

Package ‘enhancerHomologSearch’

February 19, 2025

Type Package

Title Identification of putative mammalian orthologs to given enhancer

Version 1.13.2

Description Get ENCODE data of enhancer region via H3K4me1 peaks and search homolog regions for given sequences. The candidates of enhancer homolog regions can be filtered by distance to target TSS. The top candidates from human and mouse will be aligned to each other and then exported as multiple alignments with given enhancer.

BugReports <https://github.com/jianhong/enhancerHomologSearch/issues>

URL <https://jianhong.github.io/enhancerHomologSearch>

Depends R (>= 4.1.0), methods

Imports BiocGenerics, Biostrings, BSgenome, BiocParallel, BiocFileCache, GenomeInfoDb, GenomicRanges, httr, IRanges, jsonlite, motifmatchr, Matrix, pwalk, rtracklayer, Rcpp, S4Vectors, stats, utils

Suggests knitr, rmarkdown, BSgenome.Drerio.UCSC.danRer10, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Mmusculus.UCSC.mm10, TxDb.Hsapiens.UCSC.hg38.knownGene, org.Hs.eg.db, TxDb.Mmusculus.UCSC.mm10.knownGene, org.Mm.eg.db, MotifDb, testthat, TFBSTools

biocViews Sequencing, GeneRegulation, Alignment

License GPL (>= 2)

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.2

LinkingTo Rcpp

git_url <https://git.bioconductor.org/packages/enhancerHomologSearch>

git_branch devel

git_last_commit 27b3973

git_last_commit_date 2025-01-03

Repository Bioconductor 3.21

Date/Publication 2025-02-19

Author Jianhong Ou [aut, cre] (ORCID: <<https://orcid.org/0000-0002-8652-2488>>),
Valentina Cigliola [dct],
Kenneth Poss [fnd]

Maintainer Jianhong Ou <jou@morgridge.org>

Contents

| | |
|---------------------------|-----------|
| alignment | 2 |
| alignmentOne | 4 |
| conservedMotifs | 4 |
| Enhancers-class | 6 |
| getENCODEdata | 8 |
| motifs | 9 |
| queryEncode | 9 |
| saveAlignments | 10 |
| searchTFBPS | 11 |
| shuffle | 12 |
| Index | 13 |

alignment

Output

Description

Do pairwise alignment for query enhancer to target genome

Usage

```
alignment(
  query,
  subject,
  method = c("ClustalW", "Muscle"),
  cluster = c("nj", "upgma", "upgmamax", "upgmamin", "upgmb"),
  substitutionMatrix = c("iub", "clustalw"),
  gapOpening = ifelse(method[1] == "ClustalW", 15, 400),
  gapExtension = ifelse(method[1] == "ClustalW", 6.66, 0),
  maxiters = ifelse(method[1] == "ClustalW", 3, 16),
  order = c("aligned", "input"),
  ...
)
```

Arguments

| | |
|--------------------|---|
| query | An object of DNASTringSet to represent enhancer |
| subject | An list of objects of Enhancers . |
| method | specifies the multiple sequence alignment to be used; currently, "ClustalW", and "Muscle" are supported. Default is "Muscle" |
| cluster | The clustering method which should be used. Possible values are "nj" (default) and "upgma". In the original ClustalW implementation, this parameter is called clustering. |
| substitutionMatrix | substitution matrix for scoring matches and mismatches; The valid choices for this parameter are "iub" and "clustalw". In the original ClustalW implementation, this parameter is called matrix. |
| gapOpening | gap opening penalty; the default is 400 for DNA sequences and 420 for RNA sequences. The default for amino acid sequences depends on the profile score settings: for the setting le=TRUE, the default is 2.9, for sp=TRUE, the default is 1,439, and for sv=TRUE, the default is 300. Note that these defaults may not be suitable if custom substitution matrices are being used. In such a case, a sensible choice of gap penalties that fits well to the substitution matrix must be made. |
| gapExtension | gap extension penalty; the default is 0. |
| maxiters | maximum number of iterations; the default is 16. |
| order | how the sequences should be ordered in the output object; if "aligned" is chosen, the sequences are ordered in the way the multiple sequence alignment algorithm orders them. If "input" is chosen, the sequences in the output object are ordered in the same way as the input sequences. |
| ... | Parameters can be used by Muscle, or ClustalW. |

Value

An object of [Enhancers](#).

Examples

```
library(BSgenome.Hsapiens.UCSC.hg38)
library(BSgenome.Mmusculus.UCSC.mm10)
library(BSgenome.Drerio.UCSC.danRer10)
LEN <- GRanges("chr4", IRanges(19050041, 19051709))
seqEN <- getSeq(BSgenome.Drerio.UCSC.danRer10, LEN)
aln_hs <- readRDS(system.file("extdata", "aln_hs.rds",
                             package="enhancerHomologSearch"))
genome(aln_hs) <- Hsapiens
aln_mm <- readRDS(system.file("extdata", "aln_mm.rds",
                             package="enhancerHomologSearch"))
genome(aln_mm) <- Mmusculus
al <- alignment(seqEN, list(human=aln_hs, mouse=aln_mm),
                method="ClustalW", order="input")
```

alignmentOne *Get alignment scores*

Description

Do pairwise alignment for query enhancer to target genome

Usage

```
alignmentOne(query, subject, block = 1000, bpparam = bpparam(), ...)
```

Arguments

| | |
|---------|---|
| query | An object of DNASTringSet to represent enhancer |
| subject | Output of getENCODEdata. An object of Enhancers |
| block | The size of sequences to do alignment. Increase the size will increase the memory cost. Default 1000. |
| bpparam | BiocParallel parameters. |
| ... | not used. |

Value

An object of [Enhancers](#).

Examples

```
library(BiocParallel)
bpparam <- MulticoreParam(workers = 1, tasks=200, progressbar=TRUE)
library(BSgenome.Hsapiens.UCSC.hg38)
peaks <- GRanges("chr1", IRanges(seq(5000, 50000, by=1000), width=1000))
peaks$id <- paste(seq_along(peaks), 1, sep="_")
subj <- Enhancers(genome=Hsapiens, peaks=peaks)
q <- getSeq(Hsapiens, GRanges("chr1", IRanges(90000, width=1000)))
ao <- alignmentOne(q, subj, bpparam=bpparam)
```

conservedMotifs *check the conserved motifs in the orthologs*

Description

Print the conserved motifs in the alignments

Usage

```
conservedMotifs(
  aln,
  aln_list,
  PWMs,
  queryGenome,
  background = "genome",
  ...,
  output_folder,
  format = c("txt", "html")
)
```

Arguments

| | |
|----------------------------|--|
| <code>aln</code> | alignment of multiple DNAs. Output of alignment function. |
| <code>aln_list</code> | The list of output of searchTFBPS such as for human and mouse. |
| <code>PWMs</code> | The Position Weight Matrix list represented as a numeric matrix. Object of PWMMatrixList or PFMatrixList . |
| <code>queryGenome</code> | An object of BSgenome for query enhancer. |
| <code>background</code> | Background nucleotide frequencies. Default is "genome". Refer matchMotifs for details. |
| <code>...</code> | Other parameters can be passed to to matchMotifs . |
| <code>output_folder</code> | Output folder name. |
| <code>format</code> | The format of output files with motif match positions. Available formats are 'txt' and 'html'. Default is 'txt'. |

Value

A list of [XStringViews](#)

Examples

```
library(BSgenome.Hsapiens.UCSC.hg38)
library(BSgenome.Mmusculus.UCSC.mm10)
library(BSgenome.Drerio.UCSC.danRer10)
LEN <- GRanges("chr4", IRanges(19050041, 19051709))
seqEN <- getSeq(BSgenome.Drerio.UCSC.danRer10, LEN)
aln_hs <- readRDS(system.file("extdata", "aln_hs.rds",
  package="enhancerHomologSearch"))
genome(aln_hs) <- Hsapiens
aln_mm <- readRDS(system.file("extdata", "aln_mm.rds",
  package="enhancerHomologSearch"))
genome(aln_mm) <- Mmusculus
al <- alignment(seqEN, list(human=aln_hs, mouse=aln_mm),
  method="ClustalW", order="input")
data(motifs)
conservedMotifs(al[[1]], list(human=aln_hs, mouse=aln_mm),
  motifs[["dist60"]], Drerio)
```

| | |
|-----------------|--------------------------|
| Enhancers-class | <i>Class "Enhancers"</i> |
|-----------------|--------------------------|

Description

An object of class "Enhancers" represents the output of function [getENCODEdata](#), which includes the sequences of enhancers and their genomic coordinates.

Usage

```
Enhancers(genome, peaks, TFBP, TFBP0)

## S4 method for signature 'Enhancers'
x$name

## S4 replacement method for signature 'Enhancers'
x$name <- value

## S4 method for signature 'Enhancers,ANY'
distance(x)

## S4 replacement method for signature 'Enhancers'
distance(x) <- value

## S4 method for signature 'Enhancers'
tfbp(x)

## S4 method for signature 'Enhancers'
query_tfbp(x)

## S4 method for signature 'Enhancers'
getSeq(x, ...)

## S4 method for signature 'Enhancers,ANY'
subsetByOverlaps(
  x,
  ranges,
  maxgap = -1L,
  minoverlap = 0L,
  type = c("any", "start", "end", "within", "equal"),
  invert = FALSE,
  ...
)

## S4 method for signature 'Enhancers'
subset(x, ...)
```

```

## S4 method for signature 'Enhancers'
seqinfo(x)

## S4 method for signature 'Enhancers'
genome(x)

## S4 replacement method for signature 'Enhancers'
genome(x) <- value

## S4 method for signature 'Enhancers'
peaks(x)

## S4 replacement method for signature 'Enhancers'
peaks(x) <- value

## S4 method for signature 'Enhancers'
show(object)

```

Arguments

| | |
|--|---|
| genome | An object of BSgenome . |
| peaks | An object of GRanges . |
| TFBP | An object of IgcMatrix . |
| TFBP0 | An vector of logical. "Enhancers" |
| x | An object of "Enhancers" |
| name | Slot name. |
| value | The values. |
| ... | parameters can be passed to upstream functions. |
| ranges, maxgap, minoverlap, type, invert | parameters used by subsetByOverlaps |
| object | An object of "Enhancers" |

Value

An object of Enhancers.

Slots

| | |
|--------|--|
| genome | An object of BSgenome . |
| peaks | An object of GRanges . |
| TFBP | An object of IgcMatrix . |
| TFBP0 | An vector of logical. |

Examples

```
Enhancers()
```

getENCODEdata

Download enhancer sequences from ENCODE

Description

Query enhancer peak and extract sequences from ENCODE

Usage

```
getENCODEdata(
  genome,
  markers = "H3K4me1",
  window_size = 1000L,
  step = 50L,
  output = c("Enhancer", "raw_peaks"),
  ...
)
```

Arguments

| | |
|-------------------|--|
| genome | An object of BSgenome . |
| markers | Enhancer markers. Default 'H3K4me1'. For active enhancer, it can be set as c('H3K4me1', 'H3K27ac'). If markers is a 'GRanges' object, the function will skip the download step. |
| window_size, step | The size of windows and steps to split the peaks into small pieces. These parameter is used because the width of histone marker peaks are different sizes. Break the peaks into small pieces can increase the matching score and align the matching for different peaks into same size. The window_size is also be used for overlapping detection of multiple histone markers. |
| output | Output format. If it is 'raw_peaks', the raw peaks list will be returned. Otherwise, the Enhancer object will be returned. |
| ... | Parameters can be passed to queryEncode |

Value

An object of [Enhancers](#) with genome, and peaks. The peaks is an object of GRanges. The genome is an object of BSgenome.

Examples

```
library(BSgenome.Hsapiens.UCSC.hg38)
hs <- getENCODEdata(genome=Hsapiens,
                    partialMatch=c(biosample_summary="spinal cord"))
```

`motifs`*Pre-clustered motifs from human and mouse*

Description

The data were extracted from MotifDb package (v 1.34.0) and grouped by motifStack package (v 1.37.2). The data were packaged as PFMATRIXList object by TFBSTools (v 1.30.0)

Usage

```
data(motifs)
```

Format

a list of PFMATRIXList. The names of the list is the group distance.

Source

MotifDb package. Source code for the data generation is in extdata folder

Examples

```
data(motifs)
names(motifs)
motifs[[1]]
```

`queryEncode`*query data from ENCODE by predefined criteria*

Description

Search ENCODE data by querying the ENCODE Portal using its REST API.

Usage

```
queryEncode(
  exactMatch,
  partialMatch = character(0),
  API_url = "https://www.encodeproject.org/search/",
  ...
)
```

Arguments

| | |
|--------------|---|
| exactMatch | character. Exact-match keywords refer to search results that perfectly match all the keywords in the search query, exactly as entered. It is a named character vector. The names will be the keys and characters will be the values for search. |
| partialMatch | character. Partial-match refer to search results that contain the search query. It is a named character vector. The names will be the keys and characters will be the values for search. The value starting from '!' indicates logical negation(NOT). The value starting from '>', '>=', '<', '==', '<=' indicates string comparison. |
| API_url | character. The ENCODE REST API url. |
| ... | Not used. |

Value

A list of search results

Examples

```
res <- queryEncode(
  exactMatch=c(
    target.label="H3K4me1",
    replicates.library.biosample.donor.organism.scientific_name="Homo sapiens",
    assembly="GRCh38",
    assay_term_name="ChIP-seq"),
  partialMatch=c(biosample_summary="heart"))
```

| | |
|----------------|--------------------------|
| saveAlignments | <i>output alignments</i> |
|----------------|--------------------------|

Description

Save enhancer homologs to file in phylip format.

Usage

```
saveAlignments(
  al,
  output_folder = tempdir(),
  motifConsensus = NULL,
  format = c("txt", "html")
)
```

Arguments

| | |
|----------------|---|
| al | output of alignment . |
| output_folder | output folder. |
| motifConsensus | Transcription factor binding consensus. |
| format | The format of output files. Available formats are 'txt' and 'html'. Default is 'txt'. |

Value

The I/O status.

Examples

```
al <- readRDS(system.file("extdata", "al.rds",
                        package="enhancerHomologSearch"))
tmpfolder <- tempdir()
library(MotifDb)
motifs <- query(MotifDb, "JASPAR_CORE")
consensus <- sapply(motifs, consensusString)
consensus <- DNASTringSet(gsub("\\?", "N", consensus))
saveAlignments(al, output_folder=tmpfolder, motifConsensus=consensus)
```

searchTFBPS

Transcription Factor Binding Pattern Similarity (TFBPS) search

Description

Search the TFBPs for query in subject.

Usage

```
searchTFBPS(
  query,
  subject,
  PWMs,
  queryGenome,
  background = "genome",
  ...,
  maximalShuffleEnhancers = 1000
)
```

Arguments

| | |
|-------------------------|--|
| query | An object of DNASTringSet to represent enhancer |
| subject | Output of getENCODEdata. An object of Enhancers |
| PWMs | The Position Weight Matrix list represented as a numeric matrix. Object of PWMMatrixList or PFMatrixList . |
| queryGenome | An object of BSgenome for query data. |
| background | background nucleotide frequencies. Default is "genome". Refer matchMotifs for details. |
| ... | Parameters will be passed to matchMotifs except 'out' and 'genome'. |
| maximalShuffleEnhancers | The maximal number of Shuffled enhancers. If the number of the input enhancer candidates is greater than maximalShuffleEnhancers, no shuffled enhancer sequences will be included. The shuffled enhancers will be created by shuffle . |

Value

An object of [Enhancers](#).

Examples

```
library(BSgenome.Hsapiens.UCSC.hg38)
peaks <- GRanges("chr1", IRanges(seq(5000, 50000, by=1000), width=1000))
peaks$id <- paste(seq_along(peaks), 1, sep="_")
subj <- Enhancers(genome=Hsapiens, peaks=peaks)
q <- getSeq(Hsapiens, GRanges("chr1", IRanges(90000, width=1000)))
data(motifs)
ao <- searchTFBPS(q, subj, motifs[["dist60"]], queryGenome=Hsapiens,
  maximalShuffleEnhancers = 50)
```

shuffle

shuffle reads

Description

Uses the uShuffle library to shuffle reads

Usage

```
shuffle(reads, k = 2, n = 2)
```

Arguments

| | |
|-------|---|
| reads | An object of BStringSet . |
| k | the k-let size. |
| n | the number of random sequences to generate. |

Value

An object of [BStringSet](#).

References

Jiang, M., Anderson, J., Gillespie, J. et al. uShuffle: A useful tool for shuffling biological sequences while preserving the k-let counts. *BMC Bioinformatics* 9, 192 (2008). <https://doi.org/10.1186/1471-2105-9-192>

Examples

```
library(Biostrings)
f <- DNASTringSet(c("CTC-NACCAAGTAT", "TTGA", "TACCTAGAG"))
shuffle(f)
```

Index

- * **datasets**
 - motifs, [9](#)
- \$, Enhancers-method (Enhancers-class), [6](#)
- \$<-, Enhancers-method (Enhancers-class), [6](#)

- alignment, [2](#), [5](#), [10](#)
- alignmentOne, [4](#)

- BStringSet, [12](#)

- BSgenome, [5](#), [7](#), [8](#), [11](#)
- BStringSet, [12](#)

- coerce (Enhancers-class), [6](#)
- coerce, Enhancers, GRanges-method (Enhancers-class), [6](#)
- conservedMotifs, [4](#)

- distance (Enhancers-class), [6](#)
- distance, Enhancers, ANY-method (Enhancers-class), [6](#)
- distance, Enhancers-method (Enhancers-class), [6](#)
- distance<- (Enhancers-class), [6](#)
- distance<-, Enhancers, ANY-method (Enhancers-class), [6](#)
- distance<-, Enhancers-method (Enhancers-class), [6](#)

- Enhancers, [3](#), [4](#), [8](#), [11](#), [12](#)
- Enhancers (Enhancers-class), [6](#)
- Enhancers-class, [6](#)

- genome (Enhancers-class), [6](#)
- genome, Enhancers-method (Enhancers-class), [6](#)
- genome<- (Enhancers-class), [6](#)
- genome<-, Enhancers, BSgenome-method (Enhancers-class), [6](#)
- genome<-, Enhancers-method (Enhancers-class), [6](#)
- getENCODEdata, [6](#), [8](#)

- getSeq, Enhancers-method (Enhancers-class), [6](#)
- GRanges, [7](#)

- lgCMatrix, [7](#)

- matchMotifs, [5](#), [11](#)
- motifs, [9](#)

- peaks (Enhancers-class), [6](#)
- peaks, Enhancers-method (Enhancers-class), [6](#)
- peaks<- (Enhancers-class), [6](#)
- peaks<-, Enhancers, GRanges-method (Enhancers-class), [6](#)
- peaks<-, Enhancers-method (Enhancers-class), [6](#)
- PFMatrixList, [5](#), [11](#)
- PWMMatrixList, [5](#), [11](#)

- query_tfbp (Enhancers-class), [6](#)
- query_tfbp, Enhancers, ANY-method (Enhancers-class), [6](#)
- query_tfbp, Enhancers-method (Enhancers-class), [6](#)
- queryEncode, [8](#), [9](#)

- saveAlignments, [10](#)
- searchTFBPS, [5](#), [11](#)
- seqinfo, Enhancers-method (Enhancers-class), [6](#)
- show, Enhancers-method (Enhancers-class), [6](#)
- shuffle, [11](#), [12](#)
- subset, Enhancers-method (Enhancers-class), [6](#)
- subsetByOverlaps, [7](#)
- subsetByOverlaps, Enhancers, ANY-method (Enhancers-class), [6](#)
- subsetByOverlaps, Enhancers-method (Enhancers-class), [6](#)

tfbp (Enhancers-class), [6](#)
tfbp, Enhancers, ANY-method
 (Enhancers-class), [6](#)
tfbp, Enhancers-method
 (Enhancers-class), [6](#)

XStringViews, [5](#)