Package 'KinSwingR'

November 12, 2026

Type Package

Title KinSwingR: network-based kinase activity prediction

Version 1.30.0

Description KinSwingR integrates phosphosite data derived from mass-spectrometry data and kinase-substrate predictions to predict kinase activity. Several functions allow the user to build PWM models of kinase-subtrates, statistically infer PWM:substrate matches, and integrate these data to infer kinase activity.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports data.table, BiocParallel, sqldf, stats, grid, grDevices

biocViews Proteomics, SequenceMatching, Network

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

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

git_branch RELEASE_3_23

git_last_commit aa2f6cc

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Ashley J. Waardenberg [aut, cre]

Maintainer Ashley J. Waardenberg <a.waardenberg@gmail.com>

Contents

buildPWM 2
cleanAnnotation 3

example_phosphoproteome 5
KinSwingR 5
phosphositeplus_human 6
scoreSequences 6
swing 8
viewPWM 9

Index **11**

`buildPWM` *Generate Position Weight Matrices (PWMs)*

Description

Generate Position Weight Matrices (PWMs) for a table containing centered substrate peptide sequences for a list of kinases. The output of this function is to be used for scoring PWM matches to peptides via scoreSequences()

Usage

```
buildPWM(kinase_table = NULL, wild_card = "_", substrate_length = 15,
  substrates_n = 10, pseudo = 0.01, remove_center = FALSE,
  verbose = FALSE)
```

Arguments

`kinase_table`	A data.frame of substrate sequences and kinase names. Format of data must be as follows: column 1 - kinase/kinase family name/GeneID, column 2 - centered peptide seqeuence.
`wild_card`	Letter to describe sequences that are outside of the protein after centering on the phosphosite (e.g ___MERSTRELCLNF). Default: "_".
`substrate_length`	Full length of substrate sequence (default is 15). Will be trimmed automatically or report error if sequences in kinase_table are not long enough.
`substrates_n`	Number of sequences used to build a PWM model. Low sequence counts will produce poor representative PWM models. Default: "10"
`pseudo`	Small number to add to values for PWM log transformation to prevent log transformation of zero. Default = 0.01
`remove_center`	Remove all peptide seqeuences with the central amino acid matching a character (e.g. "y"). Default = FALSE
`verbose`	Print progress to screen. Default=FALSE

Value

Output is a list containing two tables, "pwm" and "kinase". To access PWMs: pwms$pwm and Table of Kinase and sequence counts: pwms$kinase

Examples

```
## Build PWM models from phosphositeplus data with default of minimum
## of 10 substrate sequences for building a PWM model.

data(phosphositeplus_human)

##randomly sample 1000 substrates for demonstration.
set.seed(1)
sample_pwm <- phosphositeplus_human[sample(nrow(phosphositeplus_human),
1000),]
pwms <- buildPWM(sample_pwm)

## Data frame of models built and number of sequences used to build each
## PWM model:
head(pwms$kinase)
```

`cleanAnnotation` *Function for extracting peptide sequences from multimapped or complex annotated data*

Description

This function extracts unique peptide:annotation combinations from complex annotated data and formats for further analysis using KinSwingR. For instance, example input annotation may be: "A0A096MIX2|Ddx17|494|RSRYRTTSSANNPN". This function will extract the peptide sequence into a second column and associate it all annotations. See vignette for more details.

Usage

```
cleanAnnotation(input_data = NULL, annotation_delimiter = "|",
  multi_protein_delimiter = ":", multi_site_delimiter = ";",
  seq_number = 4, replace = FALSE, replace_search = "X",
  replace_with = "_", verbose = FALSE)
```

Arguments

`input_data` A data.frame of phosphopeptide data. Must contain 4 columns and the following format must be adhered to. Column 1 - Annotation, Column 2 - centered peptide sequence, Column 3 - Fold Change [-ve to +ve], Column 4 - p-value [0-1]. This will extract the peptide sequences from Column1 and replace all values in Column2 to be used in scoreSequences(). Where peptide sequences have not been extracted from the annotation, leave Column2 as NA's.

`annotation_delimiter` The character used to delimit annotations. Default="|"

`multi_protein_delimiter` The character used to delimit multi-protein assignments. Default=":". E.g. Ddx17:Ddx2

multi_site_delimiter
: The character used to delimit multi-site assignments. Default=";". E.g. 494;492

seq_number
: The annotation frame that contains the sequence after delimitation. E.g. The sequence "RSRYRTTSSANNPN" is contained in the 4th annotation frame of the following annotation: "A0A096MIX2|Ddx17|494|RSRYRTTSSANNPN" and would therefore set seq_number=4. Default=4

replace
: Replace a letter that describes sequences outside of the protein after centering on the phosphosite (e.g X in XXXMERSTRELCLNF). Use in combination with replace_search and replace_with to replace amino acids. Options are "TRUE" or "FALSE". Default="FALSE".

replace_search
: Amino Acid to search for when replacing sequences. Default="X"

replace_with
: Amino Acid to replace with when replacing sequences. Default="_"

verbose
: Print progress to screen. Default=FALSE

Value

A data.table with the peptides extracted from the annotation column

Examples

```
## Extract peptide sequences from annotation data:
data(example_phosphoproteome)

## A0A096MJ61|NA|89|PRRVRNLSAVLAART
## The following will extract all the uniquely annotated peptide
## sequences from the "annotation" column and place these in the
## "peptide" column. Where multi-mapped peptide sequences are input,
## these are placed on a new line.
##
## Here, sequences with a "X" and also replaced with a "_". This is ensure
## that PWMs are built correctly.

## Sample data for demonstration:
sample_data <- head(example_phosphoproteome)
annotated_data <- cleanAnnotation(input_data = sample_data,
                                  annotation_delimiter = "|",
                                  multi_protein_delimiter = ":",
                                  multi_site_delimiter = ";",
                                  seq_number = 4,
                                  replace = TRUE,
                                  replace_search = "X",
                                  replace_with = "_")

## Return the annotated data with extracted peptides:
head(annotated_data)
```

example_phosphoproteome *Example phosphoproteome.*

Description

A dataset containing annotated subtrate sequences derived from XXX. See original publication for more details: Engholm-Keller & Waardenberg AJ et al.

Usage

```
example_phosphoproteome
```

Format

A data frame with 6215 rows and 4 variables:

annotation Annotation of phosphorylated peptides

peptide blank - peptides need to be extracted from annotation

fc Fold Change (log2)

pval P-value for fold-change.

KinSwingR *KinSwingR: A package for predicting kinase activity*

Description

This package provides functionality for kinase-subtrate prediction, and integration with phosphopeptide fold change and signficance to assess the local connectivity (swing) of kinase-substrate networks. The final output of KinSwingR is a score that is normalised and weighted for prediction of kinase activity.

Details

Contact a.waardenberg@gmail.com for questions relating to functionality.

buildPWM function

Builds PWMs for kinases from a table of kinases and known substrate sequences.

scoreSequences function

Score kinase PWMs matches against a set of peptide seqeuences.

swing function

Integrates kinase PWMs matches against peptide seqeuences and directionality as well as significance of peptides for prediction of kinase activity.

cleanAnnotation function

Function for extracting peptides from multimapped data

---

phosphositeplus_human *Human kinase-substrates derived from PhosphositePlus.*

Description

A dataset containing human kinases and subtrate sequences. See original publication for more details: Hornbeck et al. Nucleic Acids Res. 40:D261-70, 2012

Usage

```
phosphositeplus_human
```

Format

A data frame with 11985 rows and 2 variables:

kinase human kinase gene symbol

substrate centered substrate sequence for kinase

Source

https://www.phosphosite.org/

---

scoreSequences *Score substrate sequences for matches to kinase Position Weight Matrices (PWMs)*

Description

Scores each input sequence for a match against all PWMs provided from buildPWM() and generates p-values for scores. The output of this function is to be used for building the swing metric, the predicted activity of kinases.

Usage

```
scoreSequences(input_data = NULL, pwm_in = NULL,
  background = "random", n = 1000, force_trim = FALSE,
  verbose = FALSE)
```

Arguments

input_data	A data.frame of phoshopeptide data. Must contain 4 columns and the following format must be adhered to. Column 1 - Annotation, Column 2 - centered peptide sequence, Column 3 - Fold Change [-ve to +ve], Column 4 - p-value [0-1]
pwm_in	List of PWMs created using buildPWM()
background	Option to provide a data.frame of peptides to use as background. If providing a background as a table, this must contain two columns; Column 1 - Annotation, Column 2 - centered peptide sequence. These must be centered. OR generate a random background for PWM scoring from the input list - background = random. Default: "random"
n	Number of permutations to perform for generating background. Default: "1000"
force_trim	This function will detect if a peptide sequence is of different length to the PWM models generated (provided in pwm_in) and trim the input sequences to the same length as the PWM models. If a background is provided, this will also be trimmed to the same width as the PWM models. Options are: "TRUE, FALSE". Default = FALSE
verbose	Turn verbosity on/off. To turn on, verbose=TRUE. Options are: "TRUE, FALSE". Default = FALSE

Value

A list with 3 elements: 1) PWM-substrate scores: substrate_scores$peptide_scores, 2) PWM-substrate p-values: substrate_scores$peptide_p 3) Background used for reproducibility: substrate_scores$background 4) input_data is returned in the case that it was trimmed.

Examples

```
## import data
data(example_phosphoproteome)
data(phosphositeplus_human)

## clean up the annotations
## sample 100 data points for demonstration
sample_data <- head(example_phosphoproteome, 100)
annotated_data <- cleanAnnotation(input_data = sample_data)

## build the PWM models:
set.seed(1234)
sample_pwm <- phosphositeplus_human[sample(nrow(phosphositeplus_human),
1000),]
pwms <- buildPWM(sample_pwm)

## score the PWM - substrate matches
## Using a "random" background, to calculate the p-value of the matches
## Using n=10 for demonstration
## set.seed for reproducibility
set.seed(1234)
substrate_scores <- scoreSequences(input_data = annotated_data,
                                   pwm_in = pwms,
```

```
background = "random",
n = 10)
```

`swing` *Swing statistic*

Description

This function integrates the kinase-substrate predictions, directionality of phosphopeptide fold change and signficance to assess local connectivity (swing) of kinase-substrate networks. The final score is a normalised and weighted score of predicted kinase activity. If permutations are selected, network node:edges are permutated. P-values will be calculated for both ends of the distribution of swing scores (positive and negative swing scores).

Usage

```
swing(input_data = NULL, pwm_in = NULL, pwm_scores = NULL,
  pseudo_count = 1, p_cut_pwm = 0.05, p_cut_fc = 0.05,
  permutations = 1000, return_network = FALSE, verbose = FALSE)
```

Arguments

`input_data`	A data.frame of phoshopeptide data. Must contain 4 columns and the following format must be adhered to. Column 1 - Annotation, Column 2 - centered peptide sequence, Column 3 - Fold Change [-ve to +ve], Column 4 - p-value [0-1]. This must be the same dataframe used in scoreSequences()
`pwm_in`	List of PWMs created using buildPWM()
`pwm_scores`	List of PWM-substrate scores created using scoreSequences()
`pseudo_count`	Pseudo-count acts at two levels. 1) It adds a small number to the counts to avoid zero divisions, which also 2) avoids log-zero transformations. Note that this means that pos, neg and all values in the output table include the addition of the pseudo-count. Default: "1"
`p_cut_pwm`	Significance level for determining a significant kinase-substrate enrichment. Default: "0.05"
`p_cut_fc`	Significance level for determining a significant level of Fold-change in the phosphoproteomics data. Default: "0.05"
`permutations`	Number of permutations to perform. This will shuffle the kinase-subtrate edges of the network n times. To not perform permutations and only generate the scores, set permutations=1 or permutations=FALSE. Default: "1000"
`return_network`	Option to return an interaction network for visualising in cystoscape. Default = FALSE
`verbose`	Turn verbosity on/off. To turn on, verbose=TRUE. Options are: "TRUE, FALSE". Default=FALSE

Value

A data.table of swing scores

Examples

```
## import data
data(example_phosphoproteome)
data(phosphositeplus_human)

## clean up the annotations
## sample 100 data points for demonstration
sample_data <- head(example_phosphoproteome, 100)
annotated_data <- cleanAnnotation(input_data = sample_data)

## build the PWM models:
set.seed(1234)
sample_pwm <- phosphositeplus_human[sample(nrow(phosphositeplus_human),
1000),]
pwms <- buildPWM(sample_pwm)

## score the PWM - substrate matches
## Using a "random" background, to calculate the p-value of the matches
## Using n = 100 for demonstration
## set.seed for reproducibility
set.seed(1234)
substrate_scores <- scoreSequences(input_data = annotated_data,
                                   pwm_in = pwms,
                                   background = "random",
                                   n = 100)

## Use substrate_scores and annotated_data data to predict kinase activity.
## This will permute the network node and edges 10 times for demonstration.
## set.seed for reproducibility
set.seed(1234)
swing_output <- swing(input_data = annotated_data,
                      pwm_in = pwms,
                      pwm_scores = substrate_scores,
                      permutations = 10)
```

`viewPWM` *View motif*

Description

View information content for each position of the PWM. Information content is modelled using Shannon's Entropy Model. The maximum information content is therefore log2(n), where n is the number of amino acids. Colors of Amino Acids are in accordance with the Lesk scheme.

Usage

```
viewPWM(pwm_in = NULL, which_pwm = NULL, fontsize = 10,
  view_pwm = FALSE, pseudo = 0.01, convert_PWM = FALSE,
  color_scheme = "shapely", correction_factor = NULL)
```

Arguments

pwm_in	View a PWM provided using the buildPWM. Default = NULL
which_pwm	If pwms are input (outputs of buildPWM), a kinase name must match a name in pwms$kinase$kinase list of names. Default = NULL
fontsize	Font size to use on x and y axis. Default = 10
view_pwm	View the PWM. Default = FALSE
pseudo	Small amount added to the PWM model, where zero's exist, to avoid log zero. Default = 0.01
convert_PWM	pwm_in is a matrix of counts at position. TRUE will convert this matrix to a PWM. Default = FALSE
color_scheme	Which color scheme to use for Amino Acid Groups. Options are "lesk" or "shapely". Default = "shapely"
correction_factor	Number of sequences used to infer the PWM. This can be used where a small number of sequences were used to build the model and included as E_n in the Shannon's Entropy Model. Default = NULL

Value

Visualisation of a motif, scaled on bits and two tables. 1) pwm: corresponding to the PWM from pwm and 2) pwm_bits: corresponding to the conversion to bits.

Examples

```
## Build PWM models from phosphositeplus data with default of minimum
## of 10 substrate sequences for building a PWM model.
data(phosphositeplus_human)
##randomly sample 1000 substrates for demonstration.
set.seed(1)
sample_pwm <- phosphositeplus_human[sample(nrow(phosphositeplus_human),
1000),]
pwms <- buildPWM(sample_pwm)

## Data frame of models built and number of sequences used to build each
## PWM model:
head(pwms$kinase)
## Will not visualise the motif
CAMK2A_motif <- viewPWM(pwm_in = pwms,
                        which_pwm = "CAMK2A",
                        view_pwm = FALSE)
# Use view_pwm = TRUE to view the motif
```

Index

∗ **datasets**
example_phosphoproteome, 5
phosphositeplus_human, 6

buildPWM, 2

cleanAnnotation, 3

example_phosphoproteome, 5

KinSwingR, 5
KinSwingR-package *(*KinSwingR*)*, 5

phosphositeplus_human, 6

scoreSequences, 6
swing, 8

viewPWM, 9