Package: MetaboSignal (via r-universe)

June 30, 2024

Type Package

Title MetaboSignal: a network-based approach to overlay and explore metabolic and signaling KEGG pathways

Version 1.35.0

Date 2023-12-16

Author Andrea Rodriguez-Martinez, Rafael Ayala, Joram M. Posma, Ana L. Neves, Maryam Anwar, Jeremy K. Nicholson, Marc-Emmanuel Dumas

Maintainer Andrea Rodriguez-Martinez

<andrea.rodriguez-martinez13@imperial.ac.uk>, Rafael Ayala

<rafaelayalahernandez@gmail.com>

Description MetaboSignal is an R package that allows merging, analyzing and customizing metabolic and signaling KEGG pathways. It is a network-based approach designed to explore the topological relationship between genes (signaling- or enzymatic-genes) and metabolites, representing a powerful tool to investigate the genetic landscape and regulatory networks of metabolic phenotypes.

License GPL-3

Depends R(>= 3.3)

Suggests RUnit, BiocGenerics, knitr, BiocStyle, rmarkdown

VignetteBuilder knitr

Imports KEGGgraph, hpar, igraph, RCurl, KEGGREST, EnsDb.Hsapiens.v75, stats, graphics, utils, org.Hs.eg.db, biomaRt, AnnotationDbi, MWASTools, mygene

biocViews GraphAndNetwork, GeneSignaling, GeneTarget, Network, Pathways, KEGG, Reactome, Software

NeedsCompilation no

LazyData true

Encoding UTF-8

Repository https://bioc.r-universe.dev

RemoteUrl https://github.com/bioc/MetaboSignal

RemoteRef HEAD

RemoteSha 4f711a3af7da0d0f5b53e50d207ed6d7e53169a7

Contents

`directionality_reactions`

List of KEGG reactions with incorrect/inconsistent directionality

---

Description

This matrix contains a set of KEGG reactions with incorrect/inconsistent directionality. The directionality of these reactions has been corrected based on published literature. This matrix can be updated or edited by the user if required.

Usage

```
directionality_reactions
```

Format

Matrix

Value

Matrix

---

hpaNormalTissue *Expression profiles for proteins in human tissues*

Description

This data frame contains tissue expression data of human proteins, based on the Human Protein Atlas project. This data frame was obtained from the hpar package, and it is used in MetaboSignal to filter signaling genes based on tissue expression.

Usage

```
data(hpaNormalTissue)
```

Format

Data.frame

Value

Data.frame

---

keggNet_example *KEGG network example*

Description

KEGG network generated using the metabolic and signaling pathways stored in kegg_pathways.

Usage

```
keggNet_example
```

Format

Matrix

Value

Matrix

---

kegg_pathways *Examples of metabolic and signaling human KEGG pathways*

Description

This matrix contains examples of metabolic and signaling human KEGG pathways. This matrix was generated with the function "MS_getPathIds( )".

Usage

```
kegg_pathways
```

Format

Matrix

Value

Matrix

---

mergedNet_example *Network containing KEGG, OmniPath and TRRUST interactions*

Description

Network generated by merging "keggNet_example" and "ppiNet_example" in the vignette.

Usage

```
mergedNet_example
```

Format

Matrix

Value

Matrix

MetaboSignal_table *Example of MetaboSignal network-table*

Description

This network-table was generated using two metabo_paths ("rno00010", "rno00562") and two signaling_paths ("rno04910", "rno04151"). Notice that due to KEGG udpates, this network might be different to the one generated when running the vignette.

Usage

```
data(MetaboSignal_table)
```

Format

Matrix

Value

Matrix

MS2_mergeNetworks *Merge networks*

Description

This function allows merging two network-tables of interest.

Usage

```
MS2_mergeNetworks(network_table1, network_table2)
```

Arguments

network_table1	three-column matrix where each row represents an edge between two nodes. See functions "MS_keggNetwork()" and "MS2_ppiNetwork()".
network_table2	three-column matrix where each row represents an edge between two nodes. See functions "MS_keggNetwork()" and "MS2_ppiNetwork()".

Value

A three-column matrix where each row represents an edge between two nodes.

Examples

```
data(keggNet_example)
data(ppiNet_example)

# Fast example using subsets
global_network1 <- MS2_mergeNetworks(keggNet_example[1:10, ],
                                     ppiNet_example[1:10, ])

 # Example using full datasets
global_network2 <- MS2_mergeNetworks(keggNet_example, ppiNet_example)
```

`MS2_ppiNetwork` *Build signaling-transduction network*

Description

This function generates a directed regulatory network by merging interactions reported in two literature-curated resources: OmniPath and TRRUST. The network is formalized as a three-column matrix, where each row represents an edge connecting two nodes (from source to target). The third column indicates the type of interaction, as well as the source of the interaction (OmniPath = "o_", TRRUST = "t_"). Nodes represent gene Entrez IDs.

Usage

```
MS2_ppiNetwork(datasets = "all")
```

Arguments

`datasets`	character vector indicating the datasets that will be used to build the network ("all", "omnipath","trrust"). It is also possible to select databases included within OmniPath (e.g. datasets = c("biogrid", "string"))

Value

A three-column matrix where each row represents an edge between two nodes.

Note

The dataset "regulatory_interactions" contains details regarding primary database reference(s) as well as literature reference(s) of each of the regulatory interactions. The users are fully responsible for respecting the terms of the these databases and for citing them when required. The users can edit/update this dataset if needed.

References

Ceol, A., et al. (2007). DOMINO: a database of domain-peptide interactions. Nucleic Acid Research, 35, D557-60.

Cui, Q., et al. (2007). A map of human cancer signaling. Molecular Systems Biology, 3:152.

Diella, F., et al. (2004). Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins. BMC Bioinformatics, 22, 5:79.

Dinkel, H., et al. (2012). ELM–the database of eukaryotic linear motifs. Nucleic Acid Research, 40, D242-51.

Han, H., et al. (2015). TRRUST: a reference database of human transcriptional regulatory interactions. Scientific Reports, 15, 11432.

Hornbeck, P.V., et al. (2012). PhosphoSitePlus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. Nucleic Acid Research, 40, D261-70.

Korcsmaros, T., et al. (2010). Uniformly curated signaling pathways reveal tissue-specific cross-talks and support drug target discovery. Bioinformatics, 26, 2042:2050.

Lynn, D.J., et al. (2008). InnateDB: facilitating systems-level analyses of the mammalian innate immune response. Molecular Systems Biology, 4, 218.

Orchard, S., et al. (2014). The MIntAct project–IntAct as a common curation platform for 11 molecular interaction databases. Nucleic Acid Research, 242, D358-63.

Pagel, P., et al. (2005). The MIPS mammalian protein-protein interaction database. Bioinformatics, 21, 832-834.

Papp, D., et al. (2012). The NRF2-related interactome and regulome contain multifunctional proteins and fine-tuned autoregulatory loops. FEBS Letters, 586, 1795-802.

Pawson, A.J., et al. (2014). The IUPHAR/BPS Guide to PHARMACOLOGY: an expert-driven knowledgebase of drug targets and their ligands. Nucleic Acids Research, 42, D1098-106.

Peri, S., et al. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research, 13, 2363-2371.

Turei, D., et al. (2015). Autophagy Regulatory Network - a systems-level bioinformatics resource for studying the mechanism and regulation of autophagy. Autophagy, 11, 155-165.

Turei, D., et al. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nature methods, 13, 966-967.

Sarkar, D., et al. (2015). LMPID: a manually curated database of linear motifs mediating protein-protein interactions. Database(Oxford), pii: bav014.

Shin, Y.C., et al. (2011). TRIP Database: a manually curated database of protein-protein interactions for mammalian TRP channels. Nucleic Acids Research, 39, D356-61.

Snel, B., et al. (2000). STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. Nucleic Acids Research, 28, 3442-3444.

Xenarios, I., et al. (2002). DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research, 30, 303-305.

Examples

```
# Build regulatory network using the OmniPath dataset only
omnipath_net <- MS2_ppiNetwork(datasets = "omnipath")

# Build regulatory network using the TRRUST dataset only
trrust_net <- MS2_ppiNetwork(datasets = "trrust")

# Build regulatory network using interactions from STRING and BioGRID
biogridstring_net <- MS2_ppiNetwork(datasets = c("biogrid", "string"))
```

`MS_changeNames` *Transform KEGG IDs into common names*

Description

This function allows transforming KEGG IDs of genes or compounds into their corresponding common names (for compounds) or symbols (for genes).

Usage

```
MS_changeNames(nodes, organism_code)
```

Arguments

`nodes`	character vector or matrix containing the KEEG IDs of either metabolites, genes (organism-specific or orthology), or reactions. It also converts human Entrez gene IDs into symbols.
`organism_code`	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See the function "MS_keggFinder( )". This argument is ignored when nodes are metabolites.

Value

A character string or a matrix containing the common metabolite names or gene symbols corresponding to the input KEGG IDs. Reaction IDs remain unchanged.

References

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
MS_changeNames(c("rno:84482", "K01084", "cpd:C00267"), "rno")
MS_changeNames("K01082", organism_code = "rno")
```

MS_convertGene *Transform Entrez IDs or gene symbols into KEGG IDs*

Description

This function allows transforming Entrez gene IDs or official gene symbols into KEGG IDs (orthology IDs or organism-specific gene IDs). The transformed KEGG IDs can be stored and used as source genes in the functions "MS_distances( )" or "MS_shortestpathsNetwork( )".

Usage

```
MS_convertGene(genes, organism_code, organism_name, output = "vector",
               orthology = TRUE)
```

Arguments

genes	character vector containing the Entrez IDs or official symbols of the genes of interest. All genes need to be in the same ID format (i.e. Entrez or symbols). It is preferable to use Entrez IDs rather than gene symbols, since some gene symbols are not unique.
organism_code	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See the function "MS_keggFinder( )".
organism_name	character vector containing the common name of the organism of interest (e.g. "rat", "mouse", "human", "zebrafish") or taxonomy id. For more details, check: http://docs.mygene.info/en/latest/doc/data.html#species. This argument is only required when gene symbols are used.
output	character constant indicating whether the function will return a vector containing mapped and transformed KEGG IDs (output = "vector"), or a matrix containing both mapped Entrez IDs or gene symbols and their corresponding KEGG IDs (output = "matrix").
orthology	logical scalar indicating whether the gene IDs will be transformed into orthology IDs or into organism-specific gene IDs.

Value

A character vector containing mapped and transformed KEGG IDs or a matrix containing both mapped Entrez IDs or gene symbols and their corresponding KEGG IDs.

References

Carlson, M. org.Hs.eg.db: Genome wide annotation for Human. R package version >= 3.2.3.

Mark, A., et al. (2014) mygene: Access MyGene.Info_ services. R package version >= 1.6.0.

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
# Transform gene symbol Hoga1 (293949) into rat-specific KEGG ID

MS_convertGene(genes = "Hoga1", organism_code = "rno", organism_name = "rat",
                  orthology = FALSE)

MS_convertGene(genes = "Hoga1", "rno", "rat", output = "matrix", orthology = FALSE)

# Transform entrez ID 293949 into orthology KEGG ID

MS_convertGene(genes = "293949", organism_code = "rno", output = "matrix")
```

`MS_distances` *Calculate gene-metabolite distance matrix*

Description

This function generates a distance matrix containing the length of all shortest paths from a set of genes (or reactions) to a set of metabolites. The shortest path length between two nodes is defined as the minimum number of edges between these two nodes.

Usage

```
MS_distances(network_table, organism_code, mode = "SP", source_genes = "all",
             target_metabolites = "all", names = FALSE)
```

Arguments

`network_table`	three-column matrix where each row represents an edge between two nodes. See function "MS_keggNetwork( )".
`organism_code`	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See the function "MS_keggFinder( )".
`mode`	character constant indicating whether a directed or an undirected network will be considered. "all" indicates that all the edges of the network will be considered as undirected. "out" indicates that all the edges of the network will be considered as directed. "SP" indicates that all network will be considered as directed except the edges linked to target metabolite, which will be considered as undirected. The difference between the "out" and the "SP" options, is that the latter aids reaching target metabolites that are substrates of irreversible reactions.
`source_genes`	character vector containing the genes from which the shortest paths will be calculated. Remember that Entrez IDs or gene symbols can be transformed into KEGG IDs using the function "MS_convertGene( )". By default, source_genes = "all", indicating that all the genes of the network will be used.

target_metabolites
: character vector containing the KEGG IDs of the metabolites to which the shortest paths will be calculated. Compound KEGG IDs can be obtained using the function "MS_keggFinder( )". By default, target_metabolites = "all", indicating that all the metabolites of the network will be used.

names
: logical scalar indicating whether metabolite or gene KEGG IDs will be transformed into common metabolite names or gene symbols. Reaction IDs remain unchanged.

Value

A matrix containing the shortest path length from the genes or reactions (in the rows) to the metabolites (in the columns). For unreacheable metabolites Inf is included.

References

Csardi, G. & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Examples

```
data(MetaboSignal_table)

# Distances from Ship2 (65038) and Ppp2r5b (309179) to D-glucose ("cpd:C00031")

MS_convertGene(genes = c("65038","309179"), "rno", "rat", output = "matrix")

distances_targets <- MS_distances(MetaboSignal_table, organism_code = "rno",
                                  source_genes = c("K15909", "K11584"),
                                  target_metabolites = "cpd:C00031",
                                  names = TRUE)

# Distances from all genes to all metabolites of the network

distances_all <- MS_distances(MetaboSignal_table, organism_code = "rno")
```

MS_exportCytoscape *Export network in cytoscape format*

Description

This function generates a network file and two attribute files ("NodesType.txt", "TargetNodes.txt"), which can be imported into Cytoscape to visualize the network. The first attribute file allows customizing the nodes of the network based on the molecular entity they represent: compound, reaction, metabolic-gene or signaling-gene. The second attribute file allows highlighting a set of nodes of interest.

Usage

```
MS_exportCytoscape(network_table, organism_code, names = TRUE,
                   targets = NULL, file_name = "MS")
```

Arguments

network_table	three-column matrix where each row represents and edge between two nodes. Nodes must be KEGG IDs, not common names. See function "MS_keggNetwork()". For human networks, Entrez gene IDs are also allowed.
organism_code	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See function "MS_keggFinder( )".
names	logical scalar indicating whether metabolite or gene KEGG IDs will be transformed into common metabolite names or gene symbols. Reaction IDs remain unchanged.
targets	optional character vector containing the IDs of the target nodes to be discriminated from the other nodes of the network.
file_name	character vector that allows customizing the name of the exported files.

Value

A data frame where each row represents an edge between two nodes (from source to target). The function also generates and exports a network file ("MS_Network.txt") and two attribute files ("MS_NodesType.txt", "MS_TargetNodes.txt"), which can be imported into Cytoscape to visualize the network.

References

Shannon P et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research, 13, 2498-2504.

Examples

```
data(MetaboSignal_table)
MS_exportCytoscape(MetaboSignal_table, organism_code = "rno", names = FALSE)
```

MS_findMappedNodes *Map gene IDs or metabolite IDs onto the network*

Description

This function can be used to find out if a set of genes or metabolites of interest can be mapped onto the network.

Usage

```
MS_findMappedNodes(nodes, network_table)
```

Arguments

nodes	character vector containing the IDs of the genes or the metabolites to be mapped onto the network. Remember that Entrez IDs or gene symbols can be transformed into KEGG IDs using the function "MS_convertGene( )".
network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".

Value

A list reporting which genes or metabolites can or cannot be mapped onto the network.

References

Carlson, M. org.Hs.eg.db: Genome wide annotation for Human.R package version >= 3.2.3.

Mark, A., et al.(2014) mygene: Access MyGene.Info_ services. R package version >= 1.6.0.

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
data(MetaboSignal_table)

# Map D-glucose ("cpd:C00031"), taurine ("cpd:C00245"), and aldh ("K00128") onto
# onto the network

MS_findMappedNodes(nodes = c("cpd:C00031","cpd:C00245", "K00128"), MetaboSignal_table)
```

MS_getPathIds *Get pathway identifiers of a given organism*

Description

This function retrieves the identifiers (IDs) of all metabolic and signaling KEGG pathways of a given organism. These pathway IDs can be used to build a MetaboSignal network with the function "MS_keggNetwork( )".

Usage

```
MS_getPathIds(organism_code)
```

Arguments

organism_code	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See the function "MS_keggFinder( )".

Value

This function returns a matrix, where each row contains the ID, description, category, and type (i.e. "metabolic" or "signaling") of each pathway. This matrix is also exported in a file named "organism-code_pathways.txt".

References

Tenenbaum, D. KEGGREST: Client-side REST access to KEGG. R package version >= 1.17.0.

Examples

```
rat_paths <- MS_getPathIds(organism_code = "rno")
human_paths <- MS_getPathIds(organism_code = "hsa")
```

`MS_keggFinder` *Get KEGG IDs for compounds, organisms or pathways*

Description

This function returns a list of entries corresponding to one of the following KEGG databases: "compound", "organism", "pathway". It can also find entries with matching query keywords in a given database.

Usage

```
MS_keggFinder(KEGG_database, match = NULL, organism_code)
```

Arguments

`KEGG_database`	character vector containing the name of the KEGG database of interest: "compound", "organism", "pathway".
`match`	character vector containing one or more elements (i.e. key words) to be matched as compound names.
`organism_code`	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". This argument is only required for KEGG_database = "pathway".

Value

By default, a matrix where each row contains the KEGG entries of the database of interest. When using the option "match" a list is returned, each list element containing information of matched entries.

Examples

```
MS_keggFinder(KEGG_database = "compound", match = "acetoacetic acid")

MS_keggFinder(KEGG_database = "organism", match = c("rattus","human"))

MS_keggFinder(KEGG_database = "pathway", match = c("glycol", "insulin signal", "akt"),
            organism_code = "rno")
```

MS_keggNetwork *Build MetaboSignal network-table*

Description

This function generates a directed network-table (i.e. three-column matrix), where each row represents an edge connecting two nodes (from source to target). Nodes represent different molecular entities: metabolic-genes (i.e. genes encoding enzymes that catalyze metabolic reactions), signaling-genes (e.g. kinases), reactions and compounds (metabolites, drugs or glycans). The third column of the matrix indicates the interaction type. Compound-gene (or gene-compound) interactions are designated as: "k_compound:reversible" or "kegg_compound:irreversible", depending on the direction of the interaction. Other types of interactions correspond to gene-gene interactions. When KEGG reports various types of interaction for the same gene pair, the "interaction_type" is collapsed using "/".

The network-table generated with this function can be customized based on several criteria. For instance, undesired nodes can be removed or replaced using the functions "MS_removeNode( )" or "MS_replaceNode( )" respectively. Also, the network can be filtered according to different topological parameters (e.g. node betweenness) using the function "MS_topologyFilter( )".

Usage

```
MS_keggNetwork(metabo_paths, signaling_paths, expand_genes = FALSE,
               convert_entrez = FALSE)
```

Arguments

metabo_paths	character vector containing the KEGG IDs of the metabolic pathways of interest (organism-specific). Pathway IDs take the form: "organism code + 5-digit number". For example, the ID of the rat "glycolysis/gluconeogenesis" pathway is "rno00010". See functions "MS_keggFinder( )" and "MS_getPathIds( )".
signaling_paths	character vector containing the KEGG IDs for the signaling pathways of interest (organism-specific). For example, the ID for the pathway "insulin signaling pathway" in the rat is "rno04910". See functions "MS_keggFinder( )" and "MS_getPathIds( )".
expand_genes	logical scalar indicating whether the gene nodes will represent orthology IDs (FALSE) or organism-specific gene IDs (TRUE).

convert_entrez logical scalar indicating whether the KEGG gene IDs will be transformed into Entrez IDs. This argument will be ignored if expand_genes = FALSE, or if the input paths are not human-specific.

Value

A three-column matrix where each row represents an edge between two nodes.

Note

Reaction directionality reported in KEGG has been cross-validated with published literature (Duarte et al., 2007).

References

Davidovic, L., et al. (2011). A metabolomic and systems biology perspective on the brain of the fragile X syndrome mouse model. Genome Research, 21, 2190-2202.

Duarte, N.C., et al. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proceedings of the National Academy of Sciences, 104, 1777-1782.

Posma, J.M., et al.(2014). MetaboNetworks, an interactive Matlab-based toolbox for creating, customizing and exploringsub-networks from KEGG. Bioinformatics, 30, 893-895.

Zhang, J.D. & Wiemann, S. (2009). KEGGgraph: a graph approach to KEGG PATHWAY in R and Bioconductor. Bioinformatics, 25, 1470-1471.

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
# MetaboSignal network-table with organism-specific gene nodes

MS_netIsoforms <- MS_keggNetwork(metabo_paths = c("rno00010", "rno00562"),
                                 signaling_paths = c("rno04910", "rno04151"),
                                 expand_genes = TRUE)

# MetaboSignal network-table with orthology gene nodes

MS_netK <- MS_keggNetwork(metabo_paths = c("rno00010", "rno00562"),
                          signaling_paths = c("rno04910", "rno04151"))

# MetaboSignal network-table with human Entrez gene IDs

MS_netEntrez <- MS_keggNetwork(metabo_paths = c("hsa00010", "hsa00562"),
                               signaling_paths = c("hsa04910", "hsa04151"),
                               expand_genes = TRUE, convert_entrez = TRUE)
```

MS_nodeBW *Get distribution of node betweeness*

Description

This function calculates the betweenness of each node of the network.

Usage

```
MS_nodeBW(network_table, mode = "all", normalized = TRUE)
```

Arguments

network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".
mode	character constant indicating whether a directed ("out") or undirected ("all") network will be considered.
normalized	logical scalar indicating whether to normalize the betweeness scores. If TRUE, normalized betweenness scores will be returned. If FALSE, raw betweenness scores will be returned.

Value

A numeric vector containing the betweenness of each node of the network. The function also produces and histogram showing the distribution of node betweenness.

References

Csardi, G. & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Examples

```
data(MetaboSignal_table)
MS_nodeBW(MetaboSignal_table)
```

MS_reactionNetwork *Build reaction-compound network*

Description

This function generates a directed reaction-compound network. The network is formalized as a three-column matrix, where each row represents an edge connecting two nodes (from source to target).

Usage

```
MS_reactionNetwork(metabo_paths)
```

Arguments

metabo_paths	character vector containing the KEGG IDs of the metabolic pathways of interest. See functions "MS_keggFinder( )" and "MS_getPathIds( )".

Value

A three-column matrix where each row represents an edge between two nodes.

Note

Reaction directionality reported in KEGG has been cross-validated with published literature (Duarte et al., 2007).

Examples

```
reaction_network <- MS_reactionNetwork(metabo_paths = c("rno00010", "rno00562"))
```

MS_removeDrugs *Remove edges containing drug nodes*

Description

This function allows removing edges containing drug ("dr:") nodes.

Usage

```
MS_removeDrugs(network_table)
```

Arguments

network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".

Value

A three-column matrix corresponding to the input network-table without the drug nodes.

Examples

```
data(MetaboSignal_table)

# Remove drug nodes if present

drugsRemoved <- MS_removeDrugs(MetaboSignal_table)
```

MS_removeNode *Remove undesired nodes from the network*

Description

This function allows removing undesired nodes from the network-table.

Usage

```
MS_removeNode(nodes, network_table)
```

Arguments

nodes	character vector containing the node IDs to be removed.
network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keegNetwork( )".

Value

A three-column matrix corresponding to the input network-table without the undesired nodes.

Examples

```
data(MetaboSignal_table)

# Remove glucose nodes

glucoseRemoved <- MS_removeNode(nodes = c("cpd:C00267", "cpd:C00221", "cpd:C00031"),
                                MetaboSignal_table)
```

MS_replaceNode *Replace nodes of the network*

Description

This function allows replacing node IDs of a network-table. It can be used to cluster the IDs of chemical isomers (e.g. alpha-D-glucose ("cpd:C00267"), D-glucose ("cpd:C00031"), and beta-D-glucose ("cpd:C00021")) into a single ID.

Usage

```
MS_replaceNode(node1, node2, network_table)
```

Arguments

node1	character vector containing the node IDs to be replaced.
node2	character vector containing the ID that will be used as a replacement.
network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".

Value

A three-column matrix corresponding to the input network-table with replaced nodes.

Examples

```
data(MetaboSignal_table)

# Cluster D-glucose isomers ("cpd:C00267","cpd:C00221","cpd:C00031")

glucoseClustered <- MS_replaceNode(node1 = c("cpd:C00267", "cpd:C00221"),
                                   node2 = "cpd:C00031", MetaboSignal_table)
```

MS_shortestPaths *Calculate shortest paths*

Description

This function calculates the shortest path(s) between any two reachable nodes of a network-table.

Usage

```
MS_shortestPaths(network_table, source_node, target_node, mode = "out",
                 type = "first")
```

Arguments

network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".
source_node	character vector containing the node from which the shortest paths will be calculated.
target_node	character vector containing the node to which the shortest path will be calculated.
mode	character constant indicating whether a directed or an undirected network will be considered. "all" indicates that all the edges of the network will be considered as undirected. "out" indicates that all the edges of the network will be considered as directed. "SP" indicates that all network will be considered as directed except the edges linked to target metabolite, which will be considered as undirected. The difference between the "out" and "SP" options, is that the latter aids reaching target metabolites that are substrate of irreversible reactions.

type — indicates whether all shortest paths or a single shortest path will be considered when there are several shortest paths between the source_node and the target_node. If type = "all", all shortest paths will be considered. If type = "first" a single path will be considered. If type = "bw" the path with the highest betweenness score will be considered. The betweenness score is calculated as the average betweenness of the gene nodes of the path. Using type = "bw" increases the time required to compute this function.

Value

A vector or a matrix where each row contains a shortest path from the source_node to the target_node. KEGG IDs can be transformed into common names using the function "MS_changeNames( )".

References

G. Csardi and T. Nepusz (2015). igraph package, The Comprehensive R Archive Network, v1.0.1.

Examples

```
data(MetaboSignal_table)

# Shortest path from HK ("K00844") to a-D-Glucose ("cpd:C00267")

path1 <- MS_shortestPaths(MetaboSignal_table, "K00844", "cpd:C00267", mode = "SP")
path2 <- MS_shortestPaths(MetaboSignal_table, "K00844", "cpd:C00267", mode = "out")

# Shortest paths from G6PC ("K01084") to pyruvate ("cpd:C00022")

path3 <- MS_shortestPaths(MetaboSignal_table, "K01084", "cpd:C00022", type = "all")
path4 <- MS_shortestPaths(MetaboSignal_table, "K01084", "cpd:C00022", type = "bw")
```

MS_shortestPathsNetwork

Build shortest-path subnetwork

Description

This function allows calculating the shortest paths from a set of source nodes to a set of target nodes, and representing them as a network. By default, the function exports a network file and two attribute files ("NodesType.txt", "TargetNodes.txt"), which can be imported into Cytoscape to visualize the network. The first attribute file allows customizing the nodes of the network based on the molecular entity they represent: signaling-gene, metabolic-gene, reaction or compound. The second attribute file allows highlighting the source and target nodes.

Usage

```
MS_shortestPathsNetwork(network_table, organism_code, source_nodes, target_nodes,
                    mode = "out", type = "first", distance_th = Inf, names = TRUE,
                         export_cytoscape = TRUE, file_name = "MS")
```

Arguments

network_table	three-column matrix where each row represents an edge between two nodes. See function "MS_keggNetwork( )".
organism_code	character vector containing the KEGG code for the organism of interest. For example the KEGG code for the rat is "rno". See the function "MS_keggFinder( )".
source_nodes	character vector containing the node IDs (typically genes) from which the shortest paths will be calculated. When using gene IDs make sure that they are consistent with the format of the network (i.e. organism-specific gene IDs or orthology IDs). Remember that Entrez IDs and gene symbols can be transformed into KEGG IDs with the function "MS_convertGene( )".
target_nodes	character vector containing the nodes IDs (typically compounds) to which the shortest paths will be calculated. Compound KEGG IDs can be obtained using the function "MS_keggFinder( )".
mode	character constant indicating whether a directed (mode = "out") or semi-directed (mode = "SP") network will be considered. "out" indicates that all the edges of the network will be considered as directed. "SP" indicates that all network will be considered as directed except the edges linked to target_node, which will be considered as undirected. The difference between the "out" and the "SP" options, is that the latter aids reaching target metabolites that are substrates of irreversible reactions.
type	character constant indicating whether all shortest paths or a single shortest path will be considered when there are several shortest paths between a source node and a target node. If type = "all", all shortest paths will be considered. If type = "first" a single path will be considered. If type = "bw" the path with the highest betweenness score will be considered. The betweenness score is calculated as the average betweenness of the gene nodes of the path. Note that using type = "bw" increases the time required to compute this function.
distance_th	establishes a shortest path length threshold. Only shortest paths with length below this threshold will be included in the network.
names	logical scalar indicating whether metabolite or gene KEGG IDs will be transformed into common metabolite names or gene symbols. Reaction IDs remain unchanged.
export_cytoscape	logical scalar indicating whether network and attribute Cytoscape files will be generated and exported.
file_name	character vector that allows customizing the name of the exported files.

Value

A matrix where each row represents an edge between two nodes. By default, the function also generates a network file ("MS_Network.txt") and two attribute files ("MS_NodesType.txt", "MS_TargetNodes.txt"), which can be imported into Cytoscape to visualize the network.

Note

The network-table generated with this function can be also visualized in R using the igraph package. The network-table can be transformed into an igraph object using the function "graph.data.frame( )" from igraph.

References

Csardi, G. & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Shannon, P., et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research, 13, 2498-2504.

Examples

```
data(MetaboSignal_table)

# Shortest paths from G6PC ("K01084") to pyruvate ("cpd:C00022") and
# to a-D-Glucose ("cpd:C00267")

subnet_first <- MS_shortestPathsNetwork(MetaboSignal_table, organism_code = "rno",
                                        source_nodes = "K01084",
                                        target_nodes = c("cpd:C00022", "cpd:C00267"),
                                        mode = "SP", type = "first")

subnet_all <- MS_shortestPathsNetwork(MetaboSignal_table, organism_code = "rno",
                                      source_nodes = "K01084",
                                      target_nodes = c("cpd:C00022", "cpd:C00267"),
                                      mode = "SP", type = "all")
```

MS_tissueFilter *Filter network based on tissue expression data*

Description

This function allows filtering a network based on tissue expression data from the Human Protein Atlas, by removing signaling genes that are not detected in the target tissue(s) (reliability = "approved" or "supported"). This function can be only used to filter human networks.

Usage

```
MS_tissueFilter(network_table, tissue, input_format = "kegg", expand_genes = FALSE)
```

Arguments

network_table	three-column matrix where each row represents an edge between two nodes. The gene nodes of this network must be human specific gene IDS (not orthologies). For this, use the function "MS_keggNetwork( )" with expand_genes = TRUE.
tissue	character vector indicating the tissue(s) of interest. Signaling genes (i.e. non-enzymatic genes) not detected in the target tissue(s) (reliability = "approved" or "supported") will be removed from the network. Check all possible tissues in the "hpaNormalTissue" dataset.
input_format	character vector indicating the gene format in the input network_table ("entrez" or "kegg").
expand_genes	logical scalar indicating whether the gene nodes in the filtered network will represent orthology IDs (expand_genes = FALSE) or organism-specific gene IDs (expand_genes = TRUE).

Value

A three-column matrix where each row represents an edge between two nodes.

References

Gatto, L. hpar: Human Protein Atlas in R.R package version 1.12.0.

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
# Build network

net <- MS_keggNetwork(metabo_paths = "hsa00010", signaling_paths = "hsa04014",
                      expand_genes = TRUE)

# Filter network by liver and cluster genes by orthology

net_filtered <- MS_tissueFilter(net, tissue = "liver")
```

MS_topologyFilter *Filter network based on distances or betweenness*

Description

This function allows reducing the dimensionality of a network, by removing nodes that do not meet the established distance and/or node betweenness criteria.

Usage

```
MS_topologyFilter(network_table, mode = "all", type, target_node, distance_th, bw_th)
```

Arguments

network_table	three-column matrix where each row represents and edge between two nodes. See function "MS_keggNetwork( )".
mode	character constant indicating whether a directed ("out") or undirected ("all") network will be considered.
type	character constant used to establish the criteria for filtering the network. "bw" indicates that edges (i.e. rows of the network_table) containing at least one node with betweenness below bw_th will be neglected. "distance" indicates edges containing at least one node with shortest path length to the target_node above distance_th will be neglected. "all" indicates that edges containing at least one node with either betweenness below bw_th or distance above distance_th, will be neglected.
target_node	character vector containing the ID of the node to which the distances will be calculated.
distance_th	numeric value corresponding to the distance threshold. Nodes with shortest path length to the target_node above this threshold will be removed from the network-table.
bw_th	numeric value corresponding to the normalized-betweenness threshold. Nodes with betweenness below this threshold will be removed from the network-table. See also "MS_nodeBW( )".

Value

A three-column matrix where each row represents an edge between two nodes.

References

Csardi, G. & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Examples

```
data(MetaboSignal_table)
# Remove edges containing nodes with distance to D-glucose ("cpd:C00031") > 2

network_filtered1 <- MS_topologyFilter(MetaboSignal_table, type = "distance",
                                       target_node = "cpd:C00031",
                                       distance_th = 2)

# Remove edges containing nodes with distance to D-glucose ("cpd:C00031") > 2 or
# normalized-betweenness < 0.00005

network_filtered2 <- MS_topologyFilter(MetaboSignal_table, type = "all",
                                       target_node = "cpd:C00031",
                                       distance_th = 2, bw_th = 0.00005)

# Note below that network_filtered1 has one edge more than network_filtered2. This is
# because "cpd:C00031" has betweenness = 0, and therefore it is removed in network_filtered2:
```

```
setdiff(as.vector(network_filtered1[, 1:2]),as.vector(network_filtered2[, 1:2]))
```

`ppiNet_example` *Signaling-transduction network*

Description

Signaling-transduction network generated by merging the interactions from OmniPath and TRRUST databases.

Usage

```
ppiNet_example
```

Format

Matrix

Value

Matrix

`regulatory_interactions`
Regulatory interactions from OmniPath and TRRUST

Description

This matrix contains a set of human regulatory interactions compiled from two literature-curated resources: OmniPath (directed protein-protein and signaling interactions reported in databases with an appropiate licence) and TRRUST (transcription factor-target interactions). For each interaction, both literature references and primary database references are reported. The users are responsible for respecting the terms of their licences and for citing them when required. This matrix can be edited or updated by the users if required.

Usage

```
regulatory_interactions
```

Format

Matrix

Value

Matrix

References

Ceol, A., et al. (2007). DOMINO: a database of domain-peptide interactions. Nucleic Acid Research, 35, D557-60.

Cui, Q., et al. (2007). A map of human cancer signaling. Molecular Systems Biology, 3:152.

Diella, F., et al. (2004). Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins. BMC Bioinformatics, 22, 5:79.

Dinkel, H., et al. (2012). ELM–the database of eukaryotic linear motifs. Nucleic Acid Research, 40, D242-51.

Han, H., et al. (2015). TRRUST: a reference database of human transcriptional regulatory interactions. Scientific Reports, 15, 11432.

Hornbeck, P.V., et al. (2012). PhosphoSitePlus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. Nucleic Acid Research, 40, D261-70.

Korcsmaros, T., et al. (2010). Uniformly curated signaling pathways reveal tissue-specific cross-talks and support drug target discovery. Bioinformatics, 26, 2042:2050.

Lynn, D.J., et al. (2008). InnateDB: facilitating systems-level analyses of the mammalian innate immune response. Molecular Systems Biology, 4, 218.

Orchard, S., et al. (2014). The MIntAct project–IntAct as a common curation platform for 11 molecular interaction databases. Nucleic Acid Research, 242, D358-63.

Pagel, P., et al. (2005). The MIPS mammalian protein-protein interaction database. Bioinformatics, 21, 832-834.

Papp, D., et al. (2012). The NRF2-related interactome and regulome contain multifunctional proteins and fine-tuned autoregulatory loops. FEBS Letters, 586, 1795-802.

Pawson, A.J., et al. (2014). The IUPHAR/BPS Guide to PHARMACOLOGY: an expert-driven knowledgebase of drug targets and their ligands. Nucleic Acids Research, 42, D1098-106.

Peri, S., et al. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research, 13, 2363-2371.

Turei, D., et al. (2015). Autophagy Regulatory Network - a systems-level bioinformatics resource for studying the mechanism and regulation of autophagy. Autophagy, 11, 155-165.

Turei, D., et al. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nature methods, 13, 966-967.

Sarkar, D., et al. (2015). LMPID: a manually curated database of linear motifs mediating protein-protein interactions. Database(Oxford), pii: bav014.

Shin, Y.C., et al. (2011). TRIP Database: a manually curated database of protein-protein interactions for mammalian TRP channels. Nucleic Acids Research, 39, D356-61.

Snel, B., et al. (2000). STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. Nucleic Acids Research, 28, 3442-3444.

Xenarios, I., et al. (2002). DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research, 30, 303-305.

Index