Package ‘metanetwork’

February 3, 2022

Type Package

Title Handling and representing trophic networks in space and time

Version 0.0.0

Date 2020-09-01

Author Marc Ohlmann [aut, cre]

Maintainer Marc Ohlmann <marcohlmann@live.fr>

Description A collection of tools 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 diffusion kernel and trophic levels.

Imports GGally,
network,
sna,
ggplot2,
intergraph,
dplyr,
igraph,
Matrix,
Matrix.utils,
ade4,
visNetwork,
grDevices,
RColorBrewer,
magrittr

LinkingTo

License GPL-3

URL https://gitlab.com/marcohlmann/metanetwork

NeedsCompilation no

Repository CRAN

Date/Publication

LazyData true

RoxygenNote 7.1.2

R topics documented:

append_agg_nets *append aggregated networks*

Description

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

Usage

```
append_agg_nets(metanetwork)

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

Arguments

`metanetwork`	object of class 'metanetwork'

Value

object of class 'metanetwork'

`NULL`

Examples

```
library(metanetwork)
append_agg_nets(meta_angola)
```

`append_mean_nets` *append mean networks*

Description

Append mean networks per (discrete) value of covariable

Usage

```
append_mean_nets(metanetwork)

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

Arguments

`metanetwork`	object of class 'metanetwork'

Value

object of class 'metanetwork' with mean networks appended per value of the covariable at all resolutions

`NULL`

Examples

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

`attach_layout` *compute and attach ’TL-tsne’ layout*

Description

Method to compute ’TL-tsne’ layout and save it as node attributes of the focal network. 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

Usage

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

## S3 method for class 'metanetwork'
attach_layout(
  metanetwork,
  g = NULL,
  beta = 0.1,
  mode = "TL-tsne",
  TL_tsne.config = TL_tsne.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

`mode` only ’TL-tsne’ is supported for this function

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

Value

’metanetwork’ object with layout added as node attribute of the considered network

`NULL`

Examples

```
library(metanetwork)
library(igraph)
# on angola dataset (metaweb)
meta_angola = attach_layout(meta_angola,beta = 0.05)
V(meta_angola$metaweb)$layout_beta0.05
# on a local network
meta_angola = attach_layout(meta_angola,g = meta_angola$X1986,beta = 0.05)

# getting repetitions
meta_angola = attach_layout(meta_angola,beta = 0.05)
V(meta_angola$metaweb)$layout_beta0.05
V(meta_angola$metaweb)$layout_beta0.05_1
```

`build_metanet` *Build metanetwork object*

Description

Build metanetwork object

Usage

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

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`

`covariable` a vector of class `character` indicating qualitative covariables for local networks , default is null.

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)
metaweb = meta_angola$metaweb
abTable = meta_angola$abTable
trophicTable = meta_angola$trophicTable
meta_angola = build_metanet(metaweb,abTable,trophicTable)
```

compute_TL *compute trophic levels*

Description

Compute trophic levels using graph laplacian

Usage

```
compute_TL(metanetwork)

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

Arguments

`metanetwork` object of class ’metanetwork’

Value

object of class ’metanetwork’ with a node attribude `TL`

`NULL`

Examples

```
library(metanetwork)
library(igraph)

#on angola dataset
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 a layout based on diffusion kernel, using either ’ggnet’ or ’visNetwork’ visualisation

Usage

```
diff_plot(
  metanetwork,
  g1,
  g2,
  beta = 0.1,
  mode = "TL-tsne",
  vis_tool = "ggnet",
  edge_thrs = NULL,
  layout_metaweb = F,
  flip_coords = F,
  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, ’TL-tsne’ for trophic level t-sne and ’TL-kpco’ for trophic level kernel based pco. 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 represent the difference network. to use metaweb layout = T, you need first to compute metaweb layout for this beta value using `attach_layout()`

`flip_coords` a boolean indicating wheter 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()`

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

`visNetwork.config`
: configuration list for visNetwork representation, default is visNetwork.default

Value

plot of the difference network, either using 'ggnet' or 'visNetwork'

Examples

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

diff_plot(g1 = meta_angola$X2003,g2 = meta_angola$X1986,metanetwork = meta_angola,
beta = 0.05)
#using visNetwork
diff_plot(g1 = meta_angola$X2003,g2 = meta_angola$X1986,metanetwork = meta_angola,
beta = 0.05,vis_tool = "visNetwork")

#computing and using metaweb layout for diffplot
meta_angola = attach_layout(meta_angola,beta = 0.1)
diff_plot(g1 = meta_angola$X2003,g2 = meta_angola$X1986,metanetwork = meta_angola,
beta = 0.1,layout_metaweb = TRUE)
#using visNetwork
diff_plot(g1 = meta_angola$X2003,g2 = meta_angola$X1986,metanetwork = meta_angola,
beta = 0.1,layout_metaweb = TRUE,vis_tool = "visNetwork")

##contrasting alpha
#per group
diff_plot(metanetwork = meta_angola,g1 = meta_angola$X2003,g2 = meta_angola$X1986,beta = 0.05,
          alpha_per_group = list(resolution = "Phylum",groups = c("Fish"),
alpha_focal = 0.01,alpha_hidden = 0.8))
#per node
diff_plot(metanetwork = meta_angola,g1 = meta_angola$X2003,g2 = meta_angola$X1986,beta = 0.05,
alpha_per_node = list(nodes = c("Detritus","Phytoplankton"),
alpha_focal = 0.01,alpha_hidden = 0.8))
```

extract_networks *extract networks from metanetwork*

Description

Function to extract metawebs and local networks from metanetwork

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)

extract_networks(meta_angola)
```

ggmetanet *ggmetanet*

Description

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

Usage

```
ggmetanet(
  metanetwork,
  g = NULL,
  beta = 0.1,
  legend = NULL,
  mode = "TL-tsne",
  edge_thrs = NULL,
  layout_metaweb = F,
  nrep_ly = 1,
  flip_coords = F,
  diff_plot_bool = F,
  alpha_per_group = NULL,
  alpha_per_node = NULL,
  alpha_interactive = F,
  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' for trophic level t-sne and 'TL-kpco' for trophic level kernel based pco. 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 wheter 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

Value

object of class 'ggplot'

Examples

```
library(metanetwork)
library(igraph)

g = make_ring(5,directed = TRUE)
meta0 = build_metanet(g)
meta0 = compute_TL(meta0)
ggmetanet(meta0)

# angola dataset
meta_angola = compute_TL(meta_angola)
ggmetanet(meta_angola,legend = 'Phylum',beta = 0.05)
```

---

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 14.

Examples

```
# display all default settings
ggnet.default

# create a new settings
ggnet.custom = ggnet.default
ggnet.custom$edge.size = 2
ggnet.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 wheter 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
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_r`
`compute_TL`, `attach_layout`)

Usage

```
metanet_build_pipe(metanetwork, beta = 0.1, verbose = T)

## S3 method for class 'metanetwork'
metanet_build_pipe(metanetwork, beta = 0.1, verbose = T)
```

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 wheter message along the pipeline should be printed

Value

object of class 'metanetwork'

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)

#on angola data set
meta_angola = metanet_pipe(meta_angola,beta = 0.05)
```

---

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 = T)

## S3 method for class 'metanetwork'
metanet_pipe(metanetwork, beta = 0.1, verbose = T)
```

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 wheter message along the pipeline should be printed

Value

object of class 'metanetwork'

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)

#on angola data set
meta_angola = metanet_pipe(meta_angola)
```

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

Description

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

Usage

```
meta_angola
```

Format

A object of class 'metanetwork'

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

metaweb **abTable** 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

plot_trophicTable *Plot trophic groups hierarchy*

Description

Function to represent trophic groups hierachy provided by trophicTable

Usage

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

Arguments

`metanetwork`	object of class 'metanetwork'
`res`	resolutions included in the hierachy 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'

Examples

```
library(metanetwork)

#on Angola data_set
plot_trophicTable(meta_angola)
```

print_metanet *print metanetwork*

Description

Print method for class `metanetwork`

Usage

```
print_metanet(metanetwork)

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

Arguments

`metanetwork` object of class 'metanetwork'

Value

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

NULL

Examples

```
library(metanetwork)
library(igraph)

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

#on Angola dataset
print_metanet(meta_angola)
```

TL_tsne.default *Default configuration for the diffusion kernel based t-sne*

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` *vismetaNetwork*

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 = F,
  nrep_ly = 1,
  flip_coords = F,
  diff_plot_bool = F,
  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 and 'TL-kpco' for trophic level kernel based pco. 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 wheter 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'

Examples

```
library(metanetwork)
library(igraph)

#on angola dataset
meta_angola = attach_layout(meta_angola, beta = 0.05)
vismetaNetwork(meta_angola, beta = 0.05)
# flipping coordinates and controlling x_y_range
vismetaNetwork(meta_angola, beta = 0.05,flip_coords = TRUE,x_y_range = c(50,200))
#adding a legend
vismetaNetwork(meta_angola,legend = "Phylum", beta = 0.05)
#use metaweb layout to represent a local network
vismetaNetwork(metanetwork = meta_angola,g = meta_angola$X2003,legend = "Phylum", beta = 0.05,
layout_metaweb=TRUE)

#custom visNetwork parameters
visNetwork.custom = visNetwork.default
visNetwork.custom$edge.size = 5
visNetwork.custom$label = FALSE

vismetaNetwork(meta_angola,legend = "Phylum", beta = 0.05,
visNetwork.config = visNetwork.custom)
```

`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.

Examples

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

Index