Package ‘multinets’

December 14, 2019

Title Multilevel Networks Analysis
Version 0.2.2
License GPL-3
Encoding UTF-8
LazyData true
Depends R (>= 3.4.0)
Imports igraph, Rcpp
Suggests sand, statnet, data.table, testthat, igraphdata
ByteCompile true
RoxygenNote 6.1.1
URL https://github.com/neylsoncrepalde/multinets
BugReports https://github.com/neylsoncrepalde/multinets/issues
LinkingTo Rcpp
NeedsCompilation yes
Author Neylson Crepalde [aut, cre]
Maintainer Neylson Crepalde <neylsoncrepalde@gmail.com>
Repository CRAN
Date/Publication 2019-12-14 17:20:03 UTC

R topics documented:

extract_highlevel .................................................. 2
extract_lowlevel ................................................... 3
extract_mesolevel .................................................. 4
is_multilevel ....................................................... 5
**extract_highlevel**

Extract the high level of a multilevel network

**Description**

Extract one of the three levels of a multilevel network. `extract_highlevel` will extract the higher level vertices and the edges between them, `extract_lowlevel` will extract the lower level vertices and the edges between them. On the other hand, `extract_mesolevel` will extract all the vertices but only the edges between vertices from different levels.

**Usage**

```r
extract_highlevel(x)
```

**Arguments**

- `x` a graph object. Must be a multilevel network.

**Value**

`extract_highlevel` and `extract_lowlevel` return a 1-mode network. All the vertices in the selected level and the edges between them. `extract_mesolevel` returns a bipartite (2-mode) network. All the vertices are kept. Only edges between vertices of different levels are kept. This is what the literature also calls an *affiliation* network.

**Author(s)**

Neylson Crepalde, <neylsoncrepalde@gmail.com>

**Examples**

```r
organizations <- extract_highlevel(linked_sim)

individuals <- extract_lowlevel(linked_sim)

affiliations <- extract_mesolevel(linked_sim)
```
Function: extract_lowlevel

**Description**

Extract one of the three levels of a multilevel network. `extract_highlevel` will extract the higher level vertices and the edges between them, `extract_lowlevel` will extract the lower level vertices and the edges between them. On the other hand, `extract_mesolevel` will extract all the vertices but only the edges between vertices from different levels.

**Usage**

`extract_lowlevel(x)`

**Arguments**

- `x`: a graph object. Must be a multilevel network.

**Value**

`extract_highlevel` and `extract_lowlevel` return a 1-mode network. All the vertices in the selected level and the edges between them. `extract_mesolevel` returns a bipartite (2-mode) network. All the vertices are kept. Only edges between vertices of different levels are kept. This is what the literature also calls an affiliation network.

**Author(s)**

Neylson Crepalde, <neylsoncrepalde@gmail.com>

**Examples**

```r
organizations <- extract_highlevel(linked_sim)
individuals <- extract_lowlevel(linked_sim)
affiliations <- extract_mesolevel(linked_sim)
```
**extract_mesolevel**

**Extract the meso level of a multilevel network**

**Description**

Extract one of the three levels of a multilevel network. `extract_highlevel` will extract the higher level vertices and the edges between them, `extract_lowlevel` will extract the lower level vertices and the edges between them. On the other hand, `extract_mesolevel` will extract all the vertices but only the edges between vertices from different levels.

**Usage**

```r
extract_mesolevel(x)
```

**Arguments**

- `x` a graph object. Must be a multilevel network.

**Value**

`extract_highlevel` and `extract_lowlevel` return a 1-mode network. All the vertices in the selected level and the edges between them. `extract_mesolevel` returns a bipartite (2-mode) network. All the vertices are kept. Only edges between vertices of different levels are kept. This is what the literature also calls an affiliation network.

**Author(s)**

Neylson Crepalde, <neylsoncrepalde@gmail.com>

**Examples**

```r
organizations <- extract_highlevel(linked_sim)
individuals <- extract_lowlevel(linked_sim)
affiliations <- extract_mesolevel(linked_sim)
```
is_multilevel

Check whether a graph is multilevel

Description
Checks whether a network (a graph) is multilevel according to the linked design proposed by Lazega et al (2008).

Usage
is_multilevel(x)

Arguments
x
A graph object.

Details
A multilevel network can be defined as a network with two kinds of vertex and two kinds of edges: Vertices on the lower level are, usually, individuals. Vertices on the higher level are, usually, organizations, collectivities or other kinds of social joint structures. The first kind of edges are those between the vertices of the higher level and between the vertices of the lower level. The second kind of edges are those between the vertices of different levels (affiliation ties).

Put another way, a multilevel network is a bipartite network (2-mode) that has ties between nodes on both levels.

Author(s)
Neylson Crepalde, <neylsoncrepalde@gmail.com>

References

See Also
is_bipartite

Examples
is_multilevel(linked_sim)
**Description**

Set layout coordinates for multilevel networks from a chosen algorithm.

**Usage**

```r
layout_multilevel(x, layout = igraph::layout_with_fr)
```

**Arguments**

- `x`: a graph object. Must be a multilevel network.
- `layout`: The chosen layout algorithm. A function layout of the 'igraph' package without parentheses. Default set to 'Fruchterman-Reingold'.

**Details**

In order to facilitate the visualization of multilevel networks, higher level nodes are set in the upper part of the plane and the lower level nodes are set in the lower part of the plane.

**Value**

A two- or three-column matrix, each row giving the coordinates of a vertex, according to the ids of the vertex ids.

**Author(s)**

Neylson Crepalde, <neylsoncrepalde@gmail.com>

**See Also**

`layout_`

**Examples**

```r
# Check if the network is multilevel
is_multilevel(linked_sim)

# Generate the layout (x,y) coordinates
l <- layout_multilevel(linked_sim)

# Plot the graph using the layout
plot(linked_sim, layout = l)

# Using Kamada Kawai algorithm
l.kk <- layout_multilevel(linked_sim, layout = igraph::layout_with_kk)
```
# Plot the graph with the new layout
plot(linked_sim, layout = l.kk)

linked_sim

A simulated multilevel network

Description

A simulated multilevel network following the "linked design" as proposed by Lazega et al (2008).

Usage

linked_sim

linked_sim_matrix

linked_sim_type

Format

linked_sim is a graph object with 150 vertices and 600 edges. There is one vertex attribute "type" defining to which level each vertex belongs. FALSE means lower level and TRUE means higher level.

linked_sim_matrix is a matrix object that contains the adjacency matrix for the simulated network.

linked_sim_type is a logical vector containing each vertex type.

Author(s)

Neylson Crepalde, <neylsoncrepalde@gmail.com>

Source

Elaborated by the author.

References


Description

This function is a wrapper to the `bipartite_projection` function from the 'igraph' package. It transforms a bipartite (2-mode) network to two 1-mode networks. It is recommended to use it as an analysis tool for a bipartite network that was obtained from `extract_mesolevel`.

Usage

`mode_transformation(x, which = c("both", "high", "low"))`

Arguments

- `x` a bipartite graph object. Data must be bipartite and must not be multilevel
- `which` one of "both", "high" or "low". High indicates to do the transformation to the higher level. Low indicates to do the transformation to the lower level. Both returns a list with both networks.

Details

The function counts 1 edge when two vertices of the higher level share a vertex of the lower level and 1 edge when two vertices of the lower level are affiliated to the same vertex in the higher level.

Value

If which is set to "both" (default), returns a list with two generated networks. If which is set to "high" or "low", returns a graph object.

Author(s)

Neylson Crepalde, <neylsoncrepalde@gmail.com>

Examples

```r
# First, extract the mesolevel of the multilevel network
affiliation <- extract_mesolevel(linked_sim)

# To obtain both transformed networks
transformed <- mode_transformation(affiliation)

# To obtain just one transformed network
high_transformed <- mode_transformation(affiliation, which = "high")
```
set_color_multilevel  

Description

Set vertices and edges colors for a multilevel network. Default set to blue (higher level) and red (lower level).

Usage

```r
set_color_multilevel(x, color.true = "blue", color.false = "red",
V.alpha = 0.7, E.alpha = 0.7)
```

Arguments

- `x`: a graph object. Must be a multilevel network.
- `color.true`: a character. The color to be chosen for the higher level (attribute type set to TRUE).
- `color.false`: a character. The color to be chosen for the lower level (attribute type set to FALSE).
- `V.alpha`: numeric. The factor modifying the opacity alpha for the vertices; typically in [0,1].
- `E.alpha`: numeric. The factor modifying the opacity alpha for the edges; typically in [0,1].

Value

A graph object. A multilevel network with an added color vertex attribute vector and a color edge attribute vector.

Author(s)

Neylson Crepalde, <neylsoncrepalde@gmail.com>

Examples

```r
# Check whether a graph is multilevel
is_multilevel(linked_sim)

# Set the colors for each level
linked_sim <- set_color_multilevel(linked_sim)
```
set_shape_multilevel

Set shapes for levels of a multilevel network

Description

Set vertices shapes for a multilevel network. Default set to "square" (higher level) and "circle" (lower level).

Usage

set_shape_multilevel(x, shape.true = "square", shape.false = "circle")

Arguments

x
  a graph object. Must be a multilevel network.
shape.true
  a character. The shape to be chosen for the higher level (attribute type set to TRUE).
shape.false
  a character. The shape to be chosen for the lower level (attribute type set to FALSE).

Value

A graph object. A multilevel network with an added shape vertex attribute vector.

Author(s)

Neylson Crepalde, <neylsoncrepalde@gmail.com>

Examples

# Check whether a graph is multilevel
is_multilevel(linked_sim)

# Set the shapes for each level
linked_sim <- set_shape_multilevel(linked_sim)
Index

*Topic **datasets**
  linked_sim, 7

bipartite_projection, 8
extract_highlevel, 2
extract_lowlevel, 3
extract_mesolevel, 4, 8
is_bipartite, 5
is_multilevel, 5
layout_, 6
layout_multilevel, 6
linked_sim, 7
linked_sim_matrix (linked_sim), 7
linked_sim_type (linked_sim), 7
mode_transformation, 8
set_color_multilevel, 9
set_shape_multilevel, 10