Package 'nettools'

February 20, 2015

Type Package

Title A Network Comparison Framework

Version 1.0.1

Date 2014-09-02

Maintainer Michele Filosi <filosi@fbk.eu>

Depends R (>= 2.14.1), methods

Imports parallel, rootSolve, dtw, WGCNA, minet, Matrix, minerva, combinat, igraph

Description A collection of network inference methods for co-expression networks, quantitative network distances and a novel framework for network stability analysis.

License CC BY-NC-SA 4.0

Author Michele Filosi [aut, cre], Roberto Visintainer [aut], Samantha Riccadonna [aut], Giuseppe Jurman [ctb], Cesare Furlanello [ctb]

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-12 16:16:26

R topics documented:

`nettools-package` *A Network Comparison Framework*

Description

This package provides a collection of network inference methods for co-expression networks, quantitative network distances and a novel framework for network stability analysis.

Details

Summary:

Package:
Type:
Version:
Date:
Depends: R (>= 2.14.1), methods Imports: WGCNA, minet, rootSolve, dtw, Matrix, parallel, minerva, combinat, igraph Licen

Author(s)

Michele Filosi [aut, cre], Roberto Visintainer [aut], Samantha Riccadonna [aut], Giuseppe Jurman [ctb], Cesare Furlanello [ctb]
Maintainer: Michele Filosi <filosi@fbk.eu>

References

M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello (2014) *Stability Indicators in Network Reconstruction, PLOSONE*

P. Langfelder, S. Horvath (2008) *WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 2008, 9:559*

P. E. Meyer, F. Lafitte, G. Bontempi (2008). *MINET: An open source R/Bioconductor Package for Mutual Information based Network Inference. BMC Bioinformatics*. `http://www.biomedcentral.com/1471-2105/9/461`

D. Albanese, M.Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello (2012). *minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers, Bioinformatics 2012*. `http://mpba.fbk.eu/cmine`

D. Reshef, Y. Reshef, H. Finucane, S. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, P. Sabeti. (2011) *Detecting novel associations in large datasets*. Science 334, 6062

`http://www.exploredata.net`

(SOM: Supplementary Online Material at http://www.sciencemag.org/content/suppl/2011/12/14/334.6062.1518.DC1)

See Also

Further information on specific functions for inferring the adjacency matrix could be found in **WGCNA**, **minerva** and **minet**.

mat2adj *From Data Matrix to Adjacency Matrix*

Description

mat2adj is a high level function providing different network inference methods. The function takes in input a data matrix *N by P*, with N samples on the rows and P variables on the columns. The adjacency matrix *P by P* will be computed with the specified method, using N samples to infer the interactions between the variables.

Usage

```
mat2adj(x,...)

## Default S3 method:
mat2adj(x, ...)


## S3 method for class 'data.frame'
mat2adj(x, ...)


## S3 method for class 'matrix'
mat2adj(x,method='cor', FDR=1e-3, P=6, measure=NULL,
alpha=0.6, C=15,DP=1, ...)
```

Arguments

x a matrix or data.frame of numerical values of N rows and P columns

method a character string indicating which method will be used for inferring a relationship between two variables. This must be (an abbreviation of) one of "cor" (default), "WGCNA", "WGCNAFDR", "bicor", "bicorFDR", "TOM", "ARACNE", "CLR", "MINE", "MINEFDR", "DTWMIC"

P 6 (default), integer used as soft-thresholding power for network construction, used by the "WGCNA" and "TOM" methods.

FDR 1e-3 (default), a number which indicates the number of values generated to compute the NULL hypothesis. To be used for methods "WGCNAFDR", "MINEFDR" and "bicorFDR"

measure "MIC" (default), a valid string indicating the measure of the MINE suite to compute. One of "MIC", "MCN", "MEV", "MAS" or "MICR2".

alpha 0.6 (default), the alpha argument to be passed to the function mine. See also mine

C 15 (default), an integer value to be passed at the mine function main. Only for methods "MINE" and "MINEFDR".

DP 1 (default), only for method "DTWMIC".

... Additional arguments to be passed to the downstream functions. Normally the argument passed through ... are processed by the functions which compute the inference. Not all parameters are used by all functions.

Details

mat2adj function is a high-level function which includes different methods for network inference. In particular the function infer the relation between all the possible pairwaise comparison between columns in the dataset. If the input is a data.frame, columns were first converted into a numerical matrix. Given a N by P numerical matrix, the relation between each PxP pairs of variables is inferred with the selected method.

The "FDR" corrected methods are based on a permutation estimate of the null hypothesis. A total amount of 1/("FDR") permutations are performed to asses the reliability of the inferred link; each link is set only if it is inferred in all the permutations and its weight is lower then the value on non permuted data. The default value for FDR is 1e-3.

All the available methods are the following:

cor (default) computes the interaction using the 'Pearson' correlation coefficient. Different correlation methods, such as Spearman could be passed to the function using ....

ARACNE Algorithm for the Reconstruction of Gene Regulatory Networks, see also package **minet**

CLR Context Likelihood of Relatedness see also package **minet**

WGCNA WeiGhted Correlation Network Analsysis. It is based on a correlation measure. For further details see the documentation of **WGCNA** package. The method accept parameter P which is set to 6 by default

bicor Biweighted Correlation method. It uses a biweighted correlation as described in bicor package

TOM Topological Overlap Measure inference method. For further details see the documentation of **WGCNA** package. As for WGCNA the parameter P can be set(6 by default).

MINE Maximum Information-based Non-parametric Exploration. This method uses the **minerva** implementation of the original measure. For this methods different measures are available. See **minerva** for further information. To clarify the main MINE family statistics let $D = (x, y)$ be the set of *n* ordered pairs of elements of x and y. The data space is partitioned in an *X-by-Y* grid, grouping the *x* and *y* values in *X* and *Y* bins respectively.
The value of alpha (default 0.6) has been empirically chosen by the authors of the original paper.$alpha$ is the exponent of the search-grid size $B(n) = n^{\alpha}$. It is worthwhile noting that

`alpha` and `C` are defined to obtain an heuristic approximation in a reasonable amount of time. In case of small sample size (*n*) it is preferable to increase `alpha` to 1 to obtain a solution closer to the theoretical one.
`C` determines the number of starting point of the X-by-Y search-grid. When trying to partition the x-axis into X columns, the algorithm will start with at most C x X clumps. Default value is 15.
The **Maximal Information Coefficient (MIC)** is defined as

$$\text{MIC}(D) = \max_{XY<B(n)} M(D)_{X,Y} = \max_{XY<B(n)} \frac{I^*(D,X,Y)}{log(\min X, Y)},$$

where $B(n) = n^\alpha$ is the search-grid size, $I^*(D,X,Y)$ is the maximum mutual information over all grids *X-by-Y*, of the distribution induced by D on a grid having *X* and *Y* bins (where the probability mass on a cell of the grid is the fraction of points of D falling in that cell). The other statistics of the MINE family are derived from the mutual information matrix achieved by an *X-by-Y* grid on D. The **Maximum Asymmetry Score (MAS)** is defined as

$$\text{MAS}(D) = \max_{XY<B(n)} |M(D)_{X,Y} - M(D)_{Y,X}|.$$

The **Maximum Edge Value (MEV)** is defined as

$$\text{MEV}(D) = \max_{XY<B(n)} \{M(D)_{X,Y} : X = 2\tilde{}or\tilde{}Y = 2\}.$$

The **Minimum Cell Number (MCN)** is defined as

$$\text{MCN}(D,\epsilon) = \min_{XY<B(n)} \{\log(XY) : M(D)_{X,Y} \geq (1-\epsilon)MIC(D)\}.$$

More details are provided in the supplementary material (SOM) of the original paper.

`MINEFDR` This calls an FDR corrected version of the standard MINE method. See the description for the `MINE` method. Parameter `FDR=1e-3` (default) can be set.

`bicorFDR` This calls an FDR corrected version of the `bicor` method. See the description for the `bicor`. Parameter `FDR=1e-3` (default) can be set.

`WGCNAFDR` This calls an FDR corrected version of the `WGCNA` method. Parameter `P` cannot be set for this method. Parameter `FDR=1e-3` (default) can be set.

`DTWMIC` This method uses Dynamic Time Warping transformation coupled witht the MIC statistic from the MINE family. See Details for further information. Additional parameters can be set with this method:

... `tol` 1e-5 (default), a numeric value which controls the tolerance on the variable variance. In particular this parameter is passed to a function which controls the variance of each feature. The function returns the indexes of the features with variance `<tol`. Indexes refers to 1-based column numbers of the original dataset.

`var.thr` 1e-5 (default), a numeric value which controls the tolerance parameter on the column variance for the method `MINE, MINEFDR, DTWMIC`.

Value

A P by P symmetric adjacency matrix with the diagonal set to 0. Self loop and direction of the edges are not taking into account. The values range in [0, 1].

Author(s)

Michele Filosi
Special thanks to: Samantha Riccadonna, Giuseppe Jurman, Davide Albanese and Cesare Furlanello

References

P. Langfelder, S. Horvath (2008) *WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics 2008, 9:559*

P. E. Meyer, F. Lafitte, G. Bontempi (2008). *MINET: An open source R/Bioconductor Package for Mutual Information based Network Inference. BMC Bioinformatics*

http://www.biomedcentral.com/1471-2105/9/461

Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, Timothy S Gardner. *Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles*

D. Albanese, M.Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello (2013). *minerva and minepy: a C engine for the MINE suite and its R, Python and MATLAB wrappers, Bioinformatics*

http://mpba.fbk.eu/cmine

M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello (2014)*Stability Indicators in Network Reconstruction, PLOSONE*

D. Reshef, Y. Reshef, H. Finucane, S. Grossman, G. McVean, P. Turnbaugh, E. Lander, M. Mitzenmacher, P. Sabeti. (2011) *Detecting novel associations in large datasets* Science

http://www.exploredata.net

(SOM: Supplementary Online Material at http://www.sciencemag.org/content/suppl/2011/12/14/334.6062.1518.DC1)

See Also

WGCNA, minerva, minet, cor

Examples

```
## Not run:
data(Spellman, package="minerva")
dim(Spellman)
A <- mat2adj(Spellman,method="cor", n.cores=1)
dim(A)

## End(Not run)
```

`netdist` *Distances between network*

Description

This function computes the distance between two adjacency matrix given as matrices or igraph objects.

Usage

```
netdist(x,...)

## S3 method for class 'matrix'
netdist(x, h, d = "HIM", ga = NULL, components=TRUE, ...)

## S3 method for class 'Matrix'
netdist(x, h, d = "HIM", ga = NULL, components=TRUE, ...)

## S3 method for class 'list'
netdist(x, d = "HIM", ga = NULL, components=TRUE, ...)

## S3 method for class 'igraph'
netdist(x, h, d = "HIM", ga = NULL, components=TRUE,...)
```

Arguments

`x` Adjacency matrix, igraph object or list of adjacency matrices/igraph objects

`h` Adjacency matrix or igraph object. Only when `x` is a matrix object.

`d` `HIM` (default), character string containing a valid method. Accepted values are: `"IM"`, `"ipsen"`, `"Ipsen"`, `"IpsenMikhailov"`, or `"Ipsen-Mikhailov"`, for Ipsen-Mikhailov distance and `"H"`, `"hamming"`, or `"Hamming"` for Hamming distance.

`ga` `NULL` (default), a numeric value for the gamma parameter used only if method `ipsen` is called. If `ga=NULL` an optimal value based on the number of nodes in the adjacency matrix is computed.

`components` `TRUE` (default), to obtain the components (Hamming and Ipsen-Mikhailov) for `method=HIM`; otherwise the function returns only the HIM value. If `method` is set to `hamming` or `ipsen` it will be ignored.

`...` Additional arguments to be passed to the downstream functions. Normally the argument passed through ... are processed by the functions which compute the distance. Not all parameters are used by all functions.

n.cores NULL (default), an integer number of available cores to run run the computation. This parameter used only when method is `ipsen` or `HIM`. If method `list` is called, each core take care of the computation of eigenvalues for a matrix in the input list. If `n.cores` is bigger than the number of available cores on the specific machine, than `n.cores` is set to the number of available cores - 1.

verbose `FALSE` (default), print information about the state of the computation

Details

`netdist` is a high level function which provide an interface to different network distance methods. Distance can be computed for weighted and unweighted networks either directed or undirected. Each network must have the same number of nodes, otherwise no computation will be performed and an errore message will be output.

Parallel computation is provided automatically through the **parallel** package included by default from R 2.15, only for methods `"ipsen"` and `"HIM"`. The computation can be automatically parallelized on a multi-cpu computer using the parameter `n.cores`.

Input can be either an adjacency matrix, a graph object from **igraph** package or a list of adjacency matrix or graph objects.

If the 'weight' attribute is given within the `graph` object it will be used as edge weight, otherwise binary representation of the network will be considered. In case edge weights are provided through the adjacency matrix, they should be scaled between 0-1. If some of the weights are outside the interval a scaling function is automatically applied.

The parameter `ga` is used only in the *Ipsen* and *HIM* distance. We suggest not to change it and use the automatic computation of the parameter based on the number of nodes in the network.

Value

For `components` set to `FALSE` the distance between the input networks is returned. If `components` is `TRUE` and method is `HIM` a named vector with 3 values is returned, respectively the *Hamming*(`H`) distance, the *Ipsen-Mikhailov* `IM` distance and the *HIM* distance. If a `list` object is provided as input a distance matrix with all the possible pairwise comparison is returned

Author(s)

M. Filosi, S. Riccadonna

References

G. Jurman, R. Visintainer, M. Filosi, S. Riccadonna, C. Furlanello *The HIM glocal metric and kernel for network comparison and classification arXiv 2013, arXiv:1201.2931v3*

Examples

```
a <- matrix(rnorm(200), nrow=20)
b <- matrix(rnorm(200), nrow=20)

aa <- mat2adj(a, method="cor", n.cores=1)
bb <- mat2adj(b, method="cor", n.cores=1)

dd <- netdist(aa,bb, d="HIM", n.cores=1)
print(dd)
```

```
## Use a different gamma parameter for HIM distance
dd <- netdist(aa,bb, d="HIM", ga=0.5, n.cores=1)
print(dd)

## The function can accept also list of adjacency matrices as input
alist <- list(aa,bb)
ddist <- netdist(alist, d="HIM", n.cores=1)

## Use of multiple cores
## Not run:
dd <- netdist(aa,bb,n.cores=4)

## End(Not run)
```

netSI *Network Stability Indicators*

Description

Family NetSI of stability indicators for network inference methods. The function computes a suite of 4 stability indicators of data x based on network distances.

Usage

```
netSI(x,indicator="all", d='HIM', adj.method='cor',
method="montecarlo", k=3, h=20, n.cores, save=FALSE, verbose=TRUE, ...)
```

Arguments

x	numerical matrix or data.frame to be used for network inference.
indicator	all (default) character string. Indicating all or one of the four indicators available. This must be on of the strings "S","SI", "Sw" or "Sd".
d	HIM (default), character string indicating the distance to be used for the stability computation in case of indicators S and SI.
adj.method	"cor" (default), optional character string giving one of the methods available for network inference. For details see mat2adj.
method	"montecarlo" (default), a string identifying the cross validation schema to be used for the stability computation. This must be one of the strings montecarlo (default), LOO for Leave One Out or kCV for k-fold Cross Validation.
k	3 (default) numeric value indicating the number of subset the data will be splitted in. In case method is set to montecarlo the 1-1/k samples are taken for the computation. In case cvlab is set to loo it should be set to 1, otherwise it will be automatically set to 1. In case cvlab is set to kCV h iteration are performed, each dividing the data into randomly chosen k groups

`h`	20 (default), numeric value indicates the number of iterations the `montecarlo` and `kCV` procedure should be repeated(cross validations).
`n.cores`	`NULL` (default), optional numeric value giving the number of cores to be used for parallel computation. The default behaviour try to asses the maximum number of available cores and set a parallel computation using the maximum number of cores - 1 for the computation.
`save`	`FALSE` (default), logical. Should adjacency matrices and indicator's raw value from resampling scheme be saved?
`verbose`	`TRUE` (default), logical. Print all the information about the computation. If `FALSE` messages will be suppressed.
`...`	Additional arguments to be passed to the downstream functions. Not all parameters are used by all functions. See function `netdist` and `mat2adj`. **sseed:** random number generation seed

Details

This function computes a suite of 4 stability indicators. It is based on network inference method function `mat2adj` and network distance function `netdist`. Parameters passed through ... will be used by the low-level functions `mat2adj` and `netdist`.

Indicators:

- `S`: is the global stability indicator. It asses the pertubations of the network given by the resampling: in particular it account for the distance between the network inferred using all samples and the network inferred on a subsampling.
- `SI`: is the local stability indicator. It provides a measure of the perturbations between different subsamplings. It computes the pairwise distances between networks inferred on different subsamplings.
- `Sw`: is the edge stability indicator. It asses the stability of the edges inferred in different subsamplings. In case of a binary network it take into account only the presence/absence of the link, if a "weight" attribute is provided it asses the perturbation of the edge weight given the presence/absence on the given subsampling.
- `Sd`: is the degree stability indicator. It asses the variations of the node degree given a subsampling perturbation on the data.

Parallel computation is provided automatically through the functions available in the **parallel** package included by default from R 2.15. Support for parallel computation is available only on multicore machines not on high performace computing facility. If `n.cores` is set to `NULL` it searches for multiple cores on the machine, if it finds more than 2 cores the computation is parallelized using the maximum number of cores available - 1. If `n.cores` is a numeric value it checks if the number of cores provided is suitable for the machine where R is running. If the number of cores requested is bigger than the available cores `n.cores` will be set to `NULL`, otherwise the computation will be equally distributed on the number of cores provided. In case `indicator` is set to `all` or `SI`, parameter `n.cores` will pass to the low level function which compute the mutual distances, netdist.

Value

A list with the followind elements: If `save=TRUE`:

`call` character. Function call

`ADJlist` A list with the adjacency matrices computed in the resampling scheme.

`S` The value of the global indicator "S"

`SI` The value of the global internal indicator "SI"

`Sw` The value of the edge weight stability indicator "Sw"

`Sd` The value of the node stability indicator "Sd"

`S_boot` A numerical vector with the value of the stability indicator S for each resampling.

`SI_boot` A numerical vector with the value of the stability indicator SI for each combination of pairs within all the resampling.

`Sw_boot` A numerical matrix. Each row contains the value of the weight of a particular edge on different resampling splits.

`Sd_boot` A numerical matrix number_of_resampling X number_of_nodes. Each column contain the degree of a node on different resampling splits.

If `save=FALSE` elements `call`, `ADJlist`, `ADJ`, `S_boot`, `SI_boot`, `Sw_boot` and `Sd_boot` will not be provided. For `S` and `SI` the mean over the resampling splits is computed, instead for the element `Sw` and `Sd` the range/mean is computed over resamplings.

Author(s)

Michele Filosi, Samantha Riccadonna

References

M. Filosi, R. Visintainer, S. Riccadonna, G. Jurman, C. Furlanello (2014)*Stability Indicators in Network Reconstruction, PLOSONE*

See Also

`netdist`, `mat2adj`, `WGCNA`, `minet`

Examples

```
myrawdata <- matrix(rnorm(200),ncol=20)

sstab <- netSI(myrawdata, d=”HIM”, n.cores=1, save=FALSE)
print(sstab$S)
print(sstab$SI)
```

Index