

Package ‘MSstatsBioNet’

April 9, 2025

Type Package

Title Network Analysis for MS-based Proteomics Experiments

Version 0.99.9

Description A set of tools for network analysis using mass spectrometry-based proteomics data and network databases. The package takes as input the output of MSstats differential abundance analysis and provides functions to perform enrichment analysis and visualization in the context of prior knowledge from past literature. Notably, this package integrates with INDRA, which is a database of biological networks extracted from the literature using text mining techniques.

License file LICENSE

Depends R (>= 4.4.0), MSstats

Imports RCy3, httr, jsonlite, r2r, tidyr

Suggests data.table, BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0), mockery, MSstatsConvert

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Proteomics, Software, QualityControl, NetworkEnrichment, Network

Encoding UTF-8

URL <http://msstats.org>, <https://vitek-lab.github.io/MSstatsBioNet/>

BugReports <https://groups.google.com/forum/#!forum/msstats>

Config/testthat/edition 3

RoxygenNote 7.3.2

git_url <https://git.bioconductor.org/packages/MSstatsBioNet>

git_branch devel

git_last_commit 3dbd252

git_last_commit_date 2025-02-06

Repository Bioconductor 3.21

Date/Publication 2025-04-08

Author Anthony Wu [aut, cre],

Olga Vitek [aut] (ORCID: <<https://orcid.org/0000-0003-1728-1104>>)

Maintainer Anthony Wu <wu.anthon@northeastern.edu>

Contents

<code>.populateHgncIdsInDataFrame</code>	2
<code>.populateHgncNamesInDataFrame</code>	3
<code>.populateKinaseInfoInDataFrame</code>	3
<code>.populatePhosphataseInfoInDataFrame</code>	4
<code>.populateTranscriptionFactorInfoInDataFrame</code>	4
<code>.populateUniprotIdsInDataFrame</code>	5
<code>.validateAnnotateProteinInfoFromIndraInput</code>	5
<code>annotateProteinInfoFromIndra</code>	6
<code>getSubnetworkFromIndra</code>	7
<code>visualizeNetworks</code>	8
Index	10

`.populateHgncIdsInDataFrame`

Populate HGNC IDs in Data Frame

Description

This function populates the HGNC IDs in the data frame based on the Uniprot IDs.

Usage

```
.populateHgncIdsInDataFrame(df)
```

Arguments

`df` A data frame containing protein information.

Value

A data frame with populated HGNC IDs.

`.populateHgncNamesInDataFrame`

Populate HGNC Names in Data Frame

Description

This function populates the HGNC names in the data frame based on the HGNC IDs.

Usage

```
.populateHgncNamesInDataFrame(df)
```

Arguments

`df` A data frame containing protein information.

Value

A data frame with populated HGNC names.

`.populateKinaseInfoInDataFrame`

Populate Kinase Info in Data Frame

Description

This function populates the kinase information in the data frame based on the HGNC names.

Usage

```
.populateKinaseInfoInDataFrame(df)
```

Arguments

`df` A data frame containing protein information.

Value

A data frame with populated kinase information.

```
.populatePhosphataseInfoInDataFrame
```

Populate Phosphatase Info in Data Frame

Description

This function populates the phosphatase information in the data frame based on the HGNC names.

Usage

```
.populatePhosphataseInfoInDataFrame(df)
```

Arguments

`df` A data frame containing protein information.

Value

A data frame with populated phosphatase information.

```
.populateTranscriptionFactorInfoInDataFrame
```

Populate Transcription Factor Info in Data Frame

Description

This function populates the transcription factor information in the data frame based on the HGNC names.

Usage

```
.populateTranscriptionFactorInfoInDataFrame(df)
```

Arguments

`df` A data frame containing protein information.

Value

A data frame with populated transcription factor information.

`.populateUniprotIdsInDataFrame`
Populate Uniprot IDs in Data Frame

Description

This function populates the Uniprot IDs in the data frame based on the protein ID type.

Usage

```
.populateUniprotIdsInDataFrame(df, proteinIdType)
```

Arguments

`df` A data frame containing protein information.
`proteinIdType` A character string specifying the type of protein ID. It can be either "Uniprot" or "Uniprot_Mnemonic".

Value

A data frame with populated Uniprot IDs.

`.validateAnnotateProteinInfoFromIndraInput`
Validate Annotate Protein Info Input

Description

This function validates the input data frame for the `annotateProteinInfoFromIndra` function.

Usage

```
.validateAnnotateProteinInfoFromIndraInput(df)
```

Arguments

`df` A data frame containing protein information.

Value

None. Throws an error if validation fails.

`annotateProteinInfoFromIndra`*Annotate Protein Information from Indra*

Description

This function annotates a data frame with protein information from Indra.

Usage

```
annotateProteinInfoFromIndra(df, proteinIdType)
```

Arguments

df output of `groupComparison` function's `comparisonResult` table, which contains a list of proteins and their corresponding p-values, logFCs, along with additional HGNC ID and HGNC name columns

proteinIdType A character string specifying the type of protein ID. It can be either "Uniprot" or "Uniprot_Mnemonic".

Value

A data frame with the following columns:

Protein Character. The original protein identifier.

UniprotID Character. The Uniprot ID of the protein.

HgncID Character. The HGNC ID of the protein.

HgncName Character. The HGNC name of the protein.

IsTranscriptionFactor Logical. Indicates if the protein is a transcription factor.

IsKinase Logical. Indicates if the protein is a kinase.

IsPhosphatase Logical. Indicates if the protein is a phosphatase.

Examples

```
df <- data.frame(Protein = c("CLH1_HUMAN"))
annotated_df <- annotateProteinInfoFromIndra(df, "Uniprot_Mnemonic")
head(annotated_df)
```

`getSubnetworkFromIndra`*Get subnetwork from INDRA database*

Description

Using differential abundance results from MSstats, this function retrieves a subnetwork of protein interactions from INDRA database.

Usage

```
getSubnetworkFromIndra(  
  input,  
  protein_level_data = NULL,  
  pvalueCutoff = NULL,  
  statement_types = c("IncreaseAmount", "DecreaseAmount"),  
  paper_count_cutoff = 1,  
  evidence_count_cutoff = 1,  
  correlation_cutoff = 0.3  
)
```

Arguments

<code>input</code>	output of groupComparison function's comparisonResult table, which contains a list of proteins and their corresponding p-values, logFCs, along with additional HGNC ID and HGNC name columns
<code>protein_level_data</code>	output of the dataProcess function's ProteinLevelData table, which contains a list of proteins and their corresponding abundances. Used for annotating correlation information and applying correlation cutoffs.
<code>pvalueCutoff</code>	p-value cutoff for filtering. Default is NULL, i.e. no filtering
<code>statement_types</code>	list of interaction types to filter on. Equivalent to statement type in INDRA. Default is c("IncreaseAmount", "DecreaseAmount").
<code>paper_count_cutoff</code>	number of papers to filter on. Default is 1.
<code>evidence_count_cutoff</code>	number of evidence to filter on for each paper. E.g. A paper may have 5 sentences describing the same interaction vs 1 sentence. Default is 1.
<code>correlation_cutoff</code>	if <code>protein_level_abundance</code> is not NULL, apply a cutoff for edges with correlation less than a specified cutoff. Default is 0.3

Value

list of 2 data.frames, nodes and edges

Examples

```
input <- data.table::fread(system.file(
  "extdata/groupComparisonModel.csv",
  package = "MSstatsBioNet"
))
subnetwork <- getSubnetworkFromIndra(input)
head(subnetwork$nodes)
head(subnetwork$edges)
```

<code>visualizeNetworks</code>	<i>Create visualization of network</i>
--------------------------------	--

Description

Use results from INDRA to generate a visualization of the a network on Cytoscape Desktop. Note that the Cytoscape Desktop app must be open for this function to work.

Usage

```
visualizeNetworks(
  nodes,
  edges,
  pvalueCutoff = 0.05,
  logfcCutoff = 0.5,
  node_label_column = "id",
  main_targets = c()
)
```

Arguments

<code>nodes</code>	dataframe of nodes consisting of columns <code>id</code> (character), <code>pvalue</code> (number), <code>logFC</code> (number)
<code>edges</code>	dataframe of edges consisting of columns <code>source</code> (character), <code>target</code> (character), <code>interaction</code> (character), <code>evidenceCount</code> (number), <code>evidenceLink</code> (character)
<code>pvalueCutoff</code>	p-value cutoff for coloring significant proteins. Default is 0.05
<code>logfcCutoff</code>	log fold change cutoff for coloring significant proteins. Default is 0.5
<code>node_label_column</code>	The column of the nodes dataframe to use as the node label. Default is "id". "hgncName" can be used for gene name.
<code>main_targets</code>	character vector of main targets to stand-out with a different node shape. Default is an empty vector <code>c()</code> . IDs of main targets should match the column used by the <code>node_label_column</code> parameter.

Value

cytoscape visualization of subnetwork

Examples

```
input <- data.table::fread(system.file(
  "extdata/groupComparisonModel.csv",
  package = "MSstatsBioNet"
))
subnetwork <- getSubnetworkFromIndra(input)
visualizeNetworks(subnetwork$nodes, subnetwork$edges)
```

Index

`.populateHgncIdsInDataFrame`, [2](#)
`.populateHgncNamesInDataFrame`, [3](#)
`.populateKinaseInfoInDataFrame`, [3](#)
`.populatePhosphataseInfoInDataFrame`, [4](#)
`.populateTranscriptionFactorInfoInDataFrame`,
[4](#)
`.populateUniprotIdsInDataFrame`, [5](#)
`.validateAnnotateProteinInfoFromIndraInput`,
[5](#)

`annotateProteinInfoFromIndra`, [6](#)

`dataProcess`, [7](#)

`getSubnetworkFromIndra`, [7](#)
`groupComparison`, [6](#), [7](#)

`visualizeNetworks`, [8](#)