Package ‘wTO’

July 22, 2025

Type Package

Title Computing Weighted Topological Overlaps (wTO) & Consensus wTO Network

Version 2.1

Author Deisy Morselli Gysi, Andre Voigt, Tiago Miranda Fragoso, Eivind Almaas and Katja Nowick.

Maintainer Deisy Morselli Gysi <deisy.ccnr@gmail.com>

Description Computes the Weighted Topological Overlap with positive and negative signs (wTO) networks given a data frame containing the mRNA count/ expression/ abundance per sample, and a vector containing the interested nodes of interaction (a subset of the elements of the full data frame). It also computes the cut-off threshold or p-value based on the individuals bootstrap or the values reshuffle per individual. It also allows the construction of a consensus network, based on multiple wTO networks. The package includes a visualization tool for the networks. More about the methodology can be found at <doi:10.1186/s12859-018-2351-7>.

License GPL-2

LazyData TRUE

Imports data.table, igraph, magrittr, plyr, parallel, som, visNetwork, reshape2, Rfast, HiClimR, methods

Suggests knitr, rmarkdown

RoxygenNote 7.3.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-04 22:40:01 UTC

Contents

CorrelationOverlap *CorrelationOverlap*

Description

This function computes the correlation between Nodes and the Overlapping Nodes of interest.

Usage

```
CorrelationOverlap(Data, Overlap, method)
```

Arguments

Data	data.frame containing the expression data. Nodes on the Rows, Individuals on the Columns. Don't forget to give the names to the Nodes and to the Individuals. Nodes must have the row.names() with the Node Name.
Overlap	A vector containg the names of the Nodes of interest.
method	Spearman ("s", "spearman") or Pearson ("p", "pearson") correlation

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

ExampleGRF *ExampleGRF*

Description

ExampleGRF data.frame containing data.frame containing names of GRFs.

Usage

```
data(ExampleGRF)
```

Format

data.frame 184 lines, 1 column.

`metagenomics_abundance` *metagenomics_abundance*

Description

metagenomics_abundance

Usage

```
data('metagenomics_abundance')
```

Format

data.frame from The USC Microbial Observatory. The data is public available at <https://www.ebi.ac.uk/metagenomics/proje

`Microarray_Expression1` *Microarray_Expression1*

Description

Microarray_Expression1 data.frame containing expression data for 1000 genes and 18 individuals.

Usage

```
Microarray_Expression1
```

Format

data.frame 1000 lines, 18 columns.

`Microarray_Expression2` *Microarray_Expression2*

Description

Microarray_Expression2 data.frame containing expression data for 1000 genes and 18 individuals.

Usage

```
Microarray_Expression2
```

Format

data.frame 1000 lines, 18 columns.

NetVis *NetVis*

Description

Given a set of Nodes and the weight of the edges, a cutoff for the edges, it draws the networks. Returns a list with the nodes and edges attributes. And plots the network.

Usage

```
NetVis(
  Node.1,
  Node.2,
  wTO,
  pval = NULL,
  MakeGroups = FALSE,
  padj = NULL,
  cutoff = list(kind = "Threshold", value = 0.5),
  layout = NULL,
  smooth.edges = T,
  path = NULL,
  Cluster = F,
  legend = T,
  shape = list(shape = "triangle", names = NULL),
  manipulation = F
)
```

Arguments

`Node.1`	Names of the Nodes.1 that are connected to the Nodes.2. It's the output from wTO.Complete or Consensus.
`Node.2`	Names of the Nodes.2 that are connected to the Nodes.1. It's the output from wTO.Complete or Consensus.
`wTO`	weight of the links, the wTO output from wTO.Complete or wTO.Consensus.
`pval`	p-values for the wTO value. By default it is NULL.
`MakeGroups`	algorithm to find clusters. One of the followings: walktrap, optimal, spinglass, edge.betweenness, fast_greedy, infomap, louvain, label_prop, leading_eigen. Default to FALSE.
`padj`	Adjusted p-values for the wTO value. By default it is NULL.
`cutoff`	It's a list containing the kind of cutoff to be used (pval, Threshold or pval.adj)and it's value. Example: cutoff= list(kind = "Threshold", value = 0.5)
`layout`	a layout from the igraph package.
`smooth.edges`	If the edges should be smoothed or not.
`path`	If the graph should be saved specify the name of the file.

`Cluster` TRUE or FALSE if the nodes should be clustered (double click to uncluster).

`legend` TRUE or FALSE if the legend should appear.

`shape` a list shape=list(shape = "triangle", names = NULL), with the shape and the IDs that should have a different shape, shape can be: diamond, star, triangle, triangleDown or square.

`manipulation` TRUE or FALSE if the graph should be editable.

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

Examples

```
## Not run:
 X =  wTO.Complete( k =1, n = 5, Data = Microarray_Expression1,
 Overlap = ExampleGRF$x[1:10], method = "p",  plot = FALSE)
# Plot with the default aguments.
 NetVis(Node.1 = X$wTO$Node.1, Node.2 = X$wTO$Node.2,
 wTO = X$wTO$wTO_sign, cutoff = list(kind =
"Threshold", value = 0.50))

# Plotting just the edges with p-value < 0.05, with straight edges, nodes clustered,
# no legend and mapipulation of the graph enabled.
  NetVis(Node.1 = X$wTO$Node.1, Node.2 = X$wTO$Node.2,
 wTO = X$wTO$wTO_sign, pval = X$wTO$pval_sign,
 padj = X$wTO$pval_sign,
  cutoff= list(kind = "pval", value = 0.05),
  smooth.edges = FALSE,
 Cluster = TRUE, legend = FALSE, manipulation = TRUE)
# Plotting just the edges with wTO > 0.50, no legend and the nodes:
# "ZNF738", "ZNF677" with triagle shape,
# no legend and mapipulation of the graph enabled.
 NetVis(Node.1 = X$wTO$Node.1, Node.2 = X$wTO$Node.2,
 wTO = X$wTO$wTO_sign, pval = X$wTO$pval_sign,
 padj = X$wTO$pval_sign, cutoff= list(kind = "Threshold", value = 0.5),legend = FALSE,
 shape = list(shape = "triangle", names = c("ZNF738", "ZNF677")))


## End(Not run)
```

`wTO` *wTO*

Description

Calculates the weighted topologycal overlap (wTO) between a set of Nodes and the Overlapping nodes. This function implements the method from Nowick (2009).

Usage

```
wTO(A_TF, sign = c("abs", "sign"))
```

Arguments

`A_TF`	Is the weighted adjency matrix (correlation matrix).
`sign`	("abs", "sign") if the user wants to use the absolute correlation or the signed correlation.

Value

A matrix containing the wTO values.

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

References

Katja Nowick, Tim Gernat, Eivind Almaas and Lisa Stubbs (2009) <doi:10.1073/pnas.0911376106>

`wTO.Complete` *wTO.Complete*

Description

Compute the wTO and also the bootstraps. Proposed at: arXiv:1711.04702

Usage

```
wTO.Complete(
  k = 1,
  n = 100,
  Data,
  Overlap = row.names(Data),
  method = "p",
  method_resampling = "Bootstrap",
  pvalmethod = "BH",
  savecor = F,
  expected.diff = 0.2,
  lag = NULL,
  ID = NULL,
  normalize = F,
  plot = T
)
```

Arguments

`k` Number of threads to be used for computing the weight Topological Overlap. Default is set to 1.

`n` Number of resamplings, used to compute the empirical distribuitions of the links. Default is set to 100.

`Data` data.frame containing the count / expression data for the correlation.

`Overlap` Set of nodes of interest, where the Overlapping weights will be computed.

`method` Type of the correlation that should be used. "s" / "spearman" will compute the rank spearman correlation, "p" / "pearson" will compute the linear correlation. If no value is given, the default is to use "p".

`method_resampling`
method of the resampling. Bootstrap, BlockBootstrap or Reshuffle. Bootstrap null hypothesis is that the wTO is random, and Reshuffle tests if the wTO is equal to zero.

`pvalmethod` method to compute the multiple test correction for the pvalue. for more information check the function `p.adjust`.

`savecor` T/F if need to save the correlation.

`expected.diff` Difference expected between the real wTO and resampled wTO By default, it is set to 0.2.

`lag` time dependency, lag, if you are using the BlockedBootstrap.

`ID` ID of the samples for the blocked bootstrap (for repeated measures).

`normalize` T/F Should the data be normalized?

`plot` T/F Should the diagnosis plot be plotted?

Value

a list with results.

- wTO is a data.frame containig the Nodes, the wTO computed using the signed correlations, the pvalue and the adj.pvalue.
- abs.wTO is a data.frame containig the Nodes, the wTO computed using the absolute correlations, the pvalue and the adj.pvalue.
- Correlation is a data.frame containing the correlation between all the nodes.
- Empirical.Quantile quantile values for the empirical distribution.
- Quantile quantile values for the sample distribution.

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

Examples

```
## Not run:
# Using spearman rank correlation and bonferroni correction for the pvalues.
wTO.Complete( k =8, n = 1000, Data = Microarray_Expression1,
 Overlap = ExampleGRF$x, method = "s", pvalmethod = "bonferroni")
 # Changing the resampling method to Reshuffle.
wTO.Complete( k =1, n = 1000, Data = Microarray_Expression1,
Overlap = ExampleGRF$x, method_resampling = "Reshuffle")
 # Changing the resampling method to BlockBootstrap, with a lag of 2.
 row.names(metagenomics_abundance) = metagenomics_abundance$OTU
 metagenomics_abundance = metagenomics_abundance[,-1]
wTO.Complete( k =1, n = 1000, Data = metagenomics_abundance, method = "s",
Overlap = row.names(metagenomics_abundance), method_resampling = "BlockBootstrap", lag = 2)
wTO.Complete( k =2, n = 1000, Data = Microarray_Expression1, method = "s",
Overlap = ExampleGRF$x, method_resampling = "BlockBootstrap", ID = rep(1:9,each = 2))
X = wTO.Complete( k =1, n = 1000, Data = Microarray_Expression1,
Overlap = ExampleGRF$x, method = "p", plot = FALSE)

## End(Not run)
```

---

wTO.Consensus *wTO.Consensus*

Description

Consensus requires a list of data.frame containing the pair of nodes, and the wTO values for all networks that need to be joined. Reference: arXiv:1711.04702

Usage

```
wTO.Consensus(data)
```

Arguments

`data`	list of data.frame containing the "Node.1", "Node.2" and "wTO".

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

Examples

```
## Not run:
EXAMPLE =  wTO.Complete( k =1, n = 200, Data = Microarray_Expression1,
                                       Overlap = ExampleGRF$x, method = "p")

# Constructing the consensus network
data = list(data.frame(Node.1 = EXAMPLE$wTO$Node.1,
Node.2 = EXAMPLE$wTO$Node.2,
wto_sig = EXAMPLE$wTO$wTO_sign,
pvalsig = EXAMPLE$wTO$pval_sig),
data.frame(Node.1 = EXAMPLE$wTO$Node.1,
Node.2 = EXAMPLE$wTO$Node.2,
wtoabs = EXAMPLE$wTO$wTO_abs,
pvalabs = EXAMPLE$wTO$pval_abs) )
CONS = wTO.Consensus(data)


## End(Not run)
```

---

wTO.export *wTO.export*

Description

Exports the significative interactions, the wTO weight and pvalues into a .txt file, tab separeted. This file can be imported in other visualization tools (Cytoscape for example).

Usage

```
wTO.export(DATA, path, sign = TRUE, pvalue = 0.05, padj = 0.05, prop.NA = 0.5)
```

Arguments

DATA	Output from the function wTO.Complete or wTO.Consensus.
path	Path and file name where the .txt file should be saved.
sign	Should the network contain the results for the signed network or unsigned? Only for data coming from wTO.Complete.
pvalue	cutoff p-value for the network. Only for data coming from wTO.Complete.
padj	cutoff adjusted p-value for the network. Only for data coming from wTO.Complete.
prop.NA	cutoff proportion of NAs for the network. Only for data coming from wTO.Consensus.

Examples

```
## Not run:
EXAMPLE =  wTO.Complete( k =1, n = 200, Data = Microarray_Expression2,
                                       Overlap = ExampleGRF$x, method = "p")
wTO.export(EXAMPLE , './EXAMPLE.txt')

#Selection of only the significative ones for the Consensus
Ex_k1_cor_p_boot_p005_sig = subset(EXAMPLE$wTO,
EXAMPLE$wTO$pval_sig < 0.05,
select = c("Node.1", "Node.2", "wTO_sign"))
Ex_k1_cor_p_boot_p005_abs = subset(EXAMPLE$wTO,
EXAMPLE$wTO$pval_abs < 0.05,
select = c("Node.1", "Node.2", "wTO_abs"))
# Constructing the consensus network
```

```
CN = wTO.Consensus(data = list(Ex_k1_cor_p_boot_p005_sig,
Ex_k1_cor_p_boot_p005_abs))
wTO.export(CN, './CN.txt')
### You can store the result on the workspace.
y = wTO.export(CN, './CN.txt')
head(y)

## End(Not run)
```

wTO.fast *wTO.fast*

Description

Compute the wTO and also the bootstraps. Proposed at arXiv:1711.04702. This is a quicker version of the wTO.Complete. It doesn't contain diagnose plots nor a parallel version.

Usage

```
wTO.fast(
  Data,
  Overlap = row.names(Data),
  method = "p",
  sign = "sign",
  delta = 0.2,
  n = 10,
  method_resampling = "Bootstrap",
  lag = NULL,
  ID = NULL
)
```

Arguments

`Data`	data.frame containing the count / expression data for the correlation.
`Overlap`	Set of nodes of interest, where the Overlapping weights will be computed.
`method`	Type of the correlation that should be used. "s" / "spearman" will compute the rank spearman correlation, "p" / "pearson" will compute the linear correlation. If no value is given, the default is to use "p".
`sign`	Should the wTO be signed?
`delta`	expected difference between the real wTO and the bootstraped.
`n`	Number of resamplings, used to compute the empirical distribuitions of the links. Default is set to 100.

`method_resampling`
method of the resampling. Bootstrap or BlockBootstrap.If the second is used, please give the lag (time dependency among the data).

`lag` Time dependency for the blocked bootstrap (for time series).

`ID` ID of the samples for the blocked bootstrap (for repeated measures).

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

Examples

```
 wTO.fast(Data = Microarray_Expression1,
 Overlap = ExampleGRF$x,
 method = "p")

#  For a time series with lag = 4
# wTO.fast(Data = Microarray_Expression1,
# Overlap = ExampleGRF$x,
# method = "p",
# method_resampling = 'BlockBootstrap',
# lag = 4)

# For a study where the individuals were measured multiple times.
# wTO.fast(Data = Microarray_Expression1,
# Overlap = ExampleGRF$x,
# method = "p",
# method_resampling = 'BlockBootstrap',
# ID = rep(1:9, each= 2))
```

`wTO.in.line` *wTO.in.line*

Description

Transforms a correlation matrix into the line format.

Transforms a correlation matrix into the line format.

Usage

```
wTO.in.line(d)
```

Arguments

`d` correlation matrix to be converted into the line format.

Value

the wTO matrix into a data.frame: Node1, Node2 and wTO.

the wTO matrix into a data.frame: Node1, Node2 and wTO.

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

wTO.rep_measure *wTO.rep_measure*

Description

Compute the wTO for a repeated measures expermiment and also the bootstraps. Proposed at arXiv:1711.04702. This is a quicker version of the wTO.Complete. It doesn'T contain diagnose plots nor a parallel version.

Usage

```
wTO.rep_measure(
  Data,
  Overlap = row.names(Data),
  ID,
  sign = "sign",
  delta = 0.2,
  n = 10
)
```

Arguments

`Data` data.frame containing the count / expression data for the correlation.

`Overlap` Set of nodes of interest, where the Overlapping weights will be computed.

`ID` a vector with the individuals identification

`sign` Should the wTO be signed?

`delta` expected difference between the real wTO and the bootstraped.

`n` Number of resamplings, used to compute the empirical distribuitions of the links. Default is set to 100.

Author(s)

Deisy Morselli Gysi <deisy at bioinf.uni-leipzig.de>

Examples

```
#wTO.rep_measure(Data = Microarray_Expression1, ID = rep(c(1:9),2),
#Overlap = ExampleGRF$x)
```

Index