Package ‘RTNduals’

October 18, 2017

Type Package

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

Version 1.0.3

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 (>= 3.4), methods, RTN

Imports graphics, 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 6.0.1

NeedsCompilation no

R topics documented:

`RTNduals-package` *RTNduals: An R/Bioconductor package for analysis of co-regulatory network motifs and inference of 'dual regulons'.*

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

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

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

References

Fletcher M.N.C. et al., *Master regulators of FGFR2 signalling and breast cancer risk.* Nature Communications, 4:2464, 2013.

Castro M.A.A. et al., *Regulators of genetic risk of breast cancer identified by integrative network analysis.* Nature Genetics, 48:12-21, 2016.

`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, infered '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 'duals 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

```
## 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 numeric value, representing a quantile threshold 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".
`pAdjustMethod`	A single character value specifying the p-value adjustment method to be used (see 'p.adjust' function 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 correspoding statistics.

Examples

```
##--- load a dataset for demonstration
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")]

##--- run mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)

##--- run mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)

##--- run mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)

##--- run mbrAssociation
rmbr <- mbrAssociation(rmbr, prob=0.75)
```

`mbrBootstrap,MBR-method`
Inference of consensus transcriptional networks.

Description

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

Usage

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

```
##--- load a dataset for demonstration
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")]

##--- run mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)
```

```
##--- run mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)

##--- run mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
```

---

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

```
## 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

```
##--- load a dataset for demonstration
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")]

##--- run mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)

##--- run mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)

##--- run mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)

##--- run 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

```
## S4 method for signature 'MBR'
mbrDuals(object, supplementaryTable = NULL,
  evidenceColname = NULL, verbose = TRUE)
```

Arguments

`object`	A processed object of class MBR evaluated by the method `mbrAssociation`.
`supplementaryTable`	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 'supplementaryTable'.
`verbose`	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

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

Examples

```
##--- load a dataset for demonstration
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")]

##--- run mbrAssociation
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)
rmbr <- mbrPermutation(rmbr, nPermutations=10)
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)
rmbr <- mbrAssociation(rmbr, prob=0.75)
rmbr <- mbrDuals(rmbr)

##--- check results
results <- mbrGet(rmbr, what="dualsInformation")
```

```
##--- add supplementary evidences
## here we build a 'toy' example using the 'rnorm' function for
## demonstration purposes only!
supplementaryTable <- results[,c("Regulon1","Regulon2")]
supplementaryTable$ToyEvidence <- rnorm(nrow(results))
supplementaryTable

##--- add supplementary evidences with brDuals function
rmbr <- mbrDuals(rmbr, supplementaryTable=supplementaryTable,
evidenceColname = "ToyEvidence")

##--- check updated results
mbrGet(rmbr, what="dualsInformation")
```

`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

```
## 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" and "dualsInformation"

Value

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

Examples

```
##--- load a dataset for demonstration
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")]

##--- run 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

```
## 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

```
##--- load a dataset for demonstration
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")]

##--- run mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)

##--- run 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

```
mbrPlotDuals(object, names.duals = NULL, filepath = NULL, alpha = 0.8,
  lncols = c("darkgreen", "darkorange3"))
```

Arguments

`object`	A processed object of class MBR evaluated by the method `mbrAssociation`.
`names.duals`	A vector with 'dual regulon' indentifiers from the 'dualsInformation' 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.

Value

A plot with the shared target clouds between dual regulons.

Examples

```
##--- load a dataset for demonstration
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")]

##--- run mbrPreprocess
rmbr <- mbrPreprocess(gexp=gexp, regulatoryElements1 = tfs1,
regulatoryElements2=tfs2, gexpIDs=annot)

##--- run mbrPermutation
rmbr <- mbrPermutation(rmbr, nPermutations=10)

##--- run mbrBootstrap
rmbr <- mbrBootstrap(rmbr, nBootstrap=10)

##--- run mbrAssociation
rmbr <- mbrAssociation(rmbr, prob=0.75)

##--- run mbrDuals
rmbr <- mbrDuals(rmbr)

##--- get inferred duals and plot the shared cloud of targets
duals <- mbrGet(rmbr, what="dualRegulons")
```

```
mbrPlotDuals(rmbr, names.duals=duals[1])
```

---

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

```
##--- load a dataset for demonstration
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")]

##--- run 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)
```

Arguments

`TNI1`	A 'TNI' class object.
`TNI2`	Another 'TNI' class object
`regulatoryElements1`	A character vector specifying which 'TNI1' regulatory elements should be evaluated.
`regulatoryElements2`	A character vector specifying which 'TNI2' regulatory elements should be evaluated.
`verbose`	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

An MBR object.

Examples

```
#--- load a dataset for demonstration
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")]

## Not run:

##--- compute a TNI for tfs1
tni1 <- new("TNI", gexp=gexp, transcriptionFactors=tfs1)
tni1 <- tni.preprocess(tni1, gexpIDs=annot)
tni1 <-tni.permutation(tni1)
tni1 <-tni.bootstrap(tni1)

##--- compute a TNI for tfs2
tni2 <- new("TNI", gexp=gexp, transcriptionFactors=tfs2)
tni2 <- tni.preprocess(tni2, gexpIDs=annot)
tni2 <-tni.permutation(tni2)
```

```
tni2 <-tni.bootstrap(tni2)

##--- run tni2mbrPreprocess
rmbr <- tni2mbrPreprocess(tni1, tni2)

## End(Not run)
```

Index