Package ‘metanetwork’

December 5, 2022

Type Package

Title Handling and Representing Trophic Networks in Space and Time

Version 0.7.0

Date 2022-12-04

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.

License GPL-3

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

NeedsCompilation no

Repository CRAN

Date/Publication 2022-12-05 14:10:02 UTC

RoxygenNote 7.2.0

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

biocViews

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

Author Marc Ohlmann [aut, cre] (<https://orcid.org/0000-0003-3772-3969>), Jimmy Garnier [aut, ctb], Laurent Vuillon [aut, ctb]

Maintainer Marc Ohlmann <marcohlmann@live.fr>

Depends R (>= 3.5.0)
Description

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

Usage

```r
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.

References


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:

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

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


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.
**compute_TL**

**Description**

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

**Usage**

```r
compute_TL(metanetwork)
```

**Arguments**

- `metanetwork` object of class 'metanetwork'

**Examples**

```r
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)
```
**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 $v = \text{indegree}(G) - \text{outdegree}(G)$ 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**

```r
library(metanetwork)
library(igraph)

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

**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,
  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 attach_layout()).
               Default is 'TL-tsne'
vis_tool      a character indicating the visualisation tool, either 'ggnet' or visNetwork
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 rep-
               resent the difference network. to use metaweb layout = T, you need first to com-
               pute 'TL-tsne' layout for the metaweb for this beta value using attach_layout()
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 list(resolutions
               = "XX",groups = XX, alpha_focal = XX, alpha_hidden = XX), see example
alpha_per_node
               controlling alpha per node (only for 'ggnet' vis), a list of format list(nodes =
               XX, alpha_focal = XX, alpha_hidden = XX), see example
TL_tsne.config configuration list for mode 'TL-tsne', default is TL_tsne.default
nrep_ly        If several layouts for this beta value are attached to the metaweb (if layout_metaweb
               = T), index of the layout to use, see attach_layout()
extract_networks

extract networks from a metanetwork object

**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**
```r
# 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)
```
Examples

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

Description

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

Usage

```r
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 attach_layout()

nrep_ly If several layouts for this beta value are attached to the metaweb (if layout_metaweb = T), index of the layout to use, see attach_layout()

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 list(resolutions = "XX", groups = XX, alpha_focal = XX, alpha_hidden = XX), see example

alpha_per_node controlling alpha per node (only for 'ggnet' vis), a list of format list(nodes = XX, alpha_focal = XX, alpha_hidden = XX), 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 parameters for 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 documentation).

**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

Description

Default configuration for 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 Build and execute 'metanetwork' pipeline

Description

Method executing the whole metanetwork pipeline, including building 'metanetwork' object (build_metanet, append_agg_nets, 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)
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)

ggmeganet(meta0)

Description


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

metawebTable  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)
meta_norway

Source


---

meta_norway


Description


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

**metatable** 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

Description


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

metarichicTable 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


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)

---

**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

```r
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)
```

Description

A list with parameters customizing configuration for the diffusion kernel based t-sne (see `tsne` R package documentation).

Usage

```r
TL_tsne.default
```

Format

An object of class `metanetwork_config` of length 11.

Examples

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

Usage
```r
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 attach_layout()
- `nrep_ly`: If several layouts for this beta value are attached to the metaweb (if layout_metaweb = T), index of the layout to use, see attach_layout()
- `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
visNetwork.default

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

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

* datasets
  ggnet.default, 12
  group_layout.default, 13
  meta_angola, 16
  meta_norway, 17
  meta_vrtb, 18
  TL_tsne.default, 20
  visNetwork.default, 22
%>%, 23
append_agg_nets, 2
attach_layout, 3
attach_layout(), 9, 11
build_metanet, 5
compute_TL, 6
diff_plot, 7
extract_networks, 9
ggmetanet, 10
ggmetanet(), 5
ggnet.default, 12, 13
group_layout.default, 5, 13
is.metanetwork, 13
meta_angola, 16
meta_norway, 17
meta_vrtb, 18
metanet_build_pipe, 14
metanet_pipe, 15
plot_trophicTable, 18
plot_trophicTable(), 3
print, 19
TL_tsne.default, 20
vismetaNetwork, 21
vismetaNetwork(), 5
visNetwork.default, 22