Package ‘atSNP’

November 13, 2025

Type Package

Title Affinity test for identifying regulatory SNPs

Version 1.27.0

Date 2021-4-26

Description
atSNP performs affinity tests of motif matches with the SNP or the reference genomes and SNP-led changes in motif matches.

License GPL-2

LinkingTo Rcpp

Depends R (>= 3.6)

Suggests BiocStyle, knitr, rmarkdown

Imports BSgenome, BiocFileCache, BiocParallel, Rcpp, data.table, ggplot2, grDevices, graphics, grid, motifStack, rappdirs, stats, testthat, utils, lifecycle

RoxygenNote 6.1.1

biocViews Software, ChIPSeq, GenomeAnnotation, MotifAnnotation, Visualization

URL https://github.com/sunyoungshin/atSNP

BugReports https://github.com/sunyoungshin/atSNP/issues

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/atSNP

git_branch devel

git_last_commit 0b80e8a

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-11-13

Author Chandler Zuo [aut],
Sunyoung Shin [aut, cre],
Sunduz Keles [aut]

Maintainer Sunyoung Shin <sunyoung.shin@utdallas.edu>

Contents

atSNP-package . . . 2
ComputeMotifScore . . . 3
ComputePValues . . . 4
dtMotifMatch . . . 5
encode_motif . . . 7
encode_motifinfo . . . 7
GetIUPACSequence . . . 8
jaspar_motif . . . 8
jaspar_motifinfo . . . 9
LoadFastaData . . . 9
LoadMotifLibrary . . . 10
LoadSNPData . . . 11
MatchSubsequence . . . 14
motif_library . . . 15
motif_match . . . 16
motif_scores . . . 16
plotMotifMatch . . . 17
prior . . . 18
snpInfo . . . 18
snp_tbl . . . 19
transition . . . 19

Index **20**

`atSNP-package` *atSNP: affinity tests for regulatory SNP detection*

Description

atSNP implements the affinity test for large sets of SNP-motif interactions using the importance sampling algorithm. Users may identify SNPs that potentially may affect binding affinity of transcription factors. Given a set of SNPs and a library of motif position weight matrices (PWMs), atSNP provides two main functions for analyzing SNP effects: (i) the binding affinity score for each allele and each PWM and the p-values for allele-specific binding affinity scores (ii) the p-values for affinity score changes between the two alleles for each SNP. Compared to other bioinformatics tools that provide similar functionalities, atSNP is highly scalable.

The atSNP main functions are:

1. `LoadMotifLibrary` - Load position weight matrices
2. `LoadSNPData` - Load the SNP information and code the genome sequences around the SNP locations
3. `LoadFastaData` - Load the SNP data from fasta files
4. `ComputeMotifScore` - Compute the scores for SNP effects on motifs
5. `ComputePValues` - Compute p-values for affinity scores

Some helper functions are:

1. MatchSubsequence - Compute the matching subsequence
2. GetIUPACSequence - Get the IUPAC sequence of a motif
3. dtMotifMatch - Compute the augmented matching subsequence on SNP and reference alleles

The composite logo plotting function is:

1. plotMotifMatch - Plot sequence logos of the position weight matrix of the motif and sequences of its corresponding best matching augmented subsequence on the reference and SNP allele

Author(s)

Chandler Zuo Sunyoung Shin <sunyoung.shin@utdallas.edu>

References

Zuo, Chandler, Shin, Sunyoung, and Keles, Sunduz. (2015). atSNP: Transcription factor binding affinity testing for regulatory SNP detection. Bioinformatics 31 (20): 3353-5.

See Also

atSNP vignette for more information

ComputeMotifScore *Compute the scores for SNP effects on motifs.*

Description

Compute the log-likelihood scores for motifs.

Usage

```
ComputeMotifScore(motif.lib, snp.info, ncores = 1)
```

Arguments

motif.lib	A list object with the output format of function LoadMotifLibrary.
snp.info	A list object with the output format of function LoadSNPData.
ncores	An integer for the number of parallel process. Default: 1.

Details

This function computes the binding affinity scores for both alleles at each SNP window. For each pair of SNP and motif, it finds the subsequence from both strand that maximizes the affinity binding score. It returns both the matching positions and the maximized affinity scores.

Value

A list of two data.frame's. Field `snp.tbl` contains:

snpid	SNP id.
ref_seq	Reference allele nucleotide sequence.
snp_seq	SNP allele nucleotide sequence.
ref_seq_rev	Reference allele nucleotide sequence on the reverse strand.
snp_seq_rev	SNP allele nucleotide sequence on the reverse strand.

Field `motif.score` contains:

motif	Name of the motif.
motif_len	Length of the motif.
ref_start, ref_end, ref_strand	Location of the best matching subsequence on the referen
snp_start, snp_end, snp_strand	Location of the best matching subsequence on the SNP
log_lik_ref	Log-likelihood score for the reference allele.
log_lik_snp	Log-likelihood score for the SNP allele.
log_lik_ratio	The log-likelihood ratio.
log_enhance_odds	Difference in log-likelihood ratio between SNP allele and reference allele based on the best r
log_reduce_odds	Difference in log-likelihood ratio between reference allele and SNP allele based on the bes

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
data(example)
ComputeMotifScore(motif_library, snpInfo, ncores = 2)
```

ComputePValues *Compute p-values for affinity scores.*

Description

This function computes the p-values for allele-specific affinity scores and between-allele affinity score changes using the importance sampling technique.

Usage

```
ComputePValues(motif.lib, snp.info, motif.scores, ncores = 1,
  testing.mc = FALSE, figdir = NULL)
```

Arguments

`motif.lib`	A list object with the output format of function `LoadMotifLibrary`.
`snp.info`	A list object with the output format of function `LoadSNPData`.
`motif.scores`	A data.frame object containing at least the following columns:

motif	The name of the motif.
log_lik_ref	The log-likelihood score for the reference allele.
log_lik_snp	The log-likelihood score for the SNP allele.

`ncores`	An integer for the number of parallel process. Default: 1.
`testing.mc`	Monte Carlo sample size of 200 is considered. Do not change the default unless conducting a quick test. Default: FALSE
`figdir`	A string for the path to print p-value plots for monitoring results. Default: NULL (no figure).

Value

A data.frame extending `motif.scores` by the following additional columns:

pval_ref	P-values for scores on the reference allele.
pval_snp	P-values for scores on the SNP allele.
pval_cond_ref	Conditional p-values for scores on the reference allele.
pval_cond_snp	Conditional p-values for scores on the SNP allele.
pval_diff	P-values for the difference in scores between the reference and the SNP alleles.
pval_rank	P-values for the log rank ratio between the reference and the SNP alleles.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
data(example)
ComputePValues(motif_library, snpInfo, motif_scores$motif.scores, ncores = 2, testing.mc=TRUE)
```

`dtMotifMatch` *Compute the augmented matching subsequence on SNP and reference allele s.*

Description

Calculate the best matching augmented subsequences on both SNP and reference alleles for motifs. Obtain extra unmatching position on the best matching augmented subsequence of the reference and SNP alleles.

Usage

```
dtMotifMatch(snp.tbl, motif.scores, snpids = NULL, motifs = NULL,
  motif.lib, ncores = 2)
```

Arguments

`snp.tbl` A data.frame with the following information:

snpid	SNP id.
ref_seq	Reference allele nucleobase sequence.
snp_seq	SNP allele nucleobase sequence.
ref_seq_rev	Reference allele nucleobase sequence on the reverse strand.
snp_seq_rev	SNP allele nucleobase sequence on the reverse strand.

`motif.scores` A data.frame with the following information:

motif	Name of the motif.
motif_len	Length of the motif.
ref_start, ref_end, ref_strand	Location of the best matching subsequence on the referen
snp_start, snp_end, snp_strand	Location of the best matching subsequence on the SNP
log_lik_ref	Log-likelihood score for the reference allele.
log_lik_snp	Log-likelihood score for the SNP allele.
log_lik_ratio	The log-likelihood ratio.
log_enhance_odds	Difference in log-likelihood ratio between SNP allele and reference allele based on the best r
log_reduce_odds	Difference in log-likelihood ratio between reference allele and SNP allele based on the bes

`snpids` A subset of snpids to compute the subsequences. Default: NULL, when all snps are computed.

`motifs` A subset of motifs to compute the subsequences. Default: NULL, when all motifs are computed.

`motif.lib` A list of named position weight matrices.

`ncores` The number of cores used for parallel computing. Default: 10

Value

A data.frame containing all columns from the function, `MatchSubsequence`. In addition, the following columns are added:

snp_ref_start, snp_ref_end, snp_ref_length	Location and Length of the best matching augmented subsequence on both the re
ref_aug_match_seq_forward	Best matching augmented subsequence or its corresponding sequence to the forw
snp_aug_match_seq_forward	Best matching augmented subsequence or its corresponding sequence to the forw
ref_aug_match_seq_reverse	Best matching augmented subsequence or its corresponding sequence to the rever
snp_aug_match_seq_reverse	Best matching augmented subsequence or its corresponding sequence to the rever
ref_location	SNP location of the best matching augmented subsequence on the reference allele
snp_location	SNP location of the best matching augmented subsequence on the SNP allele. Sta

ref_extra_pwm_left	Left extra unmatching position on the best matching augmented subsequence of t
ref_extra_pwm_right	Right extra unmatching position on the best matching augmented subsequence of
snp_extra_pwm_left	Left extra unmatching position on the best matching augmented subsequence of t
snp_extra_pwm_right	Right extra unmatching position on the best matching augmented subsequence of

Author(s)

Sunyoung Shin<sunyoung.shin@utdallas.edu>

Examples

```
data(example)
dtMotifMatch(motif_scores$snp.tbl, motif_scores$motif.scores,
motif.lib = motif_library)
```

`encode_motif` *A motif library containing 2065 motifs downloaded from http://compbio.mit.edu/encode-motifs/motifs.txt.*

Description

This motif library can be loaded by 'data(encode_library)'.

Format

A list object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

`encode_motifinfo` *The information for the motif library downloaded from http://compbio.mit.edu/encode-motifs/motifs.txt.*

Description

This is a character vector that be loaded by 'data(encode_library)'. The names of this vector are the same as the names for `encode_motif`. The entries of this vector are the corresponding motif information parsed from the raw file.

Format

A character vector.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

GetIUPACSequence *Get the IUPAC sequence of a motif.*

Description

Convert the posotion weight matrix of a motif to the IUPAC sequence.

Usage

```
GetIUPACSequence(pwm, prob = 0.25)
```

Arguments

pwm	The position weight matrix, with the columns representing A, C, G, T.
prob	The probability threshold. Default: 0.25.

Value

A character string.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
data(example)
GetIUPACSequence(motif_library[[1]], prob = 0.2)
```

jaspar_motif *A motif library containing 593 motifs downloaded from http://jaspar.genereg.net/html/DOWNLOAD/JASPAR_CORE/pfm/nonredundant/pfm_all.txt.*

Description

This motif library can be loaded by 'data(jaspar_library)'.

Format

A list object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

jaspar_motifinfo — *The information for the motif library downloaded from http://jaspar.genereg.net/html/DOWNLOAD/JASPAR_CORE/pfm/nonredundant/pfm_all.txt.*

Description

This is a character vector that be loaded by 'data(jaspar_library)'. The names of this vector are the same as the names for `jaspar_motif`. The entries of this vector are the corresponding motif information parsed from the raw file.

Format

A character vector.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

LoadFastaData — *Load the SNP data from fasta files.*

Description

Load SNP data.

Usage

```
LoadFastaData(ref.filename = NULL, snp.filename = NULL,
  ref.urlname = NULL, snp.urlname = NULL, snpids = NULL,
  default.par = FALSE)
```

Arguments

`ref.filename`	a fastq file name for the reference allele sequences.
`snp.filename`	a fastq file name for the SNP allele sequences.
`ref.urlname`	URL of a fastq file for the reference allele sequences.
`snp.urlname`	URL of a fastq file for the SNP allele sequences.
`snpids`	SNP IDs
`default.par`	A boolean for whether using the default Markov parameters. Default: FALSE.

Value

A list object containing the following components:

sequence_matrix	A list of integer vectors representing the deroxyribose sequence around each SNP.
a1	An integer vector for the deroxyribose at the SNP location on the reference genome.
a2	An integer vector for the deroxyribose at the SNP location on the SNP genome.

The results are coded as: "A"-1, "C"-2, "G"-3, "T"-4.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
LoadFastaData(
ref.urlname="http://pages.stat.wisc.edu/~keles/atSNP-Data/sample_1.fasta",
snp.urlname="http://pages.stat.wisc.edu/~keles/atSNP-Data/sample_2.fasta")
```

LoadMotifLibrary *Load position weight matrices.*

Description

Load the file for position weight matrices for motifs.

Usage

```
LoadMotifLibrary(filename = NULL, urlname = NULL, tag = "MOTIF",
  transpose = FALSE, field = 2, sep = c("\t", " "), skipcols = 0,
  skiprows = 2, pseudocount = 0)
```

Arguments

filename	a MEME format file name.
urlname	URL containing a MEME format file.
tag	A string that marks the description line of the position weight matrix.
transpose	If TRUE (default), then the position weight matrix should have 4 columns. Otherwise, it should have 4 rows.
field	The index of the field in the description line, seperated by space, that indicates the motif name.
sep	A vector of chars for the string separators to parse each lines of the matrix. Default: c(" ", "\t").
skipcols	Number of columns to be skipped in the position weight matrix.
skiprows	Number of description lines before each position weight matrix.
pseudocount	An integer for the pseudocount added to each of the original matrices. Default: 0. Recommended to be 1 if the original matrices are position frequency matrices.

Details

This function reads the formatted file containing motif information and convert them into a list of position weight matrices. The list of arguments should provide enough flexibility of importing a varying number of formats. Some examples are the following: For MEME format, the suggested arguments are: tag = 'Motif', skiprows = 2, skipcols = 0, transpose = FALSE, field = 2, sep = ' '; For motif files from JOHNSON lab (i.e. http://johnsonlab.ucsf.edu/mochi_files/JASPAR_motifs_H_sapiens.txt), the suggested arguments are: tag = '/NAME', skiprows = 1, skipcols = 0, transpose = FALSE, field = 2, sep = "\t"; For JASPAR pfm matrices (i.e. http://jaspar.genereg.net/download/CORE/JASPAR 2018_CORE_vertebrates_non-redundant_pfms_jaspar.txt), the suggested arguments are: tag = ">", skiprows = 1, skipcols = 0, transpose = TRUE, field = 1, sep = "\t"; For the TRANSFAC library provided by UCF bioinformatics groups (i.e. http://gibbs.biomed.ucf.edu/PreDREM/download/nonredundantmotif.transfac), the suggested arguments are: tag = "DE", skiprows = 1, skipcols = 1, transpose = FALSE, field = 2, sep = "\t".

Value

A list object of position weight matrices.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
pwms <- LoadMotifLibrary(
urlname="http://pages.stat.wisc.edu/~keles/atSNP-Data/pfm_vertebrates.txt",
tag = ">", transpose = FALSE, field = 1, sep = c("\t", " ", ">"),
skipcols = 1, skiprows = 1, pseudocount = 1)
```

LoadSNPData — *Load the SNP information and code the genome sequences around the SNP locations.*

Description

Load the SNP data.

Usage

```
LoadSNPData(filename = NULL,
  genome.lib = "BSgenome.Hsapiens.UCSC.hg38",
  snp.lib = "SNPlocs.Hsapiens.dbSNP144.GRCh38", snpids = NULL,
  half.window.size = 30, default.par = FALSE, mutation = FALSE, ...)
```

Arguments

`filename` A table containing the SNP information. Must contain at least five columns with exactly the following names:

chr	chromosome.
snp	The nucleotide position of the SNP.
snpid	The names of the SNPs.
a1	The deoxyribose for one allele.
a2	The deoxyribose for the other allele.

If this file exists already, it is used to extract the SNP information. Otherwise, SNP information extracted using argument 'snpids' is outputted to this file.

`genome.lib` A string of the library name for the genome version. Default: "BSgenome.Hsapiens.UCSC.hg38".

`snp.lib` A string of the library name to obtain the SNP information based on rs ids. Default: "SNPlocs.Hsapiens.dbSNP144.GRCh38".

`snpids` A vector of rs ids for the SNPs. This argument is overidden if the file with name `filename` exists.

`half.window.size` An integer for the half window size around the SNP within which the motifs are matched. Default: 30.

`default.par` A boolean for whether using the default Markov parameters. Default: FALSE.

`mutation` A boolean for whether this is mutation data. See details for more information. Default: FALSE.

`...` Other parameters passed to `read.table`.

Details

This function extracts the nucleotide sequence within a window around each SNP and code them using 1-A, 2-C, 3-G, 4-T.

There are two ways of obtaining the nucleotide sequences. If `filename` is not NULL and the file exists, it should contain the positions and alleles for each SNP. Based on such information, the sequences around SNP positions are extracted using the Bioconductor annotation package specified by `genome.lib`. Users should make sure that this annotation package corresponds to the correct species and genome version of the actual data. Alternatively, users can also provide a vector of rs ids via the argument `snpids`. The SNP locations and allele information is then obtained via the Bioconductor annotation package specified by `snp.lib`, and passed on to the package specified by `genome.lib` to further obtain the nucleotide sequences.

If `mutation=FALSE` (default), this function assumes that the data is for SNP analysis, and the reference genome should be consistent with either the a1 or a2 nucleotide. When extracting the genome sequence around each SNP position, this function compares the nucleotide at the SNP location on the reference genome with both a1 and a2 to distinguish between the reference allele and the SNP allele. If the nucleotide extracted from the reference genome does not match either a1 or a2, the SNP is discarded. The discarded SNPs are in the 'rsid.rm' field in the output.

Alternatively, if `mutation=TRUE`, this function assumes that the data is for general single nucleotide mutation analysis. After extracting the genome sequence around each SNP position, it replaces

the nucleotide at the SNP location by the a1 nucleotide as the 'reference' allele sequence, and by the a2 nucleotide as the 'snp' allele sequence. It does NOT discard the sequence even if neither a1 or a2 matches the reference genome. When this data set is used in other functions, such as `ComputeMotifScore`, `ComputePValues`, all the results (i.e. affinity scores and their p-values) for the reference allele are indeed for the a1 allele, and results for the SNP allele are indeed for the a2 allele.
If the input is a list of rsid's, the SNP information extracted from `snp.lib` may contain more than two alleles for a single location. For such cases, `LoadSNPData` first extracts all pairs of alleles associated with those locations. If 'mutation=TRUE', all those pairs are considered as pairs of reference and SNP alleles, and their information is contained in 'sequence_matrix', 'a1', 'a2' and 'snpid'. If 'mutation=FALSE', `LoadSNPData` further filters these pairs based on whether one allele matches to the reference genome nucleotide extracted from `genome.lib`. Only those pairs with one allele matching the reference genome nucleotide is considered as pairs of reference and SNP alleles, with their information contained in 'sequence_matrix', 'a1', 'a2' and 'snpid'.

Value

A list object containing the following components:

sequence_matrix	A list of integer vectors representing the deroxyribose sequence around each SNP.
a1	An integer vector for the deroxyribose at the SNP location on the reference genome.
a2	An integer vector for the deroxyribose at the SNP location on the SNP genome.
snpid	A string vector for the SNP rsids.
rsid.missing	If the data source is a list of rsids, this field records rsids for SNPs that are discarded because they are not i
rsid.duplicate	If the data source is a list of rsids, this field records rsids for SNPs that based on the SNPlocs package, this
rsid.na	This field records rsids for SNPs that are discarded because the nucleotide sequences contain none ACGT c
rsid.rm	If the data source is a table and `mutation=FALSE`, this field records rsids for SNPs that are discarded becau

The results are coded as: "A"-1, "C"-2, "G"-3, "T"-4.

Author(s)

Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
## Not run: LoadSNPData(snpids = c("rs53576", "rs7412"),
genome.lib ="BSgenome.Hsapiens.UCSC.hg38", snp.lib =
"SNPlocs.Hsapiens.dbSNP144.GRCh38", half.window.size = 30, default.par = TRUE
, mutation = FALSE)
## End(Not run)
```

`MatchSubsequence` *Compute the matching subsequence.*

Description

This function combines the SNP set, the motif library and the affinity score table and produce the matching subsequence found at each SNP location for each motif.

Usage

```
MatchSubsequence(snp.tbl, motif.scores, motif.lib, snpids = NULL,
  motifs = NULL, ncores = 1)
```

Arguments

`snp.tbl` A data.frame with the following information:

snpid	SNP id.
ref_seq	Reference allele nucleotide sequence.
snp_seq	SNP allele nucleotide sequence.
ref_seq_rev	Reference allele nucleotide sequence on the reverse strand.
snp_seq_rev	SNP allele nucleotide sequence on the reverse strand.

`motif.scores` A data.frame with the following information:

motif	Name of the motif.
motif_len	Length of the motif.
ref_start, ref_end, ref_strand	Location of the best matching subsequence on the referen
snp_start, snp_end, snp_strand	Location of the best matching subsequence on the SNP
log_lik_ref	Log-likelihood score for the reference allele.
log_lik_snp	Log-likelihood score for the SNP allele.
log_lik_ratio	The log-likelihood ratio.
log_enhance_odds	Difference in log-likelihood ratio between SNP allele and reference allele based on the best r
log_reduce_odds	Difference in log-likelihood ratio between reference allele and SNP allele based on the bes

`motif.lib` A list of the position weight matrices for the motifs.

`snpids` A subset of snpids to compute the subsequences. Default: NULL, when all snps are computed.

`motifs` A subset of motifs to compute the subsequences. Default: NULL, when all motifs are computed.

`ncores` The number of cores used for parallel computing.

Value

A data.frame containing all columns in both `snp.tbl` and `motif.scores`. In addition, the following columns are added:

ref_match_seq	Best matching subsequence on the reference allele.
snp_match_seq	Best matching subsequence on the SNP allele.
ref_seq_snp_match	Subsequence on the reference allele corresponding to the best matching location on the SNP allele.
snp_seq_ref_match	Subsequence on the SNP allele corresponding to the best matching location on the reference allele.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Examples

```
data(example)
MatchSubsequence(motif_scores$snp.tbl, motif_scores$motif.scores,
motif_library, ncores=2)
```

---

motif_library *A sample motif library.*

Description

A list of the position weight matrices corresponding to motifs, loaded by 'data(example)'.

Format

A list object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

motif_match — *Composit logo plotting input containing motif scores, the matching subsequences and the augmented matching subsequences on SNP and reference allele*

Description

This data.frame object loaded by 'data(example)' contains information about MYC_disc1 match to rs53576.

Format

A data.frame object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

motif_scores — *Scores for the sample snp data computed based on the motif data.*

Description

This list object loaded by 'data(example)' contains two fields:

snp.tbl	A data.frame containing the sequence of nucleobases around each SNP.
motif.scores	A data.frame containing the likelihood scores computed for each SNP and each motif.

Format

A data.frame object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

`plotMotifMatch` *Plot sequence logos of the position weight matrix of the motif and sequences of its corresponding best matching augmented subsequence on the reference and SNP allele.*

Description

Plot the best matching augmented subsequences on the reference and SNP alleles. Plot sequence logos of the position weight matrix of the motif to the corresponding positions of the best matching subsequences on the references and SNP alleles.

Usage

```
plotMotifMatch(motif.match, motif.lib, cex.main = 2, ...)
```

Arguments

`motif.match`	a single row ofdtMotifMatch output in data.frame format
`motif.lib`	A list of position weight matrices
`cex.main`	The size of the main title.
`...`	Other parameters passed to plotMotifLogo.

Value

Sequence logo stacks: Reference subsequences, sequence logo of reference allele matching potision weight matrix, SNP subsequences, sequence logo of SNP allele matching potision weight matrix

Author(s)

Sunyoung Shin<sunyoung.shin@utdallas.edu>

Examples

```
data(example)
plotMotifMatch(motif_match, motif.lib = motif_library)
```

`prior` *Default stationary distribution for nucleotide sequences in the reference genome.*

Description

This parameter is fitted using 61bp windowns around the SNPs in the NHGRI catalog. Loaded by 'data(default_par)'.

Format

A numeric vector.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

`snpInfo` *A data set for SNP information.*

Description

This list object loaded by 'data(example)' contains three fields :

sequence_matrix	A sequence matrix, coded by 1-A, 2-C, 3-G, 4-T, with each column corresponding to a subsequence of 61 b
transition	The transition matrix used in Markov model.
prior	The stationary distribution used in the Markov model.

Format

A list object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

snp_tbl — *A data frame for SNP information.*

Description

This data frame is loaded by 'data(example)'. It is a table including the following columns:

chr	The chromosome.
snp	The SNP location coordinate.
snpid	The SNP label.
a1,a2	The nucleotide on the reference and SNP allele.

Format

A data.frame object.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

transition — *Default transition probability matrix for nucleotide sequences in the reference genome.*

Description

This parameter is fitted using 61bp windowns around the SNPs in the NHGRI catalog. Loaded by 'data(default_par)'.

Format

A 4 by 4 numeric matrix.

Author(s)

Sunyoung Shin <sunyoung.shin@utdallas.edu>, Chandler Zuo <chandler.c.zuo@gmail.com>

Index

∗ **GenomeAnnotation**
atSNP-package, 2
∗ **LogoPlot**
atSNP-package, 2
∗ **MotifAnnotation**
atSNP-package, 2

atSNP-package, 2

ComputeMotifScore, *2*, 3, *13*
ComputePValues, *2*, 4, *13*

dtMotifMatch, *3*, 5

encode_motif, *7*, 7
encode_motifinfo, 7

GetIUPACSequence, *3*, 8

jaspar_motif, 8, *9*
jaspar_motifinfo, 9

LoadFastaData, *2*, 9
LoadMotifLibrary, *2, 3*, *5*, 10
LoadSNPData, *2, 3*, *5*, 11, *13*

MatchSubsequence, *3*, *6*, 14
motif_library, 15
motif_match, 16
motif_scores, 16

plotMotifMatch, *3*, 17
prior, 18

read.table, *12*

snp_tbl, 19
snpInfo, 18

transition, 19