

Package: miREA (via r-universe)

October 16, 2026

Title Network-Based Edge-Centric MicroRNA-Oriented Enrichment Analysis

Version 0.99.2

Description miREA performs microRNA-oriented enrichment analysis by modeling miRNA-gene interaction (MGI) networks directly, rather than collapsing them to miRNA- or gene-level sets. It implements four node-based methods and five edge-based methods spanning over-representation, scoring-based, topology-aware, and network-propagation approaches, using expression-informed MGI edge scores to characterize regulatory strength. The package also provides heatmap, Sankey, and summary visualizations to explore miRNA-gene-pathway regulatory mechanisms discovered by the enrichment results.

License Artistic-2.0

URL <https://github.com/laixn/miREA>

BugReports <https://github.com/laixn/miREA/issues>

biocViews miRNA, GeneExpression, GeneSetEnrichment, Network, NetworkEnrichment, Pathways, SystemsBiology

Encoding UTF-8

LazyData false

Roxygen list(markdown = FALSE)

Depends R (>= 4.5.0)

Imports BiocGenerics, ComplexHeatmap, circlize, clusterProfiler, data.table, dplyr, ggalluvial, ggnewscale, ggplot2, grDevices, graphics, grid, igraph, Matrix, methods, miRBaseConverter, parallel, patchwork, reshape2, RhpCBLASctl, scales, stats, tibble, utils, withr

Suggests BiocStyle, GSEABase, knitr, rmarkdown, SummarizedExperiment, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev perl zlib1g-dev

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

Date/Publication 2026-10-16 18:34:01 UTC

RemoteUrl <https://github.com/BiocStaging/miREA>

RemoteRef HEAD

RemoteSha 04147f83822ac5d6acb9ae7f559500f319ba0e3e

Contents

default_GGI_source	2
default_method_list	3
default_pAdjMethod	4
default_pathway_list	4
default_pvalueType	5
default_scoreFun	5
default_TP_cancer	6
Edge_Network	6
Edge_ORA	8
Edge_Score1D	9
Edge_Score2D	10
Edge_Topology	12
getAllInputData	14
MiR_ORA	16
MiR_Score	17
miREA	18
miREA_example_data	20
miREAResult-class	22
plotHeatmap	23
plotHeatmapSankey	25
plotSummary	27
TG_ORA	28
TG_Score	29
Index	31

default_GGI_source	<i>Supported gene-gene interaction sources</i>
--------------------	--

Description

Character vector of the background gene-gene interaction (GGI) sources supported by the ‘GGI_source’ argument of [getAllInputData()].

Usage

```
default_GGI_source
```

Format

A character vector of length 2.

Value

A character vector of length 2.

Examples

```
default_GGI_source
```

default_method_list	<i>Supported enrichment methods</i>
---------------------	-------------------------------------

Description

Character vector of the nine node-based and edge-based enrichment methods supported by [miREA()] and [getAllInputData()].

Usage

```
default_method_list
```

Format

A character vector of length 9.

Value

A character vector of length 9.

Examples

```
default_method_list
```

default_pAdjMethod	<i>Supported p-value adjustment methods</i>
--------------------	---

Description

Character vector of the multiple-testing correction methods supported by the ‘pAdjMethod’ argument of [miREA()] and the individual enrichment method functions (see [stats::p.adjust()]).

Usage

```
default_pAdjMethod
```

Format

A character vector of length 8.

Value

A character vector of length 8.

Examples

```
default_pAdjMethod
```

default_pathway_list	<i>Supported predefined pathway sets</i>
----------------------	--

Description

Character vector naming the pathway sets shipped with the miREA manuscript’s companion data deposit.

Usage

```
default_pathway_list
```

Format

A character vector of length 3.

Value

A character vector of length 3.

Examples

```
default_pathway_list
```

default_pvalueType	<i>Supported GSEA score types</i>
--------------------	-----------------------------------

Description

Character vector of the score types supported by the 'pvalueType' argument of the scoring-based enrichment methods ('TG_Score', 'MiR_Score', 'Edge_Score1D', 'Edge_Score2D', 'Edge_Topology').

Usage

```
default_pvalueType
```

Format

A character vector of length 3.

Value

A character vector of length 3.

Examples

```
default_pvalueType
```

default_scoreFun	<i>Supported score transformations</i>
------------------	--

Description

Character vector of the score transformations supported by the 'scoreFun' argument of [getAllInputData()].

Usage

```
default_scoreFun
```

Format

A character vector of length 4.

Value

A character vector of length 4.

Examples

```
default_scoreFun
```

default_TP_cancer	<i>Cancer types used for the TP pathway set</i>
-------------------	---

Description

Character vector of TCGA cancer-type abbreviations used to build the ""TP"" (cancer-specific) pathway set in [default_pathway_list].

Usage

```
default_TP_cancer
```

Format

A character vector of length 16.

Value

A character vector of length 16.

Examples

```
default_TP_cancer
```

Edge_Network	<i>Edge-Network (MGI Edge Network Enrichment Analysis)</i>
--------------	--

Description

Perform miRNA-gene interaction (MGI) network enrichment analysis, based on Random Walk with Restart (RWR), by evaluating how relevant the differentially expressed MGI set is to the pathway MGI set.

Usage

```
Edge_Network(
  DEMGI,
  pathway_gene,
  pathway_MGI,
  pathway_GGI,
  MGI,
  GGI,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL,
```

```

    iter = 1000,
    ncores = NULL,
    seed = NULL
  )

```

Arguments

DEMRI	A character vector contains a list of interested MGIs in the format of [miRNA]:[gene].
pathway_gene	A two-column dataframe, including pathway and gene. Equals to input_data\$pathway\$TG.
pathway_MGI	A three-column dataframe, including pathway, miRNA, gene. Equals to input_data\$pathway\$Edge.
pathway_GGI	A three column dataframe, including pathway, from, to. Equals to input_data\$pathway_GGI.
MGI	A two-column dataframe, including MGIs across all DEMGIs and pathways built from background.
GGI	A two-column dataframe, including GGIs across all DEMGIs and pathways, including internal GGIs and GGIs linking different pathways and DEMGIs.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs less than the minSize threshold.
maxSize	The maximum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs more than the maxSize threshold.
iter	Number of iterations for permutation test
ncores	Number of cores for permutation test (parallel computing random RWR procedure).
seed	Optional integer seed for the permutation test. If NULL (the default), no internal seed is set and the permutation draws depend on the caller's current random state. If supplied, the permutation is made reproducible via [withr::with_seed()], which does not alter the caller's global random state.

Value

A dataframe containing the enrichment result for Edge-Network.

Examples

```

data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata)
data(toy_gene_mat); data(toy_mir_mat)
data(toy_background_MGI); data(toy_background_GGI)
input_data <- getAllInputData(
  methods = "Edge_Network", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI,

```

```

background_GGI = toy_background_GGI, GGI_source = "Omnipath",
gene_mat = toy_gene_mat, mir_mat = toy_mir_mat
)
res <- Edge_Network(
  DEMGI = input_data$data$Edge_Network, pathway_gene = input_data$pathway$TG,
  pathway_MGI = input_data$pathway$Edge, pathway_GGI = input_data$pathway_GGI,
  MGI = input_data$MGI, GGI = input_data$GGI,
  minSize = 2, maxSize = 5000, iter = 20, ncores = 1, pvalueCutoff = 1
)
head(res)

```

Edge_ORA	<i>Edge-ORA (MiRNA-Gene Interaction Edge Over-Representation Analysis)</i>
----------	--

Description

Perform miRNA-gene interaction (MGI) enrichment analysis based on the overlap between differentially expressed MGI and pathway MGI set. Pathway MGI set was generated by linked pathway gene set and background MGI list.

Usage

```

Edge_ORA(
  DEMGI,
  background = NULL,
  pathway,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)

```

Arguments

DEMRI	A character vector contains selected differentially expressed miRNA-gene interactions (MGI).
background	A character vector contains background MGIs in the format of [miRNA]:[gene].
pathway	A three-column dataframe contains pathway miRNA gene.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs less than the minSize threshold.

maxSize The maximum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs more than the maxSize threshold.

Value

A dataframe containing the enrichment result for Edge-ORA.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "Edge_ORA", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI,
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat
)
background <- unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
res <- Edge_ORA(
  DEMGI = input_data$data$Edge_ORA, background = background,
  pathway = input_data$pathway$Edge, pvalueCutoff = 1
)
head(res)
```

Edge_Score1D	<i>Edge-Score1D (MiRNA-Gene Interaction Edge Score-based Analysis)</i>
--------------	--

Description

Perform miRNA-gene interaction (MGI) enrichment analysis based on whether the pathway MGI are located unexpectedly forward towards the ranked MGI list.

Usage

```
Edge_Score1D(
  MGIList,
  pathway,
  pvalueType = "neg",
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)
```

Arguments

MGIList A named vector shows the ranked score of MGIs.

pathway A three-column dataframe contains pathway, miRNA, gene.

pvalueType	How to calculate p value. See default_pvalueType for all possible options. Default pvalueType is "pos". The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment). The "std" is used when interested in both positive and negative scores.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs less than the minSize threshold.
maxSize	The maximum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs more than the maxSize threshold.

Value

A dataframe containing the enrichment result for Edge-Score1D.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "Edge_Score1D", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI,
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat, scoreFun = "rank"
)
res <- Edge_Score1D(
  MGIList = input_data$data$Edge_Score1D, pathway = input_data$pathway$Edge,
  pvalueType = "neg", pvalueCutoff = 1, minSize = 2, maxSize = 5000
)
head(res)
```

Edge_Score2D	<i>Edge-Score2D (MiRNA-Gene Interaction 2D Edge Score-based Analysis)</i>
--------------	---

Description

Perform miRNA-gene interaction (MGI) enrichment analysis based on a two-dimensional edge score (expression correlation and differential-expression-informed ratio), using a permutation test against randomized correlation values.

Usage

```
Edge_Score2D(
  MGI2D,
  pathway,
  background = NULL,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL,
  iter = 1000,
  ncores = NULL,
  seed = NULL
)
```

Arguments

MGI2D	A five-column dataframe, including MGI miRNA gene cor(x) ranknormratio(y).
pathway	A three-column dataframe contains pathway miRNA gene.
background	A character vector contains background MGIs in the format of [miRNA]:[gene].
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs less than the minSize threshold.
maxSize	The maximum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs more than the maxSize threshold.
iter	Number of iterations for permutation test
ncores	Number of cores for calculating randomized r during the permutation test.
seed	Optional integer seed for the permutation test. If NULL (the default), no internal seed is set and the permutation draws depend on the caller's current random state. If supplied, the permutation is made reproducible via [withr::with_seed()], which does not alter the caller's global random state.

Value

a dataframe contains enrichment result for Edge-Score2D

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "Edge_Score2D", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
```

```

gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI,
gene_mat = toy_gene_mat, mir_mat = toy_mir_mat, scoreFun = "rank"
)
background <- unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
res <- Edge_Score2D(
  MGI2D = input_data$data$Edge_Score2D, pathway = input_data$pathway$Edge,
  background = background, iter = 100, ncores = 1, pvalueCutoff = 1
)
head(res)

```

Edge_Topology

Edge-Topology (MGI Edge Pathway Topology Enrichment Analysis)

Description

Perform miRNA-gene interaction (MGI) network enrichment analysis, based on internal pathway topological importance and edge score importance, by evaluating whether the MGIs with stronger regulatory strength also have more topological importance inside the pathway.

Usage

```

Edge_Topology(
  MGI,
  pvalueType = "neg",
  background,
  pathway,
  pathway_GGI,
  dict_MGI,
  iter = 1000,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL,
  ncores = NULL,
  seed = NULL
)

```

Arguments

MGI	A four-column dataframe, including MGI miRNA gene strength.
pvalueType	How to calculate p value. See default_pvalueType for all possible options. Default pvalueType is "pos". The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment). The "std" is used when interested in both positive and negative scores.
background	A character vector contains background MGI in the format of [miRNA]:[gene].

pathway	A three-column dataframe, including pathway, miRNA, gene. Equals to input_data\$pathway\$Edge.
pathway_GGI	A three column dataframe, including pathway, from, to. Equals to input_data\$pathway_GGI.
dict_MGI	A four-column dataframe, including MGI miRNA gene strength. Containing all possible MGIs that could be used in permutation test. Equals to input_data\$dict_MGI.
iter	Number of iterations for permutation test.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values.
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs less than the minSize threshold.
maxSize	The maximum MGI set sizes allowed for analysis, filtering out pathways with number of member MGIs more than the maxSize threshold.
ncores	Number of cores for permutation test (parallel computing different pathways).
seed	Optional integer seed for the permutation test. If NULL (the default), no internal seed is set and the permutation draws depend on the caller's current random state. If supplied, the permutation is made reproducible via [withr::with_seed()], which does not alter the caller's global random state.

Value

A dataframe containing the enrichment result for Edge-Topology.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata)
data(toy_gene_mat); data(toy_mir_mat)
data(toy_background_MGI); data(toy_background_GGI)
input_data <- getAllInputData(
  methods = "Edge_Topology", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI,
  background_GGI = toy_background_GGI, GGI_source = "OmniPath",
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat, scoreFun = "rank"
)
background <- unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
res <- Edge_Topology(
  MGI = input_data$data$Edge_Topology, background = background,
  pathway = input_data$pathway$Edge, pathway_GGI = input_data$pathway_GGI,
  dict_MGI = input_data$dict_MGI, pvalueType = "neg",
  minSize = 2, maxSize = 5000, iter = 20, ncores = 1, pvalueCutoff = 1
)
head(res)
```

getAllInputData	<i>Generate input data for miREA</i>
-----------------	--------------------------------------

Description

Functions to generate input data for miREA, based on the original expression and differential expression results for miRNAs and genes.

Usage

```
getAllInputData(
  methods,
  pathway,
  mir_DEdata,
  gene_DEdata = NULL,
  background_MGI = NULL,
  background_GGI = NULL,
  GGI_source = "Omnipath",
  gene_mat = NULL,
  mir_mat = NULL,
  scoreFun = NULL
)
```

Arguments

methods	compulsory, see default_method_list for all supportive methods. You could either input one method, or a vector contains several methods you want to try.
pathway	compulsory. It should be a two-column pathway, containing pathway and gene, or a GSEABase GeneSetCollection object.
mir_DEdata	compulsory, should be a four-column dataframe, containing miRNA, log2FC, stat, padj
gene_DEdata	optional, except for TG-ORA, MiR-ORA, and MiR-Score methods, should be a four-column dataframe, containing gene, log2FC, stat, padj
background_MGI	required for MiR_ and Edge_ methods. It should be either a character vector in the format of [miRNA]:[gene], or a two-column dataframe, containing miRNA gene. miREA does not ship a bundled default.
background_GGI	optional, it should be a two-column data frame that contains source gene and target gene. Combined with *GGI_source*. If NULL, we would use GGI_source to determine the background GGIs.
GGI_source	optional, select from "Omnipath" or "Reactome". Used when background_GGI is not provided. The default is Omnipath.
gene_mat	optional, needed for Edge_ method, used for calculating correlations between miRNAs and genes. It should be a raw gene expression matrix (each row a gene, each column a sample), either as a dataframe/matrix or as a Summarized-Experiment object.

<code>mir_mat</code>	optional, needed for <code>Edge_</code> method, used for calculating correlations between miRNAs and genes. It should be a raw miRNA expression matrix (each row a miRNA, each column a sample), either as a dataframe/matrix or as a <code>SummarizedExperiment</code> object.
<code>scoreFun</code>	optional, needed for <code>Edge_Score1D</code> , <code>Edge_Score2D</code> , <code>Edge_Topology</code> . See <code>default_scoreFun</code> for all supportive methods.

Value

A list that contains all the data formats needed for miREA.

Note

`gene_mat` and `mir_mat` should have same sample name format. If the number of shared samples below 50

References

Zhang Z, Lai X (2026). miREA: a network-based tool for microRNA-oriented enrichment analysis. *bioRxiv*. doi:[10.64898/2026.02.27.708509](https://doi.org/10.64898/2026.02.27.708509)

Examples

```
data(toy_pathway)
data(toy_gene_DEdata)
data(toy_mir_DEdata)
data(toy_gene_mat)
data(toy_mir_mat)
data(toy_background_MGI)
data(toy_background_GGI)
input_data <- getAllInputData(
  methods = default_method_list,
  pathway = toy_pathway,
  mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata,
  background_MGI = toy_background_MGI,
  background_GGI = toy_background_GGI,
  GGI_source = "Omnipath",
  gene_mat = toy_gene_mat,
  mir_mat = toy_mir_mat,
  scoreFun = "rank"
)
names(input_data)
```

MiR_ORA

*MiR-ORA (MiRNA Over-Representation Analysis)***Description**

Perform miRNA enrichment analysis based on the overlap between differentially expressed miRNA set and pathway miRNA set. Pathway miRNA set was mapped from pathway gene set.

Usage

```
MiR_ORA(
  DEmiR,
  background = NULL,
  pathway,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)
```

Arguments

DEmiR	A character vector contains selected differential expressed miRNAs.
background	A character vector contains background miRNAs.
pathway	A two-column dataframe contains pathway and their member miRNAs.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum miRNA set sizes allowed for analysis, filtering out pathways with number of member miRNAs less than the minSize threshold.
maxSize	The maximum miRNA set sizes allowed for analysis, filtering out pathways with number of member miRNAs more than the maxSize threshold.

Value

A dataframe containing the enrichment result for MiR-ORA.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "MiR_ORA", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI
```

```

)
res <- MiR_ORA(
  DEmiR = input_data$data$MiR_ORA, background = unique(input_data$pathway$MiR$miRNA),
  pathway = input_data$pathway$MiR, pvalueCutoff = 1
)
head(res)

```

MiR_Score

MiR-Score (MiRNA Score-based Analysis)

Description

Perform miRNA enrichment analysis based on whether the pathway miRNA are located unexpectedly forward towards the ranked miRNA list.

Usage

```

MiR_Score(
  miRList,
  pathway,
  pvalueType = "std",
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)

```

Arguments

miRList	A named vector shows the ranked score of miRNAs.
pathway	A two-column dataframe contains pathway and their member miRNAs.
pvalueType	How to calculate p value. See default_pvalueType for all possible options. Default pvalueType is "pos". The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment). The "std" is used when interested in both positive and negative scores.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum miRNA set sizes allowed for analysis, filtering out pathways with number of member miRNAs less than the minSize threshold.
maxSize	The maximum miRNA set sizes allowed for analysis, filtering out pathways with number of member miRNAs more than the maxSize threshold.

Value

A dataframe containing the enrichment result for MiR-Score.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "MiR_Score", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI
)
res <- MiR_Score(
  miRList = input_data$data$MiR_Score, pathway = input_data$pathway$MiR,
  pvalueType = "std", pvalueCutoff = 1
)
head(res)
```

miREA	<i>MGI edge-based microRNA-oriented enrichment analysis</i>
-------	---

Description

The main entry point for miREA, implementing four node-based methods and five edge-based methods for miRNA-oriented enrichment analysis.

Usage

```
miREA(
  methods,
  input_data,
  background = NULL,
  minSize = NULL,
  maxSize = NULL,
  pvalueType = list(TG = "std", MiR = "std", Edge = "neg"),
  pAdjMethod = "BH",
  pvalueCutoff = 1,
  iter = 1000,
  ncores = NULL,
  seed = NULL
)
```

Arguments

methods	Method used for enrichment analysis, see default_method_list for all supportive options. Methods could be several, like c("Edge_ORA", "Edge_Score1D").
input_data	Input data list, containing the input data for all methods, including data, pathway, pathway_GGI, GGI, and dict_MGI. Use getAllInputData() function to generate. You could also generate it by yourself. input_data\$data: each sub-element is named by the method, containing the input data format. TG_ORA:

	DEG, MiR_ORA: DEmiR, Edge_ORA: DEMGI, Edge_Network: DEMGI All of them is a character vector, containing all selected differential expressed items (genes/miRNAs/MGI) TG_Score: geneList, MiR_Score: miRList, Edge_Score1D: MGIList All of them is a named numeric vector, where the names are items (genes/miRNAs/MGI), corresponding value is their score, which is ranked by the score. Edge_Topology: MGIdata A five-column dataframe, containing MGI miRNA gene strength. Edge_Score2D: MGI2D A five-column dataframe, including MGI miRNA gene cor(x) ranknormratio(y). input_data\$pathway: each sub-element is named by the object, which could be TG, MiR, and Edge. input_data\$pathway_GGI: a dataframe containing pathway and their directed GGI involved. Only useful when methods contain Edge_Topology and Edge_Network. input_data\$MGI: a dataframe containing MGIs across all the pathways and DEMGIs. Only useful for Edge_Network. input_data\$GGI: a dataframe containing directed GGIs across all the pathways and DEMGIs. Only useful for Edge_Network. input_data\$dict_MGI: a dataframe containing all possible MGIs and their expression profiles. Only useful for Edge_Topology.
background	Contains background items (genes for TG_ method, miRNAs for MiR_ method, MGIs for Edge_ method). If methods contains more than one targets, it should be a list where each sub-element is a character vector named by the target (TG, MiR, or Edge). If methods contains only one target, it could also be a character vector. If NULL, use the global background instead.
minSize	The minimum pathway set sizes allowed for analysis, filtering out pathways with number of members less than the minSize threshold.
maxSize	The maximum pathway set sizes allowed for analysis, filtering out pathways with number of members more than the maxSize threshold.
pvalueType	How to calculate p value. See default_pvalueType for all possible options. Default pvalueType is list(TG = "std", MiR = "std", Edge = "neg"). The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment). The "std" is used when interested in both positive and negative scores, which is the same as the original GSEA algorithm. Only used in Score-based method (Edge_Score2D, Edge_TopoScore, TG_Score, MiR_Score).
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 1.
iter	Number of iterations when perform permutation test. Default iter is 1000.
ncores	Number of cores for parallel computing. Only useful if methods contain Edge_Score2D, Edge_Topology, and Edge_Network. It accepts either a integer that is universal for all methods, or a list specify the number of cores for each method separately. Our recommendation is to specify different nodes for each method. E.g., list(Edge_Score2D = 4, Edge_Topology = 32, Edge_Network = 8). If NULL, unparallel computing will be used.

seed Optional integer seed for the permutation-based methods (Edge_Score2D, Edge_Topology, Edge_Network). If NULL (the default), no internal seed is set and the permutation draws depend on the caller's current random state. If supplied, it is forwarded to all three methods via [withr::with_seed()], which does not alter the caller's global random state.

Value

A [miREAResult-class] object containing the per-method result tables ('result'), per-method elapsed time ('time'), the methods that completed without error ('valid_methods'), and the call parameters ('parameter'). Supports '\$' access to these four fields for backward compatibility.

References

Zhang Z, Lai X (2026). miREA: a network-based tool for microRNA-oriented enrichment analysis. bioRxiv. doi:10.64898/2026.02.27.708509

Examples

```
data(toy_pathway); data(toy_gene_DEdata); data(toy_mir_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI); data(toy_background_GGI)
input_data <- getAllInputData(
  methods = default_method_list, pathway = toy_pathway,
  mir_DEdata = toy_mir_DEdata, gene_DEdata = toy_gene_DEdata,
  background_MGI = toy_background_MGI, background_GGI = toy_background_GGI,
  GGI_source = "Omnipath", gene_mat = toy_gene_mat, mir_mat = toy_mir_mat,
  scoreFun = "rank"
)
background <- list(
  TG = unique(toy_pathway$gene),
  MiR = unique(input_data$pathway$MiR$miRNA),
  Edge = unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
)
result <- miREA(
  methods = default_method_list, input_data = input_data, background = background,
  minSize = NULL, maxSize = NULL,
  pvalueType = list(TG = "std", MiR = "std", Edge = "neg"),
  pAdjMethod = "BH", pvalueCutoff = 1, iter = 20, ncores = 1
)
result$valid_methods
```

miREA_example_data

Toy BLCA example data for miREA

Description

A small, self-contained example dataset derived from the real BLCA (bladder urothelial carcinoma) data used in the miREA manuscript. Gene and miRNA differential expression are computed

(DESeq2, tumor vs. normal) from 80 tumor + 19 normal TCGA samples. The resulting DE-testable gene/miRNA universe is used to subset 16 bladder-cancer-specific MSigDB gene sets (e.g. KEGG_BLADDER_CANCER, LINDGREN_BLADDER_CANCER_*) and the manuscript's background_MGI/background_GGI. 80 tumor samples is the smallest size found to give every method, including the edge-based methods' stricter correlation-significance filters, real non-empty results, so every exported function in the package can run on real (not synthetic) data. 'toy_gene_mat'/'toy_mir_mat' use only the tumor samples (correlation is computed within the tumor population, as in the manuscript). See 'data-row/generate_toy_data.R' for the full generation script. For the full datasets used in the manuscript, see the Zenodo deposit at <https://zenodo.org/records/18803209>.

Format

- toy_pathway** A data.frame with columns 'pathway', 'gene': pathway-gene membership for 16 bladder-cancer-specific MSigDB gene sets.
- toy_gene_DEdata** A data.frame with columns 'gene', 'log2FoldChange', 'stat', 'padj': differential expression results for the gene universe.
- toy_mir_DEdata** A data.frame with columns 'miRNA', 'log2FoldChange', 'stat', 'padj': differential expression results for the miRNA universe.
- toy_gene_mat** A gene-by-sample expression matrix (data.frame, genes as rownames), the 80 tumor TCGA samples.
- toy_mir_mat** A miRNA-by-sample expression matrix (data.frame, miRNAs as rownames), the same 80 tumor TCGA samples as 'toy_gene_mat'.
- toy_background_MGI** A data.frame with columns 'miRNA', 'gene': curated miRNA-gene interactions restricted to the toy universe.
- toy_background_GGI** A data.frame with columns 'source_genesymbol', 'target_genesymbol', 'interact': Omnipath gene-gene interactions restricted to the toy gene universe.

Source

Derived from TCGA BLCA data and MSigDB bladder-cancer-specific gene sets, as processed for the miREA manuscript ([doi:10.64898/2026.02.27.708509](https://doi.org/10.64898/2026.02.27.708509)).

Examples

```
data(toy_pathway)
data(toy_gene_DEdata)
data(toy_mir_DEdata)
data(toy_gene_mat)
data(toy_mir_mat)
data(toy_background_MGI)
data(toy_background_GGI)
```

miREAResult-class	<i>S4 class for miREA() results</i>
-------------------	-------------------------------------

Description

A lightweight container bundling the per-method enrichment result tables together with the timing/validity/parameter metadata returned by [miREA()]. Supports '\$' and '[' access for backward compatibility with the function's previous plain-list return value.

Usage

```
miREAResult(result, time, valid_methods, parameter)

## S4 method for signature 'miREAResult'
x$name

## S4 method for signature 'miREAResult'
x[[i, j, ...]]

## S4 method for signature 'miREAResult'
show(object)

## S4 method for signature 'miREAResult'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

result	Named list, one enrichment result data.frame per requested method.
time	A data.frame recording the elapsed time (in seconds) for each method.
valid_methods	Character vector of methods that completed without error.
parameter	A list of the parameters [miREA()] was called with.
x	A [miREAResult-class] object.
name	The slot to access: one of "result", "time", "valid_methods", "parameter".
i	The slot to access: one of "result", "time", "valid_methods", "parameter".
j	Not used.
...	Not used.
object	A [miREAResult-class] object.
row.names	Not used.
optional	Not used.

Value

A [miREAResult-class] object.

Slots

result Named list, one enrichment result data.frame per requested method.
time A data.frame recording the elapsed time (in seconds) for each method.
valid_methods Character vector of methods that completed without error.
parameter A list of the parameters [miREA()] was called with.

Examples

```
res <- miREAResult(
  result = list(TG_ORA = data.frame(pathway = "PW1", padj = 0.01)),
  time = data.frame(method = "TG_ORA", time = 0.1),
  valid_methods = "TG_ORA",
  parameter = list(n_pathway = 1)
)
res
res$valid_methods
res[["time"]]
as.data.frame(res)
```

plotHeatmap

Visualize an enrichment result as a heatmap

Description

Draws a heatmap of the miRNA-gene-pathway enrichment result for one edge-based method, showing expression fold-changes and significance. Only supports one method each time.

Usage

```
plotHeatmap(
  method,
  result,
  input_data,
  n_mir_heatmap = 10,
  n_pathway = 10,
  plot_path = NULL,
  penrichCutoff = 0.05,
  height = NULL,
  width = NULL,
  gene_annot = NULL,
  mir_annot = NULL,
  annot_color = "#A6DDBA"
)
```

Arguments

method	a character that specifies only one method which you have the corresponding enrichment result (only five edge-based methods are supported).
result	the output of miREA() function, used here as the unprocessed result list.
input_data	the output of getAllInputData() function, used here as the unprocessed input_data list.
n_mir_heatmap	The number of miRNAs included in the heatmap. Default is 10.
n_pathway	The number of top enriched pathways included in the heatmap. Default it 10. If the number of enriched pathways is less than n_pathway, all enriched pathways will be included.
plot_path	The directory to save the plot. If not null, it will generate a pdf file named heatmap_[method].pdf at the plot path. If null, the plot will be drawn at the current window.
penrichCutoff	The threshold for deciding enrichment of pathways. Default is 0.05.
height	The height of the plot. If not, it will be set to default values.
width	The width of the plot. If not, it will be set to default values.
gene_annot	A list contains the annotations of genes, where each sub-element denotes a character vector containing a list of genes. It will generate a annotation bar on the top of the heatmap to show whether the genes hit in the annotation list. Default is NULL, which means no annotation bar will be generated.
mir_annot	Same format as gene_annot. If not NULL, it will generate an annotation bar for miRNAs at the left side of the heatmap, to show whether miRNAs hit in the given list.
annot_color	The color of the annotation bars for genes and miRNAs

Value

A heatmap plot either in the current window, or saved as heatmap_[method].pdf in the plot_path.

Examples

```
data(toy_pathway); data(toy_gene_DEdata); data(toy_mir_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
# Edge_Score2D is included so input_data carries the MGI "strength" scores
# plotHeatmap() needs for its heatmap cells, without running it in miREA().
input_data <- getAllInputData(
  methods = c("Edge_ORA", "Edge_Score2D"), pathway = toy_pathway,
  mir_DEdata = toy_mir_DEdata, gene_DEdata = toy_gene_DEdata,
  background_MGI = toy_background_MGI,
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat, scoreFun = "rank"
)
background <- list(
  Edge = unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
)
result <- miREA(
  methods = "Edge_ORA", input_data = input_data, background = background,
```

```

    minSize = NULL, maxSize = NULL, pvalueCutoff = 1
  )
  plotHeatmap(method = "Edge_ORA", result = result, input_data = input_data, penrichCutoff = 1)

```

plotHeatmapSankey

Visualize an enrichment result as a combined heatmap-Sankey plot

Description

Draws a heatmap of the miRNA-gene-pathway enrichment result for one edge-based method, combined with a Sankey diagram showing the miRNA-gene-pathway flow. Only supports one method each time.

Usage

```

plotHeatmapSankey(
  method,
  result,
  input_data,
  n_mir_heatmap = 20,
  n_mir_sankey = 10,
  n_pathway = 10,
  plot_path = NULL,
  penrichCutoff = 0.05,
  height = NULL,
  width = NULL,
  sankey_prop = 1.5,
  gene_annot = NULL,
  mir_annot = NULL,
  annot_color = "#A6DDBA"
)

```

Arguments

method	a character that specifies only one method which you have the corresponding enrichment result (only five edge-based methods are supported).
result	the output of miREA() function, used here as the unprocessed result list.
input_data	the output of getAllInputData() function, used here as the unprocessed input_data list.
n_mir_heatmap	The number of miRNAs included in the heatmap. Default is 20.
n_mir_sankey	The number of miRNAs included in the Sankey diagram. Default is 10.
n_pathway	The number of top enriched pathways included in the heatmap. Default it 10. If the number of enriched pathways is less than n_pathway, all enriched pathways will be included.

<code>plot_path</code>	The directory to save the plot. If not null, it will generate a pdf file named <code>heatmap_sankey_[method].pdf</code> at the plot path. If null, the plot will be drawn at the current window.
<code>penrichCutoff</code>	The threshold for deciding enrichment of pathways. Default is 0.05.
<code>height</code>	The height of the plot. If not, it will be set to default values.
<code>width</code>	The width of the plot. If not, it will be set to default values.
<code>sankey_prop</code>	The ratio of the height of Sankey diagram and heatmap. Default is 1.5.
<code>gene_annot</code>	A list contains the annotations of genes, where each sub-element denotes a character vector containing a list of genes. It will generate an annotation bar on the top of the heatmap to show whether the genes hit in the annotation list. Default is NULL, which means no annotation bar will be generated.
<code>mir_annot</code>	Same format as <code>gene_annot</code> . If not NULL, it will generate an annotation bar for miRNAs at the left side of the heatmap, to show whether miRNAs hit in the given list.
<code>annot_color</code>	The color of the annotation bars for genes and miRNAs

Value

A heatmap-sankey plot either in the current window, or saved as `heatmap_sankey_[method].pdf` in the `plot_path`.

Examples

```
data(toy_pathway); data(toy_gene_DEdata); data(toy_mir_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
# Edge_Score2D is included so input_data carries the MGI "strength" scores
# plotHeatmapSankey() needs for its heatmap cells, without running it in miREA().
input_data <- getAllInputData(
  methods = c("Edge_ORA", "Edge_Score2D"), pathway = toy_pathway,
  mir_DEdata = toy_mir_DEdata, gene_DEdata = toy_gene_DEdata,
  background_MGI = toy_background_MGI,
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat, scoreFun = "rank"
)
background <- list(
  Edge = unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
)
result <- miREA(
  methods = "Edge_ORA", input_data = input_data, background = background,
  minSize = NULL, maxSize = NULL, pvalueCutoff = 1
)
plotHeatmapSankey(method = "Edge_ORA", result = result, input_data = input_data, penrichCutoff = 1)
```

plotSummary

*Visualize and compare enrichment results across methods***Description**

Draws summary plots comparing enrichment results across the different miREA methods: a positive-rate/PCA/pairwise-overlap panel, and a p-value density plot.

Usage

```
plotSummary(
  result,
  penrichCutoff = NULL,
  plot_path = NULL,
  fill_col = c(TG_ORA = "#97D7F2", TG_Score = "#07AEE3", MiR_ORA = "#B3D49D", MiR_Score =
    "#35B257", Edge_ORA = "#EDC194", Edge_Score1D = "#F09137", Edge_Score2D = "#626FB3",
    Edge_Topology = "#EAA5C2", Edge_Network = "#FA8072")
)
```

Arguments

result	the output of miREA() function, used here as the unprocessed result list.
penrichCutoff	The threshold for deciding enrichment of pathways. Default is 0.05.
plot_path	The directory to save the plot. If not null, it will generate two pdf file named summary.pdf and pvalue_density.pdf at the plot path. If null, the plot will be drawn at the current window.
fill_col	a named vector to specify fill colors for different methods.

Value

Two plots, either in the current window if plot_path is NULL, or saved as summary.pdf and pvalue_density.pdf at the plot_path.

Examples

```
data(toy_pathway); data(toy_gene_DEdata); data(toy_mir_DEdata)
data(toy_gene_mat); data(toy_mir_mat); data(toy_background_MGI)
methods <- c("TG_ORA", "TG_Score", "MiR_ORA", "MiR_Score", "Edge_ORA")
input_data <- getAllInputData(
  methods = methods, pathway = toy_pathway,
  mir_DEdata = toy_mir_DEdata, gene_DEdata = toy_gene_DEdata,
  background_MGI = toy_background_MGI,
  gene_mat = toy_gene_mat, mir_mat = toy_mir_mat
)
background <- list(
  TG = unique(toy_pathway$gene),
  MiR = unique(input_data$pathway$MiR$miRNA),
```

```

    Edge = unique(paste0(toy_background_MGI$miRNA, ":", toy_background_MGI$gene))
  )
  result <- miREA(
    methods = methods, input_data = input_data, background = background,
    minSize = NULL, maxSize = NULL,
    pvalueType = list(TG = "std", MiR = "std", Edge = "neg"),
    pvalueCutoff = 1
  )
  plotSummary(result, penrichCutoff = 1)

```

TG_ORA

TG-ORA (Target Gene Over-Representation Analysis)

Description

Perform target gene enrichment analysis based on the overlap between differentially expressed gene set and pathway gene set.

Usage

```

TG_ORA(
  TG,
  background = NULL,
  pathway,
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)

```

Arguments

TG	A character vector contains selected target genes of DEmiRs.
background	A character vector contains background genes.
pathway	A two-column dataframe contains pathway and their member genes.
pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum gene set sizes allowed for analysis, filtering out pathways with number of member genes less than the minSize threshold.
maxSize	The maximum gene set sizes allowed for analysis, filtering out pathways with number of member genes more than the maxSize threshold.

Value

A dataframe containing the enrichment result for TG-ORA.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "TG-ORA", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI
)
res <- TG_ORA(
  TG = input_data$data$TG_ORA, background = unique(toy_pathway$gene),
  pathway = input_data$pathway$TG, pvalueCutoff = 1
)
head(res)
```

TG_Score	<i>TG-Score (Target Gene Score-based Analysis)</i>
----------	--

Description

Perform target gene enrichment analysis based on whether the pathway genes are located unexpectedly forward towards the ranked gene list.

Usage

```
TG_Score(
  geneList,
  pathway,
  pvalueType = "std",
  pAdjMethod = "BH",
  pvalueCutoff = 0.05,
  minSize = NULL,
  maxSize = NULL
)
```

Arguments

- geneList A named vector shows the ranked score of target genes.
- pathway A two-column dataframe contains pathway and their member genes.
- pvalueType How to calculate p value. See default_pvalueType for all possible options. Default pvalueType is "pos". The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment). The "std" is used when interested in both positive and negative scores.

pAdjMethod	The method used for multiple testing correction to adjust p-values and control the false discovery rate. Use default_pAdjMethod to find all possible values. If not specified, Benjamini & Hochberg (BH) method will be automatically used. If no adjustment is desired, set pAdjMethod = "none".
pvalueCutoff	The threshold for statistical significance, filtering out pathways with padj values below the specified pvalueCutoff. Default pvalueCutoff is 0.05.
minSize	The minimum gene set sizes allowed for analysis, filtering out pathways with number of member genes less than the minSize threshold.
maxSize	The maximum gene set sizes allowed for analysis, filtering out pathways with number of member genes more than the maxSize threshold.

Value

A dataframe containing the enrichment result for TG-Score.

Examples

```
data(toy_pathway); data(toy_mir_DEdata); data(toy_gene_DEdata); data(toy_background_MGI)
input_data <- getAllInputData(
  methods = "TG_Score", pathway = toy_pathway, mir_DEdata = toy_mir_DEdata,
  gene_DEdata = toy_gene_DEdata, background_MGI = toy_background_MGI
)
res <- TG_Score(
  geneList = input_data$data$TG_Score, pathway = input_data$pathway$TG,
  pvalueType = "std", pvalueCutoff = 1
)
head(res)
```

Index

*** datasets**
 miREA_example_data, [20](#)
[[, miREAResult-method
 (miREAResult-class), [22](#)
\$, miREAResult-method
 (miREAResult-class), [22](#)

as.data.frame, miREAResult-method
 (miREAResult-class), [22](#)

default_GGI_source, [2](#)
default_method_list, [3](#)
default_pAdjMethod, [4](#)
default_pathway_list, [4](#)
default_pvalueType, [5](#)
default_scoreFun, [5](#)
default_TP_cancer, [6](#)

Edge_Network, [6](#)
Edge_ORA, [8](#)
Edge_Score1D, [9](#)
Edge_Score2D, [10](#)
Edge_Topology, [12](#)

getAllInputData, [14](#)

MiR_ORA, [16](#)
MiR_Score, [17](#)
miREA, [18](#)
miREA_example_data, [20](#)
miREAResult (miREAResult-class), [22](#)
miREAResult-class, [22](#)

plotHeatmap, [23](#)
plotHeatmapSankey, [25](#)
plotSummary, [27](#)

show, miREAResult-method
 (miREAResult-class), [22](#)

TG_ORA, [28](#)

TG_Score, [29](#)
toy_background_GGI
 (miREA_example_data), [20](#)
toy_background_MGI
 (miREA_example_data), [20](#)
toy_gene_DEdata (miREA_example_data), [20](#)
toy_gene_mat (miREA_example_data), [20](#)
toy_mir_DEdata (miREA_example_data), [20](#)
toy_mir_mat (miREA_example_data), [20](#)
toy_pathway (miREA_example_data), [20](#)