Package 'rmimp'

February 19, 2018

Type Package

Title Predicting the impact of mutations on kinase-substrate phosphorylation

Version 1.2

Date 2018-02-19

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 sequence specificities and predicts whether SNVs disrupt existing phosphorylation sites or create new sites. This helps discover mutations that modify protein function by altering kinase networks and provides insight into disease biology and therapy development.

License LGPL

RoxygenNote 6.0.1

R topics documented:

computeBinding *Score wt and mt sequences for a pwm*

Description

Score wt and mt sequences for a pwm

Usage

```
computeBinding(obj, mut_ss, mut_location, prob.thresh = 0.5,
  log2.thresh = 1)
```

Arguments

obj	MIMP object containing PWM, GMM parameters, and etc.
mut_ss	snvs data frame containing wt and mt sequences computed from SNVs function
mut_location	list of mutation locations
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).

dohtml *Helper function for* results2html

Description

Helper function for results2html

Usage

```
dohtml(x, LOGO_DIR, HL_DIR, logoExt = ".svg", .webserver = F)
```

Arguments

x	Data frame resulting from mimp call.
LOGO_DIR	Directory containing sequence logo images.
HL_DIR	Directory containing overlays
logoExt	Extension of logo files
.webserver	Request coming from webserver?

dohtmlSh3 *Helper function for* results2html *and sh3 binding*

Description

Helper function for results2html and sh3 binding

Usage

```
dohtmlSh3(x, LOGO_DIR, HL_DIR, logoExt = ".svg", .webserver = F)
```

Arguments

x	Data frame resulting from mimp call.
LOGO_DIR	Directory containing sequence logo images.
HL_DIR	Directory containing overlays
logoExt	Extension of logo files
.webserver	Request coming from webserver?

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, central = T, domain = "phos", species = "human",
  psites = NULL, terminal.range = 5, 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")

`central` Whether the mutation site is at the central residue of the sequence

`domain` Which binding domain to run mimp for

`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

`terminal.range` The number of amino acids used for predicting terminal domain binding.

`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).

`display.results` If TRUE results are visualised in an html document after analysis is complete

`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` (Optional) Position of the phosphosite, if domain = "phos"

`mut_dist` (Optional) Distance of the mutation relative to the central residue, if domain = "phos"

`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	(Optional, available only if domain = "phos") 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	(Optional, available only if domain = "phos") 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)
```

---

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
psite.file = system.file("extdata", "sample_phosphosites.tab", package = "rmimp")

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

# Run for all kinases
results_all = predictKinasePhosphosites(psite.file, seq.file)

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

results2html *Display MIMP results interactively in browser*

Description

Display MIMP results interactively in browser

Usage

```
results2html(x, domain = "phos", max.rows = 5000)
```

Arguments

x	Data frame resulting from mimp call.
domain	Which binding domain to run mimp for
max.rows	If data contains more rows than this value, results won't be displayed.

scoreArrayRolling *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

```
scoreArrayRolling(seqs, pwm)
```

Arguments

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

Examples

```
# No Examples
```

scoreWTSequence *Score wt sequence using PWMs in the model*

Description

Score wt sequence using PWMs in the model

Usage

```
scoreWTSequence(wt_seqs, central = T, domain = "phos", species = "human",
  model.data = "hconf", cores = 2)
```

Arguments

wt_seqs	A list of sequences to be scored
central	Whether the mutation site is at the central residue of the sequence
cores	Number of cores the function could use

SNVs *Find non-central variants (SNVs)*

Description

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

Usage

```
SNVs(md, seqdata, flank)
```

Arguments

md	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.
seqdata	Phosphorylation data as a data frame of two columns (1) name of gene or protein (2) Position of the phosphorylated residue
flank	Number of amino acids flanking the site to be considered

Examples

```
# No examples
```

`trainModel` *Train GMM model and return as a list to be used later. If file is passed, the model will also be save to a .mimp file.*

Description

Train GMM model and return as a list to be used later. If file is passed, the model will also be save to a .mimp file.

Usage

```
trainModel(pos.dir, neg.dir, kinase.domain = F, cores = 2, file = NULL,
  threshold = 10, min.auc = 0.65, priors)
```

Arguments

`pos.dir`	the path to the directory contains positive entries
`neg.dir`	the path to the directory contains negative entries
`kinase.domain`	Whether the domain to be trained is a kinase domain.
`cores`	(optional) the number of CPU cores that can be used to train the model
`file`	(optional) the path to save the model
`threshold`	(optional) the minimum number of scores needed for each domain to train the model
`min.auc`	(optional) the minimum number of AUC needed for each domain to train the model
`priors`	Named character vector containing priors of amino acids.

Value

a GMM model

Examples

```
No examples
```

tSNVs *Find terminal variants (tSNVs)*

Description

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

Usage

```
tSNVs(md, seqdata, terminal)
```

Arguments

md	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.
seqdata	Phosphorylation data as a data frame of two columns (1) name of gene or protein (2) Position of the phosphorylated residue
terminal	Number of amino acids flanking the site to be considered

Examples

```
# No examples
```