Package ‘RTNduals’

April 26, 2017

Type Package

Title Analysis of co-regulatory network motifs and inference of ‘dual regulons’

Version 1.0.0

Author Vinicius S. Chagas, Clarice S. Groeneveld, Gordon Robertson, Kerstin B. Meyer, Mauro A. A. Castro

Maintainer Vinicius Chagas <vinicius.chagas@ufpr.br>, Mauro Castro
          <mauro.castro@ufpr.br>, Clarice Groeneveld
          <clari.groeneveld@gmail.com>

Depends R (>= 3.3), methods, RTN, graphics

Imports grDevices, stats, utils

Suggests knitr, rmarkdown, BiocStyle, RUnit, BiocGenerics

Description RTNduals is a tool that searches for possible co-regulatory loops between regulon pairs generated by the RTN package. It compares the shared targets in order to infer ‘dual regulons’, a new concept that tests whether regulon pairs agree on the predicted downstream effects.

License Artistic-2.0

biocViews GeneRegulation, GeneExpression, GeneticVariability,
            GeneSetEnrichment, NetworkEnrichment, NetworkInference,
            GraphAndNetwork

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

  RTNduals-package .................................................. 2
  MBR-class .......................................................... 3
  mbrAssociation, MBR-method ...................................... 4
  mbrBootstrap, MBR-method ......................................... 5
  mbrDpiFilter, MBR-method .......................................... 6
  mbrDuals, MBR-method ............................................... 7
RTNduals-package

mbrGet, MBR-method .................................................. 8
mbrPermutation, MBR-method ........................................... 9
mbrPlotDuals ............................................................ 9
mbrPreprocess, matrix-method ....................................... 10
tni2mbrPreprocess, TNI-method .................................... 11

Index 13

<table>
<thead>
<tr>
<th>RTNduals-package</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTNduals: An R/Bioconductor package for analysis of co-regulatory network motifs and inference of 'dual regulons'.</td>
</tr>
</tbody>
</table>

**Description**

RTNduals is a tool that searches for possible co-regulatory loops between regulon pairs generated by the RTN package. It compares the shared targets in order to infer "dual regulons", a new concept that tests whether regulon pairs agree on the predicted downstream effects.

**Details**

Package: RTNduals  
Type: Package  
Depends: R (>= 3.4.0), methods, RTN  
Imports: grDevices, stats, utils  
Suggests: knitr, rmarkdown, BiocStyle, RUnit, BiocGenerics  
License: Artistic-2.0  
biocViews: NetworkInference, NetworkEnrichment, GeneRegulation, GeneExpression, GraphAndNetwork

**Index**

<table>
<thead>
<tr>
<th>MBR-class:</th>
</tr>
</thead>
<tbody>
<tr>
<td>an S4 class for co-regulation analysis and inference of 'dual regulons'.</td>
</tr>
</tbody>
</table>

| mbrPreprocess:  |
| a preprocessing function for objects of class MBR. |

| mbrPermutation:  |
| inference of transcriptional networks. |

| mbrBootstrap:  |
| inference of consensus transcriptional networks. |

| mbrDpiFilter:  |
| a filter based on the Data Processing Inequality (DPI) algorithm. |

| mbrAssociation:  |
| motifs analysis and inference of "dual regulons". |

| mbrDuals:  |
| a summary for results from the MBR methods. |

| tni2mbrPreprocess:  |
| a preprocessing function for objects of class MBR. |

Further information is available in the vignettes by typing `vignette("RTNduals")`. Documented topics are also available in HTML by typing `help.start()` and selecting the RTNduals package from the menu.

**Author(s)**

Viniclus S. Chagas, Clarice S. Groeneveld, Kerstin B Meyer, Gordon Robertson, Mauro A. A. Castro
**MBR-class**

**References**


---

**MBR-class**  
**MBR objects**

---

**Description**

MBR: an S4 class for co-regulation analysis and inference of 'dual regulons'.

**Details**

The MBR class is a container for results from the MBR methods. The class slots are used to store information of different transcriptional networks, regulator annotation, inferred 'dual regulons' and parameters used in the analysis. All the information is stored in nine slots.

**Slots**

- **TNI1** a 'TNI' object created by the RTN package.
- **TNI2** another 'TNI' object created by the RTN package.
- **testedElementsTNI1** regulatory elements listed in the TNI1.
- **testedElementsTNI2** regulatory elements listed in the TNI2.
- **dualRegulons** all possible 'dual regulons' computed by `mbrAssociation`.
- **results** a list, results from the MBR methods.
- **para** a list, parameters used in the MBR methods.
- **summary** a list, summary for 'para' and 'results'.
- **status** a character vector specifying the status of the MBR object based on the available methods.

**Constructor**

There are two constructors to create an MBR object, users can opt for one of the following: (1) `mbrPreprocess`; (2) `tni2mbrPreprocess`.

- (1): It is used to create an MBR object without any pre-computed transcriptional network.
- (2): It is used to create an MBR object using available transcriptional networks.
mbrAssociation, MBR-method

Motifs analysis and inference of 'dual regulons'.

Description

This function takes an MBR object and compares the shared regulon targets in order to test whether regulon pairs agree on the predicted downstream effects.

Usage

```r
## S4 method for signature 'MBR'
mbrAssociation(object, regulatoryElements1 = NULL,
               regulatoryElements2 = NULL, minRegulonSize = 30, prob = 0.95,
               estimator = "spearman", pAdjustMethod = "BH", verbose = TRUE)
```

Arguments

- **object**: A processed object of class MBR evaluated by the methods `mbrPermutation`, `mbrBootstrap` and `mbrDpiFilter`.
- **regulatoryElements1**: An optional character vector specifying which 'TNI1' regulatory elements should be evaluated. If 'NULL' all regulatory elements will be evaluated.
- **regulatoryElements2**: An optional character vector specifying which 'TNI2' regulatory elements should be evaluated. If 'NULL' all regulatory elements will be evaluated.
- **minRegulonSize**: A single integer or numeric value specifying the minimum number of elements in a regulon. Gene sets with fewer than this number are removed from the analysis.
- **prob**: A quantile filter applied to the association metric used to infer 'dual regulons'.
- **estimator**: A character value specifying the estimator used in the association analysis. One of "spearman" (default), "kendall", or "pearson", can be abbreviated.
- **pAdjustMethod**: A single character value specifying the p-value adjustment method to be used (see 'p.adjust' for details).
- **verbose**: A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

An MBR object with two data.frames in the slot 'results' listing the inferred 'dual regulons' and a hypergeometric test for each 'dual regulon'.

Examples

```r
data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LM04","ZNF552")]
##---mbrPreprocess
```
mbrBootstrap, MBR-method

Inference of consensus transcriptional networks.

Description

This function takes an MBR object and computes two consensus transcriptional networks.

Usage

```r
mbrBootstrap(object, verbose = TRUE, ...
```

Arguments

- `object`: A processed objec of class MBR evaluated by the method `mbrPermutation`.
- `verbose`: A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
- `...`: Additional arguments passed to the `tni.bootstrap` function.

Value

An MBR object with two consensus mutual information matrices, one in each "TNI" slot.

Examples

```r
data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNF552")]
##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)
##---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
##---mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
##---mbrDpiFilter
rmbr <- mbrDpiFilter(rmbr)
##---mbrAssociation
rmbr <- mbrAssociation(rmbr, prob=0.75)
```
mbrDpiFilter, MBR-method

A filter based on the Data Processing Inequality (DPI) algorithm.

Description

This function takes an MBR object and computes two transcriptional networks filtered by the data processing inequality algorithm.

Usage

```r
## S4 method for signature 'MBR'

mbrDpiFilter(object, verbose = TRUE, ...)
```

Arguments

- `object`: A processed object of class MBR evaluated by the methods `mbrPermutation` and `mbrBootstrap`.
- `verbose`: A single logical value specifying to display detailed messages (when `verbose=TRUE`) or not (when `verbose=FALSE`).
- `...`: Additional arguments passed to the `tni.dpi.filter` function.

Value

An MBR object with two DPI-filtered mutual information matrices, one in each "TNI" slot.

Examples

data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("TCF1","STAT4","STAT1","LMP6","ZNF552")]

###---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1, regulatoryElements2=tfs2, gexpIDs=annot)
###---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
###---mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
###---mbrDpiFilter
rmbr <- mbrDpiFilter(rmbr)
mbrDuals, MBR-method

A summary for results from the MBR methods.

Description

This function lists the inferred 'dual regulons' and, if available, adds external evidences.

Usage

```r
## S4 method for signature 'MBR'
mbrDuals(object, supplementary.table = NULL,
         evidenceColname = NULL, verbose = TRUE)
```

Arguments

- **object**: A processed object of class `MBR` evaluated by the method `mbrAssociation`.
- **supplementary.table**: An optional `data.frame` with three columns representing (1) regulatory elements of 'TNI1', (2) regulatory elements of 'TNI2', and (3) external evidences between the regulatory elements.
- **evidenceColname**: A single character value specifying a column in the 'supplementary.table'.
- **verbose**: A single logical value specifying to display detailed messages (when `verbose=TRUE`) or not (when `verbose=FALSE`).

Value

An `MBR` object with a data.frame in the slot 'results' listing the input additional evidences.

Examples

```r
data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNF552")]##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
                         regulatoryElements2=tfs2, gexpIDs=annot)
##---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
##---mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
##---mbrDpiFilter
rmbr <- mbrDpiFilter(rmbr)
##---mbrAssociation
rmbr <- mbrAssociation(rmbr, prob=0.75)
##---a 'toy' table with supplementary evidences
motifsInformation <- mbrGet(rmbr, what="motifsInformation")
n <- nrow(motifsInformation)
supplementaryTable <- motifsInformation[1:n,c("Regulon1","Regulon2")]
supplementaryTable$ToyEvidence <- rnorm(n)
```
### mbrGet, MBR-method

Get information from individual slots in MBR object.

#### Description

Get information from individual slots in an MBR object and any available results from previous analysis.

#### Usage

```r
## S4 method for signature 'MBR'
mbrGet(object, what = "status")
```

#### Arguments

- `object`: A preprocessed object of class MBR
- `what`: a single character value specifying which information should be retrieved from the slots. Options: "TNI1", "TNI2", "testedElementsTNI1", "testedElementsTNI2", "dualRegulons", "results", "para", "summary", "status", "motifsInformation" and "hyperResults"

#### Value

A slot content from a object of class 'MBR' MBR object

#### Examples

```r
data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNF552")]
## mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1, regulatoryElements2=tfs2, gexpIDs=annot)
## get the 'TNI1' slot using 'mbrGet'
tni1 <- mbrGet(rmbr, what="TNI1")
```
mbrPermutation, MBR-method

Inference of transcriptional networks.

Description

This function takes an MBR object and computes two transcriptional networks inferred by mutual information (with multiple hypothesis testing corrections).

Usage

```r
## S4 method for signature 'MBR'
mbrPermutation(object, verbose = TRUE, ...)
```

Arguments

- `object`: A preprocessed object of class MBR.
- `verbose`: A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
- `...`: Additional arguments passed on to the `tni.permutation` function.

Value

An MBR object with two mutual information matrices, one in each "TNI" slot.

Examples

```r
data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8", "IRF1", "PRDM1", "AFF3", "E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1", "STAT4", "STAT1", "LMO4", "ZNF552")]
##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
                      regulatoryElements2=tfs2, gexpIDs=annot)
##---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
```

mbrPlotDuals

Plot shared target clouds between dual regulons.

Description

This function plots the shared target clouds between a regulon pair.

Usage

```r
mbrPlotDuals(object, names.duals = NULL, filepath = NULL, alpha = 0.8,
             lncols = c("darkgreen", "darkorange3"), lwd = 0.7, Pvalue = FALSE)
```
Arguments

object
A processed object of class MBR evaluated by the method mbrAssociation.

names.duals
A vector with 'dual regulon' identifiers from the 'motifsInformation' table.

filepath
A character string indicating the file path where the plot should be saved.

alpha
The alpha transparency, a number in [0,1].

lncols
A vector of length 2 indicating the colors of the negative and positive target clouds, respectively.

lwd
Line width, a decimal value (between 0 and 1).

Pvalue
A Boolean value that indicates whether the 'dual regulon' p-value will be showed in the plot.

Value
A plot with the shared target clouds between dual regulons.

Examples

data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$expIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNF552")]  
##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1, regulatoryElements2=tfs2, expIDs=annot)
##---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
##---mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
##---mbrDpiFilter
rmbr <- mbrDpiFilter(rmbr)
##---mbrAssociation
rmbr <- mbrAssociation(rmbr, prob=0.75)
##---mbrDuals
rmbr <- mbrDuals(rmbr)
###

dual <- mbrGet(rmbr, what="dualRegulons")[1]
mbrPlotDuals(rmbr, names.duals=dual)  

mbrPreprocess, matrix-method
A preprocessing function for objects of class MBR.

Description

A preprocessing function for objects of class MBR.
Usage

## S4 method for signature 'matrix'
mbrPreprocess(gexp, regulatoryElements1, regulatoryElements2, verbose = TRUE, ...)

Arguments

gexp A numerical matrix, typically with mRNA and/or miRNA expression values.
regulatoryElements1 A named vector with regulatory elements listed in 'gexp' rownames.
regulatoryElements2 A named vector with regulatory elements listed in 'gexp' rownames.
verbose A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).
... Additional arguments passed on to tni.preprocess function.

Value

A preprocessed 'MBR-class' object.

Examples

data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNFI552")]
##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1, regulatoryElements2=tfs2, gexpIDs=annot)

tni2mbrPreprocess,TNI-method

A preprocessing function for objects of class MBR.

Description

This function merges two TNI class objects and creates one MBR class object.

Usage

## S4 method for signature 'TNI'
tni2mbrPreprocess(TNI1, TNI2, regulatoryElements1 = NULL, regulatoryElements2 = NULL, verbose = TRUE)
tni2mbrPreprocess, TNI-method

Arguments

TNI1 A 'TNI' class object.

TNI2 Another 'TNI' class object

regulatoryElements1 A character vector specifying which 'TNI1' regulatory elements should be evaluated. It is obligatory when the 'TNI2' is missing.

regulatoryElements2 A character vector specifying which 'TNI1' regulatory elements should be evaluated. It is obligatory when the 'TNI2' is missing and can not be equal to the 'regulatoryElements1' vector.

verbose A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

An MBR object.

Examples

data("dt4rtn", package = "RTN")
gexp <- dt4rtn$gexp
annot <- dt4rtn$gexpIDs
tfs1 <- dt4rtn$tfs[c("IRF8","IRF1","PRDM1","AFF3","E2F3")]
tfs2 <- dt4rtn$tfs[c("HCLS1","STAT4","STAT1","LMO4","ZNF552")]
##---mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)
##---mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)
##---mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
rmbr <- mbrDpiFilter(rmbr)
##---tni2mbrPreprocess
tni1 <- mbrGet(rmbr, what="TNI1")
tni2 <- mbrGet(rmbr, what="TNI2")
rmbr <- tni2mbrPreprocess(tni1, tni2)
Index

*Topic package

  RTNduals-package, 2

 MBR, 4–10, 12
 MBR (MBR-class), 3
 MBR-class, 2, 3
 mbrAssociation, 2, 3, 7, 10
 mbrAssociation
   (mbrAssociation, MBR-method), 4
 mbrAssociation, MBR-method, 4
 mbrBootstrap, 2, 4, 6
 mbrBootstrap (mbrBootstrap, MBR-method), 5
 mbrBootstrap, MBR-method, 5
 mbrDpiFilter, 2, 4
 mbrDpiFilter (mbrDpiFilter, MBR-method), 6
 mbrDpiFilter, MBR-method, 6
 mbrDuals, 2
 mbrDuals (mbrDuals, MBR-method), 7
 mbrDuals, MBR-method, 7
 mbrGet (mbrGet, MBR-method), 8
 mbrGet, MBR-method, 8
 mbrPermutation, 2, 4–6
 mbrPermutation
   (mbrPermutation, MBR-method), 9
 mbrPermutation, MBR-method, 9
 mbrPlotDuals, 9
 mbrPreprocess, 2, 3
 mbrPreprocess
   (mbrPreprocess, matrix-method), 10
 mbrPreprocess, matrix-method, 10
 RTNduals (RTNduals-package), 2
 RTNduals-package, 2

tni.bootstrap, 5
 tni.dpi.filter, 6
 tni.permutation, 9
 tni.preprocess, 11
 tni2mbrPreprocess, 2, 3
 tni2mbrPreprocess
   (tni2mbrPreprocess, TNI-method), 11