Package ‘MSstatsBioNet’

November 14, 2025

Type Package

Title Network Analysis for MS-based Proteomics Experiments

Version 1.3.1

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 7.3.3

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

git_branch devel

git_last_commit 5afd016

git_last_commit_date 2025-11-05

Repository Bioconductor 3.23

Date/Publication 2025-11-13

Author Anthony Wu [aut, cre] (ORCID: <https://orcid.org/0009-0001-7391-9902>),
Olga Vitek [aut] (ORCID: <https://orcid.org/0000-0003-1728-1104>)

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

Contents

`.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.

.populatePhophataseInfoInDataFrame

Populate Phosphatase Info in Data Frame

Description

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

Usage

```
.populatePhophataseInfoInDataFrame(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)
```

exportNetworkToHTML *Export network data with Cytoscape visualization*

Description

Convenience function that takes nodes and edges data directly and creates both the configuration and HTML export in one step.

Usage

```
exportNetworkToHTML(
  nodes,
  edges,
  filename = "network_visualization.html",
  displayLabelType = "id",
  ...
)
```

Arguments

`nodes`	Data frame with node information
`edges`	Data frame with edge information
`filename`	Output HTML filename
`displayLabelType`	Type of label to display ("id" or "hgncName")
`...`	Additional arguments passed to exportCytoscapeToHTML()

Value

Invisibly returns the file path of the created HTML file

generateCytoscapeConfig *Generate Cytoscape visualization configuration*

Description

This function creates a complete Cytoscape configuration object that can be used to render a network visualization. It's decoupled from any specific UI framework.

Usage

```
generateCytoscapeConfig(
  nodes,
  edges,
  display_label_type = "id",
  container_id = "network-cy",
  event_handlers = NULL,
  layout_options = NULL
)
```

Arguments

`nodes`	List of nodes from getSubnetworkFromIndra
`edges`	List of edges from getSubnetworkFromIndra
`display_label_type`	column of nodes table for displaying node names
`container_id`	ID of the HTML container element (default: 'network-cy')
`event_handlers`	Optional list of event handler configurations
`layout_options`	Optional list of layout configuration options

Value

List containing: - elements: Combined node and edge elements - style: Cytoscape style configuration - layout: Layout configuration - container_id: Container element ID - js_code: Complete JavaScript code (for backward compatibility)

`generateJavaScriptCode` *Generate JavaScript code from Cytoscape configuration*

Description

Internal function to convert configuration object to JavaScript code

Usage

```
generateJavaScriptCode(config)
```

Arguments

`config`	Configuration object from generateCytoscapeConfig()

Value

Character string containing JavaScript code 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 = NULL,
  paper_count_cutoff = 1,
  evidence_count_cutoff = 1,
  correlation_cutoff = 0.3,
  sources_filter = NULL,
  logfc_cutoff = NULL,
  force_include_other = NULL,
  filter_by_curation = FALSE,
  api_key = ""
)
```

Arguments

`input`	output of `groupComparison` function's comparisionResult table, which contains a list of proteins and their corresponding p-values, logFCs, along with additional HGNC ID and HGNC name columns
`protein_level_data`	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.
`pvalueCutoff`	p-value cutoff for filtering. Default is NULL, i.e. no filtering
`statement_types`	list of interaction types to filter on. Equivalent to statement type in INDRA. Default is NULL.
`paper_count_cutoff`	number of papers to filter on. Default is 1.
`evidence_count_cutoff`	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.
`correlation_cutoff`	if protein_level_abundance is not NULL, apply a cutoff for edges with correlation less than a specified cutoff. Default is 0.3

sources_filter filtering only on specific sources. Default is no filter, i.e. NULL. Otherwise, should be a list, e.g. c('reach', 'medscan').

logfc_cutoff absolute log fold change cutoff for filtering proteins. Only proteins with |logFC| greater than this value will be retained. Default is NULL, i.e. no logFC filtering.

force_include_other
character vector of identifiers to include in the network, regardless if those ids are in the input data. Should be formatted as "namespace:identifier", e.g. "HGNC:1234" or "CHEBI:4911".

filter_by_curation
logical, whether to filter out statements that have been curated as incorrect in INDRA. Default is FALSE.

api_key string of INDRA API key for accessing curated statements.

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)
```

---

previewNetworkInBrowser

Preview network in browser

Description

Creates a temporary HTML file and opens it in the default web browser

Usage

```
previewNetworkInBrowser(nodes, edges, displayLabelType = "id", ...)
```

Arguments

nodes Data frame with node information

edges Data frame with edge information

displayLabelType
Type of label to display ("id" or "hgncName")

... Additional arguments passed to exportCytoscapeToHTML()

visualizeNetworks *Create visualization of network*

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

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