

Package: metanetwork (via r-universe)

October 27, 2024

Type Package

Title Handling and Representing Trophic Networks in Space and Time

Version 0.8.0

Date 2022-12-08

Description A toolbox to handle and represent trophic networks in space or time across aggregation levels. This package contains a layout algorithm specifically designed for trophic networks, using dimension reduction on a diffusion graph kernel and trophic levels. Importantly, this package provides a layout method applicable for large trophic networks. The package also implements network diversity indices at different aggregation levels and connectance computation.

License GPL-3

URL <https://github.com/MarcOhlmann/metanetwork>,
<https://marcohlmann.github.io/metanetwork/>

NeedsCompilation no

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

BugReports <https://github.com/MarcOhlmann/metanetwork/issues>

Imports GGally, network, ggplot2, intergraph, dplyr, igraph, Matrix,
visNetwork, grDevices, RColorBrewer, magrittr, ggimage, rlang,
sna

Suggests spelling, testthat, covr, knitr, markdown, rmarkdown

Language en-US

Config/testthat/edition 3

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

RemoteUrl <https://github.com/marcohlmann/metanetwork>

RemoteRef HEAD

RemoteSha 1d3a7b6659de085bb5b289666267861f5bd32352

Contents

append_agg_nets	2
attach_layout	3
build_metanet	5
compute_dis	6
compute_div	8
compute_metrics	10
compute_TL	12
diff_plot	13
extract_networks	15
ggmetanet	15
ggnet.default	18
group_layout.default	19
is.metanetwork	19
metanet_build_pipe	20
metanet_pipe	21
meta_angola	22
meta_norway	23
meta_vrtb	23
plot_trophicTable	24
print	25
TL_tsne.default	26
vismetaNetwork	26
visNetwork.default	28
%>%	28
Index	29

append_agg_nets	<i>append aggregated networks</i>
-----------------	-----------------------------------

Description

Method to append aggregated metawebs and local networks using the hierarchy described in trophicTable

Usage

```
append_agg_nets(metanetwork)
```

```
## S3 method for class 'metanetwork'
append_agg_nets(metanetwork)
```

Arguments

metanetwork object of class 'metanetwork'

Details

It uses the network aggregation method developed in Ohlmann et al. 2019. It computes group abundances and edge probabilities of the aggregated networks.

Value

an object of class 'metanetwork', with aggregated networks appended to the network list.

NULL

References

Ohlmann, M., Miele, V., Dray, S., Chalmandrier, L., O'connor, L., & Thuiller, W. 2019. Diversity indices for ecological networks: a unifying framework using Hill numbers. *Ecology letters*, 22 4 , 737-747.

See Also

[plot_trophicTable\(\)](#)

Examples

```
library(metanetwork)
data(meta_angola)
meta_angola = append_agg_nets(meta_angola)
names(meta_angola)
```

attach_layout

compute and attach metanetwork layouts

Description

Method to compute 'TL-tsne' and 'group-TL-tsne' layouts and save it as node attributes of the focal network.

Usage

```
attach_layout(
  metanetwork,
  g = NULL,
  beta = 0.1,
  mode = "TL-tsne",
  TL_tsne.config = TL_tsne.default,
  res = NULL,
  group_layout.config = group_layout.default
)

## S3 method for class 'metanetwork'
```

```
attach_layout(
  metanetwork,
  g = NULL,
  beta = 0.1,
  mode = "TL-tsne",
  TL_tsne.config = TL_tsne.default,
  res = NULL,
  group_layout.config = group_layout.default
)
```

Arguments

metanetwork	object of class 'metanetwork'
g	character indicating the name of the network for which the 'TL-tsne' layout is computed, default is 'metaweb'
beta	the diffusion parameter of the diffusion kernel, a positive scalar controlling the squeezing of the network, default is 0.1
mode	'TL-tsne' or 'group-TL-tsne', default is 'TL-tsne'.
TL_tsne.config	configuration list for mode 'TL-tsne', default is TL_tsne.default
res	resolution for the 'group-TL-tsne' layout
group_layout.config	configuration list for mode 'group-TL-tsne', default is group_layout.default

Details

The 'TL-tsne' layout is a diffusion based layout algorithm specifically designed for trophic networks. In metanetwork, first axis is the trophic level (see compute_TL method) whereas the second axis is computed using a diffusion graph kernel (Kondor & Lafferty 2002) and tsne dimension reduction algorithm to (see van der Maaten & Hinton (2008) and 'tsne' R package). Let A be the adjacency matrix of the considered network and D its degree diagonal matrix. The Laplacian matrix of the symmetrised network is defined by:

$$L = D - A - t(A)$$

The diffusion graph kernel is:

$$K = \exp(-beta * L)$$

It is a similarity matrix between nodes according to a diffusion process. beta is the diffusion constant, it must be provided by the user. beta parameter influences the layout by grouping together similar paths (see pyramid vignette). Each node of the focal network has an attribute layout_beta_VALUE. If this function is run several times for a given beta value, repetitions of the layout algorithm will be stored as node attributes.

The 'group-TL-tsne' layout is a variation of 'TL-tsne' layout. For a focal network, it mixes 'TL-tsne' layout at the desired aggregated level with the layout_with_graphopt function from igraph. It clusters nodes belonging to the same group. 'group-TL-tsne' layout is recommended for large networks since you only need to compute 'TL-tsne' at the aggregated network that is

much smaller than the focal network. `group_layout.config` allows controlling the overall size of the groups.

Value

an object of class 'metanetwork', with the computed layout added as node attribute of the considered network

NULL

References

Kondor, R. I., & Lafferty, J. (2002, July). Diffusion kernels on graphs and other discrete structures. In Proceedings of the 19th international conference on machine learning (Vol. 2002, pp. 315-322).
Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

See Also

[ggmetanet\(\)](#), [vismetaNetwork\(\)](#), [group_layout.default](#)

Examples

```
library(metanetwork)
library(igraph)
# on angola dataset (metaweb)
data("meta_angola")
meta_angola = attach_layout(meta_angola,beta = 0.05)
V(meta_angola$metaweb)$layout_beta0.05
```

build_metanet

Build metanetwork object

Description

Build metanetwork object

Usage

```
build_metanet(
  metaweb,
  abTable = NULL,
  trophicTable = NULL,
  compute_local_nets = TRUE
)
```

Arguments

metaweb	metaweb of the metanetwork, object of class 'graph', 'matrix', 'data.frame' or 'dgCMatrix'. Metaweb needs to be directed and connected. This argument must be non-null.
abTable	node abundances in local networks, matrix of class 'matrix', columns must have names corresponding to node labels of the metaweb, rows are node abundances in local networks. Default is null, in that case, uniform abundances are assigned
trophicTable	a 'matrix' or 'data.frame' indicating hierarchy of the nodes. Names of the columns correspond to the different resolutions. It indicates the membership of each node of the metaweb. Default is null.
compute_local_nets	a boolean, indicates whether local networks must be computed or not. Default is TRUE

Value

an object of S3 class 'metanetwork'

Examples

```
library(metanetwork)
library(igraph)
#with a single metaweb
g = igraph::make_ring(5,directed = TRUE)
meta = build_metanet(g)

#on Angola dataset (re-building the dataset)
data("meta_angola")
metaweb = meta_angola$metaweb
abTable = meta_angola$abTable
trophicTable = meta_angola$trophicTable
meta_angola = build_metanet(metaweb,abTable,trophicTable)
print(meta_angola)
```

compute_dis

compute pairwise network dissimilarity indices

Description

Function to compute pairwise network dissimilarity indices based on Hill numbers following the method described in Ohlmann et al. 2019

Usage

```
compute_dis(metanetwork, q = 1, res = NULL, ncores = 4)
```

Arguments

metanetwork	object of class 'metanetwork'
q	viewpoint parameter controlling the weight given to abundant species/groups and links, default is 1
res	a vector containing the resolutions at which the diversities are computed
ncores	number of cores used for the computation, default is 4

Details

This function compute pairwise dissimilarity indices using Hill numbers on node and link abundances. Importantly, a viewpoint parameters q allows giving more weight to abundant nodes/links. Given a network, we note p_q the abundance of node q (stored as node attribute ab) and π_{ql} interaction probability between nodes q and l (stored as edge attribute weight). The link abundance L_{ql} between nodes q and l is then:

$$L_{ql} = \pi_{ql} p_q p_l$$

Node diversity (for $q = 1$) is then computed as:

$$D(p) = \exp\left(\sum_q -p_q \log p_q\right)$$

Link diversity is computed as:

$$D(L) = \exp\left(\sum_{ql} -\frac{L_{ql}}{C} \log L_{ql} C\right)$$

where C is the weighted connectance

$$C = \sum_{ql} \pi_{ql} p_q p_l$$

For $q = 1$, the pairwise node dissimilarity indices are computed from pairwise diversities as:

$$\delta_P = \frac{\log(G_P) - \log(A_P)}{\log 2}$$

where G_P is node γ -diversity and A_P the node α -diversity

Pairwise link diversity is:

$$\delta_L = \frac{\log(G_L) - \log(A_L)}{\log 2}$$

where G_P is the link γ -diversity and A_P the link α -diversity

For more details on α -, β - and γ -diversity, see Ohlmann et al. 2019.

Value

a list of data.frame containing node and link pairwise dissimilarities

References

Ohlmann, M., Miele, V., Dray, S., Chalmandrier, L., O'connor, L., & Thuiller, W. (2019). Diversity indices for ecological networks: a unifying framework using Hill numbers. *Ecology letters*, 22(4), 737-747.

See Also

[compute_div\(\)](#)

Examples

```
library(metanetwork)
library(igraph)

#on angola dataset
data(meta_angola)
compute_dis(meta_angola,q = 1,ncores = 1)

#computing dissimilarities only at Phylum level
compute_dis(meta_angola,q = 1,res = "Phylum",ncores = 1)
```

compute_div	<i>compute network diversity indices</i>
-------------	--

Description

Function to compute network diversity indices based on Hill numbers following the method described in Ohlmann et al. 2019

Usage

```
compute_div(metanetwork, q = 1, res = NULL)
```

Arguments

metanetwork	object of class 'metanetwork'
q	viewpoint parameter controlling the weight given to abundant species/groups and links, default is 1
res	a vector containing the resolutions at which the diversities are computed

Details

This function compute diversity indices using Hill numbers on node and link abundances. Importantly, a viewpoint parameters q allows giving more weight to abundant nodes/links. Given a network, we note p_q the abundance of node q (stored as node attribute `ab`) and π_{ql} interaction probability between nodes q and l (stored as edge attribute `weight`). The link abundance L_{ql} between nodes q and l is then:

$$L_{ql} = \pi_{ql} p_q p_l$$

Node diversity (for $q = 1$) is then computed as:

$$D(p) = \exp\left(\sum_q -p_q \log p_q\right)$$

Link diversity is computed as:

$$D(L) = \exp\left(\sum_{ql} -\frac{L_{ql}}{C} \log L_{ql} C\right)$$

where C is the weighted connectance

$$C = \sum_{ql} \pi_{ql} p_q p_l$$

The overall α -diversity in node abundances is (for $q = 1$):

$$A_P = \exp\left(\sum_{q=1}^Q \sum_k -\frac{P_{qk}}{P_{++}} \log\left(\frac{P_{qk}}{P_{++}}\right) - \log(K)\right)$$

The overall α -diversity in node abundances is (for $q = 1$):

$$A_P = \exp\left(\sum_{q=1}^Q \sum_{k=1}^K -\frac{P_{qk}}{P_{++}} \log\left(\frac{P_{qk}}{P_{++}}\right) - \log(K)\right)$$

where $P_{++} = \sum_k \sum_q P_{qk}$. The overall α -diversities in link abundances and link probabilities are equal to:

$$A_L = \exp\left(\sum_{q,l=1}^Q \sum_{k=1}^K -\frac{L_{qlk}}{L_{++++}} \log\left(\frac{L_{qlk}}{L_{++++}}\right) - \log(K)\right)$$

where $L_{++++} = \sum_k \sum_{q,l} L_{qlk}$. The γ -diversity in node and link abundances are (for $q = 1$):

$$G_P = \exp\left(\sum_{q=1}^Q -\frac{P_{q+}}{P_{++}} \log\left(\frac{P_{q+}}{P_{++}}\right)\right)$$

$$G_L = \exp\left(\sum_{q,l=1}^Q -\frac{L_{ql+}}{L_{++++}} \log\left(\frac{L_{ql+}}{L_{++++}}\right)\right)$$

The β -diversity is then defined in a multiplicative way:

$$B_P = \frac{G_P}{A_P}$$

$$B_L = \frac{G_P}{A_L}$$

For more details on α -, β - and γ -diversity, see Ohlmann et al. 2019.

Value

a data.frame

References

Ohlmann, M., Miele, V., Dray, S., Chalmandrier, L., O'connor, L., & Thuiller, W. (2019). Diversity indices for ecological networks: a unifying framework using Hill numbers. *Ecology letters*, 22(4), 737-747.

See Also

[compute_dis\(\)](#)

Examples

```
library(metanetwork)
library(igraph)

#on angola dataset
data("meta_angola")
compute_div(meta_angola,q = 1)

#computing diversities only at Phylum level
compute_div(meta_angola,q = 1,res = "Phylum")
```

compute_metrics	<i>compute network metrics</i>
-----------------	--------------------------------

Description

Function to compute (some) network metrics on the metaweb and local networks

Usage

```
compute_metrics(metanetwork, res = NULL)
```

Arguments

metanetwork	object of class 'metanetwork'
res	a vector containing the resolutions at which the metrics are computed

Details

This function computes network metrics for the metaweb and local networks at the available resolutions. This function computes the following metrics:

- weighted connectance: connectance weighted by abundances and link probability (see Ohlmann et al. 2019) Denoting p_q abundance of node q and π_{ql} the interaction probability between nodes q and l , then the weighted connectance C is:

$$C = \sum_{q,l} \pi_{ql} p_q p_l$$

- mean_TL and max_TL: mean and max trophic level of each network, using values computed from compute_TL() method.
- mean_shortest_path_length: mean shortest path length of each network computed using igraph mean_distance() function.

Value

a data.frame

References

- Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. InterJournal, complex systems, 1695(5), 1-9.
- Ohlmann, M., Miele, V., Dray, S., Chalmardrier, L., O'connor, L., & Thuiller, W. (2019). Diversity indices for ecological networks: a unifying framework using Hill numbers. Ecology letters, 22(4), 737-747.

See Also

[compute_TL\(\)](#), [compute_div\(\)](#)

Examples

```
library(metanetwork)
library(igraph)

#on angola dataset
data("meta_angola")
compute_metrics(meta_angola)

#computing metrics only at Phylum level
compute_metrics(meta_angola,res = "Phylum")
```

compute_TL	<i>compute trophic levels</i>
------------	-------------------------------

Description

Method to compute trophic levels using graph Laplacian using the method described in MacKay et al 2020.

Usage

```
compute_TL(metanetwork)

## S3 method for class 'metanetwork'
compute_TL(metanetwork)
```

Arguments

metanetwork object of class 'metanetwork'

Details

Let A be the adjacency matrix of the considered network and D its degree diagonal matrix. The Laplacian matrix of the symmetrised network is defined by:

$$L = D - A - t(A)$$

With $v = indegree(G) - outdegree(G)$ the imbalance degree vector, the trophic level x is defined as the solution of:

$$Lx = v$$

For a connected network, the solution is unique up to a translation. We then fix the minimum trophic level value at 0 thus fixing trophic levels of all others species. Local networks may be disconnected due to sampling effect. In that case, we fix the minimum value on each connected component.

Value

an object of class 'metanetwork', with computed trophic levels stored as node attribute TL
 NULL

References

MacKay, R. S., Johnson, S., & Sansom, B. (2020). How directed is a directed network?. Royal Society open science, 7(9), 201138.

Examples

```

library(metanetwork)
library(igraph)

#on angola dataset
data(meta_angola)
meta_angola = compute_TL(meta_angola)
V(meta_angola$metaweb)$TL

```

diff_plot

*plot difference network***Description**

Function to represent difference between two networks belonging to a metanetwork with specific layout ('TL-tsne' or group 'TL-tsne') using either 'ggnet' or 'visNetwork' visualisation. This function represent the difference between g1 and g2 (g1-g2).

Usage

```

diff_plot(
  metanetwork,
  g1,
  g2,
  beta = 0.1,
  mode = "TL-tsne",
  vis_tool = "ggnet",
  edge_thrs = NULL,
  edge_diff_thrs = NULL,
  layout_metaweb = FALSE,
  flip_coords = FALSE,
  alpha_per_group = NULL,
  alpha_per_node = NULL,
  TL_tsne.config = TL_tsne.default,
  nrep_ly = 1,
  ggnet.config = ggnet.default,
  visNetwork.config = visNetwork.default
)

```

Arguments

metanetwork	object of class 'metanetwork'
g1	network (of class 'igraph') of metanetwork
g2	network (of class 'igraph') of metanetwork
beta	the diffusion parameter of the diffusion kernel, a positive scalar controlling the squeezing of the network

mode	mode used for layout, either 'TL-tsne' or 'group-TL-tsne' (see <code>attach_layout()</code>). Default is 'TL-tsne'
vis_tool	a character indicating the visualisation tool, either 'ggnet' or <code>visNetwork</code>
edge_thrs	if non-null, a numeric (between 0 and 1) indicating an edge weight threshold for the representation
edge_diff_thrs	if non-null, a numeric (between 0 and 1) indicating an edge weight difference threshold under which edge weights of the two networks are considered equal
layout_metaweb	a boolean indicating whether the layout of the metaweb should be used to represent the difference network. to use metaweb layout = T, you need first to compute 'TL-tsne' layout for the metaweb for this beta value using <code>attach_layout()</code>
flip_coords	a boolean indicating whether coordinates should be flipped. In that case, y-axis is the trophic level and x-axis is the layout axis
alpha_per_group	controlling alpha per group (only for 'ggnet' vis), a list of format <code>list(resolutions = "XX", groups = XX, alpha_focal = XX, alpha_hidden = XX)</code> , see example
alpha_per_node	controlling alpha per node (only for 'ggnet' vis), a list of format <code>list(nodes = XX, alpha_focal = XX, alpha_hidden = XX)</code> , see example
TL_tsne.config	configuration list for mode 'TL-tsne', default is <code>TL_tsne.default</code>
nrep_ly	If several layouts for this beta value are attached to the metaweb (if <code>layout_metaweb = T</code>), index of the layout to use, see <code>attach_layout()</code>
ggnet.config	configuration list for ggnet representation, default is <code>ggnet.default</code>
visNetwork.config	configuration list for <code>visNetwork</code> representation, default is <code>visNetwork.default</code>

Value

an object of class `ggplot` or `visNetwork`, representation of the difference network

See Also

[attach_layout\(\)](#)

Examples

```
#on Angola dataset
library(igraph)
library(metanetwork)

data(meta_angola)

diff_plot(g1 = meta_angola$X2003, g2 = meta_angola$X1986, metanetwork = meta_angola,
beta = 0.05)
```

extract_networks	<i>extract networks from a metanetwork object</i>
------------------	---

Description

Function to extract metawebs and local networks from a metanetwork object

Usage

```
extract_networks(metanetwork)
```

Arguments

metanetwork the object whose networks need to be extracted

Details

Return a list of 'igraph' objects

Value

a list of igraph objects with attributes computed by metanetwork

Examples

```
library(metanetwork)
data("meta_angola")
nets = extract_networks(meta_angola)
sapply(nets,class)
```

ggmetanet	<i>ggmetanet</i>
-----------	------------------

Description

Function that provides network static representation (using 'ggnet') from a 'metanetwork' object using 'TL-tsne' or 'group-TL-tsne' layout.

Usage

```
ggmetanet(
  metanetwork,
  g = NULL,
  beta = 0.1,
  legend = NULL,
  mode = "TL-tsne",
  edge_thrs = NULL,
  layout_metaweb = FALSE,
  nrep_ly = 1,
  flip_coords = FALSE,
  diff_plot_bool = FALSE,
  alpha_per_group = NULL,
  alpha_per_node = NULL,
  alpha_interactive = FALSE,
  ggnet.config = ggnet.default,
  TL_tsne.config = TL_tsne.default
)
```

Arguments

metanetwork	object of class metanetwork
g	network (igraph object) to represent, default is metaweb
beta	the diffusion parameter of the diffusion kernel, a positive scalar controlling the vertical squeezing of the network
legend	resolution for the legend, legend resolution must be a coarser resolution than the resolution of g, default is NULL
mode	mode used for layout, 'TL-tsne' or 'group-TL-tsne' Default is 'TL-tsne'. This argument can also be a two-column matrix for custom layout.
edge_thrs	if non-null, a numeric (between 0 and 1) indicating an edge threshold for the representation
layout_metaweb	a boolean indicating whether the layout of the metaweb should be used to represent the network to use metaweb layout = TRUE, you need first to compute metaweb layout for this beta value using <code>attach_layout()</code>
nrep_ly	If several layouts for this beta value are attached to the metaweb (if <code>layout_metaweb = T</code>), index of the layout to use, see <code>attach_layout()</code>
flip_coords	a boolean indicating whether coordinates should be flipped.
diff_plot_bool	boolean, do not edit by hand
alpha_per_group	controlling alpha per group (only for 'ggnet' vis), a list of format <code>list(resolutions = "XX", groups = XX, alpha_focal = XX, alpha_hidden = XX)</code> , see example
alpha_per_node	controlling alpha per node (only for 'ggnet' vis), a list of format <code>list(nodes = XX, alpha_focal = XX, alpha_hidden = XX)</code> , see example In that case, y-axis is the trophic level and x-axis is the layout axis

alpha_interactive a boolean indicating whether alpha (that is node transparency) should be asked in interactive mode to the user

ggnet.config configuration list for ggnet representation, default is ggnet.default

TL_tsne.config configuration list for mode 'TL-tsne', default is TL_tsne.default

Details

At each call of the function with 'TL-tsne' layout, it computes a layout for the current beta value. If a layout is already attached to the current network, it uses directly this layout (without computing). This function provides many static visualisation tools:

- customising ggnet parameters wrapped in ggnet.config
- legending using the trophicTable
- playing on group transparency (alpha)
- using the metaweb layout
- building a legend for large networks.

Value

an object of class ggplot, the current network representation

See Also

[attach_layout\(\)](#), [ggnet.default](#)

Examples

```
library(metanetwork)
library(igraph)

#lattice example
g = make_lattice(dim = 2,length = 4,directed = TRUE)
#building metanetwork and computing trophic levels
meta0 = build_metanet(g)
meta0 = compute_TL(meta0)
ggmetanet(meta0)
#storing layout
meta0 = attach_layout(meta0)
ggmetanet(meta0)

#custom ggnet parameters
ggnet.custom = ggnet.default
ggnet.custom$label = TRUE
ggnet.custom$edge.alpha = 0.5
ggnet.custom$alpha = 0.7
ggnet.custom$arrow.size = 1
ggnet.custom$max_size = 12

# using pre-computed layout and custom ggnet parametersfor vertebrates metaweb
```

```
data("meta_vrtb")
#custom ggnet parameters
ggnet.custom = ggnet.default
ggnet.custom$label = TRUE
ggnet.custom$edge.alpha = 0.5
ggnet.custom$alpha = 0.7
ggnet.custom$arrow.size = 1
ggnet.custom$max_size = 12
#at SBM group level
beta = 0.005
ggmetanet(meta_vrtb,g = meta_vrtb$metaweb_group,flip_coords = TRUE,
          beta = beta,legend = "group",
          ggnet.config = ggnet.custom,edge_thrs = 0.1)
```

ggnet.default

Default configuration for ggnet

Description

A list with parameters customizing ggmetanet representation (see ggnet documentations)

Usage

```
ggnet.default
```

Format

An object of class metanetwork_config of length 16.

Examples

```
# display all default settings
ggnet.default

# create a new settings
ggnet.custom = ggnet.default
ggnet.custom$edge.size = 2
ggnet.custom
```

group_layout.default *Default configuration for group-TL-tsne layout*

Description

A list with parameters customizing group-TL-tsne layout

Usage

```
group_layout.default
```

Format

An object of class list of length 3.

Examples

```
# display all default settings
group_layout.default

# create a new settings object with n_neighbors set to 5
group_layout.custom = group_layout.default
group_layout.custom$group_height = 10
group_layout.custom
```

is.metanetwork *Test of belonging to class metanetwork*

Description

Return a boolean indicating whether the object belongs to class metanetwork

Usage

```
is.metanetwork(metanetwork)

## S3 method for class 'metanetwork'
is.metanetwork(metanetwork)
```

Arguments

metanetwork the object to test

Value

a boolean indicating whether the object belongs to class metanetwork
 NULL

Examples

```
library(metanetwork)
library(igraph)

g = make_ring(5,directed = TRUE)
meta = build_metanet(g)
is.metanetwork(meta)
#on Angola dataset
data("meta_angola")
is.metanetwork(meta_angola)
```

metanet_build_pipe	<i>Build and execute 'metanetwork' pipeline</i>
--------------------	---

Description

Method executing the whole metanetwork pipeline, including building 'metanetwork' object (build_metanet,append_agg_r, compute_TL, attach_layout)

Usage

```
metanet_build_pipe(
  metaweb,
  abTable = NULL,
  trophicTable = NULL,
  compute_local_nets = TRUE,
  verbose = TRUE,
  beta = 0.1
)
```

Arguments

metaweb	metaweb of the metanetwork, object of class 'graph', 'matrix', 'data.frame' or 'dgCMatrix'. Metaweb needs to be directed and connected. This parameter must be non-null.
abTable	abundances of nodes in local networks, matrix of class 'matrix', columns must have names corresponding to node labels of the metaweb, rows are node abundances in local networks. Default is null, in that case, uniform abundances are assigned
trophicTable	a 'matrix' or 'data.frame' indicating hierarchy of the nodes. Names of the columns correspond to the different resolutions. It indicates the membership of each node of the metaweb. Default is null.

compute_local_nets a boolean, indicates whether local networks must be computed or not. Default is TRUE

verbose a boolean indicating whether message along the pipeline should be printed

beta the diffusion parameter of the diffusion kernel, a positive scalar controlling the squeezing of the network

Value

object of class 'metanetwork', with computed layout stored as node attribute

Examples

```
library(metanetwork)
library(igraph)

g = make_lattice(dimvector = c(4,4),2,3,directed = TRUE)
meta0 = metanet_build_pipe(g)
ggmetanet(meta0)
```

metanet_pipe *Execute 'metanetwork' pipeline*

Description

Method executing the whole metanetwork pipeline for the initial metanetwork object (append_agg_nets, compute_TL, attach_layout)

Usage

```
metanet_pipe(metanetwork, beta = 0.1, verbose = TRUE)

## S3 method for class 'metanetwork'
metanet_pipe(metanetwork, beta = 0.1, verbose = TRUE)
```

Arguments

metanetwork object of class 'metanetwork'

beta the diffusion parameter of the diffusion kernel, a positive scalar controlling the squeezing of the network

verbose a boolean indicating whether message along the pipeline should be printed

Value

object of class 'metanetwork', with computed trophic levels and layout stored as node attribute
NULL

Examples

```

library(metanetwork)
library(igraph)

g = make_lattice(dimvector = c(4,4),2,3,directed = TRUE)
meta0 = build_metanet(g)
meta0 = metanet_pipe(meta0)
ggmetanet(meta0)

```

meta_angola	<i>Angola fishery metanetwork metanetwork built from: Angelini & Velho 2011, Data from: Angelini, R., Velho, VF. (2011) Ecosystem structure and trophic analysis of Angolan fishery landings. Scientia Marina 75(2)</i>
-------------	---

Description

Angola fishery metanetwork metanetwork built from: Angelini & Velho 2011, Data from: Angelini, R., Velho, VF. (2011) Ecosystem structure and trophic analysis of Angolan fishery landings. Scientia Marina 75(2)

Usage

```
data(meta_angola)
```

Format

A object of class 'metanetwork'

The metaweb from Angelini & Velho 2011, containing 28 groups and 127 interactions, a igraph object

metanetTable Abundance table built from biomass at two dates: 1986 and 2003, a matrix

trophicTable Taxonomic table, a three column data.frame with three different taxonomic levels (species (or group), phylum and kingdom)

Source

<https://scientiamarina.revistas.csic.es/index.php/scientiamarina/article/view/1254>

meta_norway	<i>Norway soil metanetwork metanetwork built from: Calderon-Sanou et al. 2021, Data from: Calderon-Sanou, I., Munkemuller, T., Zinger, L., Schimann, H., Yoccoz, N. G., Gielly, L., ... & Thuiller, W. (2021). Cascading effects of moth outbreaks on subarctic soil food webs. Scientific reports, 11(1), 1-12.</i>
-------------	--

Description

Norway soil metanetwork metanetwork built from: Calderon-Sanou et al. 2021, Data from: Calderon-Sanou, I., Munkemuller, T., Zinger, L., Schimann, H., Yoccoz, N. G., Gielly, L., ... & Thuiller, W. (2021). Cascading effects of moth outbreaks on subarctic soil food webs. Scientific reports, 11(1), 1-12.

Usage

```
data(meta_norway)
```

Format

A object of class 'metanetwork'

The metaweb from Calderon-Sanou et al. 2021, containing 40 groups and 204 interactions, a igraph object

metanetworkTable Abundance table built from eDNA data in disturbed (moth outbreaks) and non-disturbed sites, a matrix

trophicTable Trophic table, a three column data.frame with three different taxonomic levels (trophic_group, trophic_class and taxa)

Source

<https://www.nature.com/articles/s41598-021-94227-z>

meta_vrtb	<i>European vertebrates metanetwork metanetwork built using data from: O'Connor, L. M., Pollock, L. J., Braga, J., Ficetola, G. F., Maiorano, L., Martinez-Almoyna, C., ... & Thuiller, W. (2020). Unveiling the food webs of tetrapods across Europe through the prism of the Eltonian niche. Journal of Biogeography, 47(1), 181-192. and Maiorano, L., Montemaggiore, A., Ficetola, G. F., O'connor, L., & Thuiller, W. (2020). TETRA-EU 1.0: a species-level trophic metaweb of European tetrapods. Global Ecology and Biogeography, 29(9), 1452-1457.</i>
-----------	--

Description

European vertebrates metanetwork metanetwork built using data from: O'Connor, L. M., Pollock, L. J., Braga, J., Ficetola, G. F., Maiorano, L., Martinez-Almoyna, C., ... & Thuiller, W. (2020). Unveiling the food webs of tetrapods across Europe through the prism of the Eltonian niche. *Journal of Biogeography*, 47(1), 181-192. and Maiorano, L., Montemaggiori, A., Ficetola, G. F., O'connor, L., & Thuiller, W. (2020). TETRA-EU 1.0: a species-level trophic metaweb of European tetrapods. *Global Ecology and Biogeography*, 29(9), 1452-1457.

Usage

```
data(meta_vrtb)
```

Format

A object of class 'metanetwork'

The metaweb from Maiorano et al. 2020, O'Connor et al 2020, containing 1101 species and 49013 interactions, a igraph object

metatrophicTable Trophic table, a two columns data.frame with a column containing species name and a column containing Stochastic Block Model groups inferred in O'Connor et al 2020

Source

<https://onlinelibrary.wiley.com/doi/abs/10.1111/geb.13138>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/jbi.13773>

plot_trophicTable *Plot trophic groups hierarchy*

Description

Function to represent trophic groups hierarchy provided by trophicTable

Usage

```
plot_trophicTable(metanetwork, res = "all", ggnet.config = ggnet.default)
```

Arguments

metanetwork	object of class 'metanetwork'
res	resolutions included in the hierarchy representation. Default is "all" (all resolutions are then included) but can be also a vector of given resolutions
ggnet.config	configuration list for ggnet representation, default is ggnet.default

Value

object of class 'ggnet', representation of group hierarchy

Examples

```
library(metanetwork)

#on Angola data_set
data("meta_angola")
plot_trophicTable(meta_angola)
```

print	<i>print metanetwork</i>
-------	--------------------------

Description

Print method for class metanetwork

Usage

```
print(metanetwork)

## S3 method for class 'metanetwork'
print(metanetwork)
```

Arguments

metanetwork object of class 'metanetwork'

Value

character indicating number of nodes and edges of the metaweb, available resolutions and number of local networks

NULL

Examples

```
library(metanetwork)
library(igraph)

g = make_ring(5,directed = TRUE)
meta = build_metanet(g)
print(meta)

#on Angola dataset
data("meta_angola")
print(meta_angola)

#on Norway dataset
data("meta_norway")
print(meta_norway)
```

TL_tsne.default	<i>Default configuration for the diffusion kernel based t-sne</i>
-----------------	---

Description

A list with parameters customizing configuration for the diffusion kernel based t-sne (see 'tsne' R package documentation)

Usage

```
TL_tsne.default
```

Format

An object of class `metanetwork_config` of length 11.

Examples

```
# display all default settings
TL_tsne.default

# create a new settings object with n_neighbors set to 5
TL_tsne.custom = TL_tsne.default
TL_tsne.custom$max_iter = 5
TL_tsne.custom
```

vismetaNetwork	<i>vismetaNetwork</i>
----------------	-----------------------

Description

Function that provides network dynamic representation (using 'visNetwork') from a 'metanetwork' object with a layout based on a diffusion kernel

Usage

```
vismetaNetwork(
  metanetwork,
  g = NULL,
  beta = 0.1,
  legend = NULL,
  mode = "TL-tsne",
  edge_thrs = NULL,
  layout_metaweb = FALSE,
  nrep_ly = 1,
```

```

    flip_coords = FALSE,
    diff_plot_bool = FALSE,
    x_y_range = c(100, 100),
    visNetwork.config = visNetwork.default,
    TL_tsne.config = TL_tsne.default
  )

```

Arguments

metanetwork	object of class metanetwork
g	network (igraph object) to represent, default is metaweb
beta	the diffusion parameter of the diffusion kernel, a positive scalar controlling the vertical squeezing of the network
legend	resolution for the legend, legend resolution must be a coarser resolution than the resolution of g, default is NULL
mode	mode used for layout, 'TL-tsne' for trophic level t-sne. Default is 'TL-tsne'
edge_thrs	if non-null, a numeric (between 0 and 1) indicating an edge threshold for the representation
layout_metaweb	a boolean indicating whether the layout of the metaweb should be used to represent the network to use metaweb layout = T, you need first to compute metaweb layout for this beta value using <code>attach_layout()</code>
nrep_ly	If several layouts for this beta value are attached to the metaweb (if <code>layout_metaweb = T</code>), index of the layout to use, see <code>attach_layout()</code>
flip_coords	a boolean indicating whether coordinates should be flipped. In that case, y-axis is the trophic level and x-axis is the layout axis
diff_plot_bool	boolean, do not edit by hand
x_y_range	a two dimension numeric vector, indicating dilatation of x,y axis
visNetwork.config	configuration list for visNetwork representation, default is visNetwork.default
TL_tsne.config	configuration list for mode 'TL-tsne', default is TL_tsne.default

Value

object of class 'visNetwork', dynamic representation of the current network

Examples

```

library(metanetwork)
library(igraph)
data("meta_angola")
## Return htmlwidget
# on angola dataset
meta_angola = attach_layout(meta_angola, beta = 0.05)
vismetaNetwork(meta_angola, beta = 0.05)

```

`visNetwork.default` *Default configuration for visNetwork*

Description

A list with parameters customizing visNetwork visualisation (see visNetwork documentations)

Usage

```
visNetwork.default
```

Format

An object of class `metanetwork_config` of length 4.

Examples

```
# display all default settings
visNetwork.default

# create a new settings
visNetwork.custom = visNetwork.default
visNetwork.custom$label.size = 10
visNetwork.custom
```

`%>%` *Pipe*

Description

Like `dplyr`, `metanetwork` also uses the pipe function, `%>%` to turn function composition into a series of imperative statements.

Value

an object of the class of the output of the last called method/function

Examples

```
library(metanetwork)
data("meta_angola")
meta_angola %>% attach_layout() %>% ggmetanet()
```

Index

* datasets

ggnet.default, 18
group_layout.default, 19
meta_angola, 22
meta_norway, 23
meta_vrtb, 23
TL_tsne.default, 26
visNetwork.default, 28
%>%, 28

append_agg_nets, 2
attach_layout, 3
attach_layout(), 14, 17

build_metanet, 5

compute_dis, 6
compute_dis(), 10
compute_div, 8
compute_div(), 8, 11
compute_metrics, 10
compute_TL, 12
compute_TL(), 11

diff_plot, 13

extract_networks, 15

ggmetanet, 15
ggmetanet(), 5
ggnet.default, 17, 18
group_layout.default, 5, 19

is.metanetwork, 19

meta_angola, 22
meta_norway, 23
meta_vrtb, 23
metanet_build_pipe, 20
metanet_pipe, 21

plot_trophicTable, 24

plot_trophicTable(), 3
print, 25

TL_tsne.default, 26

vismetaNetwork, 26
vismetaNetwork(), 5
visNetwork.default, 28