Package 'CINNA'

January 20, 2025

Title Deciphering Central Informative Nodes in Network Analysis

Version 1.2.2

Date 2023-08-07

Description Computing, comparing, and demonstrating top informative centrality measures within a network. ``CINNA: an R/CRAN package to decipher Central Informative Nodes in Network Analysis'' provides a comprehensive overview of the package functionality Ashtiani et al. (2018) <doi:10.1093/bioinformatics/bty819>.

License GPL-3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Suggests knitr, rmarkdown

Imports igraph, network, sna, centiserve, FactoMineR, factoextra, GGally, pheatmap, corrplot, dendextend, viridis, ggplot2, Rtsne, plyr, circlize, intergraph, utils, stats

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Author Minoo Ashtiani [aut, cre],
Mehdi Mirzaie [ctb],
Mohieddin Jafari [ctb]

Maintainer Minoo Ashtiani <ashtiani.minoo@gmail.com>

Date/Publication 2023-08-08 16:40:02 UTC

Contents

---

calculate_centralities

Centrality measure calculation

Description

This function computes multitude centrality measures of an igraph object.

Usage

```
calculate_centralities(x, except = NULL, include = NULL, weights = NULL)
```

Arguments

x	the component of a network as an igraph object
except	A vector containing names of centrality measures which could be omitted from the calculations.
include	A vector including names of centrality measures which should be computed.
weights	A character scalar specifying the edge attribute to use. (default=NULL)

Details

This function calculates various types of centrality measures which are applicable to the network topology and returns the results as a list. In the "except" argument, you can specify centrality measures that are not necessary to calculate.

Value

A list containing centrality measure values. Each column indicates a centrality measure, and each row corresponds to a vertex. The structure of the output is as follows: - The list has named elements, where each element represents a centrality measure. - The value of each element is a numeric vector, where each element of the vector corresponds to a vertex in the network. - The order of vertices in the numeric vectors matches the order of vertices in the input igraph object. - The class of the output is "centrality".

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Bonacich, P., & Lloyd, P. (2001). Eigenvector like measures of centrality for asymmetric relations. Social Networks, 23(3), 191–201.

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. The Journal of Mathematical Sociology, 2(1), 113–120.

Bonacich, P. (1987). Power and Centrality: A Family of Measures. American Journal of Sociology, 92(5), 1170–1182.

Burt, R. S. (2004). Structural Holes and Good Ideas. American Journal of Sociology, 110(2), 349–399.

Batagelj, V., & Zaversnik, M. (2003). An O(m) Algorithm for Cores Decomposition of Networks, 1–9. Retrieved from

Seidman, S. B. (1983). Network structure and minimum degree. Social Networks, 5(3), 269–287.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5), 604–632.

Wasserman, S., & Faust, K. (1994). Social network analysis : methods and applications. American Ethnologist (Vol. 24). Barrat, A., Barthélemy, M., Pastor Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. Proceedings of the National Academy of Sciences of the United States of America , 101(11), 3747–3752.

Brin, S., & Page, L. (2010). The Anatomy of a Large Scale Hypertextual Web Search Engine The Anatomy of a Search Engine. Search, 30(June 2000), 1–7.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

Brandes, U. (2001). A faster algorithm for betweenness centrality*. The Journal of Mathematical Sociology, 25(2), 163–177.

Estrada E., Rodriguez-Velazquez J. A.: Subgraph centrality in Complex Networks. Physical Review E 71, 056103.

Freeman, L. C., Borgatti, S. P., & White, D. R. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. Social Networks, 13(2), 141–154.

Brandes, U., & Erlebach, T. (Eds.). (2005). Network Analysis (Vol. 3418). Berlin, Heidelberg: Springer Berlin Heidelberg.

Stephenson, K., & Zelen, M. (1989). Rethinking centrality: Methods and examples. Social Networks, 11(1), 1–37.

Wasserman, S., & Faust, K. (1994). Social network analysis : methods and applications. American Ethnologist (Vol. 24).

Brandes, U. (2008). On variants of shortest path betweenness centrality and their generic computation. Social Networks, 30(2), 136–145.

Goh, K.-I., Kahng, B., & Kim, D. (2001). Universal Behavior of Load Distribution in Scale Free Networks. Physical Review Letters, 87(27), 278701.

Shimbel, A. (1953). Structural parameters of communication networks. The Bulletin of Mathematical Biophysics, 15(4), 501–507.

Assenov, Y., Ramrez, F., Schelhorn, S.-E., Lengauer, T., & Albrecht, M. (2008). Computing topological parameters of biological networks. Bioinformatics, 24(2), 282–284.

Diekert, V., & Durand, B. (Eds.). (2005). STACS 2005 (Vol. 3404). Berlin, Heidelberg: Springer Berlin Heidelberg.

Gräßler, J., Koschützki, D., & Schreiber, F. (2012). CentiLib: comprehensive analysis and exploration of network centralities. Bioinformatics (Oxford, England), 28(8), 1178–9.

Latora, V., & Marchiori, M. (2001). Efficient Behavior of Small World Networks. Physical Review Letters, 87(19), 198701.

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245–251.

Estrada, E., Higham, D. J., & Hatano, N. (2009). Communicability betweenness in complex networks. Physica A: Statistical Mechanics and Its Applications, 388(5), 764–774.

Hagberg, Aric, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using NetworkX. No. LA-UR-08-05495; LA-UR-08-5495. Los Alamos National Laboratory (LANL), 2008.

Kalinka, A. T., & Tomancak, P. (2011). linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. Bioinformatics, 27(14), 2011–2012.

Faghani, M. R., & Nguyen, U. T. (2013). A Study of XSS Worm Propagation and Detection Mechanisms in Online Social Networks. IEEE Transactions on Information Forensics and Security, 8(11), 1815–1826.

Brandes, U., & Erlebach, T. (Eds.). (2005). Network Analysis (Vol. 3418). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lin, C.-Y., Chin, C.-H., Wu, H.-H., Chen, S.-H., Ho, C.-W., & Ko, M.-T. (2008). Hubba: hub objects analyzer–a framework of interactome hubs identification for network biology. Nucleic Acids Research, 36(Web Server), W438–W443.

Chin, C., Chen, S., & Wu, H. (2009). cyto Hubba: A Cytoscape Plug in for Hub Object Analysis in Network Biology. Genome Informatics . . . , 5(Java 5), 2–3.

Qi, X., Fuller, E., Wu, Q., Wu, Y., & Zhang, C.-Q. (2012). Laplacian centrality: A new centrality measure for weighted networks. Information Sciences, 194, 240–253.

Joyce, K. E., Laurienti, P. J., Burdette, J. H., & Hayasaka, S. (2010). A New Measure of Centrality for Brain Networks. PLoS ONE, 5(8), e12200.

Lin, C.-Y., Chin, C.-H., Wu, H.-H., Chen, S.-H., Ho, C.-W., & Ko, M.-T. (2008). Hubba: hub objects analyzer–a framework of interactome hubs identification for network biology. Nucleic Acids Research, 36(Web Server), W438–W443.

Hubbell, C. H. (1965). An Input Output Approach to Clique Identification. Sociometry, 28(4), 377.

Dangalchev, C. (2006). Residual closeness in networks. Physica A: Statistical Mechanics and Its Applications, 365(2), 556–564.

Brandes, U. & Erlebach, T. (2005). Network Analysis: Methodological Foundations, U.S. Government Printing Office.

Korn, A., Schubert, A., & Telcs, A. (2009). Lobby index in networks. Physica A: Statistical Mechanics and Its Applications, 388(11), 2221–2226.

White, S., & Smyth, P. (2003). Algorithms for estimating relative importance in networks. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD '03 (p. 266). New York, New York, USA: ACM Press.

Cornish, A. J., & Markowetz, F. (2014). SANTA: Quantifying the Functional Content of Molecular Networks. PLoS Computational Biology, 10(9), e1003808.

Scardoni, G., Petterlini, M., & Laudanna, C. (2009). Analyzing biological network parameters with CentiScaPe. Bioinformatics, 25(21), 2857–2859.

Lin, N. (1976). Foundations of Social Research. Mcgraw Hill.

Borgatti, S. P., & Everett, M. G. (2006). A Graph theoretic perspective on centrality. Social Networks, 28(4), 466–484.

Newman, M. (2010). Networks. Oxford University Press.

Junker, Bjorn H., Dirk Koschutzki, and Falk Schreiber(2006). "Exploration of biological network centralities with CentiBiN." BMC bioinformatics 7.1 : 219.

Pal, S. K., Kundu, S., & Murthy, C. A. (2014). Centrality measures, upper bound, and influence maximization in large scale directed social networks. Fundamenta Informaticae, 130(3), 317–342.

Lin, C.-Y., Chin, C.-H., Wu, H.-H., Chen, S.-H., Ho, C.-W., & Ko, M.-T. (2008). Hubba: hub objects analyzer–a framework of interactome hubs identification for network biology. Nucleic Acids Research, 36(Web Server), W438–W443.

Scardoni, G., Petterlini, M., & Laudanna, C. (2009). Analyzing biological network parameters with CentiScaPe. Bioinformatics, 25(21), 2857–2859.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social Networks, 1(3), 215–239.

Chen, D.-B., Gao, H., L?, L., & Zhou, T. (2013). Identifying Influential Nodes in Large Scale Directed Networks: The Role of Clustering. PLoS ONE, 8(10), e77455.

Jana Hurajova, S. G. and T. M. (2014). Decay Centrality. In 15th Conference of Kosice Mathematicians. Herlany.

Viswanath, M. (2009). ONTOLOGY BASED AUTOMATIC TEXT SUMMARIZATION. Vishweshwaraiah Institute of Technology.

Przulj, N., Wigle, D. A., & Jurisica, I. (2004). Functional topology in a network of protein interactions. Bioinformatics, 20(3), 340–348.

del Rio, G., Koschtzki, D., & Coello, G. (2009). How to identify essential genes from molecular networks BMC Systems Biology, 3(1), 102.

Scardoni, G. and Carlo Laudanna, C.B.M.C., 2011. Network centralities for Cytoscape. University of Verona.

BOLDI, P. & VIGNA, S. 2014. Axioms for centrality. Internet Mathematics, 00-00.

MARCHIORI, M. & LATORA, V. 2000. Harmony in the small-world. Physica A: Statistical Mechanics and its Applications, 285, 539-546.

OPSAHL, T., AGNEESSENS, F. & SKVORETZ, J. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32, 245-251.

OPSAHL, T. 2010. Closeness centrality in networks with disconnected components (http://toreopsahl.com/2010/03/20/closeness-centrality-in-networks-with-disconnected-components/)

Michalak, T.P., Aadithya, K.V., Szczepanski, P.L., Ravindran, B. and Jennings, N.R., 2013. Efficient computation of the Shapley value for game-theoretic network centrality. Journal of Artificial Intelligence Research, 46, pp.607-650.

Macker, J.P., 2016, November. An improved local bridging centrality model for distributed network analytics. In Military Communications Conference, MILCOM 2016-2016 IEEE (pp. 600-605). IEEE. DOI: 10.1109/MILCOM.2016.7795393

DANGALCHEV, C. 2006. Residual closeness in networks. Physica A: Statistical Mechanics and its Applications, 365, 556-564. DOI: 10.1016/j.physa.2005.12.020

Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, Alessandro Vespignani: The architecture of complex weighted networks, Proc. Natl. Acad. Sci. USA 101, 3747 (2004)

See Also

alpha.centrality, bonpow, constraint, centr_degree, eccentricity, eigen_centrality, coreness, authority_score, hub_score, transitivity, page_rank, betweenness, subgraph.centrality, flowbet, infocent, loadcent, stresscent, graphcent, topocoefficient, closeness.currentflow, closeness.latora, communibet, communitycent, crossclique, entropy, epc, laplacian, leverage, mnc, hubbell, semilocal, closeness.vitality, closeness.residual, lobby, markovcent, radiality, lincent, geokpath, katzcent, diffusion.degree, dmnc, centroid, closeness.freeman, clusterrank, decay, barycenter, bottleneck, averagedis, local_bridging_centrality, wiener_index_centrality, group_centrality, dangalchev_closeness_centrality, harmonic_centrality, strength

Examples

```
data("zachary")
p <- proper_centralities(zachary)
calculate_centralities(zachary, include = "Degree Centrality")
```

---

cortex *Macaque Visual Cortex Network*

Description

A graph describing the macaque visual cortex network. Nodes are neocortical areas, 25 of them are participated in visual function in the macaque, and 7 of which are associated with them

Usage

```
data("cortex")
```

Format

an igraph object with "gml" format

References

D.J. Felleman and D.C. van Essen, "Distributed hierarchical processing in the primate cerebral cortex." Cerebral Cortex 1(1), 1-47 (1991).

Examples

```
data("cortex")
print(cortex)
```

---

dangalchev_closeness_centrality
Dangalchev Closeness Centrality

Description

This function computes the Dangalchev Closeness Centrality for nodes in a network. The Dangalchev Closeness Centrality measures closeness by removing nodes and edges, allowing for easier evaluation and handling of unconnected graphs.

Usage

```
dangalchev_closeness_centrality(
  x,
  vids = V(x),
  mode = c("all", "out", "in"),
  weights = NULL
)
```

Arguments

x	An igraph or a network object.
vids	Nodes to be considered in the calculation.
mode	A character value indicating whether the shortest paths "in" or "out" of the nodes in directed graphs should be considered. For undirected graphs, use "all".
weights	A numeric vector indicating the weights of the edges.

Value

A numeric vector including the centrality values for each node. The centrality values represent the Dangalchev Closeness Centrality measure for each node in the network.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Dangalchev, C. (2006). Residual closeness in networks. Physica A: Statistical Mechanics and its Applications, 365, 556-564. DOI: 10.1016/j.physa.2005.12.020

See Also

closeness.residual

Examples

```
data(zachary)

dangalchev_closeness_centrality(zachary)
```

---

drugTarget *Drug Target Network*

Description

A bipartite graph extracted from DrugBank 1.0 database. The network includes two set of nodes including Food and Drug Administration (FDA)-approved drugs and their corresponding protein targets designated by their Uniprot ID. The 1080 drugs and their 519 target proteins nodes are connected via 3766 interactions. Please note that it is a shrunken network in which metabolizing enzymes, carriers and transporters associated with drug metabolism are filtered and solely targets directly related to their pharmacological effects are included. It is also an example of unconnected graphs.

Usage

```
drugTarget
```

Format

an igraph object with "gml" format

References

Barneh, F., Jafari, M., & Mirzaie, M. (2015). Updates on drug–target network; facilitating polypharmacology and data integration by growth of DrugBank database. Briefings in Bioinformatics, bbv094. https://doi.org/10.1093/bib/bbv094

Examples

```
data("drugTarget")
print(drugTarget)
```

---

giant_component_extract

Giant component extraction of a graph

Description

This function extracts the largest connected component or the giant component of the input graph, which can be an "igraph" object or a "network" object, and converts it to an "igraph" object. For bipartite graphs, this function can perform projection before extracting the components.

Usage

```
giant_component_extract(
  x,
  directed = TRUE,
  bipartite_proj = FALSE,
  num_proj = 1
)
```

Arguments

x	An igraph or a network object.
directed	Whether to create a directed graph. (default = TRUE)
bipartite_proj	Whether the bipartite network must be projected or not. (default = FALSE)
num_proj	A number indicating the specific projection to work with, especially for bipartite graphs. (default = 1)

Details

This function identifies the largest component of an "igraph" or a "network" object and returns it as a list containing the giant component as an igraph object and its corresponding edgelist. If the input graph is bipartite and the "bipartite_proj" parameter is set to TRUE, the function can perform projection before extracting the components. You can specify which projection to work with using the "num_proj" parameter.

Value

A list containing the giant component of the input graph. The first element is an igraph object representing the giant component, and the second element is the corresponding edgelist.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Newman, M. (2010). Networks. Oxford University Press.

See Also

`induced.subgraph`, `clusters`

Examples

```
# A graph with 4 vertices

data(zachary)
giant_component_extract(zachary)
```

---

group_centrality   *Group Centrality*

Description

This function computes the Group Centrality for nodes in a network. Group centrality considers a consistent ranking of each node to be calculated, taking into account the diverse possible synergies among possible groups of vertices.

Usage

```
group_centrality(x, vids = V(x))
```

Arguments

x	An igraph or a network object.
vids	Nodes to be considered in the calculation.

Value

A numeric vector of centrality values for each node. The length of the vector is equal to the number of nodes in the network.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Michalak, T.P., Aadithya, K.V., Szczepanski, P.L., Ravindran, B. and Jennings, N.R., 2013. Efficient computation of the Shapley value for game-theoretic network centrality. Journal of Artificial Intelligence Research, 46, pp.607-650.

https://www.civilica.com/Paper-IBIS07-IBIS07_127.html

Examples

```
data(zachary)

group_centrality(zachary)
```

---

harmonic_centrality   *Harmonic Centrality*

Description

This function computes the Harmonic Centrality for nodes in a network. The harmonic centrality metric is defined as the denormalized reciprocal of the harmonic mean of all distances.

Usage

```
harmonic_centrality(
  x,
  vids = V(x),
  mode = c("all", "out", "in"),
  weights = NULL
)
```

Arguments

x	An igraph or a network object.
vids	Nodes to be considered in the calculation.
mode	A character value, indicating the type of degree to consider ("out" for out-degree, "in" for in-degree, "total" for the sum of the two). For undirected graphs, this argument is ignored. The default value is "total".
weights	Numeric vector indicating weights of the edges.

Value

A numeric vector of centrality values for each node. The length of the vector is equal to the number of nodes in the network.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

BOLDI, P. & VIGNA, S. 2014. Axioms for centrality. Internet Mathematics, 00-00.

MARCHIORI, M. & LATORA, V. 2000. Harmony in the small-world. Physica A: Statistical Mechanics and its Applications, 285, 539-546.

OPSAHL, T., AGNEESSENS, F. & SKVORETZ, J. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32, 245-251.

OPSAHL, T. 2010. Closeness centrality in networks with disconnected components (http://toreopsahl.com/2010/03/20/closer centrality-in-networks-with-disconnected-components/)

Examples

```
data(zachary)

harmonic_centrality(zachary)
```

---

kangaroo   *Kangaroo Network*

Description

An undirected graph based on interactions between free-ranging grey kangaroos. A node displays a kangaroo and an edge between two kangaroos demonstrates an interaction. The weights indicate the total count of interactions.

Usage

```
kangaroo
```

Format

an igraph object with "gml" format

References

Kangaroo network dataset – KONECT, October 2016.

TR Grant. Dominance and association among members of a captive and a free-ranging group of grey kangaroos (Macropus giganteus). Animal Behaviour, 21(3):449–456, 1973.

Examples

```
data("kangaroo")
print(kangaroo)
```

---

local_bridging_centrality

Local Bridging Centrality

Description

This function computes the Local Bridging Centrality for nodes in a network. The Local Bridging Centrality classifies nodes based on their structural links among the dense components.

Usage

```
local_bridging_centrality(x, vids = V(x))
```

Arguments

x	An igraph or a network object.
vids	Nodes to be considered in the calculation.

Value

A numeric vector including the centrality values for each node. The centrality values represent the Local Bridging Centrality measure for each node in the network.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Macker, J.P. (2016). An improved local bridging centrality model for distributed network analytics. In Military Communications Conference, MILCOM 2016-2016 IEEE (pp. 600-605). IEEE. DOI: 10.1109/MILCOM.2016.7795393

See Also

betweenness

Examples

```
data(zachary)

local_bridging_centrality(zachary)
```

misc_extract_components

Component extraction of miscellaneous graph formats

Description

This function extracts all components of the input with various formats and converts them to "igraph" objects.

Usage

```
misc_extract_components(
  x,
  directed = TRUE,
  mode = "directed",
  weighted = NULL,
  unibipartite = FALSE,
  diag = TRUE
)
```

Arguments

x	The input can be an edgelist, an adjacency matrix, or a graphNEL object.
directed	Whether to create a directed graph. (default = TRUE)
mode	Character scalar, explains how to interpret the supplied matrix. Possible values are: "directed", "undirected", "upper", "lower", "max", "min", "plus". (default = "directed")
weighted	An argument for specifying whether the graph should be weighted or not. If it is NULL, then an unweighted graph is created. (default = NULL)
unibipartite	A boolean parameter describing whether the input edge list corresponds to a bipartite graph. A TRUE value specifies a bipartite graph, and vice versa. (default = FALSE)
diag	Logical scalar, whether to consider the diagonal of the matrix or not. If it is FALSE, then the diagonal is treated as zeros. (default = TRUE)

Details

This function extracts components from the input object, which can be an edgelist, an adjacency matrix, or a graphNEL object. The result is a list including the components represented as separate graphs.

Value

A list including the components of the input graph as igraph objects. Each element of the list represents a component and is an igraph object. The components are disconnected subgraphs of the input graph.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

See Also

`induced.subgraph`, `components`, `graph_from_adjacency_matrix`

---

pca_centralities   *PCA Centrality Measures*

Description

This function performs Principal Component Analysis (PCA) on centrality measures. It computes the contributions of variables to the principal components and visualizes them.

Usage

```
pca_centralities(
  x,
  scale.unit = TRUE,
  cut.off = 80,
  ncp = 2,
  graph = FALSE,
  axes = c(1, 2)
)
```

Arguments

x	a list containing the computed centrality values
scale.unit	a boolean constant indicating whether data should be scaled to unit variance (default = TRUE)
cut.off	The intensity that must be exceeded in cumulative percentage of variance of eigen values (default = 80)
ncp	number of dimensions in the final results (default = 5)
graph	a boolean constant indicating whether the graph should be displayed (default = FALSE)
axes	a length 2 vector describing the number of components to plot (default = c(1, 2))

Value

A plot illustrating the contributions of variables to the principal components. The x-axis represents the centrality measures, and the y-axis represents the contribution. The higher the contribution value, the more important the centrality measure is in the ranking. The plot helps in identifying the most influential centrality measures.

Examples

```
# Create a data frame with multiple observations
centralities <- data.frame(
  Betweenness = c(0.2, 0.3, 0.5),
 Closeness = c(0.4, 0.2, 0.6),
  Degree = c(0.3, 0.1, 0.4),
  Eigenvector = c(0.1, 0.5, 0.2)
)
pca_centralities(centralities)
```

---

proper_centralities   *Proper centrality measure representation*

Description

This function indicates the proper centrality measures of an igraph object based on the network topology.

Usage

```
proper_centralities(x)
```

Arguments

x   An igraph object.

Details

This function returns a list including the names of centrality measures that are applicable for the input graph based on its topology.

Value

A list including the names of centrality measures that are suitable for the input graph.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

See Also

[calculate_centralities](#)

Examples

```
data("zachary")
proper_centralities(zachary)
```

---

rhesus   *Moreno Rhesus Network*

Description

A directed graph including observed grooming episodes between free ranging rhesus macaques (Macaca mulatta) in Cayo Santiago during a two month period in 1963. Cayo Santiago is an island off the coast of Puerto Rico, which also is named as Isla de los monos (Island of the monkeys). A node indicates a monkey and a directed edge in which a rhesus macaque groomed another rhesus macaque. The weights of edges demonstrates how often this behaviour was seen.

Usage

```
rhesus
```

Format

an igraph object with "gml" format

References

Rhesus network dataset – KONECT, October 2016.

DS Sade. Sociometrics of macaca mulatta I. linkages and cliques in grooming matrices. Folia Primatologica, 18(3-4):196–223, 1972.

Examples

```
data("rhesus")
print(rhesus)
```

---

summary_calculate_centralities
                    Summarize centrality measure calculation results

---

Description

This function computes the minimum, first quartile, median, mean, third quartile, and maximum values of the computed centrality measures.

Usage

```
summary_calculate_centralities(x)
```

Arguments

x   Centrality measure calculation results, typically a numeric vector.

Value

A list summarizing the computed centrality measures. The list includes the following elements: - "minimum": The minimum value of the centrality measures. - "first_quartile": The first quartile (25th percentile) value of the centrality measures. - "median": The median (50th percentile) value of the centrality measures. - "mean": The mean value of the centrality measures. - "third_quartile": The third quartile (75th percentile) value of the centrality measures. - "maximum": The maximum value of the centrality measures.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

---

summary_graph_extract_components

Summarize component extraction of a graph

Description

This function summarizes all components of the input which can be an "igraph" object or a "network" object.

Usage

```
summary_graph_extract_components(
  x,
  directed = TRUE,
  bipartite_proj = FALSE,
  num_proj = 1
)
```

Arguments

x	An igraph or a network object.
directed	A boolean value indicating whether to create a directed graph (default = TRUE).
bipartite_proj	A boolean value indicating whether the bipartite network should be projected (default = FALSE).
num_proj	A number indicating the number of projects specifically for bipartite graphs (default = 1).

Value

A list of igraph objects representing the extracted components of the graph. Each element in the list corresponds to a component, and each component is represented by an igraph object. The components are extracted based on the connectivity of the graph and can include single nodes, disconnected subgraphs, or connected subgraphs. If the graph is directed and the directed parameter is set to FALSE, the function returns the weakly connected components. If the graph is bipartite and the bipartite_proj parameter is set to TRUE, the function returns the projected graph components. If the graph is neither directed nor bipartite, the function returns the connected components.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

---

summary_pca_centralities

Summarize PCA result related to centrality measures

Description

This function summarizes the PCA result related to centrality measures.

Usage

```
summary_pca_centralities(x, scale.unit = TRUE, ncp = 5)
```

Arguments

x	A list containing the computed centrality values.
scale.unit	A boolean value indicating whether the data should be scaled to unit variance (default = TRUE).
ncp	The number of dimensions in the final results (default = 5).

Value

The result of the `pca_centralities` function, which includes the PCA analysis results such as eigenvalues, variance explained, and scores. The returned value is an object of class "PCA" from the FactoMineR package. It contains the following components:

eig	A numeric vector of eigenvalues, indicating the amount of variance explained by each principal component.
var	A numeric vector of proportions of variance explained by each principal component.
ind	A data frame of individual scores, where each row represents an individual and each column represents a principal component.
call	The function call used to create the PCA object.
call2	The call used to compute the PCA analysis.
call3	The call used to project the individuals.
svd	The singular value decomposition of the data matrix.
cor	The correlation matrix of the variables.
cos2	The squared cosines of the variables.
contrib	The contributions of the variables to the principal components.
proj	The projected coordinates of the individuals on the principal components.
quali.sup	The results of the supplementary qualitative variables analysis.
quali.sup.ind	The results of the supplementary individuals analysis.
quanti.sup	The results of the supplementary quantitative variables analysis.
quanti.sup.var	The results of the supplementary variables analysis.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

---

summary_tsne_centralities

Summarize t-Distributed Stochastic Neighbor Embedding (t-SNE) on centrality measures

---

Description

This function summarizes the t-SNE analysis results on centrality measures.

Usage

```
summary_tsne_centralities(x, dims = 2, perplexity = 5, scale = TRUE)
```

Arguments

x	A list containing the computed centrality values.
dims	An integer specifying the number of output dimensions (default = 2).
perplexity	A numeric value representing a flexible measure of the efficient number of neighbors. The performance of t-SNE is fairly robust to changes in perplexity, and typical values are between 5 and 50 (default = 5).
scale	A logical value indicating whether the centrality values should be scaled or not (default = TRUE).

Value

A list containing the following elements: - "Y": A matrix containing the new representations for the objects after t-SNE. - "costs": The cost for every object after the final iteration of t-SNE. The "costs" provide information about the optimization process.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

See Also

Rtsne

---

tsne_centralities	*t-Distributed Stochastic Neighbor Embedding (t-SNE) on centrality measures*

Description

This function applies t-SNE, a dimensionality reduction algorithm, to centrality measures.

Usage

```
tsne_centralities(x, dims = 2, perplexity = 5, scale = TRUE)
```

Arguments

x	A list containing the computed centrality values.
dims	An integer specifying the number of output dimensions (default = 2).
perplexity	A numeric value representing a flexible measure of the efficient number of neighbors. The performance of t-SNE is fairly robust to changes in perplexity, and typical values are between 5 and 50 (default = 5).
scale	A logical value indicating whether the centrality values should be scaled or not (default = TRUE).

Details

t-SNE is a non-linear dimensionality reduction algorithm used for exploring high-dimensional data. It maps multi-dimensional centrality measure data to a lower-dimensional space suitable for analysis and visualization.

Value

A cost plot of t-SNE results, which displays centralities in order of their corresponding costs. The cost plot provides information about the optimization process and the quality of the embedding.

A cost plot of t-SNE results, which displays centralities in order of their corresponding costs. The cost plot is a ggplot object that represents the optimization process and the quality of the embedding. The x-axis represents the iterations of the t-SNE algorithm, and the y-axis represents the cost associated with each iteration. The cost measures the discrepancy between the original high-dimensional space and the low-dimensional embedding. By examining the cost plot, you can assess the convergence and stability of the t-SNE algorithm and evaluate the quality of the embedding.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

van der Maaten, L. (2014). Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research, 15, 3221–3245. Van Der Maaten, L. J. P., & Hinton, G. E. (2008). Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

See Also

Rtsne

---

visualize_association   *Pairwise association plot between centrality measures*

Description

This function computes regression between pair of centrality measures to show more details of association among them.

Usage

```
visualize_association(x, y, scale = TRUE)
```

Arguments

x	a vector containing a centrality measure as independent variable
y	a vector containing a centrality measure as dependent variable
scale	Whether the centrality values should be scaled or not

Details

This function applies regression analysis on two different centrality values in order to find out the corresponding association between them. Regression analysis is a kind of statitiscal method for approximation the association between variables.It asserts that the value of dependent variable changes when the value of independent variable varies.

Value

The function returns a regression plot and the results of the regression analysis between two centrality measures. The function takes two vectors, 'x' and 'y', which represent the independent and dependent centrality measures, respectively. If the 'scale' parameter is set to 'TRUE', the centrality values will be scaled before performing the regression analysis. The function applies regression analysis to estimate the association between the two centrality measures. It creates a scatter plot of the centrality values, with a regression line representing the relationship between the variables. Additionally, the plot includes confidence intervals around the regression line to show the uncertainty of the estimates. The function returns the regression plot and the computed regression results.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Chambers, J. M. (1992). Statistical Models in S. Wadsworth. Pacific Grove, California. Retrieved from [Link to the reference]

Wilkinson, G. N., & Rogers, C. E. (1973). Symbolic Description of Factorial Models for Analysis of Variance. Applied Statistics, 22(3), 392.

---

visualize_correlations

Correlation plot between centrality measures

Description

This function draw correlation plot between pair of centrality measures

Usage

```
visualize_correlations(x, scale = TRUE, method = "pearson")
```

Arguments

x	a list indicating calculated centrality measures
scale	Whether the centrality values should be scaled or not(default=TRUE)
method	a character string describing the type of correlation coefficient (or covariance) to be computed. The proper values are "pearson", "kendall", or "spearman". (default="pearson")

Details

This function illustrates pairwise correlation plot of computed centrality measures. The names of centralities shown in the result plot is abbreviated and compelete names can be seen in "proper_centralities" function. Colors from red to blue indicate the intensity of correlation value. If two centrality measures have an inverse relationship then their correspnding color in plot have to be red and vice versa.

Value

The function generates a correlation plot between pairs of centrality measures. The function takes a list x as input, which contains the computed centrality measures. If the scale parameter is set to TRUE, the centrality values will be scaled before computing the correlations. The method parameter specifies the type of correlation coefficient to be computed: "pearson" (default), "kendall", or "spearman". The function creates a pairwise correlation plot, where each cell represents the correlation between a pair of centrality measures. The color of each cell indicates the strength and direction of the correlation, ranging from red (negative correlation) to blue (positive correlation). The function returns the pairwise correlation plot.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

See Also

ggpairs

---

visualize_dendrogram   *Dendrogram plot among centrality measures*

Description

This function demonstrates the vertice dendrogram of a graph based on a centrality type.

Usage

```
visualize_dendrogram(
  x,
  centrality.type = "Degree Centrality",
  computed_centrality_value = NULL,
  k = 4
)
```

Arguments

x   an igraph object

centrality.type

                         The type of centrality which should be considered.(default="Degree Centrality")

computed_centrality_value

                         A vector containing the values of calculated centrality measure for each node.(default=NULL)

k   number of clusters(default=4)

Value

The function generates a dendrogram plot of a graph's vertices based on a centrality measure. The function takes an igraph object 'x' as input, representing the graph. The 'centrality.type' parameter specifies the type of centrality measure to be considered (default is "Degree Centrality"). If the 'computed_centrality_value' parameter is provided, it should be a vector containing the computed centrality values for each node. Otherwise, the function will compute the specified centrality measure for the graph. The 'k' parameter specifies the number of clusters to be formed from the dendrogram (default is 4). The function creates a dendrogram plot of the vertices, where the branching structure represents the hierarchical clustering based on the centrality measure. The function returns the dendrogram plot.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Galili, T. (2015). dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical clustering. Bioinformatics, 31(22), 3718–3720.

---

visualize_graph	*Graph visualization based on a specific centrality measure*

Description

This function demonstrates the input graph in which the size of nodes indicates calculated centrality value.

Usage

```
visualize_graph(
  x,
  computed_centrality_value = NULL,
  centrality.type = "Degree Centrality"
)
```

Arguments

x   an igraph object

computed_centrality_value

        A vector containing the values of calculated centrality measure for each node.

centrality.type

        The type of centrality which should be calculated.

Details

This function represents the graph in which size of nodes are based on computed centrality value. If the values of wanted centrality measure were computed then by placing them in computed_centrality_value argument to use it for drawing the plot. Otherwise, by only giving the name of favorite centrality measure in centrality.type argument, this function will calculate it and then demonstrates the corresponding graph.

Value

A plot illustrating the input graph, where the size of nodes represents the calculated centrality values. The function takes an igraph object as input and either a vector of precomputed centrality values or the name of the desired centrality measure. If precomputed centrality values are provided in the computed_centrality_value argument, the graph will be plotted based on those values. If only the name of the desired centrality measure is provided in the centrality.type argument, the function will calculate the centrality values and then plot the graph. The plot shows the nodes of the graph, with their sizes indicating the centrality values. The larger the node, the higher the centrality value. This plot helps visualize the distribution of centrality values across the nodes of the graph. The function returns the plot and does not modify the input graph.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

---

visualize_heatmap   *Heatmap plot between centrality measures*

Description

This function draws heatmap between pair of centrality measures

Usage

```
visualize_heatmap(x, scale = TRUE)
```

Arguments

x   a list indicating calculated centrality measures

scale   Whether the centrality values should be scaled or not

Details

This function illustrates the heatmap plot of computed centrality measures.

Value

The function generates a heatmap plot between pairs of centrality measures. The function takes
a list x as input, which contains the computed centrality measures. If the scale parameter is set
to TRUE, the centrality values will be scaled before plotting the heatmap. The function creates a
heatmap plot, where each cell represents the correlation between a pair of centrality measures. The
color of each cell indicates the strength and direction of the correlation. The function returns the
heatmap plot.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

visualize_pair_correlation

Pairwise correlation plot between two centrality measures

Description

This function computes and plots correlation between pair of centrality measures and histogram plots.

Usage

```
visualize_pair_correlation(x, y, scale = TRUE)
```

Arguments

x	a vector containing a centrality measure
y	a vector containing another centrality measure
scale	Whether the centrality values should be scaled or not

Details

This function illustrates the correlation value between two centrality measures and their corresponding scatterplot and histograms.

Value

The function computes and plots the correlation between two centrality measures. The function takes two vectors, 'x' and 'y', representing the centrality measures to be correlated. If the 'scale' parameter is set to 'TRUE', the centrality values will be scaled before computing the correlation. The function creates a correlation plot, which includes a scatterplot of the centrality values and histograms of each measure. The correlation coefficient between the two measures is displayed on the plot. The function returns the correlation plot.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Emerson, J. W., Green, W. A., Schloerke, B., Crowley, J., Cook, D., Hofmann, H., & Wickham, H. (2013). The Generalized Pairs Plot. Journal of Computational and Graphical Statistics, 22(1), 79–91.

See Also

ggpairs

---

wiener_index_centrality

Wiener Index Centrality

Description

This function computes the Wiener Index Centrality for nodes in a network. The Wiener index is a measure of centrality that computes the sum of all shortest paths between a node and all other related nodes in the graph. In essence, it is similar to closeness centrality, but without taking the reciprocal, resulting in a measure with the opposite meaning.

Usage

```
wiener_index_centrality(
  x,
  vids = V(x),
  mode = c("all", "out", "in"),
  weights = NULL
)
```

Arguments

x	An igraph or a network object.
vids	Nodes to be considered in the calculation.
mode	A character value indicating whether the shortest paths "in" or "out" of the nodes in directed graphs should be considered. For undirected graphs, "all" is used.
weights	Numeric vector indicating weights of the edges.

Value

A numeric vector including the centrality values for each node. The centrality values represent the Wiener Index Centrality measure for each node in the network.

Author(s)

Minoo Ashtiani, Mehdi Mirzaie, Mohieddin Jafari

References

Scardoni, G. and Carlo Laudanna, C.B.M.C. (2011). Network centralities for Cytoscape. University of Verona.

Examples

```
data(zachary)

wiener_index_centrality(zachary)
```

---

zachary   *Zachary Karate Club Network*

Description

A graph describing friendships among members of a university karate club. Includes metadata for faction membership after a social partition.

Usage

```
zachary
```

Format

an igraph object with "gml" format

References

W. W. Zachary, "An information flow model for conflict and fission in small groups." Journal of Anthropological Research 33, 452-473 (1977).

Examples

```
data("zachary")
print(zachary)
```

Index