Package 'enrichViewNet'

November 11, 2026

Type Package

Version 1.10.0

Date 2023-08-12

Title From functional enrichment results to biological networks

Description This package enables the visualization of functional enrichment results as network graphs. First the package enables the visualization of enrichment results, in a format corresponding to the one generated by gprofiler2, as a customizable Cytoscape network. In those networks, both gene datasets (GO terms/pathways/protein complexes) and genes associated to the datasets are represented as nodes. While the edges connect each gene to its dataset(s). The package also provides the option to create enrichment maps from functional enrichment results. Enrichment maps enable the visualization of enriched terms into a network with edges connecting overlapping genes.

Encoding UTF-8

License Artistic-2.0

Depends R (>= 4.2.0)

Imports gprofiler2, strex, RCy3, jsonlite, stringr, enrichplot, igraph, reshape2, methods

Suggests BiocStyle, knitr, rmarkdown, ggplot2, scatterpie, ggtangle, ggrepel, testthat, ggnetwork, magick

biocViews BiologicalQuestion, Software, Network, NetworkEnrichment, GO

VignetteBuilder knitr

URL https://github.com/adeschen/enrichViewNet, https://adeschen.github.io/enrichViewNet/

BugReports https://github.com/adeschen/enrichViewNet/issues

RoxygenNote 7.3.3

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

git_branch RELEASE_3_23

git_last_commit d4a98cc

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Astrid Deschênes [aut, cre] (ORCID:
<https://orcid.org/0000-0001-7846-6749>),
Pascal Belleau [aut] (ORCID: <https://orcid.org/0000-0002-0802-1071>),
Robert L. Faure [aut] (ORCID: <https://orcid.org/0000-0003-1798-4723>),
Maria J. Fernandes [aut] (ORCID:
<https://orcid.org/0000-0002-3973-025X>),
Alexander Krasnitz [aut],
David A. Tuveson [aut] (ORCID: <https://orcid.org/0000-0002-8017-2712>)

Maintainer Astrid Deschênes <adeschen@hotmail.com>

Contents

enrichViewNet-package . . . 3
createBasicEmap . . . 4
createBasicEmapAsIgraph . . . 5
createCompareResultDataFrame . . . 6
createCXJSONForCytoscape . . . 8
createEnrichMap . . . 9
createEnrichMapAsIgraph . . . 10
createEnrichMapMultiBasic . . . 12
createEnrichMapMultiBasicAsIgraph . . . 14
createEnrichMapMultiComplex . . . 17
createEnrichMapMultiComplexAsIgraph . . . 19
createMetaDataSectionCXJSON . . . 22
createMultiEmap . . . 23
createMultiEmapAsIgraph . . . 24
createNetwork . . . 25
createNetworkForCytoscape . . . 27
demoGOST . . . 28
extractInformationWhenIntersection . . . 29
extractInformationWhenNoIntersection . . . 30
extractNodesAndEdgesInformation . . . 31
filterResults . . . 32
formatInformationForCXJSON . . . 33
isCytoscapeRunning . . . 34
manageNameDuplicationInEmap . . . 35
manageQueryDuplicationInEmap . . . 36
parentalNapaVsDMSODEG . . . 37
parentalNapaVsDMSOEnrichment . . . 39
removeRootTerm . . . 40
rosaNapaVsDMSODEG . . . 41
rosaNapaVsDMSOEnrichment . . . 43
similarityJaccard . . . 44

validateCreateEnrichMapArguments 45
validateCreateEnrichMapAsIgraphArg 47
validateCreateEnrichMapMultiBasicArgs 48
validateCreateEnrichMapMultiBasicAsIgraphArgs 50
validateCreateEnrichMapMultiBasicGOSTArgs 52
validateCreateEnrichMapMultiComplexArg 53
validateCreateEnrichMapMultiComplexAsIgraphArg 55
validateCreateEnrichMapMultiComplexGostPartOne 56
validateCreateEnrichMapMultiComplexGostPartTwo 58
validateCreateEnrichMapSubSectionArguments 59
validateCreateNetworkArguments 60

Index **62**

`enrichViewNet-package` *From functional enrichment results to biological networks*

Description

The `enrichViewNet` package enables the visualization of enrichment results, in a format corresponding to the one generated by `gprofiler2` (https://cran.r-project.org/web/packages/gprofiler2/index.html) under the form of `Cytoscape` network (https://cytoscape.org/).

Details

In those networks, both gene datasets (GO terms/pathways/protein complexes) and genes are represented as nodes. A edge connect a gene to its datasets. In the current version, only genes present in at least one gene dataset are retained.

Author(s)

Astrid Deschênes, Pascal Belleau, Robert L Faure, Maria J Fernandes, David A Tuveson

Maintainer: Astrid Deschênes <adeschen@hotmail.com>

See Also

- `createNetwork` for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- `createEnrichMap` for transforming functional enrichment results from gprofiler2 into an enrichment map in ggplot format
- `createEnrichMapAsIgraph` for transforming functional enrichment results from gprofiler2 into an enrichment map in igraph format
- `createEnrichMapMultiComplex` for transforming functional enrichment results from multiple gprofiler2 enrichment results into an enrichment map in ggplot format
- `createEnrichMapMultiComplexAsIgraph` for transforming functional enrichment results from multiple gprofiler2 enrichment results into an enrichment map in igraph format

createBasicEmap *Create a basic enrichment map*

Description

The function creates a basic enrichment map using functional enrichment results.

Usage

```
createBasicEmap(
  gostResults,
  backgroundGenes,
  showCategory,
  categoryLabel,
  categoryNode,
  significantMethod,
  line,
  ...
)
```

Arguments

gostResults	a data.frame containing the enrichment results to be plot.
backgroundGenes	a vector of character string representing the name of the genes present in the request.
showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).
categoryNode	a positive numeric representing the amount by which plotting category nodes should be scaled relative to the default (1).
significantMethod	a character string representing the name of the multiple testing correction method used on the results.
line	a non-negative numeric representing the scale of line width.
...	additional arguments that will be pass to the emapplot function.

Value

a ggplot object representing the enrichment map.

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retain the results section
gostResults <- as.data.frame(parentalNapaVsDMSOEnrichment$result)

## Limit the results to Wikipathways
## and remove the root term
gostResults <- gostResults[which(gostResults$source == "WP"),]
gostResults <- gostResults[which(gostResults$term_name != "WIKIPATHWAYS"),]

## Extract meta data information
meta <- parentalNapaVsDMSOEnrichment$meta

## Get all background genes
backgroundGenes <- meta$query_metadata$queries[["parental_napa_vs_DMSO"]]

## Get significant method
significantMethod <- meta$query_metadata$significance_threshold_method

## Create basic enrichment map using Wikipathways terms
enrichViewNet:::createBasicEmap(gostResults=gostResults,
    backgroundGenes=backgroundGenes, showCategory=30L,
    categoryLabel=1, categoryNode=1,
    significantMethod=significantMethod, line=1)
```

`createBasicEmapAsIgraph`

Create a basic enrichment map

Description

The function creates a basic enrichment map using functional enrichment results.

Usage

```
createBasicEmapAsIgraph(
  gostResults,
  backgroundGenes,
  showCategory,
  similarityCutOff
)
```

Arguments

`gostResults`	a `data.frame` containing the enrichment results to be plot.

backgroundGenes
: a vector of character string representing the name of the genes present in the request.

showCategory
: a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.

similarityCutOff
: a positive numeric between 0 and 1 indicating the minimum level of similarity between two terms to have an edge linking the terms.

Value

a igraph object representing the enrichment map.

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retain the results section
gostResults <- as.data.frame(parentalNapaVsDMSOEnrichment$result)

## Limit the results to Wikipathways
## and remove the root term
gostResults <- gostResults[which(gostResults$source == "WP"),]
gostResults <- gostResults[which(gostResults$term_name != "WIKIPATHWAYS"),]

## Extract meta data information
meta <- parentalNapaVsDMSOEnrichment$meta

## Get all background genes
backgroundGenes <- meta$query_metadata$queries[["parental_napa_vs_DMSO"]]

## Create basic enrichment map, as an igraph, using Wikipathways terms
enrichViewNet:::createBasicEmapAsIgraph(gostResults=gostResults,
    backgroundGenes=backgroundGenes, showCategory=30L,
    similarityCutOff=0.2)
```

---

createCompareResultDataFrame

Create a data frame containing the compiled enrichment results from multiple queries

---

Description

The function creates a data frame object using the list of queries and list of enrichment results given be the user.

Usage

```
createCompareResultDataFrame(queryList, gostResultsList)
```

Arguments

queryList	a list containing the query names.
gostResultsList	a list of data.frame containing the enrichment results to be plot with different group identification.

Value

a data.frame object

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retain the results section
gostResults <- as.data.frame(parentalNapaVsDMSOEnrichment$result)

## Limit the results subsection of REACTOME and KEGG
## Kegg is replicated to show shared results between queries
gostResultsREAC <- gostResults[which(gostResults$source == "REAC"),]
gostResultsREAC <- gostResultsREAC[1:13, ]
gostResultsKEGG <- gostResults[which(gostResults$source == "KEGG"),]
gostResultsKEGG2 <- gostResultsKEGG[1:6,]

## List of query names with duplicated names
queryList <- list("parental_vs_DMSO", "rosa_vs_DMSO", "parental_vs_DMSO",
    "rosa_vs_DMSO", "parental_vs_Control", "rosa_vs_DMSO")

## Extract meta data information
queryList <- list("parental - REACTOME", "parental - KEGG - v1",
    "parental - KEGG - v2")

## Change the query names for the duplicated names
enrichViewNet:::createCompareResultDataFrame(queryList=queryList,
    gostResultsList=list(gostResultsREAC, gostResultsKEGG,
            gostResultsKEGG2))
```

createCXJSONForCytoscape *Create CX JSON text representing the network*

Description

Create a CX JSON text that represent the network which includes information about nodes and edges present in the network.

Usage

```
createCXJSONForCytoscape(nodeEdgeInfo, title)
```

Arguments

nodeEdgeInfo	a list created by gprofiler2 that contains the results from an enrichment analysis. TODO
title	a character string representing the name assigned to the network.

Value

character string that represent the network in a CX JSON format.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retained the GO Molecular Function results
results <- parentalNapaVsDMSOEnrichment$result[
        parentalNapaVsDMSOEnrichment$result$source == "GO:MF", ]

## Extract node and edge information
information <- enrichViewNet:::extractInformationWhenIntersection(
        gostResults=results)

jsonFormat <- enrichViewNet:::createCXJSONForCytoscape(
                nodeEdgeInfo=information, title="WikiPathways")
```

createEnrichMap *Using functional enrichment results in gprofiler2 format to create an enrichment map*

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph.

Usage

```
createEnrichMap(
  gostObject,
  query,
  source = c("TERM_ID", "GO:MF", "GO:CC", "GO:BP", "KEGG", "REAC", "TF", "MIRNA", "HPA",
    "CORUM", "HP", "WP"),
  termIDs = NULL,
  removeRoot = TRUE,
  showCategory = 30L,
  categoryLabel = 1,
  categoryNode = 1,
  line = 1,
  ...
)
```

Arguments

gostObject	a list corresponding to gprofiler2 enrichment output that contains and that contains the results from an enrichment analysis.
query	a character string representing the name of the query that is going to be used to generate the graph. The query must exist in the gostObject object.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network. Default: NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present). Default: TRUE.

showCategory	a positive integer representing the maximum number of terms to display. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed. Default: 30L.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1). Default: 1.s
categoryNode	a positive numeric representing the amount by which plotting category nodes should be scaled relative to the default (1). Default: 1.
line	a non-negative numeric representing the scale of line width. Default: 1.
...	additional arguments that will be pass to the emapplot function.

Value

a ggplot object which is the enrichment map for enrichment results.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Extract query information (only one in this dataset)
query <- unique(parentalNapaVsDMSOEnrichment$result$query)

## Create graph for Gene Ontology - Cellular Component related results
createEnrichMap(gostObject=parentalNapaVsDMSOEnrichment,
    query=query, source="GO:CC", removeRoot=TRUE)
```

---

createEnrichMapAsIgraph

Using functional enrichment results in gprofiler2 format to create an enrichment map in an igraph format

---

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph. The output is an igraph.

Usage

```
createEnrichMapAsIgraph(
  gostObject,
  query,
  source = c("TERM_ID", "GO:MF", "GO:CC", "GO:BP", "KEGG", "REAC", "TF", "MIRNA", "HPA",
    "CORUM", "HP", "WP"),
  termIDs = NULL,
  removeRoot = TRUE,
  showCategory = 30L,
  similarityCutOff = 0.2
)
```

Arguments

gostObject	a list corresponding to gprofiler2 enrichment output that contains and that contains the results from an enrichment analysis.
query	a character string representing the name of the query that is going to be used to generate the graph. The query must exist in the gostObject object.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network. Default: NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present). Default: TRUE.
showCategory	a positive integer representing the maximum number of terms to display. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed. Default: 30L.
similarityCutOff	a positive numeric between 0 and 1 indicating the minimum level of similarity between two terms to have an edge linking the terms. Default: 0.20.

Value

a igraph object which is the enrichment map for enrichment results. The node have 2 attributes: "name" and "size". The "name" corresponds to the term description. While the "size" corresponds to the number of genes found in the specific gene set. The edges have 3 attributes: "similarity", "width", and "weight". All those 3 attributes correspond to the Jaccard coefficient.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Extract query information (only one in this dataset)
query <- unique(parentalNapaVsDMSOEnrichment$result$query)

## Create graph for Gene Ontology - Cellular Component related results
mapG <- createEnrichMapAsIgraph(gostObject=parentalNapaVsDMSOEnrichment,
    query=query, source="GO:CC", removeRoot=TRUE,
    showCategory=30L, similarityCutOff=0.20)

## Required library igraph to show the graph
if(requireNamespace("igraph", quietly=TRUE)) {
    ## Using library igraph to show the graph
    library(igraph)
    plot(mapG)
}

if (requireNamespace("ggplot2", quietly=TRUE) &&
        requireNamespace("igraph", quietly=TRUE) &&
        requireNamespace("ggtangle", quietly=TRUE) &&
        requireNamespace("ggnetwork", quietly=TRUE)) {
    ## Using more complex set of libraries to display personalized graph
    library(ggplot2)
    library(igraph)
    library(ggnetwork)
    library(ggtangle)
    emapG <- ggplot(mapG, layout=layout_with_fr) +
                geom_edge(color="gray", linewidth=1) +
                geom_nodes(aes(size=size)) +
                geom_nodetext(aes(label=name), color="black", size=3) +
                theme_void()
    emapG
}
```

`createEnrichMapMultiBasic` *Using functional enrichment results in gprofiler2 format to create an enrichment map with multiple groups from different enrichment analyses*

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph.

Usage

```
createEnrichMapMultiBasic(
  gostObjectList,
  queryList,
  source = c("TERM_ID", "GO:MF", "GO:CC", "GO:BP", "KEGG", "REAC", "TF", "MIRNA", "HPA",
    "CORUM", "HP", "WP"),
  termIDs = NULL,
  removeRoot = TRUE,
  showCategory = 30L,
  categoryLabel = 1,
  categoryNode = 1,
  line = 1,
  ...
)
```

Arguments

gostObjectList	a list of gprofiler2 objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the queryList parameter.
queryList	a list of character strings representing the names of the queries that are going to be used to generate the graph. The query names must exist in the associated gostObjectList objects and follow the same order. The number of entries must correspond to the number of entries for the gostObjectList parameter.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network. Default: NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present). Default: TRUE.
showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed. Default: 30L.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1). Default: 1.
categoryNode	a positive numeric representing the amount by which plotting category nodes should be scaled relative to the default (1). Default: 1.
line	a non-negative numeric representing the scale of line width. Default: 1.
...	additional arguments that will be pass to the emapplot function.

Value

a ggplot object which is the enrichment map for enrichment results.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing results from 2 enrichment analyses done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Extract query information (only one in each dataset)
query1 <- unique(parentalNapaVsDMSOEnrichment$result$query)[1]
query2 <- unique(rosaNapaVsDMSOEnrichment$result$query)[1]

## Create graph for KEGG related results from
## 2 enrichment analyses
createEnrichMapMultiBasic(gostObjectList=list(parentalNapaVsDMSOEnrichment,
    rosaNapaVsDMSOEnrichment),
    queryList=list(query1, query2), source="KEGG", removeRoot=TRUE)
```

---

createEnrichMapMultiBasicAsIgraph

Using functional enrichment results in gprofiler2 format to create an enrichment map with multiple groups from different enrichment analyses as an igraph

---

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph. The output is an enrichment map in igraph format.

Usage

```
createEnrichMapMultiBasicAsIgraph(
  gostObjectList,
  queryList,
  source = c("TERM_ID", "GO:MF", "GO:CC", "GO:BP", "KEGG", "REAC", "TF", "MIRNA", "HPA",
    "CORUM", "HP", "WP"),
  termIDs = NULL,
  removeRoot = TRUE,
```

```
  showCategory = 30L,
  similarityCutOff = 0.2
)
```

Arguments

gostObjectList	a list of gprofiler2 objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the queryList parameter.
queryList	a list of character strings representing the names of the queries that are going to be used to generate the graph. The query names must exist in the associated gostObjectList objects and follow the same order. The number of entries must correspond to the number of entries for the gostObjectList parameter.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network. Default: NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present). Default: TRUE.
showCategory	a positive integer representing the maximum number of terms to display. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed. Default: 30L.
similarityCutOff	a positive numeric between 0 and 1 indicating the minimum level of similarity between two terms, calculated using the Jaccard coefficient, to have an edge linking the terms. Default: 0.20.

Value

a igraph object which is the enrichment map for enrichment results. The node have 5 attributes: "name", "size", "pie", "cluster", and "pieName". The "name" corresponds to the term description. While the "size" corresponds to the number of unique genes found in the specific gene set when looking at all the experiments. The edges have 3 attributes: "similarity", "width", and "weight". All those 3 attributes correspond to the Jaccard coefficient.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing results from 2 enrichment analyses done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Extract query information (only one in each dataset)
query1 <- unique(parentalNapaVsDMSOEnrichment$result$query)[1]
query2 <- unique(rosaNapaVsDMSOEnrichment$result$query)[1]

## Create graph for KEGG related results from
## 2 enrichment analyses
emapG <- createEnrichMapMultiBasicAsIgraph(gostObjectList=
    list(parentalNapaVsDMSOEnrichment, rosaNapaVsDMSOEnrichment),
    queryList=list(query1, query2), source="KEGG", removeRoot=TRUE,
    similarityCutOff=0.3)

if (requireNamespace("ggplot2", quietly=TRUE) &&
        requireNamespace("igraph", quietly=TRUE) &&
        requireNamespace("scatterpie", quietly=TRUE) &&
        requireNamespace("ggtangle", quietly=TRUE) &&
        requireNamespace("ggrepel", quietly=TRUE)) {
    ## Create a visual representation of the enrichment map
    ## by default
    library(igraph)
    plot(emapG)

    ## Add see to reproduce the same graph
    set.seed(12)

    library(ggplot2)
    library(ggtangle)
    library(scatterpie)
    library(ggrepel)

    emapGraph <- ggplot(emapG, layout=layout_with_fr) +
                geom_edge(color="gray", linewidth=1)

    pieInfo <- as.data.frame(do.call(rbind, V(emapG)$pie))
    colnames(pieInfo) <- V(emapG)$pieName[[1]]

    ## Add information about the groups associated with each node in the
    ## ggplot object so that the node can be colored accordingly
    for (i in seq_len(ncol(pieInfo))) {
        emapGraph$data[colnames(pieInfo)[i]] <- pieInfo[, i]
    }

    ## Using scatterpie, ggtangle and ggrepel to generate the graph
    ## geom_scatterpie() allows to have scatter pie plot
    ## geom_text_repel() allows to have minimum overlying terms
    ## coord_fixed() forces the plot to have a 1:1 aspect ratio
    emapGraph +
```

```
        geom_scatterpie(aes(x=x, y=y, r=size/50),
            cols=c(colnames(pieInfo)), legend_name = "Group", color=NA) +
        geom_scatterpie_legend(radius=emapGraph$data$size/50, n=4,
            x=max(emapGraph$data$x), y=min(emapGraph$data$y),
            labeller=function(x) {round(x*50)}, label_position="right") +
        geom_text_repel(aes(x=x, y=y, label=label), max.overlaps=20) +
        coord_fixed()
}
```

`createEnrichMapMultiComplex` *Using functional enrichment results in gprofiler2 format to create an enrichment map with multiple groups from same or different enrichment analyses*

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph.

Usage

```
createEnrichMapMultiComplex(
  gostObjectList,
  queryInfo,
  showCategory = 30L,
  categoryLabel = 1,
  categoryNode = 1,
  line = 1,
  ...
)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.
`queryInfo`	a `data.frame` contains one row per group being displayed. The number of rows must correspond to the number of entries for the `gostObjectList` parameter. The mandatory columns are: • `queryName`: a `character` string representing the name of the query retained for this group). The query names must exist in the associated `gostObjectList` objects and follow the same order.

- source: a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
- removeRoot: a logical that specified if the root terms of the selected source should be removed (when present).
- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

showCategory — a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed. Default: 30L.

categoryLabel — a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1). Default: 1.

categoryNode — a positive numeric representing the amount by which plotting category nodes should be scaled relative to the default (1). Default: 1.

line — a non-negative numeric representing the scale of line width. Default: 1.

... — additional arguments that will be pass to the emapplot function.

Value

a ggplot object which is the enrichment map for enrichment results.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing results from 2 enrichment analyses done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Create list of enrichment objects required to generate the enrichment map
gostObjectList=list(parentalNapaVsDMSOEnrichment,
    parentalNapaVsDMSOEnrichment, rosaNapaVsDMSOEnrichment,
    rosaNapaVsDMSOEnrichment)

## Create data frame containing required information enabling the
## selection of the retained enriched terms for each enrichment analysis.
```

```
## One line per enrichment analyses present in the gostObjectList parameter
## With this data frame, the enrichment results will be split in 4 groups:
## 1) KEGG significant terms from parental napa vs DMSO (no root term)
## 2) REACTOME significant terms from parental napa vs DMSO (no root term)
## 3) KEGG significant terms from rosa napa vs DMSO (no root term)
## 4) REACTOME significant terms from rosa napa vs DMSO (no root term)
queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
        "parental_napa_vs_DMSO", "rosa_napa_vs_DMSO", "rosa_napa_vs_DMSO"),
    source=c("KEGG", "REAC", "KEGG", "REAC"),
    removeRoot=c(TRUE, TRUE, TRUE, TRUE), termIDs=c("", "", "", ""),
    groupName=c("parental - KEGG", "parental - Reactome",
        "rosa - KEGG", "rosa - Reactome"), stringsAsFactors=FALSE)

## Create graph for KEGG and REACTOME significant results from
## 2 enrichment analyses
createEnrichMapMultiComplex(gostObjectList=gostObjectList,
    queryInfo=queryDataFrame, line=1.5)
```

`createEnrichMapMultiComplexAsIgraph`

Using functional enrichment results in gprofiler2 format to create an enrichment map with multiple groups from same or different enrichment analyses in an igraph format

---

Description

User selected enrichment terms are used to create an enrichment map. The selection of the term can by specifying by the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The map is only generated when there is at least on significant term to graph.

Usage

```
createEnrichMapMultiComplexAsIgraph(
  gostObjectList,
  queryInfo,
  showCategory = 30L,
  similarityCutOff = 0.2
)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.
`queryInfo`	a `data.frame` contains one row per group being displayed. The number of rows must correspond to the number of entries for the `gostObjectList` parameter. The mandatory columns are:

- queryName: a character string representing the name of the query retained for this group). The query names must exist in the associated gostObjectList objects and follow the same order.
- source: a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
- removeRoot: a logical that specified if the root terms of the selected source should be removed (when present).
- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

showCategory a positive integer or NULL. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed. Default: 30L.

similarityCutOff
a positive numeric, larger than zero and small than 1 that represent the minimum similarity level between two nodes (terms) to be linked by an edge. Default: 0.2.

Value

a igraph object which is the enrichment map for enrichment results. The node have 5 attributes: "name", "size", "pie", "cluster", and "pieName". The "name" corresponds to the term description. While the "size" corresponds to the number of unique genes found in the specific gene set when looking at all the experiments. The edges have 3 attributes: "similarity", "width", and "weight". All those 3 attributes correspond to the Jaccard coefficient.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing results from 2 enrichment analyses done with
## gprofiler2 queries
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## The graph will be split in 4 groups
## Groups 1 and 2 are using the parental Napa vs DMSO dataset
## Group 3 and 4 are using the rosa Napa vs DMSO dataset
gostObjectList=list(parentalNapaVsDMSOEnrichment,
    parentalNapaVsDMSOEnrichment, rosaNapaVsDMSOEnrichment,
```

```
    rosaNapaVsDMSOEnrichment)

## Create data frame containing required information enabling the
## selection of the retained enriched terms for each enrichment analysis.
## One line per enrichment analyses present in the gostObjectList parameter
## With this data frame, the enrichment results will be split in 4 groups:
## 1) KEGG significant terms from parental napa vs DMSO (no root term)
## 2) REACTOME significant terms from parental napa vs DMSO (no root term)
## 3) KEGG significant terms from rosa napa vs DMSO (no root term)
## 4) REACTOME significant terms from rosa napa vs DMSO (no root term)
queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
        "parental_napa_vs_DMSO", "rosa_napa_vs_DMSO", "rosa_napa_vs_DMSO"),
    source=c("KEGG", "REAC", "KEGG", "REAC"),
    removeRoot=c(TRUE, TRUE, TRUE, TRUE), termIDs=c("", "", "", ""),
    groupName=c("parental - KEGG", "parental - Reactome",
        "rosa - KEGG", "rosa - Reactome"), stringsAsFactors=FALSE)

## Create graph for KEGG and REACTOME significant results from
## 2 enrichment analyses in an igraph format
emap <- createEnrichMapMultiComplexAsIgraph(gostObjectList=gostObjectList,
    queryInfo=queryDataFrame, showCategory=5)

if (requireNamespace("ggplot2", quietly=TRUE) &&
        requireNamespace("igraph", quietly=TRUE) &&
        requireNamespace("scatterpie", quietly=TRUE) &&
        requireNamespace("ggtangle", quietly=TRUE) &&
        requireNamespace("ggrepel", quietly=TRUE)) {
    ## Create a visual representation of the enrichment map
    ## by default
    library(igraph)
    plot(emap)

    ## Add see to reproduce the same graph
    set.seed(12)

    library(ggplot2)
    library(ggtangle)
    library(scatterpie)
    library(ggrepel)

    emapG <- ggplot(emap, layout=layout_with_fr) +
                geom_edge(color="gray", linewidth=1)

    pieInfo <- as.data.frame(do.call(rbind, V(emap)$pie))
    colnames(pieInfo) <- V(emap)$pieName[[1]]

    ## Add information about the groups associated with each node in the
    ## ggplot object so that the node can be colored accordingly
    for (i in seq_len(ncol(pieInfo))) {
        emapG$data[colnames(pieInfo)[i]] <- pieInfo[, i]
    }

    ## Using scatterpie, ggtangle and ggrepel to generate the graph
```

```
    ## geom_scatterpie() allows to have scatter pie plot
    ## geom_text_repel() allows to have minimum overlying terms
    ## coord_fixed() forces the plot to have a 1:1 aspect ratio
    emapG +
        geom_scatterpie(aes(x=x, y=y, r=size/50),
            cols=c(colnames(pieInfo)), legend_name = "Group", color=NA) +
        geom_scatterpie_legend(radius=emapG$data$size/50, n=4,
            x=max(emapG$data$x), y=max(emapG$data$y),
            labeller=function(x) {round(x*50)}, label_position="right") +
        geom_text_repel(aes(x=x, y=y, label=label), max.overlaps=20) +
        coord_fixed()
}
```

`createMetaDataSectionCXJSON` *Create meta data section for the CX JSON file*

Description

Create meta data section for the CX JSON file that contains the network information

Usage

```
createMetaDataSectionCXJSON()
```

Value

a `JSON` object that contains the meta data section related to the network

Author(s)

Astrid Deschênes

Examples

```
## Create the JSON object that contains the meta data information
enrichViewNet:::createMetaDataSectionCXJSON()
```

createMultiEmap *Create a basic enrichment map*

Description

The function creates a basic enrichment map using functional enrichment results.

Usage

```
createMultiEmap(
  gostResultsList,
  queryList,
  showCategory,
  categoryLabel,
  categoryNode,
  line,
  ...
)
```

Arguments

gostResultsList	a list of data.frame containing the enrichment results to be plot with different group identification.
queryList	a list of character string representing the name of query retained for each enrichment results present in the gostResultsList parameter. The query should be present in its associated enrichment results.
showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).
categoryNode	a positive numeric representing the amount by which plotting category nodes should be scaled relative to the default (1).
line	a non-negative numeric representing the scale of line width.
...	additional arguments that will be pass to the emapplot function.

Value

a ggplot object representing the enrichment map with different colors for each group of enrichment results.

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retain the results section
gostResults <- as.data.frame(parentalNapaVsDMSOEnrichment$result)

## Limit the results subsection of REACTOME and KEGG
gostResultsREAC <- gostResults[which(gostResults$source == "REAC"),]
gostResultsREAC <- gostResultsREAC[1:13, ]
gostResultsKEGG <- gostResults[which(gostResults$source == "KEGG"),]

## Extract meta data information
queryList <- list("parental - REACTOME", "parental - KEGG")

## Create basic enrichment map using Wikipathways terms
enrichViewNet:::createMultiEmap(gostResultsList=list(gostResultsREAC,
    gostResultsKEGG), queryList=queryList, showCategory=30L,
    categoryLabel=1, categoryNode=1, line=1.4)
```

`createMultiEmapAsIgraph` *Create a complex enrichment map as an igraph*

Description

The function creates a complex enrichment map, as an igraph, using functional enrichment results.

Usage

```
createMultiEmapAsIgraph(
  gostResultsList,
  queryList,
  showCategory,
  similarityCutOff
)
```

Arguments

`gostResultsList`	a `list` of `data.frame` containing the enrichment results to be plot with different group identification.
`queryList`	a `list` of `character` string representing the name of query retained for each enrichment results present in the `gostResultsList` parameter. The query should be present in its associated enrichment results.
`showCategory`	a positive `integer` or a `vector` of `characters` representing terms. If a `integer`, the first `n` terms will be displayed. If `NULL`, all terms will be displayed.

similarityCutOff
: a positive numeric, larger than zero and small than 1 that represent the minimum similarity level between two nodes (terms) to be linked by an edge.

Value

a igraph object representing the enrichment map with different colors for each group of enrichment results.

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retain the results section
gostResults <- as.data.frame(parentalNapaVsDMSOEnrichment$result)

## Limit the results subsection of REACTOME and KEGG
## Kegg is replicated to show shared results between queries
gostResultsREAC <- gostResults[which(gostResults$source == "REAC"),]
gostResultsREAC <- gostResultsREAC[1:13, ]
gostResultsKEGG <- gostResults[which(gostResults$source == "KEGG"),]
gostResultsKEGG2 <- gostResultsKEGG[1:6,]

## Extract meta data information
queryList <- list("parental - REACTOME", "parental - KEGG - v1",
    "parental - KEGG - v2")

## Create basic enrichment map using Wikipathways terms
igraph <- enrichViewNet:::createMultiEmapAsIgraph(
    gostResultsList=list(gostResultsREAC, gostResultsKEGG,
            gostResultsKEGG2),
    queryList=queryList, showCategory=30L, similarityCutOff=0.5)
```

---

createNetwork *Using functional enrichment results from gprofiler2 to create a Cytoscape network*

Description

User selected enrichment terms are used to create a Cytoscape network where the selected terms and the genes that where part of the enrichment analysis are all represented as nodes. Edges are linking the genes to their terms. The selection of the term can by specifying the source of the terms (GO:MF, REAC, TF, etc.) or by listing the selected term IDs. The network is only generated when there is at least on significant term to graph. When the enrichment analysis contains more than one query, only one query can be selected to generate the network.

Usage

```
createNetwork(
  gostObject,
 source = c("TERM_ID", "GO:MF", "GO:CC", "GO:BP", "KEGG", "REAC", "TF", "MIRNA", "HPA",
    "CORUM", "HP", "WP"),
  termIDs = NULL,
  removeRoot = TRUE,
  query = NULL,
  title = "gprofiler network",
  collection = "enrichment results",
  fileName = "gprofilerNetwork.cx"
)
```

Arguments

gostObject	a list created by gprofiler2 that contains the results from an enrichment analysis.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network. Default: NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present). Default: TRUE.
query	a character string that specified the retained query to generate the network. When NULL, the query present in the result is retained; NULL cannot be used when more than one query is present. Default: NULL.
title	a character string representing the name assigned to the network. Default: "gprofiler network".
collection	a character string representing the collection name assigned to the network. Default: "enrichment results".
fileName	a character string representing the name of the CX JSON file that is created when Cytoscape is not running. The name must have a '.cx' extension. Default: "gprofilerNetwork_01.cx".

Value

TRUE

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Some of the enrichment results present in the dataset
head(parentalNapaVsDMSOEnrichment$result)

## Create network for Gene Ontology - Molecular Function related results
## in Cytoscape (when the application is opened)
## Otherwise, create a CX file in the temporary directory
## The file can be opened in Cytoscape
createNetwork(gostObject=parentalNapaVsDMSOEnrichment, source="KEGG",
    removeRoot=FALSE, title="KEGG Graph",
    fileName=file.path(tempdir(), "KEGG_demo.cx"))
```

---

createNetworkForCytoscape

Create network and load it into Cytoscape

Description

Create network from gprofiler2 results and load it into Cytoscape

Usage

```
createNetworkForCytoscape(nodeEdgeInfo, title, collection)
```

Arguments

nodeEdgeInfo	a TODO
title	a character string representing the name assigned to the network.
collection	a character string representing the collection name assigned to the network.

Value

TRUE

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retained the GO Molecular Function results
results <- parentalNapaVsDMSOEnrichment$result[
        parentalNapaVsDMSOEnrichment$result$source == "GO:MF", ]

## Extract node and edge information
information <- enrichViewNet:::extractInformationWhenIntersection(
        gostResults=results)

## The creation of the network can only be done when Cytoscape
## is up and running
## A network using GO - Molecular Function enriched terms will be
## generated and loaded into Cytoscape
if (enrichViewNet:::isCytoscapeRunning()) {
    enrichViewNet:::createNetworkForCytoscape(nodeEdgeInfo=information,
        title="Test", collection="New Collection")
}
```

---

demoGOST — *The result of a functional enrichment analysis done with* gprofiler2 *(https://cran.r-project.org/web/packages/gprofiler2/vignettes/gprofiler2.html).*

Description

The object is a `list` with 2 entries. It contains the results of the enrichment analysis as well as the metadata related to the analysis.

Usage

```
data(demoGOST)
```

Format

The `list` contains two entries. The `result` entry contains a `data.frame` with the significant results obtained by an enrichment analysis done with `gprofiler2`. The `meta` entry contains a named list with all the metadata for the query.

Details

This dataset can be used to test the `createNetwork` function.

Value

A `list` containing two entries. The `result` entry contains a `data.frame` with the significant results obtained by an enrichment analysis done with `gprofiler2`. The `meta` entry contains a named list with all the metadata for the query.

See Also

- createNetwork for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- createEnrichMap for transforming functional enrichment results from gprofiler2 into an enrichment map

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(demoGOST)

## Create network for WikiPathways results
## in Cytoscape (if the application is open)
## Otherwise, create a CX file in the temporary directory
## The file can be opened in Cytoscape
createNetwork(gostObject=demoGOST, source="WP", title="Wikipathways",
    fileName=file.path(tempdir(), "Wikipathways_Demo.cx"))
```

---

extractInformationWhenIntersection

Extract node and edge information to be used to create network when interaction column is present

---

Description

Create a list containing all node and edge information needed to create the network

Usage

```
extractInformationWhenIntersection(gostResults)
```

Arguments

`gostResults`	a `data.frame` containing the terms retained for the creation of the network. The `data.frame` must contain a column called "intersection".

Value

`list` containing 2 entries:

- `"geneNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are genes.
- `"termNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are terms.
- `"edges"`: a `data.frame` containing the information about the edges present in the network. The edges connect one gene to one term.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retained the GO Molecular Function results
results <- parentalNapaVsDMSOEnrichment$result[
        parentalNapaVsDMSOEnrichment$result$source == "GO:MF", ]

## Extract node and edge information
information <-
    enrichViewNet:::extractInformationWhenIntersection(
        gostResults=results)
```

---

extractInformationWhenNoIntersection

Extract information about nodes and edges to be used to create network when interaction column is missing

---

Description

Create a list containing all node and edge information needed to create the network

Usage

```
extractInformationWhenNoIntersection(gostResults, gostObject)
```

Arguments

`gostResults`	a `data.frame` containing the terms retained for the creation of the network. The `data.frame` does not contain a column called "intersection".
`gostObject`	a `list` created by gprofiler2 that contains the results and meta-data from an enrichment analysis.

Value

`list` containing 2 entries:

- `"geneNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are genes.
- `"termNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are terms.
- `"edges"`: a `data.frame` containing the information about the edges present in the network. The edges connect one gene to one term.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(demoGOST)

## Only retained the WikiPathways results
results <- demoGOST$result[demoGOST$result$source == "WP", ]

information <- enrichViewNet:::extractInformationWhenNoIntersection(
                gostResults=results, gostObject=demoGOST)
```

`extractNodesAndEdgesInformation`
Extract node and edge information from the enrichment results

---

Description

Create a list containing all node and edge information needed to create the network

Usage

```
extractNodesAndEdgesInformation(gostResults, gostObject)
```

Arguments

`gostResults`	a `data.frame` containing the terms retained for the creation of the network. The `data.frame` does not contain a column called "intersection".
`gostObject`	a `list` created by gprofiler2 that contains the results from an enrichment analysis.

Value

`list` containing 2 entries:

- `"geneNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are genes.
- `"termNodes"`: a `data.frame` containing the information about the nodes present in the network. The nodes are terms.
- `"edges"`: a `data.frame` containing the information about the edges present in the network. The edges connect one gene to one term.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retained the GO Molecular Function results
results <- parentalNapaVsDMSOEnrichment$result[
        parentalNapaVsDMSOEnrichment$result$source == "GO:MF", ]

## Extract node and edge information
information <-
    enrichViewNet:::extractNodesAndEdgesInformation(
        gostResults=results, gostObject=parentalNapaVsDMSOEnrichment)
```

`filterResults` *Filter the results to retain only the selected terms*

Description

Filter the enrichment results to retain only the selected terms and remove root term if requested.

Usage

```
filterResults(gostResults, source, termIDs, removeRoot)
```

Arguments

`gostResults`	a `data.frame` containing the enriched terms that should be filtered.

source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.
termIDs	a vector of character strings that contains the term IDS retained for the creation of the network or NULL.
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present).

Value

a data.frame of filtered terms with or without the root term.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(demoGOST)

## Only retained the GO - Molecular Function results
results <- demoGOST$result

## Remove WIKIPATHWAYS root term
selected <- enrichViewNet:::filterResults(gostResults=results, source="WP",
    termIDs=NULL, removeRoot=TRUE)
```

---

formatInformationForCXJSON

Transform the node and edge information into a format easy to process to create a CX JSON text file

---

Description

Extract information about nodes and edges that is necessary to create the CX JSON text representing the network

Usage

```
formatInformationForCXJSON(results)
```

Arguments

results	a list containing the information about the nodes and edges present in the network. TODOS

Value

a list containing 4 entries:

- "nodes": a data.frame containing the information about the nodes present in the network.
- "edges": a data.frame containing the information about the edges present in the network.
- "nodeAttributes": a data.frame containing the attributes associated to the nodes present in the network.
- "edgesAttributes": a data.frame containing the attributes associated to the edges present in the network

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(parentalNapaVsDMSOEnrichment)

## Only retained the GO Molecular Function results
results <- parentalNapaVsDMSOEnrichment$result[
        parentalNapaVsDMSOEnrichment$result$source == "GO:MF", ]

## Extract node and edge information
info <- enrichViewNet:::extractNodesAndEdgesInformation(gostResults=results,
            gostObject=parentalNapaVsDMSOEnrichment)

## Format node and edge information
information <- enrichViewNet:::formatInformationForCXJSON(result=info)
```

---

isCytoscapeRunning *Verifying that Cytoscape is running*

Description

Verifying that Cytoscape is running

Usage

```
isCytoscapeRunning()
```

Value

a `logical` indicating if Cytoscape is running.

Author(s)

Astrid Deschênes

Examples

```
## Test if Cytoscape is running
enrichViewNet:::isCytoscapeRunning()
```

---

manageNameDuplicationInEmap

Change name description in data frame when more than one term as the same name

---

Description

The function adds the ID of the term at the name of the name description when terms with different ID share the same name. For example, "MAPK pathway" would become "MAPK pathway (KEGG:04010)".

Usage

```
manageNameDuplicationInEmap(clProfDF)
```

Arguments

`clProfDF`	a `data.frame` containing the enrichment terms that are going to be graphed as an enrichment map. The `data.frame` must contain at least one case of duplicated name descriptions for different term IDs.

Value

a `data.frame` with the name descriptions modified for terms with duplicated name descriptions but different term IDs.

Author(s)

Astrid Deschênes

Examples

```
## Data frame with duplicated name descriptions for different IDs
clustData <- data.frame(Cluster=c("group 1" , "group 1", "group 2"),
    ID=c("WP:WP4925", "WP:WP382", "KEGG:04010"),
    Description=c("Unfolded protein response",
                  rep("MAPK signaling pathway", 2)),
    GeneRatio=c("4/157", "3/157", "3/157"),
    BgRatio=c("4/24022", "3/24022", "3/24022"),
    pvalues=c(1.55e-4, 8.13e-8, 4.33e-5),
    p.adjust=c(1e-3, 1e-3, 1.4e-3), qvalue=c(1e-3, 1e-3, 1.4e-3),
    geneID=c("ENSG000107968/ENSG000120129/ENSG000123358/ENSG000158050",
        "ENSG000107968/ENSG000120129/ENSG000158050",
        "ENSG000107968/ENSG000120129/ENSG000158050"),
    Count=c(4, 3, 3))

## Change the name descriptions for the duplicated terms
enrichViewNet:::manageNameDuplicationInEmap(clProfDF=clustData)
```

`manageQueryDuplicationInEmap` *Change query name when more than one queries have the same name*

Description

The function adds an unique ID at the name of the query when more than one query have the same name.

Usage

```
manageQueryDuplicationInEmap(queryList)
```

Arguments

`queryList`	a `list` containing the query names.

Value

a `list` with the query names modified for queries with duplicated names.

Author(s)

Astrid Deschênes

Examples

```
## List of query names with duplicated names
queryList <- list("parental_vs_DMSO", "rosa_vs_DMSO", "parental_vs_DMSO",
    "rosa_vs_DMSO", "parental_vs_Control", "rosa_vs_DMSO")

## Change the query names for the duplicated names
enrichViewNet:::manageQueryDuplicationInEmap(queryList=queryList)
```

`parentalNapaVsDMSODEG` *The result of a differential expression analysis done between napabucasin treated and DMSO control parental MiaPaCa2 cells. The cells were treated for 2 hours with 0.5 uM napabucasin. The protocol to generate the RNA-seq is described in Froeling F.E.M. et al 2019.*

Description

The object is a `data.frame` with 24184 rows and 4 columns. Each row correspond to a tested gene.

Usage

```
data(parentalNapaVsDMSODEG)
```

Format

a `data.frame` containing the results of a differential expression analysis between napabucasin treated and DMSO control parental MiaPaCa2 cells for all 24184 genes tested. The 4 columns are:

- `"EnsemblID"`: a `character` string representing the unique Ensembl identifier for the tested gene
- `"log2FoldChange"`: a `numeric` representing the expression difference (in log2FoldChange) between the napabucasin treatment and the DMSO control for the tested gene
- `"padj"`: a `numeric` representing the adjusted p-value associated to the difference in expression for the tested gene
- `"GeneName"`: a `character` string representing the name of the tested gene

Details

The differentially expressed genes between napabucasin-treated cells (0.5 uM) and DMSO as vehicle control are reprinted from Clinical Cancer Research, 2019, 25 (23), 7162–7174, Fieke E.M. Froeling, Manojit Mosur Swamynathan, Astrid Deschênes, Iok In Christine Chio, Erin Brosnan, Melissa A. Yao, Priya Alagesan, Matthew Lucito, Juying Li, An-Yun Chang, Lloyd C. Trotman, Pascal Belleau, Youngkyu Park, Harry A. Rogoff, James D. Watson, David A. Tuveson, Bioactivation of napabucasin triggers reactive oxygen species–mediated cancer cell death, with permission from AACR.

Value

a `data.frame` containing the results of a differential expression analysis between napabucasin treated and DMSO control parental MiaPaCa2 cells for all 24184 genes tested. The 4 columns are:

- `"EnsemblID"`: a `character` string representing the unique Ensembl identifier for the tested gene
- `"log2FoldChange"`: a `numeric` representing the expression difference (in log2FoldChange) between the napabucasin treatment and the DMSO control for the tested gene
- `"padj"`: a `numeric` representing the adjusted p-value associated to the difference in expression for the tested gene
- `"GeneName"`: a `character` string representing the name of the tested gene

Source

The original RNA-sequencing data is available at the Gene Expression Omnibus (GEO) under the accession number GSE135352.

See Also

- createNetwork for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- createEnrichMap for transforming functional enrichment results from gprofiler2 into an enrichment map

Examples

```
## Required library
library(gprofiler2)

## Loading data set containing the results of a differentially expressed
## analysis between 2-hour treatment with 0.5 uM napabucasin and
## DMSO vehicle control parental MiaPaCa2 cells
data(parentalNapaVsDMSODEG)

allGenes <- unique(parentalNapaVsDMSODEG$EnsemblID)

## Select the significantly differentially expressed genes
selection <- which(abs(parentalNapaVsDMSODEG$log2FoldChange) > 1 &
                           parentalNapaVsDMSODEG$padj < 0.05)

selectedGenes <- unique(parentalNapaVsDMSODEG$EnsemblID[selection])

## Run an enrichment analysis using WikiPathways dataset
gostres <- gost(query = list(parental_napa_vs_DMSO=selectedGenes),
    organism="hsapiens",
    correction_method = "g_SCS",
    sources=c("WP"), significant=TRUE, evcodes=TRUE,
    custom_bg=allGenes, exclude_iea=TRUE)
```

parentalNapaVsDMSOEnrichment

The result of an enrichment analysis has been done using the significantly differentially expressed genes between napabucasin treated and DMSO control parental MiaPaCa2 cells. The cells were treated for 2 hour with 0.5 uM napabucasin. The protocol to generate the RNA-seq is described in Froeling F.E.M. et al 2019.

Description

The enrichment analysis was done with gprofile2 package (Kolberg L et al 2020) with database version 'e109_eg56_p17_1d3191d' and g:SCS multiple testing correction method applying significance threshold of 0.05 (Raudvere U et al 2019). All tested genes were used as background.

Usage

```
data(parentalNapaVsDMSOEnrichment)
```

Format

a `list` created by gprofiler2 that contains the results from the enrichment analysis:

- `"result"`: a `data.frame` with the significantly enriched terms
- `"meta"`: a `list` with the meta-data information

Details

The object is a named `list` with 2 entries. The 'result' entry contains a `data.frame` with the enrichment analysis results and the 'meta' entry contains metadata information.

The dataset used for the enrichment analysis is associated to this publication:

Froeling F.E.M. et al.Bioactivation of Napabucasin Triggers Reactive Oxygen Species–Mediated Cancer Cell Death. Clin Cancer Res 1 December 2019; 25 (23): 7162–7174

The enrichment analysis has been done with gprofile2 package (Kolberg L et al 2020) with database version 'e109_eg56_p17_1d3191d' and g:SCS multiple testing correction method applying significance threshold of 0.05 (Raudvere U et al 2019). All tested genes were used as background.

Value

a `list` containing 2 entries:

- `"result"`: a `data.frame` with the significantly enriched terms
- `"meta"`: a `list` with the meta-data information

Source

The original RNA-sequencing data is available at the Gene Expression Omnibus (GEO) under the accession number GSE135352.

See Also

- createNetwork for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- createEnrichMap for transforming functional enrichment results from gprofiler2 into an enrichment map
- createEnrichMapAsIgraph for transforming functional enrichment results from gprofiler2 into an enrichment map in a igraph format

Examples

```
## Loading dataset containing the results of the enrichment analysis
## done on a differentially expressed
## analysis between 2-hour treatment with 0.5 uM napabucasin and
## DMSO vehicle control parental MiaPaCa2 cells
data(parentalNapaVsDMSOEnrichment)

## Create an enrichment map for the GO:CC terms
createEnrichMap(gostObject=parentalNapaVsDMSOEnrichment,
    query="parental_napa_vs_DMSO", source="GO:CC")
```

---

removeRootTerm *Remove root term if present in the list of selected terms*

Description

Remove root term if present in the list of selected terms

Usage

```
removeRootTerm(gostResult)
```

Arguments

gostResult	a data.frame containing the terms retained for the creation of the network.

Value

a data.frame of selected terms without the root term.

Author(s)

Astrid Deschênes

Examples

```
## Loading dataset containing result from an enrichment analysis done with
## gprofiler2
data(demoGOST)

## Only retained the WikiPathways results
results <- demoGOST$result[demoGOST$result$source == "WP", ]

## Remove WIKIPATHWAYS root term
enrichViewNet:::removeRootTerm(gostResult=results)
```

`rosaNapaVsDMSODEG` *The result of a differential expression analysis done between napabucasin treated and DMSO control MiaPaCa2 cells stably expressing the Rosa26 control vector. The cells were treated for 2 hours with 0.5 uM napabucasin. The protocol to generate the RNA-seq is described in Froeling F.E.M. et al 2019.*

Description

The object is a `data.frame` with 23542 rows and 4 columns. Each row correspond to a tested gene.

Usage

```
data(rosaNapaVsDMSODEG)
```

Format

a `data.frame` containing the results of a differential expression analysis between napabucasin treated and DMSO control MiaPaCa2 cells stably expressing the Rosa26 control vector for all 23542 genes tested. The 4 columns are:

- `"EnsemblID"`: a `character` string representing the unique Ensembl identifier for the tested gene
- `"log2FoldChange"`: a `numeric` representing the expression difference (in log2FoldChange) between the napabucasin treatment and the DMSO control for the tested gene
- `"padj"`: a `numeric` representing the adjusted p-value associated to the difference in expression for the tested gene
- `"GeneName"`: a `character` string representing the name of the tested gene

Details

The differentially expressed genes between napabucasin-treated cells (0.5 uM) and DMSO as vehicle control are reprinted from Clinical Cancer Research, 2019, 25 (23), 7162–7174, Fieke E.M. Froeling, Manojit Mosur Swamynathan, Astrid Deschênes, Iok In Christine Chio, Erin Brosnan, Melissa A. Yao, Priya Alagesan, Matthew Lucito, Juying Li, An-Yun Chang, Lloyd C. Trotman, Pascal Belleau, Youngkyu Park, Harry A. Rogoff, James D. Watson, David A. Tuveson, Bioactivation of napabucasin triggers reactive oxygen species–mediated cancer cell death, with permission from AACR.

Value

a `data.frame` containing the results of a differential expression analysis between napabucasin treated and DMSO control MiaPaCa2 cells stably expressing the Rosa26 control vector for all 23542 genes tested. The 4 columns are:

- `"EnsemblID"`: a `character` string representing the unique Ensembl identifier for the tested gene
- `"log2FoldChange"`: a `numeric` representing the expression difference (in log2FoldChange) between the napabucasin treatment and the DMSO control for the tested gene
- `"padj"`: a `numeric` representing the adjusted p-value associated to the difference in expression for the tested gene
- `"GeneName"`: a `character` string representing the name of the tested gene

Source

The original RNA-sequencing data is available at the Gene Expression Omnibus (GEO) under the accession number GSE135352.

See Also

- createNetwork for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- createEnrichMap for transforming functional enrichment results from gprofiler2 into an enrichment map

Examples

```
## Required library
library(gprofiler2)

## Loading dataset containing the results of a differentially expressed
## analysis between 2-hour treatment with 0.5 uM napabucasin and
## DMSO vehicle control MiaPaCa2 cells stably expressing the
## Rosa26 control vector
data(rosaNapaVsDMSODEG)

allGenes <- unique(rosaNapaVsDMSODEG$EnsemblID)

## Select the significantly differentially expressed genes
```

```
selection <- which(abs(rosaNapaVsDMSODEG$log2FoldChange) > 1 &
                           rosaNapaVsDMSODEG$padj < 0.05)

selectedGenes <- unique(rosaNapaVsDMSODEG$EnsemblID[selection])

## Run an enrichment analysis using Transfac dataset (transcription factor)
gostres <- gost(query = list(rosa_napa_vs_DMSO=selectedGenes),
    organism="hsapiens",
    correction_method = "g_SCS",
    sources=c("TF"), significant=TRUE, evcodes=TRUE,
    custom_bg=allGenes, exclude_iea=TRUE)
```

`rosaNapaVsDMSOEnrichment`

The result of an enrichment analysis that has been done using the significantly differentially expressed genes between napabucasin treated and DMSO control MiaPaCa2 cells stably expressing the Rosa26 control vector. The cells were treated for 2 hour with 0.5 uM napabucasin. The protocol to generate the RNA-seq is described in Froeling F.E.M. et al 2019.

Description

The enrichment analysis was done with gprofile2 package (Kolberg L et al 2020) with database version 'e109_eg56_p17_1d3191d' and g:SCS multiple testing correction method applying significance threshold of 0.05 (Raudvere U et al 2019). All tested genes were used as background.

Usage

```
data(rosaNapaVsDMSOEnrichment)
```

Format

a `list` created by gprofiler2 that contains the results from the enrichment analysis:

- `"result"`: a `data.frame` with the significantly enriched terms
- `"meta"`: a `list` with the meta-data information

Details

The object is a named `list` with 2 entries. The 'result' entry contains a `data.frame` with the enrichment analysis results and the 'meta' entry contains metadata information.

The dataset used for the enrichment analysis is associated to this publication:

Froeling F.E.M. et al.Bioactivation of Napabucasin Triggers Reactive Oxygen Species–Mediated Cancer Cell Death. Clin Cancer Res 1 December 2019; 25 (23): 7162–7174

The enrichment analysis has been done with gprofile2 package (Kolberg L et al 2020) with database version 'e109_eg56_p17_1d3191d' and g:SCS multiple testing correction method applying significance threshold of 0.05 (Raudvere U et al 2019). All tested genes were used as background.

Value

a `list` created by gprofiler2 that contains the results from the enrichment analysis:

- `"result"`: a `data.frame` with the significantly enriched terms
- `"meta"`: a `list` with the meta-data information

Source

The original RNA-sequencing data is available at the Gene Expression Omnibus (GEO) under the accession number GSE135352.

See Also

- createNetwork for transforming functional enrichment results from gprofiler2 into a Cytoscape network
- createEnrichMap for transforming functional enrichment results from gprofiler2 into an enrichment map

Examples

```
## Loading dataset containing the enrichment analysis done on the
## differentially expressed analysis between 2-hour treatment with 0.5 uM
## napabucasin and DMSO control MiaPaCa2 cells stably expressing
## the Rosa26 control vector
data(rosaNapaVsDMSOEnrichment)

## Create an enrichment map for the KEGG terms
createEnrichMap(gostObject=rosaNapaVsDMSOEnrichment,
    query="rosa_napa_vs_DMSO", source="KEGG")
```

---

similarityJaccard *Calculate the Jaccard similarity coefficient for the terms present in a data frame*

Description

The function calculates the Jaccard similarity coefficient for the enriched terms present i a data frame.

Usage

```
similarityJaccard(resultDF)
```

Arguments

resultDF	a data.frame containing the enrichment terms that are going to be graphed as an enrichment map. The data.frame must contain the geneID and Description columns.

Value

a matrix with the lower triangle section containing the Jaccard similarity coefficients as numeric. The upper triangle section is filled with NA as well as the diagonal.

Author(s)

Astrid Deschênes

Examples

```
## Data frame with the enriched terms
resultData <- data.frame(
    ID=c("REAC:R-HSA-9031628", "REAC:R-HSA-9648895", "REAC:R-HSA-9614085",
            "REAC:R-HSA-166520"),
    Description=c("NGF-stimulated transcription",
            "Response of EIF2AK1 (HRI) to heme deficiency",
            "FOXO-mediated transcription", "Signaling by NTRKs"),
    geneID=c(paste0("ENSG00000120738/ENSG00000122877/ENSG00000125740/",
        "ENSG00000135625/ENSG00000170345/ENSG00000171223/ENSG00000173334/",
        "ENSG00000179388/ENSG0000019857"),
        paste0("ENSG00000087074/ENSG00000128272/ENSG00000128965/",
        "ENSG00000162772/ENSG00000172216/ENSG00000175197"),
        paste0("ENSG00000105327/ENSG00000113916/ENSG00000124762/",
        "ENSG00000133639/ENSG00000136826/ENSG00000153094/ENSG00000175197"),
        paste0("ENSG00000105327/ENSG00000113916/ENSG00000124762/",
        "ENSG00000133639/ENSG00000136826/ENSG00000153094/ENSG0000017519/",
        "ENSG00000120738/ENSG00000122877/ENSG00000125740/ENSG00000135625/",
        "ENSG00000170345/ENSG00000171223/ENSG00000173334/ENSG00000179388/",
        "ENSG00000198576")))

## Calculate the Jaccard similarity coefficient for all pairwise
## term combination
enrichViewNet:::similarityJaccard(resultDF=resultData)
```

---

validateCreateEnrichMapArguments

Validate arguments passed to createEnrichMap() function

Description

Validate the arguments passed to createEnrichMap() function. First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapArguments(
  gostObject,
  query,
  source,
  termIDs,
  removeRoot,
  showCategory,
  categoryLabel,
  categoryNode,
  line
)
```

Arguments

gostObject	a list created by gprofiler2 that contains the results from an enrichment analysis.
query	a character string representing the name of the query that is going to be used to generate the graph. The query must exist in the gostObject object.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.
termIDs	a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present).
showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).
categoryNode	a positive numeric representing he amount by which plotting category nodes should be scaled relative to the default (1).
line	a non-negative numeric representing the scale of line width.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(demoGOST)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapArguments(gostObject=demoGOST,
    query="query_1", source="GO:BP", termIDs=NULL, removeRoot=FALSE,
    showCategory=20, categoryLabel=1.1, categoryNode=1, line=1.2)
```

---

validateCreateEnrichMapAsIgraphArg

Validate arguments passed to createEnrichMapAsIgraph() function

---

Description

Validate the arguments passed to createEnrichMapAsIgraph() function.First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapAsIgraphArg(
  gostObject,
  query,
  source,
  termIDs,
  removeRoot,
  showCategory,
  similarityCutOff
)
```

Arguments

gostObject	a list created by gprofiler2 that contains the results from an enrichment analysis.
query	a character string representing the name of the query that is going to be used to generate the graph. The query must exist in the gostObject object.

source — a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.

termIDs — a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".

removeRoot — a logical that specified if the root terms of the selected source should be removed (when present).

showCategory — a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.

similarityCutOff — a positive numeric between 0 and 1 indicating the minimum level of similarity between two terms to have an edge linking the terms.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(demoGOST)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapAsIgraphArg(
    gostObject=demoGOST, query="query_1", source="GO:BP", termIDs=NULL,
    removeRoot=FALSE, showCategory=20, similarityCutOff=0.5)
```

---

validateCreateEnrichMapMultiBasicArgs

Validate arguments passed to createEnrichMapMultiBasic() function

---

Description

Validate the arguments passed to createEnrichMapMultiBasic() function. First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiBasicArgs(
  gostObjectList,
  queryList,
  source,
  termIDs,
  removeRoot,
  showCategory,
  categoryLabel,
  categoryNode,
  line
)
```

Arguments

gostObjectList	a list of gprofiler2 objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the queryList parameter.
queryList	a list of character strings representing the names of the queries that are going to be used to generate the graph. The query names must exist in the associated gostObjectList objects and follow the same order. The number of entries must correspond to the number of entries for the gostObjectList parameter.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.
termIDs	a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present).
showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).
categoryNode	a positive numeric representing he amount by which plotting category nodes should be scaled relative to the default (1).
line	a non-negative numeric representing the scale of line width.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiBasicArgs(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                            rosaNapaVsDMSOEnrichment),
    queryList=list("parental_napa_vs_DMSO", "rosa_napa_vs_DMSO"),
    source="GO:BP", termIDs=NULL, removeRoot=FALSE,
    showCategory=20, categoryLabel=1.1, categoryNode=1, line=1.2)
```

---

validateCreateEnrichMapMultiBasicAsIgraphArgs

Validate arguments passed to createEnrichMapMultiBasic() function

---

Description

Validate the arguments passed to createEnrichMapMultiBasic() function. First, the object containing the enrichment results must correspond to a object created by `gprofiler2` software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiBasicAsIgraphArgs(
  gostObjectList,
  queryList,
  source,
  termIDs,
  removeRoot,
  showCategory,
  similarityCutOff
)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.

queryList — a list of character strings representing the names of the queries that are going to be used to generate the graph. The query names must exist in the associated gostObjectList objects and follow the same order. The number of entries must correspond to the number of entries for the gostObjectList parameter.

source — a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.

termIDs — a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".

removeRoot — a logical that specified if the root terms of the selected source should be removed (when present).

showCategory — a positive integer representing the maximum number of terms to display. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed.

similarityCutOff — a positive numeric between 0 and 1 indicating the minimum level of similarity between two terms to have an edge linking the terms.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiBasicAsIgraphArgs(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                            rosaNapaVsDMSOEnrichment),
    queryList=list("parental_napa_vs_DMSO", "rosa_napa_vs_DMSO"),
    source="GO:BP", termIDs=NULL, removeRoot=FALSE,
    showCategory=20, similarityCutOff=0.3)
```

validateCreateEnrichMapMultiBasicGOSTArgs *Validate arguments passed to createEnrichMapMultiBasic() function*

Description

Validate the arguments passed to createEnrichMapMultiBasic() function. First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiBasicGOSTArgs(
  gostObjectList,
  queryList,
  source,
  termIDs,
  removeRoot
)
```

Arguments

gostObjectList	a list of gprofiler2 objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the queryList parameter.
queryList	a list of character strings representing the names of the queries that are going to be used to generate the graph. The query names must exist in the associated gostObjectList objects and follow the same order. The number of entries must correspond to the number of entries for the gostObjectList parameter.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.
termIDs	a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present).

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiBasicGOSTArgs(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                        rosaNapaVsDMSOEnrichment),
    queryList=list("parental_napa_vs_DMSO", "rosa_napa_vs_DMSO"),
    source="GO:BP", termIDs=NULL, removeRoot=TRUE)
```

---

validateCreateEnrichMapMultiComplexArg

Validate arguments passed to createEnrichMapMultiComplex() function

---

Description

Validate the arguments passed to createEnrichMapMultiComplex() function. First, the object containing the enrichment results must correspond to a object created by `gprofiler2` software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiComplexArg(
  gostObjectList,
  queryInfo,
  showCategory,
  categoryLabel,
  categoryNode,
  line
)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.
`queryInfo`	a `data.frame` contains one row per group being displayed. The number of rows must correspond to the number of entries for the `gostObjectList` parameter. The mandatory columns are:

- queryName: a character string representing the name of the query retained for this group). The query names must exist in the associated gostObjectList objects and follow the same order.
- source: a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
- removeRoot: a logical that specified if the root terms of the selected source should be removed (when present).
- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

showCategory — a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.

categoryLabel — a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).

categoryNode — a positive numeric representing he amount by which plotting category nodes should be scaled relative to the default (1).

line — a non-negative numeric representing the scale of line width.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
    "rosa_napa_vs_DMSO"), source=c("KEGG", "WP"), removeRoot=c(TRUE, TRUE),
    termIDs=c("", ""), groupName=c("parental - KEGG", "rosa - WP"),
    stringsAsFactors=FALSE)

## Check that all arguments are valid
```

```
enrichViewNet:::validateCreateEnrichMapMultiComplexArg(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                        rosaNapaVsDMSOEnrichment),
    queryInfo=queryDataFrame,
    showCategory=20, categoryLabel=1.1, categoryNode=1, line=1.2)
```

`validateCreateEnrichMapMultiComplexAsIgraphArg` *Validate arguments passed to validateCreateEnrichMapMultiComplex() function*

Description

Validate the arguments passed to validateCreateEnrichMapMultiComplex() function. First, the object containing the enrichment results must correspond to a object created by `gprofiler2` software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiComplexAsIgraphArg(
  gostObjectList,
  queryInfo,
  showCategory,
  similarityCutOff = similarityCutOff
)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.
`queryInfo`	a `data.frame` contains one row per group being displayed. The number of rows must correspond to the number of entries for the `gostObjectList` parameter. The mandatory columns are: • `queryName`: a `character` string representing the name of the query retained for this group). The query names must exist in the associated `gostObjectList` objects and follow the same order. • `source`: a `character` string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the `termIDs` parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".

- removeRoot: a logical that specified if the root terms of the selected source should be removed (when present).
- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

showCategory a positive integer or NULL. If a integer, the first n terms will be displayed. If NULL, all terms will be displayed.

similarityCutOff
a positive numeric, larger than zero and small than 1 that represent the minimum similarity level between two nodes (terms) to be linked by an edge.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
    "rosa_napa_vs_DMSO"), source=c("KEGG", "WP"), removeRoot=c(TRUE, TRUE),
    termIDs=c("", ""), groupName=c("parental - KEGG", "rosa - WP"),
    stringsAsFactors=FALSE)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiComplexAsIgraphArg(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                        rosaNapaVsDMSOEnrichment),
    queryInfo=queryDataFrame, showCategory=20, similarityCutOff=0.2)
```

---

validateCreateEnrichMapMultiComplexGostPartOne

Validate arguments related to GOST passed to validateCreateEnrichMapMultiComplexArg() function

---

Description

Validate the arguments passed to validateCreateEnrichMapMultiComplexArg() function. First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiComplexGostPartOne(gostObjectList, queryInfo)
```

Arguments

gostObjectList	a list of gprofiler2 objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the queryList parameter.
queryInfo	a data.frame contains one row per group being displayed. The number of rows must correspond to the number of entries for the gostObjectList parameter. The mandatory columns are:

- queryName: a character string representing the name of the query retained for this group). The query names must exist in the associated gostObjectList objects and follow the same order.
- source: a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID".
- removeRoot: a logical that specified if the root terms of the selected source should be removed (when present).
- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
    "rosa_napa_vs_DMSO"), source=c("KEGG", "WP"), removeRoot=c(TRUE, TRUE),
    termIDs=c("", ""), groupName=c("parental - KEGG", "rosa - WP"),
```

```
    stringsAsFactors=FALSE)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiComplexGostPartOne(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                        rosaNapaVsDMSOEnrichment),
    queryInfo=queryDataFrame)
```

`validateCreateEnrichMapMultiComplexGostPartTwo` *Validate arguments related to GOST passed to validateCreateEnrichMapMultiComplexArg() function*

Description

Validate the arguments passed to validateCreateEnrichMapMultiComplexArg() function. First, the object containing the enrichment results must correspond to a object created by `gprofiler2` software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateEnrichMapMultiComplexGostPartTwo(gostObjectList, queryInfo)
```

Arguments

`gostObjectList`	a `list` of `gprofiler2` objects that contain the results from an enrichment analysis. The list must contain at least 2 entries. The number of entries must correspond to the number of entries for the `queryList` parameter.
`queryInfo`	a `data.frame` contains one row per group being displayed. The number of rows must correspond to the number of entries for the `gostObjectList` parameter. The mandatory columns are: • `queryName`: a `character` string representing the name of the query retained for this group). The query names must exist in the associated `gostObjectList` objects and follow the same order. • `source`: a `character` string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the `termIDs` parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways. Default: "TERM_ID". • `removeRoot`: a `logical` that specified if the root terms of the selected source should be removed (when present).

- termIDs: a character strings that contains the term IDS retained for the creation of the network separated by a comma ',' when the "TERM_ID" source is selected. Otherwise, it should be a empty string ("").
- groupName: a character strings that contains the name of the group to be shown in the legend. Each group has to have a unique name.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(parentalNapaVsDMSOEnrichment)
data(rosaNapaVsDMSOEnrichment)

queryDataFrame <- data.frame(queryName=c("parental_napa_vs_DMSO",
    "rosa_napa_vs_DMSO"), source=c("KEGG", "WP"), removeRoot=c(TRUE, TRUE),
    termIDs=c("", ""), groupName=c("parental - KEGG", "rosa - WP"),
    stringsAsFactors=FALSE)

## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapMultiComplexGostPartTwo(
    gostObjectList=list(parentalNapaVsDMSOEnrichment,
                        rosaNapaVsDMSOEnrichment),
    queryInfo=queryDataFrame)
```

---

validateCreateEnrichMapSubSectionArguments

Validate common arguments passed to createEnrichMap() and createEnrichMapMultiBasic() functions

---

Description

Validate the common arguments passed to createEnrichMap() and createEnrichMapMultiBasic() functions.

Usage

```
validateCreateEnrichMapSubSectionArguments(
  showCategory,
  categoryLabel,
  categoryNode,
  line
)
```

Arguments

showCategory	a positive integer or a vector of characters representing terms. If a integer, the first n terms will be displayed. If vector of terms, the selected terms will be displayed.
categoryLabel	a positive numeric representing the amount by which plotting category nodes label size should be scaled relative to the default (1).
categoryNode	a positive numeric representing he amount by which plotting category nodes should be scaled relative to the default (1).
line	a non-negative numeric representing the scale of line width.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Check that all arguments are valid
enrichViewNet:::validateCreateEnrichMapSubSectionArguments(
    showCategory=20, categoryLabel=1.1, categoryNode=1,
    line=0.5)
```

---

validateCreateNetworkArguments

Validate arguments passed to creatNetwork() function

Description

Validate the arguments passed to creatNetwork() function. First, the object containing the enrichment results must correspond to a object created by gprofiler2 software. Second, the selected source must at least have one enriched term in the results. Then, if the source is 'TERM_ID', the listed terms must be present in the enrichment results.

Usage

```
validateCreateNetworkArguments(
  gostObject,
  source,
  termIDs,
  removeRoot,
  query,
  title,
```

```
  collection,
  fileName
)
```

Arguments

gostObject	a list created by gprofiler2 that contains the results from an enrichment analysis.
source	a character string representing the selected source that will be used to generate the network. To hand-pick the terms to be used, "TERM_ID" should be used and the list of selected term IDs should be passed through the termIDs parameter. The possible sources are "GO:BP" for Gene Ontology Biological Process, "GO:CC" for Gene Ontology Cellular Component, "GO:MF" for Gene Ontology Molecular Function, "KEGG" for Kegg, "REAC" for Reactome, "TF" for TRANSFAC, "MIRNA" for miRTarBase, "CORUM" for CORUM database, "HP" for Human phenotype ontology and "WP" for WikiPathways.
termIDs	a vector of character strings that contains the term IDs retained for the creation of the network. This parameter is only used when source is set to "TERM_ID".
removeRoot	a logical that specified if the root terms of the selected source should be removed (when present).
query	a character string that specified the retained query to generate the network or NULL.
title	a character string representing the name assigned to the network.
collection	a character string representing the collection name assigned to the network.
fileName	a character string representing the name of the CX JSON file that is created when Cytoscape is not running. The name must have a '.cx' extension.

Value

TRUE when all arguments are valid

Author(s)

Astrid Deschênes

Examples

```
## Load the result of an enrichment analysis done with gprofiler2
data(demoGOST)

## Check that all arguments are valid
enrichViewNet:::validateCreateNetworkArguments(gostObject=demoGOST,
    source="GO:BP", termIDs=NULL, removeRoot=FALSE, query=NULL,
    title="Network graph Test",
    collection="test collection", fileName="test.cx")
```

Index

∗ **datasets**
demoGOST, 28
parentalNapaVsDMSODEG, 37
parentalNapaVsDMSOEnrichment, 39
rosaNapaVsDMSODEG, 41
rosaNapaVsDMSOEnrichment, 43
∗ **internal**
createBasicEmap, 4
createBasicEmapAsIgraph, 5
createCompareResultDataFrame, 6
createCXJSONForCytoscape, 8
createMetaDataSectionCXJSON, 22
createMultiEmap, 23
createMultiEmapAsIgraph, 24
createNetworkForCytoscape, 27
extractInformationWhenIntersection, 29
extractInformationWhenNoIntersection, 30
extractNodesAndEdgesInformation, 31
filterResults, 32
formatInformationForCXJSON, 33
isCytoscapeRunning, 34
manageNameDuplicationInEmap, 35
manageQueryDuplicationInEmap, 36
removeRootTerm, 40
similarityJaccard, 44
validateCreateEnrichMapArguments, 45
validateCreateEnrichMapAsIgraphArg, 47
validateCreateEnrichMapMultiBasicArgs, 48
validateCreateEnrichMapMultiBasicAsIgraphArgs, 50
validateCreateEnrichMapMultiBasicGOSTArgs, 52
validateCreateEnrichMapMultiComplexArg, 53
validateCreateEnrichMapMultiComplexGostPartOne, 56
validateCreateEnrichMapMultiComplexGostPartTwo, 58
validateCreateEnrichMapSubSectionArguments, 59
validateCreateNetworkArguments, 60
∗ **package**
enrichViewNet-package, 3

createBasicEmap, 4
createBasicEmapAsIgraph, 5
createCompareResultDataFrame, 6
createCXJSONForCytoscape, 8
createEnrichMap, *3*, 9, *29*, *38*, *40*, *42*, *44*
createEnrichMapAsIgraph, *3*, 10, *40*
createEnrichMapMultiBasic, 12
createEnrichMapMultiBasicAsIgraph, 14
createEnrichMapMultiComplex, *3*, 17
createEnrichMapMultiComplexAsIgraph, *3*, 19
createMetaDataSectionCXJSON, 22
createMultiEmap, 23
createMultiEmapAsIgraph, 24
createNetwork, *3*, 25, *28*, *29*, *38*, *40*, *42*, *44*
createNetworkForCytoscape, 27

demoGOST, 28

emapplot, *4*, *10*, *13*, *18*, *23*
enrichViewNet *(*enrichViewNet-package*)*, 3
enrichViewNet-package, 3
extractInformationWhenIntersection, 29
extractInformationWhenNoIntersection, 30
extractNodesAndEdgesInformation, 31

filterResults, 32
formatInformationForCXJSON, 33

isCytoscapeRunning, 34

manageNameDuplicationInEmap, 35
manageQueryDuplicationInEmap, 36

parentalNapaVsDMSODEG, 37
parentalNapaVsDMSOEnrichment, 39

removeRootTerm, 40
rosaNapaVsDMSODEG, 41
rosaNapaVsDMSOEnrichment, 43

similarityJaccard, 44

validateCreateEnrichMapArguments, 45
validateCreateEnrichMapAsIgraphArg, 47
validateCreateEnrichMapMultiBasicArgs, 48
validateCreateEnrichMapMultiBasicAsIgraphArgs, 50
validateCreateEnrichMapMultiBasicGOSTArgs, 52
validateCreateEnrichMapMultiComplexArg, 53
validateCreateEnrichMapMultiComplexAsIgraphArg, 55
validateCreateEnrichMapMultiComplexGostPartOne, 56
validateCreateEnrichMapMultiComplexGostPartTwo, 58
validateCreateEnrichMapSubSectionArguments, 59
validateCreateNetworkArguments, 60