Package ‘motifr’

October 13, 2022

Title Motif Analysis in Multi-Level Networks
Version 1.0.0
Date 2020-12-01
Multi-level networks combine multiple networks in one, e.g.
social-ecological networks. Motifs are small configurations of nodes
and edges (subgraphs) occurring in networks. ‘motifr’ can visualize
multi-level networks, count multi-level network motifs and compare
motif occurrences to baseline models. It also identifies contributions
of existing or potential edges to motifs to find critical or missing
edges. The package is in many parts an R wrapper for the excellent
‘SESMotifAnalyser’ ‘Python’ package written by Tim Seppelt.

License MIT + file LICENSE

URL https://marioangst.github.io/motifr/

BugReports https://github.com/marioangst/motifr/issues

Depends R (>= 3.5.0)
Imports dplyr, ggplot2 (>= 2.1.0), ggraph, igraph, intergraph,

network, purrr, RColorBrewer, reshape2, reticulate, rlang,
scales, tibble, tidygraph

Suggests ergm, knitr, pkgdown, rmarkdown, shiny, testthat (>= 2.1.0)

VignetteBuilder knitr

Config/reticulate list( packages = list( list(package = `sma", pip =

TRUE), list(package = `networkx", pip = TRUE) ) )

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.1.1

SystemRequirements Python (>= 3.0.0), numpy, pandas

NeedsCompilation no
**R topics documented:**

- compare_to_baseline ........................................... 2
- count_motifs .................................................. 4
- critical_dyads .................................................. 5
- directed_dummy_net ............................................ 6
- dummy_net ....................................................... 6
- edge_contribution .............................................. 7
- exemplify_motif ............................................... 8
- explore_motifs ................................................ 9
- identify_gaps .................................................. 9
- induced_level_subgraph .................................... 10
- is.directed ..................................................... 11
- large_directed_dummy_net ................................... 11
- list_motifs ...................................................... 12
- ml_net ............................................................ 13
- motifs_distribution ........................................... 13
- motif_summary .................................................. 14
- plot_critical_dyads ........................................... 15
- plot_gaps ........................................................ 16
- plot_gaps_or_critical_dyads ................................. 17
- plot_mnet ....................................................... 18
- show_motif ...................................................... 19
- simulate_baseline ............................................ 20
- supported_classes ............................................. 22
- supported_signatures ....................................... 22
- tidygraph_dummy_net ......................................... 23
- to_py_graph .................................................... 24
- update_motifr .................................................. 25

**Index**

| compare_to_baseline | Compare motif occurrence in empirical network to occurrence in a baseline model |

**Description**

This function plots a comparison of the motif counts in a given network with the motif counts in a baseline model.
Usage

```r
compare_to_baseline(
  net,
  motifs,
  n = 10,
  lvl_attr = "sesType",
  assume_sparse = TRUE,
  model = "erdos_renyi",
  level = -1,
  ergm_model = NULL,
  directed = NULL
)
```

Arguments

- `net` network object
- `motifs` list of motif identifier strings
- `n` number of random graphs used in baseline model
- `lvl_attr` character vector specifying the attribute name where level information is stored in `net`.
- `assume_sparse` whether the random graphs shall be assumed to be sparse. used to find ideal counting function
- `model` baseline model to be used. Options are 'erdos_renyi', 'actors_choice', 'ergm', 'partial_ergm' and fixed_densities'. See vignette("random_baselines") for more details. Defaults to 'erdos_renyi'.
- `level` lvl_attr of the variable level for the Actor’s Choice model
- `ergm_model` ergm model as for example fitted by calling `ergm::ergm()` on the empirically observed network. Needs to be supplied when model is set to 'erdos_renyi'.
- `directed` whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

Details

Note that when using the Actor’s Choice model this function does not choose the variable level automatically. Use the `level` parameter to provide a valid level.

When using ERGM the parameter `net` is not used. Networks to create the baseline from are sampled in R using the `ergm_model` parameter.

Value

data frame with one row for each motif identifier string and one row for every computed random graph
count_motifs

Examples

## Not run:
compare_to_baseline(ml_net, list("1,2[I.C]", "1,2[II.C]"), directed = FALSE)

## End(Not run)

count_motifs Count multi-level motifs

Description

Count multi-level motifs

Usage

count_motifs(
  net,
  motifs,
  lvl_attr = c("sesType"),
  assume_sparse = TRUE,
  omit_total_result = TRUE,
  directed = NULL
)

Arguments

net A network object with a node attribute specifying the level of each node
motifs a list of motif identifiers which shall be counted, e.g. list("1,2[I.C]")
lvl_attr character vector specifying the vertex attribute name where level information is stored in net
assume_sparse whether the network shall be assumed to be sparse (for optimization), default TRUE
omit_total_result whether total results shall be omitted, default FALSE
directed whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

Value

data frame with a column containing motif identifier strings and one column containing motif counts
critical_dyads

Examples

```r
## Not run:
count_motifs(ml_net,
  lvl_attr = c("sesType"),
  motifs = list("1,2[II.C]", "1,2[II.C]", "2,1[II.C]", "2,1[II.C]"),
  directed = FALSE
)
## End(Not run)
```

---

critical_dyads  

List critical dyads

Description

Critical dyads are edges on a specified level which break motifs by being removed from the network.

Usage

```
critical_dyads(net, motif, lvl_attr = c("sesType"), level = -1)
```

Arguments

- **net**  
  network object
- **motif**  
  motif identifier
- **lvl_attr**  
  character vector specifying the attribute name where level information is stored in net
- **level**  
  level of the dyads which shall be considered, or -1 if the level shall be determined automatically.

Details

The level parameter determines on which level of the network critical dyads are analysed. Per default, when `level = -1`, the first level in the motif which provides exactly two nodes is selected. Use this parameter to specify a level manually. The procedure for determining the level is the same as for the Actor’s Choice Model, cf. vignette.

Note that this only works for undirected graphs. Regardless of whether the input graph is directed it is treated as undirected graph.

Value

data frame with three columns, listing edges and their contribution to motifs described by the motif identifier in descending order
Examples

```r
### Not run:
head(critical_dyads(ml_net, motif = "1,2[I.C"]))
```

### End(Not run)

directed_dummy_net  Two-level directed network dummy example

Description

Simple igraph network object based on dummy data

Usage

directed_dummy_net

Format

igraph network object

Source


Examples

plot_mnet(directed_dummy_net)

dummy_net  Three-level network dummy example

Description

A simple statnet network object based on dummy data.

Usage

dummy_net

Format

Statnet network object with 60 nodes and 1035 edges on three levels. The network contains two variables to describe nodes/vertices.

vertex.names node labels

sesType Categorical variable specifying network levels for every node (levels are 0, 1 and 2) ...
edge_contribution

Source

Dummy data [https://gitlab.com/t.seppelt/sesmotifanalyser/-/tree/master/test/data](https://gitlab.com/t.seppelt/sesmotifanalyser/-/tree/master/test/data)

Examples

```r
plot_mnet(dummy_net)
```

---

edge_contribution | List edge contribution

**Description**

List gaps ordered by contribution to a motif. This is a list of ties together with the number of motifs of a given class the dyad would generate by being flipped. This is a generalisation of `motifr::identify_gaps()` and `motifr::critical_dyads()`.

**Usage**

```r
edge_contribution(net, motif, lvl_attr = c("sesType"), level = -1)
```

**Arguments**

- `net` network object
- `motif` motif identifier
- `lvl_attr` character vector specifying the attribute name where level information is stored in `net`
- `level` level of the dyads which shall be considered, or -1 if the level shall be determined automatically.

**Details**

The level parameter determines on which level of the network edge contributions are analysed. Per default, when `level = -1`, the first level in the motif which provides exactly two nodes is selected. Use this parameter to specify a level manually. The procedure for determining the level is the same as for the Actor’s Choice Model, cf. vignette.

Note that this only works for undirected graphs. Regardless of whether the input graph is directed it is treated as undirected graph.

**Value**

data frame with three columns, listing edges and their contribution to motifs described by the motif identifier in descending order
exemplify_motif

Examples

## Not run:
head(edge_contribution(ml_net, "1,2[I.C]"))

## End(Not run)

---

**exemplify_motif**

*Returns an example for a motif found in a given network*

## Description

Returns an example for a motif found in a given network

## Usage

exemplify_motif(net, motif, lvl_attr = "sesType", directed = NULL)

## Arguments

- **net**: network object
- **motif**: motif identifier string for the motif
- **lvl_attr**: character vector specifying the attribute name where level information is stored in net.
- **directed**: whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

## Value

vector of nodes in the motif

## See Also

motifr::show_motif

## Examples

## Not run:
exemplify_motif(ml_net, motif = "1,2[I.C]", directed = FALSE)

## End(Not run)
explore_motifs  

Explore the motif zoo interactively in a shiny app

Description

Without any arguments, this launches a shiny app, where all available motifs in motifr can be graphically displayed by selecting signature-class combinations from a dropdown list.

Usage

explore_motifs(net = NULL, lvl_attr = c("sesType"))

Arguments

net  
optional: you may supply your own network object here (must be loaded as an R object in the global environment)

lvl_attr  
if you supply your own network object, indicate the name of the network attribute where level information is stored for each node

Details

If arguments net and lvl_attr are provided, you can load you own network into the shiny app to explore what a given motif classifier looks like for your network. Be aware that if your network does not contain a specific motif, an example of the motif can also not be shown, because motifr illustrates motifs by actually finding an example within a given network.

Value

Launches a shiny app where all available motifs can be displayed or, alternatively, all available motifs for a user-supplied network

identify_gaps  

List gaps

Description

List gaps ordered by contribution to a motif. This is a list of ties together with the number of motifs of a given class the dyad would generate by being added to the network.

Usage

identify_gaps(net, motif, lvl_attr = c("sesType"), level = -1)
induced_level_subgraph

Returns subgraph induced by one level of the network

Description

This function is intended to be used together with simulate_baseline() for partial ERGM models. Currently, only network objects are supported as input.

Usage

induced_level_subgraph(net, level, lvl_attr = "sesType")

Arguments

- `net`: the network
- `level`: the (number of the) level
- `lvl_attr`: name of the nodal attribute specifying the level
Value
induced subgraph as network object.

Examples

```r
subgraph_actors <- induced_level_subgraph(motifr::ml_net, 1)
plot_mnet(subgraph_actors, label = TRUE)
```

is.directed Checks whether the given network is directed

Description
Placeholder function for the corresponding functions of the various supported network formats. For example, this function calls network::is.directed() on network objects and igraph::is.directed() on igraph objects.

Usage

```r
is.directed(net)
```

Arguments

net the network

Value
whether the given network is directed

Examples

```r
is.directed(motifr::ml_net)
```

large_directed_dummy_net Large two-level directed network dummy example

Description
Large two-level directed network dummy example

Usage

```r
large_directed_dummy_net
```
Format

network network object

Source

Dummy data [https://gitlab.com/t.seppelt/sesmotifanalyser/-/tree/master/test/data](https://gitlab.com/t.seppelt/sesmotifanalyser/-/tree/master/test/data)

Examples

```r
plot_mnet(large_directed_dummy_net)
```

---

**list_motifs**  
*Lists motifs of a given class or all motifs with a given signature*

Description

Returns a dataframe with one row for each instance of the motif specified by the given motif identifier string. If the identifier string specifies a motif class, e.g. 1,2[I.A], then only motifs of the given class are returned. If the identifier string specifies a signature, e.g. 1,2, then a full list of all motifs of this signature is returned. In the latter case, the dataframe contains an additional column stating the classes of the motifs. The naming scheme of the columns is as follows: Each column is called `levelA_nodeB` where `A` is the `lvl_attr` of the nodes in the column and `B` the index of the nodes among the nodes on the same level. This index stems from the internal order of the nodes and does not carry any specific meaning.

Usage

```r
list_motifs(net, identifier, lvl_attr = "sesType", directed = NULL)
```

Arguments

- `net`: network object
- `identifier`: motif identifier string (with or without class, see above)
- `lvl_attr`: character vector specifying the attribute name where level information is stored in `net`.
- `directed`: whether the graph shall be treated as a directed graph. Per default (`NULL`), this is determined automatically using the structure of the provided network object.

Value

data frame with one row for each motif

Examples

```r
## Not run:
head(list_motifs(ml_net, "1,2[I.C]", directed = FALSE))
```

## End(Not run)

```r
## Not run:
head(list_motifs(ml_net, "1,2[I.C]", directed = FALSE))
```

## End(Not run)
ml_net

Two-level network example (wetlands management)

Description

A statnet network object based on empirical data about actors and their activities in a case study of Swiss wetlands management

Usage

ml_net

Format

Statnet network object with 132 nodes and 566 edges on two levels. One network level contains actors, a second network level contains activities. Links between actors indicate collaboration among actors. Links between actors and activities indicate that an actor is active in a given activity. Links between activities indicate that the activities are causally interdependent. The network contains two variables to describe nodes/vertices.

vertex.names node labels

sesType Binary variable specifying network levels for every node (1 = node is a social node (actor), 0 = node is a non-social node (an activity))...

Source

Surveys and expert interviews in a Swiss wetland. Data is anonymized and should only be used for exemplary purposes.

Examples

plot_mnet(ml_net)

motifs_distribution

Compute statistical properties (expectation and variance) of the distribution of motifs in a baseline model

Description

This function supports the Erdős-Rényi Model (erdos_renyi) and the the Actor’s Choice Model (actors_choice). The model can be specified using the model parameter. The Erdős-Rényi Model can be used without providing further parameters. In case of the Actor’s Choice Model a level of the given network can be specified which is only level assumed to be variable. All other levels are assumed to be fixed. Per default, level = -1, the first level carrying two nodes in the signature of the motif is selected as variable level. Set the level parameter to the value of the lvl_attr of the nodes in the desired level to specify the level manually.
Usage

```r
motifs_distribution(
  net,
  motifs,
  lvl_attr = "sesType",
  model = "erdos_renyi",
  level = -1,
  omit_total_result = TRUE,
  directed = NULL
)
```

Arguments

- **net**: network object
- **motifs**: list of motif identifiers describing the motifs whose distribution shall be analysed
- **lvl_attr**: character vector specifying the attribute name where level information is stored in net.
- **model**: baseline model to be used. options are "erdos_renyi" and "actors_choice". Defaults to "erdos_renyi".
- **level**: Additional parameter to set the level to vary for the actors_choice model manually. All other levels are held fixed.
- **omit_total_result**: whether total results shall be omitted
- **directed**: whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

Value

data frame with one column giving names of motif identifiers and two column giving expectation and variances per motif. For other motifs, expectations are computed but variances are returned as NaN.

Examples

```r
## Not run:
motifs_distribution(ml_net, motif = list("1,2[I.C]"), directed = FALSE)
## End(Not run)
```

motif_summary

**Summary for motif counts and Erdős-Rényi distribution**

Description

Returns a data frame with counts and statistical properties (expectation, variances) of six selected motifs in the given network. Note that this function implicitly assumes that the network is undirected, cf. `motifr::to_py_graph`. 
**plot_critical_dyads**

*Plot critical dyads in network visualisation*

**Description**

Note that this only works for undirected graphs. Regardless of whether the input graph is directed it is treated as undirected graph.

**Usage**

```r
plot_critical_dyads(
    net,
    motif,
    lvl_attr = c("sesType"),
    level = -1,
    cutoff = 2,
    subset_graph = "none",
    ...
)
```

**Arguments**

- `net`: Statnet network object
- `motif`: Motif to explore gaps in for
- `lvl_attr`: Node attribute specifying level information
- `level`: Focal level for gap analysis
plot_gaps

Plot gaps in network visualisation

Description

Note that this only works for undirected graphs. Regardless of whether the input graph is directed it is treated as undirected graph.

Usage

plot_gaps(
  net,
  motif,
  lvl_attr = c("sesType"),
  level = -1,
  cutoff = 2,
  subset_graph = "none",
  ...)

Value

A plot of gaps, sized by weight in a multilevel network

Examples

## Not run:
plot_critical_dyads(ml_net, "1,2[I.C]", level = -1)
plot_critical_dyads(ml_net, "1,2[I.C]",
  level = -1,
  subset_graph = "focal", cutoff = 4, label = TRUE)
plot_critical_dyads(ml_net, "1,2[I.C]",
  level = -1,
  subset_graph = "partial", cutoff = 4, label = TRUE)

## End(Not run)
Arguments

- **net**: Statnet network object
- **motif**: Motif to explore gaps in for
- **lvl_attr**: Node attribute specifying level information
- **level**: Focal level for gap analysis
- **cutoff**: Cut-off point in contributions of an edge to the number of motifs above which to analyse gaps
- **subset_graph**: Whether to subset the graph to only show nodes involved in gaps. One of "none" (no subset, default), "partial" (only focal level is subset) or "focal" (only focal level shown)
- **...**: list of additional parameters to be passed to plotting function (see `motifr::plot_mnet`), e.g. `label = TRUE`

Value

A plot of gaps, sized by weight in a multilevel network

Examples

```r
## Not run:
plot_gaps(ml_net, "1,2[II.C]", level = -1)
plot_gaps(ml_net, "1,2[II.C]",
  level = -1,
  subset_graph = "focal", cutoff = 4, label = TRUE
)
plot_gaps(ml_net, "1,2[II.C]",
  level = -1,
  subset_graph = "partial", cutoff = 4, label = TRUE
)
## End(Not run)
```

Description

Note that this only works for undirected graphs. Regardless of whether the input graph is directed it is treated as undirected graph.
plot_mnet

Visualize a multi-level network (using ggraph)

Description

Visualize a multi-level network, with the possibility of specifying separate layouts for each level. This is a somewhat hacky wrapper for arranging separate ggraph calls for each network level in a circle.

Usage

plot_gaps_or_critical_dyads(
  net,
  edge_contribution,
  colour,
  title,
  lvl_attr = c("sesType"),
  cutoff = 2,
  subset_graph = "none",
  ...
)

Arguments

net 
  network object
edge_contribution 
  data frame providing edge contribution data
colour 
  colour code for the weighted edges
title 
  title of the plot
lvl_attr 
  nodal attribute specifying level information
cutoff 
  Cut-off point in contributions of an edge to the number of motifs above which to analyse gaps
subset_graph 
  Whether to subset the graph to only show nodes involved in gaps. One of "none" (no subset, default), "partial" (only focal level is subset) or "focal" (only focal level shown)
...
  list of additional parameters to be passed to plotting function (see motifr::plot_mnet), e.g. label = TRUE

Value

A plot of gaps or critical edges, sized by weight in a multilevel network

See Also

plot_gaps, plot_critical_dyads.
Usage

plot_mnet(
  net,
  lvl_attr = c("sesType"),
  layouts = rep("kk", n_levels),
  label = FALSE,
  directed = NULL,
  nodesize = 3,
  edgewidth = 0.5
)

Arguments

net A tidygraph, igraph or statnet network object
lvl_attr The name of the categorical node attribute specifying at which level a node is situated
layouts A list of layouts (see ggraph::layout_ggraph) for every level e.g. for two levels list("auto", "circle")
label logical - should nodes be labelled? (defaults to false)
directed whether the network object shall be interpreted as directed network. Per default, motifr::is.directed is used to determine that.
nodesize The size of node displays, if displayed as points (if label = false)
edgewidth The width of lines illustrating edges

Details

For more extensive visualization options, it is recommended to explore the layout_as_multilevel function included in the package graphlayouts.

Value

A ggraph object

Examples

plot_mnet(net = motifr::ml_net, lvl_attr = "sesType", layouts = list("kk", "circle"))

Description

If no network is provided, a motif in a dummy network (motifr::dummy_net or motifr::large_directed_dummy_net) will be shown.
simulate_baseline

Simulate a baseline baseline model

Description

A baseline distribution of motif counts from a specified number of networks using a specified baseline model is computed. Options for the baseline model are - Erdős–Rényi - Actor’s choice - Fixed density - Providing an ERGM fit for the whole network - Providing a partial ERGM fit (for only one level)

Usage

```r
simulate_baseline(
  net,
  motifs,
  n = 10,
  lvl_attr = "sesType",
  assume_sparse = TRUE,
)```

Usage

```r
show_motif(motif, net = NULL, lvl_attr = c("sesType"), directed = NULL, ...)
```
simulate_baseline

```r
calling simulate_baseline
model = "erdos_renyi",
level = -1,
ergm_model = NULL,
directed = NULL
)
```

**Arguments**

- **net**
  - network object
- **motifs**
  - list of motif identifier strings
- **n**
  - number of random graphs
- **lvl_attr**
  - character string specifying the attribute name where level information is stored in net.
- **assume_sparse**
  - whether the random graphs shall be assumed to be sparse. used to find ideal counting function. defaults to TRUE.
- **model**
  - baseline model to be used. Options are 'erdos_renyi', 'fixed_densities', 'actors_choice', 'ergm' and 'partial_ergm'. See vignette("random_baselines") for more details. Defaults to 'erdos_renyi'.
- **level**
  - lvl_attr of the variable level for the Actor’s Choice model and for partial ERGM
- **ergm_model**
  - ergm model as for example fitted by calling `ergm::ergm()`. Used when model is set to 'ergm' or 'partial_ergm' to sample random networks.
- **directed**
  - whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

**Details**

Note that when using the Actor’s Choice model this function does not choose the variable level automatically. Use the `level` parameter to provide a valid level.

When using (partial) ERGM the parameter `net` is not used. Random networks are sampled in R using the `ergm_model` parameter.

**Value**

data frame with one column for each motif identifier string and one row for every computed random graph

**Examples**

```r
## Not run:
simulate_baseline(ml_net, list("1,2[I.C]"), n = 10, directed = FALSE)

## End(Not run)
```
**supported_classes**

*Lists all supported motif classes for a given signature*

**Description**

Returns a list with all supported motif classes for the given signature. Raises an error if the given signature is not supported.

**Usage**

`supported_classes(signature, directed)`

**Arguments**

- **signature**
  - head of a motif identifier string, i.e. string with comma-separated list specifying the signature of the motif
- **directed**
  - whether the motifs are directed.

**Value**

list of supported motif classes

**See Also**

`supported_signatures()`

**Examples**

```r
## Not run:
supported_classes("1,2", FALSE)
supported_classes("1,1", TRUE)
## End(Not run)
```

---

**supported_signatures**

*Lists all supported signatures*

**Description**

Returns a data frame with three columns: signature, a Boolean value indicating whether the motifs are directed, the number of levels which the motif spans across

**Usage**

`supported_signatures()`
tidygraph_dummy_net

Value
data frame with all supported signatures

See Also
supported_classes()

Examples

## Not run:
supported_signatures()
## End(Not run)

tidygraph_dummy_net  Two-level tidygraph network example

Description
Simple tidygraph network object for testing

Usage
tidygraph_dummy_net

Format
tidygraph network object

Source

Examples
plot_mnet(tidygraph_dummy_net)
to_py_graph

Translate multi-level statnet or igraph network object to Python networkx object

Description

The function `motifr::is.directed` is used to determine whether the provided network is directed (if `directed = FALSE`).

Usage

to_py_graph(g, lvl_attr, relabel = TRUE, directed = NULL)

Arguments

g statnet or igraph network object
lvl_attr character vector specifying the attribute name where level information is stored in `net`.
relabel should nodes be relabelled with statnet `vertex.names` or igraph nodal attribute name
directed whether the graph shall be treated as a directed graph. Per default (NULL), this is determined automatically using the structure of the provided network object

Details

The nodal attribute specified by `lvl_attr` indicates the levels of the nodes. Values are automatically converted to integers. Levels must be numbered starting with 0, 1, ....

Value

Python networkx graph object

Examples

```r
## Not run:
to_py_graph(motifr::dummy_net, lvl_attr = "sesType")

## End(Not run)
```
update_motifr

Description

It might be necessary to restart your R session after updating the sma package.

Usage

update_motifr()
Index

* datasets
    directed_dummy_net, 6
    dummy_net, 6
    large_directed_dummy_net, 11
    ml_net, 13
    tidygraph_dummy_net, 23

compare_to_baseline, 2
count_motifs, 4
critical_dyads, 5
directed_dummy_net, 6
dummy_net, 6
directed_dummy_net, 6
dummy_net, 6
directed_dummy_net, 6
dummy_net, 6
edge_contribution, 7
exemplify_motif, 8
explore_motifs, 9
identify_gaps, 9
induced_level_subgraph, 10
is.directed, 11
large_directed_dummy_net, 11
layout_as_multilevel, 19
list_motifs, 12
ml_net, 13
motif_summary, 14
motifs_distribution, 13
plot_critical_dyads, 15
plot_gaps, 16
plot_gaps_or_critical_dyads, 17
plot_mnet, 18
show_motif, 19
simulate_baseline, 20
supported_classes, 22
supported_signatures, 22
tidygraph_dummy_net, 23
to_py_graph, 24
update_motifr, 25