Package 'rmimp'

July 14, 2015

Type Package

Title Predicting the impact of mutations on kinase-substrate
phosphorylation

Version 1.1

Date 2015-07-14

Author Omar Wagih

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

Description MIMP is a machine learning method that predicts the impact of missense single-
nucleotide variants (SNVs) on kinase-substrate interactions. MIMP analyzes kinase se-
quence specificities and predicts whether SNVs disrupt existing phosphorylation sites or cre-
ate new sites. This helps discover mutations that modify protein function by altering kinase net-
works and provides insight into disease biology and therapy development.

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
```

---

computeRewiring	*Score wt and mt sequences for a pwm*

Description

Score wt and mt sequences for a pwm

Usage

```
computeRewiring(obj, mut_ps, prob.thresh = 0.5, log2.thresh = 1,
  include.cent = F, degenerate.pwms = F, .degenerate.groups = c("DE",
  "KR", "ILMV"))
```

Arguments

obj   MIMP kinase object containing PWM, auc, GMM parameters, family name, etc.

mut_ps   psnvs data frame containing wt and mt sequences computed from pSNVs function

prob.thresh   Probability threshold of gains and losses. This value should be between 0.5 and 1.

log2.thresh   Threshold for the absolute value of log ratio between wild type and mutant scores. Anything less than this value is discarded (default: 1).

include.cent   If TRUE, gains and losses caused by mutation in the central STY residue are kept

degeneratePWM	*Create a degenerate PWM i.e. for each aa group, set weight to the best weight of the group at that position e.g. R-2 has weight 0.7, K-2 has weight 0.1. Set both R-2 and K-2 to 0.7*

Description

Create a degenerate PWM i.e. for each aa group, set weight to the best weight of the group at that position e.g. R-2 has weight 0.7, K-2 has weight 0.1. Set both R-2 and K-2 to 0.7

Usage

```
degeneratePWM(pwm, dgroups = c("DE", "KR", "ILMV", "QN", "ST"))
```

Arguments

pwm	position weight matrix
dgroups	groups of amino acids

dohtml	*Helper function for* results2html

Description

Helper function for results2html

Usage

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

Arguments

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

---

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 character.

Usage

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

Arguments

seqs   Character 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 characters to extract, before and after an index

empty.char   Character used to replace out of bound flanking sequences

Examples

```
# One sequence and one index. Central character 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 multiple sequences and indices
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 = NULL, prob.thresh = 0.5, log2.thresh = 1,
  display.results = T, include.cent = F, model.data = "hconf")
```

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(GENEA="ARNDGH", GENEB="YVRRHS")
psites	Phosphorylation data file (optional): a space delimited text file OR data frame containing two columns (1) gene and (1) positions of phosphorylation sites. Example:

TP53	280
CTNNB1	29
CTNNB1	44

prob.thresh	Probability threshold of gains and losses. This value should be between 0.5 and 1.
log2.thresh	Threshold for the absolute value of log ratio between wild type and mutant scores. Anything less than this value is discarded (default: 1).
include.cent	If TRUE, gains and losses caused by mutation in the central STY residue are kept. Scores of peptides with a non-STY central residue is given a score of 0 (default: FALSE).
model.data	Name of specificity model data to use, can be "hconf" : individual experimental kinase specificity models used to scan for rewiring events. For experimental kinase specificity models, grouped by family, set to "hconf-fam". Both are considered high confidence. For lower confidence predicted specificity models , set to "lconf". NOTE: Predicted models are purely speculative and should be used with caution

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 phosphosite
mut_dist	Distance of the mutation relative to the central residue
wt	Sequence of the wildtype phosphosite (before the mutation). Score is NA if the central residue is not S, T or Y
mt	Sequence of the mutated phosphosite (after the mutation). Score is NA if the central residue is not S, T or Y
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-phosphorylation event. A high negative log ratio represents a high confidence loss-of-phosphorylation event. This ratio is NA for mutations that affect the central phosphorylation sites
pwm	Name of the kinase being rewired
pwm_fam	Family/subfamily of kinase being rewired. If a kinase subfamily is available the family and subfamily will be separated by an underscore e.g. "DMPK_ROCK". If no subfamily is available, only the family is shown e.g. "GSK"

nseqs	Number of sequences used to construct the PWM. PWMs constructed with a higher number of sequences are generally considered of better quality.
prob	Joint probability of wild type sequence belonging to the foreground distribution and mutated sequence belonging to the background distribution, for loss and vice versa for gain.
effect	Type of rewiring event, can be "loss" or "gain"

Examples

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

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

# 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, na.rm = F, ignore.central = T)
```

Arguments

seqs	Sequences to be scored
pwm	Position weight matrix
na.rm	Remove NA scores?
ignore.central	If TRUE, central residue is ignore from scoring.

Examples

```
# No Examples
```

predictKinasePhosphosites

Compute posterior probability of wild type phosphosites for kinases

Description

Compute posterior probability of wild type phosphosites for kinases

Usage

```
predictKinasePhosphosites(psites, seqs, model.data = "hconf",
  posterior_thresh = 0.8, intermediate = F, kinases)
```

Arguments

psites	phosphorylation data, see ?mimp for details
seqs	sequence data, see ?mimp for details
model.data	MIMP model used, see ?mimp for details
posterior_thresh	
	posterior probability threshold that the score belongs to the foreground distribution of the kinase, probabilities below this value are discarded (default 0.8)
intermediate	if TRUE intermediate MSS scores and likelihoods are reported (default FALSE)
kinases	vector of kinases used for the scoring (e.g. c("AURKB", "CDK2")), if this isn't provided all kinases will be used .

Value

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

gene	Gene with the rewiring event
pos	Position of the phosphosite
wt	Sequence of the wildtype phosphosite
score_wt	(intermediate value) matrix similarity score of sequence
l.wt.fg	(intermediate value) likelihood of score given foreground distribution
l.wt.bg	(intermediate value) likelihood of score given background distribution
post.wt.fg	posterior probability of score in foreground distribution
post.wt.bg	posterior probability of score in background distribution
pwm	Name of the predicted kinase
pwm_fam	Family/subfamily of the predicted kinase. If a kinase subfamily is available the family and subfamily will be seprated by an underscore e.g. "DMPK_ROCK". If no subfamily is available, only the family is shown e.g. "GSK"

If no predictions were made, function returns NULL

Examples

```
# Get the path to example phosphorylation data
psites.file = system.file("extdata", "ps_data.txt", package = "rmimp")

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

# Run for all kinases
results_all = scoreWtOnly(psites.file, seq.file)

# Run for select kinases
results_select = scoreWtOnly(psites.file, seq.file, kinases=c("AURKB", "CDK2"))
```

---

pRewiringPosterior   *Computing posterior probability - ploss and pgain*

Description

Computing posterior probability - ploss and pgain

Usage

```
pRewiringPosterior(wt.scores, mt.scores, fg.params, bg.params, auc = 1,
  intermediate = F)
```

Arguments

wt.scores	Wild type score
mt.scores	Mutant score
fg.params	Distribution parameters of GMMs (foreground). This is precomputed and comes built into mimp.
bg.params	Distribution parameters of GMMs (background). This is precomputed and comes built into mimp.
auc	AUC of the model. This is precomputed and comes built into mimp.
intermediate	If TRUE, intermediate likelihoods used to compute ploss and pgain is returned. Otherwise only ploss and pgain returned

---

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(md, pd, seqdata, flank = 7)
```

Arguments

`flank`	Number of amino acids flanking the psite to be considered
`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.

Examples

```
# No examples
```

PWM	*Construct position weight matrix*

Description

Makes a position weight matrix given aligned sequences.

Usage

```
PWM(seqs, pseudocount = 0.01, relative.freq = T, is.kinase.pwm = T,
  priors = AA_PRIORS_HUMAN, do.pseudocounts = F)
```

Arguments

`seqs`	Aligned sequences all of the same length
`pseudocount`	Pseudocount factor. Final pseudocount is background probability * this factor
`relative.freq`	Set to TRUE if each column should be divided by the sum
`is.kinase.pwm`	Set to TRUE if matrix is being built for a kinase
`priors`	Named character vector containing priors of amino acids.
`do.pseudocounts`	
	TRUE if we are to add pseudocounts

Examples

```
# No examples
```

---

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.

---

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
```

| 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
```