Package 'dupNodes'

July 22, 2025

Type Package

Title Computes DNSLbetweenness, a Betweenness Measure that Includes
Self-Loops

Version 0.3.0

Author Juan Julián Merelo-Guervós

Maintainer Juan Julián Merelo-Guervós <jjmerelo@gmail.com>

Description Computes a new measure, DNSL betweenness, via the creation
of a new graph from an existing one, duplicating nodes
with self-loops. This betweenness centrality does not drop this
essential information. Implements
Merelo & Molinari (2024) <doi:10.1007/s42001-023-00245-4>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Suggests testthat (>= 3.0.0), knitr, qpdf, rmarkdown, dogesr

Config/testthat/edition 3

Imports igraph, Rdpack

VignetteBuilder knitr

RdMacros Rdpack

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-10 12:00:02 UTC

Contents

1

Index   **9**

---

betweenness.with.duplicated.nodes

Computes betweenness of an igraph object that already has duplicated nodes

---

Description

This is an implementation of the method published in the paper "Intra-family links in the analysis of marital networks" (Merelo and Molinari 2024), which proposed a duplication of nodes that includes self-loops to get a reflect better (and include all information about) the status of all nodes in the network, computing betweenness ignoring the duplicated nodes.

Usage

```
betweenness.with.duplicated.nodes(dup.graph)
```

Arguments

dup.graph   A graph object that has already duplicated nodes with self-loops. Essentially, it
                 will recognize duplicated nodes as those with a prime at the end. Will not return
                 betweenness value for these (it will be exactly the same as the other one).

Details

This method computes betweenness centrality in the regular way for the graph that includes duplicated nodes; it will only return values for the "original" nodes, however.

This object should be created with the function dup.nodes.from.data.frame or dup.nodes.from.graph

Value

A list with betweenness values indexed by node name.

References

Merelo JJ, Molinari MC (2024). "Intra-family links in the analysis of marital networks." *Journal of computational social science*. doi:10.1007/s42001023002454, https://link.springer.com/article/10.1007/s42001-023-00245-4.

Examples

```
library(igraph)
V1 <- c("A", "A","B","C","D","E")
V2 <- c("A", "B","C","D","E","A")
df <- data.frame(V1, V2)

dup.graph <- dup.nodes.from.data.frame(df)
print(betweenness.with.duplicated.nodes(dup.graph))
```

DNSL.betweenness	*Computed betweenness via duplication of nodes with self-loops (Duplicated Nodes with Self Loops)*

Description

Computes the betweenness of nodes in a graph that includes self-loops directly.

Usage

```
DNSL.betweenness(df,first.node = "V1",second.node = "V2")
```

Arguments

df	A data frame that should include at least two columns with the names of the nodes
first.node	The column that indicates the node that's linked
second.node	The column that indicates the other node that's linked

Details

This method computes betweenness centrality of a social network described by a data frame. Use it when you want to just apply the DNSLbetweenness method (Merelo and Molinari 2024) to a data set, without using the intermediate graph obtained by duplicating nodes.

Value

A list with betweenness values indexed by node name

References

Merelo JJ, Molinari MC (2024). "Intra-family links in the analysis of marital networks." *Journal of computational social science*. doi:10.1007/s42001023002454, https://link.springer.com/article/10.1007/s42001-023-00245-4.

Examples

```
library(igraph)
V1 <- c("A", "A","B","C","D","E")
V2 <- c("A", "B","C","D","E","A")
print(DNSL.betweenness(data.frame(V1,V2)))
```

DNSLbetweenness_for_graph

Computes betweenness via duplication of nodes with self-loops for a graph

---

Description

Computes the betweenness of nodes in a graph that includes self-loops directly.

Usage

```
DNSLbetweenness_for_graph(graph.with.self.loops)
```

Arguments

graph.with.self.loops

An igraph object some of whose nodes have connections with themselves (self-loops).

Details

This method computes betweenness centrality via the duplicated nodes method (Merelo and Molinari 2024) for a 'igraph' object. Use it when you want to just apply the DNSLbetweenness method to an already built graph. The intermediate graph obtained by duplicating nodes is hidden and discarded.

Value

A list with DNSLbetweenness values for the nodes in the graph.

References

Merelo JJ, Molinari MC (2024). "Intra-family links in the analysis of marital networks." *Journal of computational social science*. doi:10.1007/s42001023002454, https://link.springer.com/article/10.1007/s42001-023-00245-4.

Examples

```
library(igraph)
V1 <- c("A", "A","B","C","D","E")
V2 <- c("A", "B","C","D","E","A")
print(DNSLbetweenness_for_graph(graph_from_data_frame(data.frame(V1,V2))))
```

---

dup.nodes.from.data.frame

Creates an igraph object from a data frame with self-loops

Description

This is an implementation of the method published in the paper "Intra-family links in the analysis of marital networks" (Merelo and Molinari 2024), which proposed a duplication of nodes that includes self-loops to get a reflect better (and include all information about) the status of all nodes in the network.

Usage

```
dup.nodes.from.data.frame(df,first.node="V1",second.node="V2")
```

Arguments

df	A data frame that should include at least two columns with the names of the nodes
first.node	The column that indicates the node that's linked
second.node	The column that indicates the other node that's linked

Details

This method duplicates nodes with self-loops, and links the duplicated nodes with the original with a weight that is equal to the number of self-loops.

This can be applied to matrimonial networks where there are marriages among members of the same family, or any other type of network with self-loops, and allows to apply centrality (and other) measures, obtaining a meaningful and more accurate result.

You can them compute "corrected" betweenness on the resulting object using the DNSL.betweenness function.

Value

An igraph object with all nodes with self-loops duplicated. These newly created nodes will have the same name as the original, with a "'" added.

References

Merelo JJ, Molinari MC (2024). "Intra-family links in the analysis of marital networks." *Journal of computational social science*. doi:10.1007/s42001023002454, https://link.springer.com/article/10.1007/s42001-023-00245-4.

Examples

```
library(igraph)
V1 <- c("A", "A","B","C","D","E")
V2 <- c("A", "B","C","D","E","A")
df <- data.frame(V1, V2)

dup.graph <- dup.nodes.from.data.frame(df)
print(incident(dup.graph,"A"))
```

---

dup.nodes.from.graph	*Creates an igraph object from a igraph some of whose nodes have self-loops*

Description

This is an implementation of the method published in the paper "Intra-family links in the analysis of marital networks" (Merelo and Molinari 2024), which proposed a duplication of nodes that includes self-loops to get a reflect better (and include all information about) the status of all nodes in the network.

Usage

```
dup.nodes.from.graph(graph.with.self.loops)
```

Arguments

```
graph.with.self.loops
```
An igraph object that includes some nodes with self-loops

Details

This method duplicates nodes in a graph if they include self-loops, and linking duplicated nodes with the original with a weight that is equal to the number of self-loops; it also duplicates all connections from the original node to other node, including the new ones created.

This can be applied to matrimonial (marital) networks where there are marriages among members of the same family, or any other type of network with self-loops, and allows to apply centrality (and other) measures, obtaining a meaningful and more accurate result.

You can them compute "corrected" betweenness on the resulting object using the DNSL.betweenness function.

Value

An igraph object with all nodes with self-loops duplicated, and edges to the original nodes duplicated in the same way. These newly created nodes will have the same name as the original, with a "'" added.

References

Merelo JJ, Molinari MC (2024). "Intra-family links in the analysis of marital networks." *Journal of computational social science.* doi:10.1007/s42001023002454, https://link.springer.com/article/10.1007/s42001-023-00245-4.

Examples

```
library(igraph)
V1 <- c("A", "A","B","C","D","E")
V2 <- c("A", "B","C","D","E","A")

graph.with.self.loops <- graph_from_data_frame(data.frame(V1, V2), directed=FALSE)

dup.graph <- dup.nodes.from.graph(graph.with.self.loops)
print(incident(dup.graph,"A"))
```

| florentine.sn | *Data on matrimonial and commercial links of Florentine families* |

Description

Florentine matrimonial links as an igraph object; it includes marriages as well as commercial links between Florentine families.

Usage

```
data("florentine.sn")
```

Details

Florentine matrimonial links as an igraph object; it includes marriages as well as commercial links between Florentine families as recorded by Padgett and Ansell (Padgett and Ansell 1993) and published by (Wang 2022). It has been enriched with the Medici family genealogy, adding a single self-loop to it.

This is an igraph object where vertices are Florentine families, and edges link two families that have some kind of relationship, either matrimonial or commercial.

Value

A list that can be used as an igraph object, with noble family names as vertices and edges indicating links.

Note

Data originally from the Wikipedia page on the Medici family, and the Padgett and Ansell paper as codified in the Wang dataset.

Author(s)

J. J. Merelo

References

Padgett JF, Ansell CK (1993). "Robust Action and the Rise of the Medici, 1400-1434." *American journal of sociology*, **98**(6), 1259–1319.

Wang D (2022). "Padgett-Florence-Families_Multiplex_Social (1).zip." doi:10.6084/m9.figshare.21545577.v1, https://figshare.com/articles/dataset/Padgett-Florence-Families_Multiplex_Social_1_zip/21545577.

See Also

```
link{doges}
```

Examples

```
library(dupNodes)
library(igraph)
data("florentine.sn")

# All families linked to the Medici
incident(florentine.sn,as.numeric(V(florentine.sn)["MEDICI"]))

# Self loops
E(florentine.sn)[which_loop(florentine.sn)]
```

Index