Package ‘MIMP’

May 17, 2014

Type Package

Title Predicting the impact of single nucleotide variants on kinase-substrate phosphorylation

Version 1.0

Date 2013-10-29

Author Omar Wagih

Maintainer Omar Wagih <wagih@ebi.ac.uk>

Description No description

License LGPL

R topics documented:

`bestSequence` *Given a position weight matrix, find the best matching sequence*

Description

Finds the amino acid at each position of the PWM with the highest occurence. Used in matrix similarity score calculation.

Usage

```
bestSequence(pwm)
```

Arguments

`pwm`	Position weight matrix

Examples

```
# No Examples
```

`dohtml` *Helper function for* `results2html`

Description

Helper function for `results2html`

Usage

```
dohtml(x, LOGO_DIR)
```

Arguments

`x`	Data frame resulting from mimp call.
`LOGO_DIR`	Directory containing sequence logo images.

`extractDigits` *Extracts digits from a string and returns them in a numerical form*

Description

Extracts digits from a string and returns them in a numerical form

Usage

```
extractDigits(string)
```

Arguments

`string`	String to be manipulated

Examples

```
extractDigits(A123F)
```

`flankingSequence` *Get flanking sequences of a position.*

Description

This function obtains the flanking sequence at one or more position. Out of bound indices are replaced by a blank charachter.

Usage

```
flankingSequence(seqs, inds, flank = 7, empty.char = "-")
```

Arguments

`seqs`	Charachter vector of sequences. If only one sequence is provided, indices from `inds` are assumed to all be from the same sequence.
`inds`	Numerical vector of positions corresponding to the sequences provided in `seqs`.
`flank`	Value indicating the number of charachters to extract, before and after an index
`empty.char`	Charachter used to replace out of bound flanking sequences

Examples

```
# One sequence and one index. Central charachter is B
flankingSequence(seqs=ABC, inds=2, flank=1)
# An example showing the use of empty.char
flankingSequence(seqs=ABC, inds=2, flank=5)
# An example with muliple sequences and indicies
flankingSequence(seqs=c(ABC, XYZ), inds=c(2, 1), flank=1)
```

`mimp` *Predict the impact of single variants on phosphorylation.*

Description

This function takes in mutation, sequence and phosphorylation data to predict the impact the mutation has on phosphorylation.

Usage

```
mimp(muts, seqs, psites, perc.bg = 90, perc.fg = 10, thresh.log2 = 0,
  display.results = T)
```

Arguments

`muts`	Mutation data file: a space delimited text file or data frame containing two columns (1) gene and (1) mutation. Example:

TP53	R282W
CTNNB1	S33C
CTNNB1	S37F

`seqs`	Sequence data file containing protein sequences in FASTA format OR named list of sequences where each list element is the uppercase sequence and the name of each element is that of the protein. Example: list(TP53="ABCXYZ", CDK2="HJKEWR")
`psites`	Phosphorylation data file (optional): a space delimited text file containing positions of phosphorylation sites. Example:

TP53	280
CTNNB1	29
CTNNB1	44

`perc.bg`	Percentile value between 0 - 100. This value is used to compute a threshold, <ce><b2> from the negative (background) distribution of scores. By default this is the 90th percentile of the background distribution of scores. Anything below the threshold is considered a negative hit.
`perc.fg`	Percentile value between 0 - 100. This value is used to compute a threshold, <ce><b1> from the positive (foreground) distribution of scores. By default this is the 10th percentile of the foreground distribution of scores. Anything above the threshold is considered a positive hit.
`thresh.log2`	Threshold for the absolute value of log ratio. Anything less than this value is discarded (default: 0).

Value

The data is returned in a `data.frame` with the following columns:

`gene`	gene with the rewiring event

`mut`	mutation causing the rewiring event
`psite_pos`	position of the central residue of the phosphosite
`mut_dist`	distance of the mutation from the central phosphosite
`wt`	sequence of the wildtype phosphosite (before the mutation)
`mt`	sequence of the mutated phosphosite (after the mutation)
`score_wt`	matrix similarity score of the wildtype phosphosite
`score_mt`	matrix similarity score of the mutated phosphosite
`log_ratio`	Log2 ratio between mutant and wildtype scores. A high positive log ratio represents a high confidence gain-of-signaling event. A high negative log ratio represents a high confidence loss-of-signaling event.
`pwm`	name of the kinase being rewiried
`perc_wt`	Percentile rank of the wt score
`perc_mt`	Percentile rank of the mutant score

Examples

```
# Get the path to example mutation data
mut.file = system.file("extdata", "mutation_data.txt", package = "MIMP")

# Get the path to example FASTA sequence data
seq.file = system.file("extdata", "sequence_data.txt", package = "MIMP")

# View the files in a text editor
browseURL(mut.file)
browseURL(seq.file)

# Run rewiring analysis
results = mimp(mut.file, seq.file, display.results=TRUE)

# Show head of results
head(results)
```

`mss` *Compute matrix similarity score as described in MATCH algorithm*

Description

Computes matrix similarity score of a PWM with a k-mer. Score ranges from 0-1, as described in [PMID: 12824369]

Usage

```
mss(seqs, pwm, is.kinase.pwm = T, na.rm = F, ignore.ind = 8)
```

Arguments

`seqs`	Sequences to be scored
`pwm`	Position weight matrix
`is.kinase.pwm`	TRUE if PWM is that of a kinase
`na.rm`	Remove NA scores?

Examples

```
# No Examples
```

`pSNVs` *Find phosphorylation related variants (pSNVs)*

Description

Given mutation data and psites, find variants that exist in the flanking regions of the psite

Usage

```
pSNVs(muts, psites, seqs, flank = 7, multicore = F)
```

Arguments

`muts`	Mutation data as data frame of two columns (1) name of gene or protein (2) mutation in the format X123Y, where X is the reference amino acid and Y is the alternative amino acid.
`psites`	Phosphorylation data as a data frame of two columns (1) name of gene or protein (2) Position of the phosphorylated residue
`seqs`	Sequence data as a name list. Names of the list correspond to the gene or protein name. Each entry contains the collapsed sequence.
`flank`	Number of amino acids flanking the psite to be considered
`multicore`	If true, will use mclapply to speed things up!

Examples

```
# No examples
```

`PWM` *Construct position weight matrix*

Description

Makes a position weight matrix given aligned sequences.

Usage

```
PWM(seqs, pseudocount = 0.001, relative.freq = T, type = "AA",
  priors = AA_PRIORS_HUMAN)
```

Arguments

`seqs`	Aligned sequences all of the same length
`pseudocount`	Pseudocount factor. Final pseudocount is background probability * this factor
`relative.freq`	TRUE if each column should be divided by the sum
`type`	Type of sequences 'AA' or 'DNA'
`priors`	Named character vector containing priors of amino acids.

Examples

```
# No examples
```

---

replaceChar — *Replace charachters at certain positions of a string with another charachter.*

Description

Replace charachters at certain positions of a string with another charachter.

Usage

```
replaceChar(string, pos, char)
```

Arguments

`string`	String to be manipulated
`pos`	One or more positions corresponding to charachters to be changed
`char`	Replacement charachter

Examples

```
replaceChar(ABC, 2, X)
```

---

results2html — *Display MIMP results interactively in browser*

Description

Display MIMP results interactively in browser

Usage

```
results2html(x, max.rows = 5000)
```

Arguments

`x`	Data frame resulting from mimp call.
`max.rows`	If data contains more rows than this value, results won't be displayed.

saveTransfac *Save a PWM matrix object to transfac format*

Description

This saves an already generated PWM matrix object in R to transfac format, which can be read in by RWebLogo

Usage

```
saveTransfac(pwm, file.out = tempfile("transfac"), type = "aa")
```

Arguments

`pwm`	PWM matrix object
`file.out`	where the transfac matrix is written
`type`	'aa', 'dna' or 'rna' depending on the namespace

scoreArray *Get weight/probability for each amino acid in a sequence*

Description

Gets weight/probability for the amino acid at each position of the sequence as an array.

Usage

```
scoreArray(seqs, pwm)
```

Arguments

`seqs`	One or more sequences to be processed
`pwm`	Position weight matrix

Examples

```
# No Examples
```

scoreWtMt *Score wt and mt sequences for a pwm*

Description

Score wt and mt sequences for a pwm

Usage

```
scoreWtMt(pwm, mut_ps, is.kinase.pwm = T, thresh.bg = 1, thresh.fg = 0,
  thresh.log2 = 0)
```

Arguments

`pwm`	Position weight matrix of interest
`mut_ps`	psnvs data frame containing wt and mt sequences computed from pSNVs function
`is.kinase.pwm`	TRUE if pwm is that of a kinase
`thresh.bg`	Anything below this threshold is considered a negative hit
`thresh.fg`	Anything above this threshold is considered a positive hit
`thresh.log2`	Threshold for the absolute value of log ratio. Anything less than this value is discarded.

unfactor *Converts all columns of a data frame of class factor to character*

Description

Converts all columns of a data frame of class factor to character

Usage

```
unfactor(df)
```

Arguments

`string`	String to be manipulated

Examples

```
unfactor( data.frame(x=c(A, B)) )
```

`worstSequence` *Given a position weight matrix, find the worst matching sequence*

Description

Finds the amino acid at each position of the PWM with the lowest occurence. Used in matrix similarity score calculation.

Usage

```
worstSequence(pwm)
```

Arguments

`pwm`	Position weight matrix

Examples

```
# No Examples
```

Index