylation data to bed format*

Description

Reorders tabular methylation data to bed-formatted files. Compatible inputs are .txt, .csv data and other textual formats that contain the following mandatory columns: chr, start, end and score.

Usage

```
reorderBed(input.data, chr.col, start.col, end.col, score.col,
  strand.col = NULL)
```

Arguments

input.data	A data frame containing input bed-formatted data
chr.col	Column number containing the chromosome name
start.col	Column number containing the chromosome's start position
end.col	Column number containing the chromosome's end position
score.col	Column number containing the methylation score values either beta or delta-beta
strand.col	Column number containing the strand in the use data file ('+' strand is assumed if strand column is missing)

Value

A valid bed-formatted file (input of the 'MeinteR::find*' functions)

sample *Sample DNA methylation dataset (data)*

Description

A data frame with 5840 methylation sites containing the chromosomal position of the methylation sites and the corresponding delta beta values.

Format

The dataset has the following 5 variables:

Chromosome A factor with valid values chr1 to chr22, chrX, chrY, chrM

Start Start position of the methylation site

End End position of the methylation site

Strand Strand of the methylation site, either + or -

Differences A numeric vector with the (group1 - group2) methylation value for each CpG site.

Examples

```
#Distribution of the methylation values
plot(sample[,5])
```

scatterConsTF *Create a scatterplot of the identified conserved transcription factors*

Description

Generates a scatterplot of the results exported by the 'findConservedTFBS' function. The scatterplot illustrates the number of binding sites per transription factor relative to the expected frequency on the reference human genome. The trascription factors with high frequency (>= 3rd quantile) in the reference genome or to the analysed data are labeled on the scatterplot.

Usage

```
scatterConsTF(df)
```

Arguments

df	The data frame exported by the 'findConservedTFBS' function

test.data *Test dataset with chromosomal position of the methylated sites and the corresponding delta beta values (data)*

Description

A valid and well-formatted sample dataset containing 401 differentially methylated data with |delta beta-values|>0.3.

Format

The dataset has the following 5 variables:

`chr` A factor with valid values chr1 to chr22, chrX, chrY and chrM

`start` Start position of the methylation site

`end` End position of the methylation site

`score` A numeric vector with the (group1 - group2) methylation value for each methylation site.

`strand` Strand of the methylation site, either + or -

Examples

```
#Distribution of the methylation values
plot(test.data$score)
```

TF.class *Transcription factor classes (data)*

Description

Tabular file containing the number of transcription factors per class.

Format

A data frame with 34 trascription factor classes and their corresponding number of factors in each class.

`class` Name of the class

`Number` Number of factors in the class

validateBed *Validate format of the input bed data*

Description

Validates input methylation data. Checks the presence of the chr, start, end, score columns. If column 'strand' is not set then '+' strand is assumed. Cleans rows with empty cells and sets numeric format to the start, end and score columns.

Usage

```
validateBed(bed.data, omit.na = TRUE)
```

Arguments

`bed.data`	A data frame containing input bed-formatted data
`omit.na`	Omit rows with empty cells (default:TRUE)

Value

A well-formatted data frame

Index