

Package: pKSEA (via r-universe)

October 18, 2024

Type Package

Title Prediction-Based Kinase-Substrate Enrichment Analysis

Version 0.0.1

Description A tool for inferring kinase activity changes from phosphoproteomics data. 'pKSEA' uses kinase-substrate prediction scores to weight observed changes in phosphopeptide abundance to calculate a phosphopeptide-level contribution score, then sums up these contribution scores by kinase to obtain a phosphoproteome-level kinase activity change score (KAC score). 'pKSEA' then assesses the significance of changes in predicted substrate abundances for each kinase using permutation testing. This results in a permutation score (pKSEA significance score) reflecting the likelihood of a similarly high or low KAC from random chance, which can then be interpreted in an analogous manner to an empirically calculated p-value. 'pKSEA' contains default databases of kinase-substrate predictions from 'NetworKIN' (NetworKINPred_db) <<http://networkin.info>> Horn, et. al (2014) <[doi:10.1038/nmeth.2968](https://doi.org/10.1038/nmeth.2968)> and of known kinase-substrate links from 'PhosphoSitePlus' (KSEAdb) <<https://www.phosphosite.org/>> Hornbeck PV, et. al (2015) <[doi:10.1093/nar/gku1267](https://doi.org/10.1093/nar/gku1267)>.

Depends R (>= 3.3.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Maintainer Peter Liao <p1121@case.edu>

Repository <https://pll21.r-universe.dev>

RemoteUrl <https://github.com/pll21/pksea>

RemoteRef HEAD

RemoteSha 7f871fb6e79a81fcd536194f17987e2faed49df1

Contents

batchrun	2
compare	3
get_matched_data	4
KSEAdb	5
NetworKINPred_db	5
run_on_matched	6
Index	7

batchrun	<i>Running pKSEA::compare() on multiple files</i>
----------	---

Description

For running compare() on multiple CSV data files in the same directory and for writing results to a folder in the designated data directory. Can receive various arguments to be passed on to downstream functions. Writes to tempdir() unless outputpath variable is specified by user (argument passed on to [results_write](#)).

Usage

```
batchrun(summaryfiledir, commonfilestring = ".csv",
predictionDB, results_folder = NULL, ...)
```

Arguments

summaryfiledir Directory containing summary statistic CSV files. Required data file columns: GN = gene name identifier that will be matched with prediction database, Peptide = unique peptide identifier (for example, sequence with modifications), Phosphosites = comma-separated phosphorylation sites (eg. "T102,S105"), pval= pairwise test p-value, fc= mean fold change, t= pairwise test t-statistic. pval and fc are used for results reporting only, all others are important for database searching, calculation, and permutation testing.

commonfilestring Common string identifying all files to be included in analysis

predictionDB Input database whose prediction scores will be used for calculations. Required columns: substrate_name= name of substrate corresponding to GN in summary_data, kinase_id = identifiers for kinase predictors, position= phosphorylated residue number, score = numeric score for strength of prediction.

results_folder if desired, a single output folder. Else each run performed on each file will have a separate output folder identified by run initiation time.

... parameters to be passed on to downstream functions, including(default): output-path (tempdir()) n_permutations (1000), seed (123), kseadb (NULL), kin_ens_table (NULL). See [run_on_matched](#), [compare](#) for details.

Examples

```
#point to data directory that contains summary .csv files
datapath <- system.file("extdata", package = "pKSEA")

#run batchrun function to analyze all files in that folder, with options
batchrun(datapath, predictionDB=NetworKINPred_db, kseadb = KSEAdb, n_permutations = 5)
```

compare	<i>Running analysis runs on known substrates, predicted substrates, and both.</i>
---------	---

Description

Performs up to three `run_on_matched()` runs on summary-prediction matcheddata from `get_matched_data()`, returning permutation significance score results. If a KSEA database is provided for filtering and comparison, one full analysis will be performed on all phosphosites, one on data with all known kinase substrates removed according to the provided KSEA database, and one on known kinase substrates only.

Usage

```
compare(matched_data, predictionDB, kseadb, ...)
```

Arguments

matched_data	File path to summary statistic phosphoproteomics CSV data file with an entry for each phosphopeptide. Required data file columns: GN = gene name identifier that will be matched with prediction database, Peptide = unique peptide identifier (for example, sequence with modifications), Phosphosites = comma-separated phosphorylation sites (eg. "T102,S105"), pval= pairwise test p-value, fc= mean fold change, t= pairwise test t-statistic. pval and fc are used for results reporting only, all others are important for database searching, calculation, and permutation testing.
predictionDB	Input database whose prediction scores will be used for calculations. Required columns: substrate_name= name of substrate corresponding to GN in summary_data, kinase_id = identifiers for kinase predictors, position= phosphorylated residue number, score = numeric score for strength of prediction.
kseadb	Optional KSEA database for filtering purposes. Containing substrate gene name "SUB_GENE" and phosphorylated residue "SUB_MOD_RSD" in standard form (ie. T302).
...	optional parameters to be passed on to downstream functions, including (default): n_permutations (1000), seed (123), kin_ens_table (NULL). See run_on_matched for details.

Examples

```
#Read in example summary statistics dataset from csv
summarydata_ex <- read.csv(system.file("extdata", "example_data1.csv", package="pKSEA"))

#Get matched data using predictions from NetworKIN
matched_data_ex <- get_matched_data(summarydata_ex, NetworKINPred_db)

#Perform comparative analysis using provided KSEadb as filter
## Not run:
compare_results_ex <- compare(matched_data_ex, kseadb = KSEadb, n_permutations = 10)

## End(Not run)
```

get_matched_data *Filtering data to matched predictions*

Description

This function reformats summary statistic phosphoproteomic data to single observations for each phosphorylation site, duplicating other fields for multiple sites on the same peptide. Next, it attempts to find predictions for each phosphorylation site in the provided database. It returns observations (phosphorylation sites) for which a prediction is detected in the database, matching based on HUGO gene name and phosphorylated residue.

Usage

```
get_matched_data(datafull, predictionDB)
```

Arguments

datafull	Statistical summary data with an entry for each phosphopeptide. Required columns: GN = gene name identifier that will be matched with prediction database, Peptide = unique peptide identifier (for example, sequence with modifications), Phosphosites = comma-separated phosphorylation sites (eg. "T102,S105"), pval= pairwise test p-value, fc= mean fold change, t= pairwise test t-statistic. pval and fc are used for results reporting only, all others are important for database searching, calculation, and permutation testing.
predictionDB	Input database whose prediction scores will be used for calculations. Required columns: substrate_name= name of substrate corresponding to GN in datafull, kinase_id = identifiers for kinase predictors, position= phosphorylated residue number, score = numeric score for strength of prediction.

Examples

```
#Read in example summary statistics dataset from csv
summarydata_ex <- read.csv(system.file("extdata", "example_data1.csv", package="pKSEA"))

#Get matched data using predictions from NetworKIN
matched_data_ex <- get_matched_data(summarydata_ex, NetworKINPred_db)
```

KSEAdb

KSEAdb

Description

A data table containing all known kinase-substrate links known in PhosphoSitePlus.

Usage

KSEAdb

Format

An object of class `data.frame` with 240749 rows and 6 columns.

Source

<https://www.phosphosite.org/staticDownloads.action>

References

Hornbeck PV, Zhang B, Murray B, Kornhauser JM, Latham V, Skrzypek E PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. *Nucleic Acids Res.* 2015 43:D512-20.

NetworkKINPred_db

NetworkKINPred_db

Description

A data table containing all precalculated NetworkKIN predictions performed on known ensemble sequences.

Usage

NetworkKINPred_db

Format

An object of class `data.frame` with 450418 rows and 4 columns.

Source

<http://networkkin.info/download.shtml>

References

Horn et al., KinomeXplorer: an integrated platform for kinome biology studies. *Nature Methods* 2014 Jun;11(6):603–4.

run_on_matched	<i>Runs pKSEA analysis on a dataset result from get_matched_data.</i>
----------------	---

Description

Calculates score contributions from summary statistics (tscore) and prediction scores, and sums contribution scores by kinase to calculate raw kinase activity change scores (KAC scores). Performs permutation test on summary statistic data to assess significance of kinase activity change scores, and reports significance as a percentile score (pKSEA significance score).

Usage

```
run_on_matched(matched_data, n_permutations = 1000, seed = 123,  
  kin_ens_table = NULL)
```

Arguments

matched_data	data after filtering against predictions (results from get_matched_data())
n_permutations	number of mutations to perform (default 1000)
seed	seed used for permutation testing
kin_ens_table	optional table for inclusion of matched ensembl ids for kinases, with columns: ens = ensembl id, kinases = kinase_id as otherwise used

Examples

```
#Read in example summary statistics dataset from csv  
summarydata_ex <- read.csv(system.file("extdata", "example_data1.csv", package="pKSEA"))  
  
#Get matched data using predictions from NetworKIN  
matched_data_ex <- get_matched_data(summarydata_ex, NetworKINPred_db)  
  
#Perform single run of pKSEA analysis  
single_run_results_ex <- run_on_matched(matched_data_ex, n_permutations = 10)
```

Index

* datasets

KSEAdb, [5](#)

NetworKINPred_db, [5](#)

batchrun, [2](#)

compare, [2](#), [3](#)

get_matched_data, [4](#)

KSEAdb, [5](#)

NetworKINPred_db, [5](#)

results_write, [2](#)

run_on_matched, [2](#), [3](#), [6](#)