Package ‘streamDAG’

July 20, 2022

Version 0.1.0-2

Date 2022-6-28

Title Descriptors and Methods for Stream DAGs

Author Ken Aho [aut, cre], Arya Legg [ctb]

Maintainer Ken Aho <kenaho1@gmail.com>

Depends R (>= 4.0), igraph

Imports asbio, graphics, stats, plotrix

Suggests ggplot2, sf

Description Provides indices and tools for directed acyclic graphs (DAGs), particularly DAG representations of intermittent streams.

LazyLoad yes

NeedsCompilation no

R topics documented:

`A` *Arcs of a directed graph*

Description

This function and its documentation have been lifted from the *igraph* function `E` with arguments according to DAG conventions. An arc sequence is a vector containing numeric arc ids, with a special class attribute that allows custom operations: selecting subsets of arcs based on attributes, or graph structure, creating the intersection, union of arcs, etc.

Usage

```
A(G, P, path)
```

Arguments

`G`	Graph object of class `igraph`. See `graph_from_literal`.
`P`	A list of node to select arcs via pairs of nodes. The first and second nodes select the first arc, the third and fourth node select the second arc, etc.
`path`	A list of nodes, to select arcs along a path. Note that this only works reliable for simple graphs. If the graph has multiple arcs, one of them will be chosen arbitrarily to be included in the arc sequence.

Details

Arc sequences are usually used as function arguments that refer to arcs of a graph.

An arc sequence is tied to the graph it refers to: it really denoted the specific arcs of that graph, and cannot be used together with another graph.

An arc sequence is most often created by the `A()` function. The result includes arcs in increasing arc id order by default (if none of the `P` and `path` arguments are used). An arc sequence can be indexed by a numeric vector, just like a regular R vector.

Value

An arc sequence of the graph.

Author(s)

Gabor Csardi

See Also

See E

Examples

```
G <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e, e --+ j, j --+ m, f --+ g, g --+ i,
 h --+ i, i --+ k, k --+ l, l --+ m, m --+ n,  n --+ o)

A(G)
```

---

arc.pa.from.nodes *Obtain arc activity outcomes based on bounding nodes*

Description

Given nodal presence absence outcomes for a graph, G, the function calculates arc activity probabilities. For the kth arc with bounding nodes u and v, there are three possibilities for the arc: 1.0, if both u and v are present, 0.0 if both u and v are absent, and 0.5 if only one of u or v is present and the other is absent.

Usage

```
arc.pa.from.nodes(G, node.pa)
```

Arguments

G	Graph object of class igraph. See graph_from_literal.
node.pa	A data frame or matrix of nodal presence absence data with column names corresponding to node names in G.

Author(s)

Ken Aho

Examples

```
murphy_spring <- graph_from_literal(IN_N --+ M1984 --+ M1909, IN_S --+ M1993,
M1993 --+ M1951 --+ M1909 --+ M1799 --+ M1719 --+ M1653 --+ M1572 --+ M1452,
M1452 --+ M1377 --+ M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823,
M823 --+ M759 --+ M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153,
M153 --+ M91 --+ OUT)

data(mur_node_pres_abs)
arc.pa.from.nodes(murphy_spring, mur_node_pres_abs[400:405,][,-1])
```

assort *Assortativity*

Description

Calculates graph assortativity

Usage

```
assort(G, mode = "in.out")
```

Arguments

G	Graph object of class igraph. See graph_from_literal.
mode	One of "in.in", "in.out", "out.out", "out.in", or "all".

Details

The definitive measure of graph assortativity is the assortativity coefficient which is the Pearson correlation coefficient of the degree of pairs of adjacent nodes (Newman, 2002). Let $\overrightarrow{u_i v_i}$ define nodes and directionality of the ith arc, $i = 1, 2, 3, \ldots, m$, let $\gamma, \tau \in -, +$ index the degree type: $- = in, + = out$, and let (u_i^γ, v_i^τ), be the $\gamma-$ and $\tau-$degree of the *i*th arc. Then, the general form of assortativity index is:

$$r\left(\gamma, \tau\right) = m^{-1} \frac{\sum_{i=1}^{m} (u_i^\gamma - \bar{u}^\gamma)(v_i^\tau - \bar{v}^\tau)}{s^\gamma s^\tau}$$

where $\bar{u}^\gamma$ and $\bar{v}^\gamma$ are the arithmetic means of the u_i^γs and v_i^τs, and s^γ and s^τ are the population standard deviations of the u_i^γs and v_i^τs. Under this framework, there are four possible forms to $r\left(\gamma, \tau\right)$ (Foster et al., 2010). These are: $r\left(+, -\right), r\left(-, +\right), r\left(-, -\right)$, and $r\left(+, +\right)$.

Value

Assortativity coefficeint outcome(s)

Author(s)

Ken Aho, , Gabor Csardi wrote degree

References

Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters, 89*(20), 208701.

Foster, J. G., Foster, D. V., Grassberger, P., & Paczuski, M. (2010). Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences, 107*(24), 10815-10820.

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e, e --+ j,
j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l, l --+ m, m --+ n,
n --+ o)
assort(network_a)
```

bern.length — *Botter and Durighetto Bernoulli stream length*

Description

A simple function for calculating the dot product of a vector of stream arc lengths and a corresponding vector of either binary (stream presence or absence) outcomes, probabilities of stream presence or inverse probabilites of stream presence.

Usage

```
bern.length(lengths, pa, mode = "local")
```

Arguments

`lengths`	A numeric vector of stream arc lengths
`pa`	A numeric vector of either binary (stream presence or absence) outcomes, probabilities of stream presence or inverse probabilites of stream presence. A Vector outcome in `lengths` should correspond to an outcome for the same arc in `pa`.
`mode`	One of `"local"` of `"global"`

Value

When `pa` is a vector of binary (stream presence or absence) data, the function provides a measure of instantaneous stream length (in the units used in `lengths`). When `pa` is a vector of probabilities of stream presence, the function provides average stream length (in units used in `lengths`). When `pa` is a vector of inverse probabilites of stream presence, the function provides average communication distance (in units used in `lengths`).

Author(s)

Ken Aho

References

Botter, G., & Durighetto, N. (2020). The stream length duration curve: A tool for characterizing the time variability of the flowing stream length. *Water Resources Research, 56*(8), e2020WR027282.

Examples

```
lengths <- rexp(10, 10)
pa <- rbinom(10, 11, 0.4)
bern.length(lengths, pa)
```

`beta.posterior` *Posterior Beta and Inverse-beta summaries*

Description

Calculates summaries for beta and inverse-beta posteriors given prior probabilities for success, binary data and prior weight specification. Summaries include beta and inverse beta posterior means and variances and stream length and communication distance summaries given that stream `length` is provided for intermittent stream segments.

Usage

```
beta.posterior(p.prior, dat, length = NULL, w = 0.5)
```

Arguments

`p.prior`	Prior probability for success for the beta prior. The beta prior for the probability of success (e.g., stream presence) for *k*th outcome (e.g., stream segment) is defined as: $\theta_k \sim BETA(\alpha, \beta = t\alpha)$, where $\frac{1}{1+t} = p_{prior}$. This results in: $E(\theta_k) = p_{prior}$.
`dat`	An $n \times s$ matrix of binary outcomes, where *n* is the number of observations (e.g., stream observations over time) and *s* is the number experimental units observed, (e.g., stream segments).
`length`	An optional $n \times 1$ vector containing stream segement lengths to allow calculation of mean stream Bernoulli stream length and mean communication distance.
`w`	Weight for the prior distribution compared to the actual data (generally a proportion).

Details

As our Bayesian framework we assume a conjugate beta prior $\theta_k \sim BETA(\alpha, \beta)$ and binomial likelihood $\boldsymbol{x}_k \mid \theta_k \sim BIN(n, \theta_k)$ resulting in the posterior $\theta_k \mid \boldsymbol{x}_k \sim BETA(\alpha + \sum \boldsymbol{x}_k, \beta + n - \sum \boldsymbol{x}_k)$.

Value

Returns a list with components:

`alpha`	The α shape parameters for the beta and inverse beta posteriors.
`beta`	The β shape parameters for the beta and inverse beta posteriors.
`mean`	The means of the beta posteriors.
`var`	The variances of the beta posteriors.
`mean.inv`	The means of the inverse-beta posteriors.
`var.inv`	The variances of the inverse-beta posteriors.
`Com.dist`	If `length` is supplied, the mean communication distances of the network.
`Length`	If `length` is supplied, the mean stream length of the network.
`x`	The observed number of Bernoulli successes over *n* trials observed in `dat`.

Author(s)

Ken Aho

See Also

dinvbeta.

---

biv.bern *Bivariate Bernoulli Distribution*

Description

Densities (probabilities) of a bivariate Bernoulli distribution, Y_1, Y_2.

Usage

```
biv.bern(p11, p10, p01, p00, y1, y2)
```

Arguments

p11	The probability that $y_1, y_2 = 1, 1$.
p10	The probability that $y_1, y_2 = 1, 0$.
p01	The probability that $y_1, y_2 = 0, 1$.
p00	The probability that $y_1, y_2 = 0, 0$.
y1	Outcome for Y_1.
y2	Outcome for Y_2.

Value

Densities (probability) of the joint Bernoulli distribution.

Author(s)

Ken Aho

Examples

```
biv.bern(0.25,0.25,0.25,0.25,1,0)
biv.bern(0.1,0.4,0.3,0.2,1,0)
```

`degree.dists` *Potential degree distributions*

Description

Calculates degree distribution probability density. By default calculates an uncorrelated (random) density for a given degree.

Usage

```
degree.dists(d, exp.lambda = 3/2, normalize = TRUE)
```

Arguments

`d`	degree
`exp.lambda`	if not NULL, allows specification of chaotic `exp.lambda` < `3/2` and correlated stochastic processes `exp.lambda` < `3/2`.
`normalize`	ensures that sum of demsities = 1

Details

In general $f(d) = \exp(-\lambda d)$ where d is the degree. For random degree distributions, $\lambda = \log(3/2)$.

Value

Returns a density plot for a degree.

Author(s)

Ken Aho

See Also

`degree.distribution`, `plot_degree.dist`.

`delete.arcs.pa` *Delete arcs based on presence absence data*

Description

Create a new graph after deleting stream graph arcs based on presence/absence data, e.g., data based on outcomes from STIC (Stream Temperature, Intermittency, and Conductivity) loggers.

Usage

```
delete.arcs.pa(G, pa)
```

Arguments

`G`	A graph object of class "igraph", see `graph_from_literal`
`pa`	A vector of binary = 0,1 values indicating the absence or presence of arcs from `G`.

Author(s)

Ken Aho, Gabor Csardi wrote `delete.edges`

Examples

```
G <- graph_from_literal(a--+b--+c--+d--+e)
delete.arcs.pa(G, c(0,0,1,1))
```

`delete.nodes.pa` *Delete nodes based on presence absence data*

Description

Create a new graph after deleting stream graph nodes based on presence/absence data, e.g., data based on outcomes from STIC (Stream Temperature, Intermittency, and Conductivity) loggers.

Usage

```
delete.nodes.pa(G, pa)
```

Arguments

`G`	A graph object of class "igraph", see `graph_from_literal`
`pa`	A vector of binary = 0,1 values indicating the absence or presence of nodes from `G`.

Author(s)

Ken Aho, Gabor Csardi wrote `delete.vertices`

Examples

```
G <- graph_from_literal(a--+b--+c--+d--+e)
delete.nodes.pa(G, c(0,0,1,1,1))
```

dinvbeta *Inverse Beta Distribution*

Description

Calculates density (dinvbeta), lower-tailed probability (pinvbeta) and obtains random outcomes (rinvbeta) for an inverse beta distribution

Usage

```
dinvbeta(x, alpha, beta)
pinvbeta(x, alpha, beta)
rinvbeta(n, alpha, beta)
```

Arguments

x	Quantile vector or scalar at which to evaluate density or probability.
alpha	Alpha parameter
beta	Beta parameter
n	Number of random outcomes to be generated.

Value

Returns density, probability, and random outcomes for an inverse beta distribution.

Author(s)

Ken Aho and Dwayne Derryberry

See Also

See Also dbeta.

Examples

```
dinvbeta(1,1,1)
pinvbeta(1,1,1)
rinvbeta(1,1,1)
```

`global.summary` *Global Summary*

Description

This function calculates useful DAG global summaries including (but not limited to) size, diameter, number of paths to sink, mean path centrality, mean path length, Randic index, first Zagreb Index, second Zagreb index, atom-bond connectivity, geometric-arithmatic index, harmonic index, assortativity correlation (+, -), and assortativity correlation (+, +).

Usage

```
global.summary(G, sink, inf.paths = FALSE)
```

Arguments

`G`	graph object of class "igraph". See `graph_from_literal`.
`sink`	sink node from graph G
`inf.paths`	logical, consider infinite paths?

Details

Simple global graph measures of complexivity and/or connectivity of a stream DAG include size, diameter,and number of paths to a sink. The size of stream metric is equal to the number of arcs in the stream network. The diameter equals the lenght of the longest path, *height* of the sink, and *in eccentricity* of the sink. The number of paths to sink is equivalent to the number of nodes from which the sink node is reachable, which will be n-1 for a fully active stream. For more information see `I.D`

Author(s)

Ken Aho

References

Kunkler, S. J., LaMar, M. D., Kincaid, R. K., & Phillips, D. (2013). Algorithm and complexity for a network assortativity measure. arXiv Preprint *arXiv:1307.0905*.

Das, K. C., Gutman, I., & Furtula, B. (2011). On atom-bond connectivity index. *Chemical Physics Letters, 511*(4-6), 452-454.

Li, X., & Shi, Y. (2008). A survey on the randic index. *MATCH Commun. Math. Comput. Chem, 59*(1), 127-156.

See Also

`degree`

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)

global.summary(network_a, sink ="o")
```

`I.D` *Generalized DAG indices*

Description

Calculates global generalized topopological indices for a digraph

Usage

```
I.D(G, mode = "gen.rand", alpha = -1/2, mult = FALSE)
```

Arguments

`G`	Graph object of class. See `graph_from_literal`.
`mode`	One of `"gen.rand"`,`"sum.con"`, `"ABC"`, `"GA"`, `"harm"`.
`alpha`	Exponent value for forms of omega with alpha exponent.
`mult`	Logical if `TRUE` use experimental multiplicative measures.

Details

For an arc $a = \overrightarrow{uv}$, $a \in A$, we denote the out degree of u as d_u^+, and the in degree of v as d_v^-. Now let $I(D)$ represent a generalized topopological index for a digraph, D (cf. Deng et al., 2021) that depends on d_u^+ and d_v^-:

$$I(D) = 1/2 \sum_{uv \in A} \omega(d_u^+, d_v^-)$$

Five basic configurations for $I(D)$ can be recognized:

1. If $\omega(x, y) = (xy)^\alpha$, for $\alpha \neq 0$, then $I(D)$ is the *general directed Randic index* (Kincaid et al., 2016) for D. Specific variants include the *Randic index* ($\alpha = -1/2$), the *second Zagreb index* ($\alpha = 1$) and the *second modified Zagreb index* ($\alpha = -1$) (Anthony & Marr, 2021).

2. If $\omega(x, y) = (x + y)^\alpha$, then $I(D)$ is the *first general sum-connectivity index* for D (Deng et al., 2021). Further, if $\omega(x, y) = 2(x + y)^\alpha$, then $I(D)$ is the *sum connectivity* (Zhou & Trinajstic, 2009), and the directed *first Zagreb index* (Anthony & Marr, 2021) for $\alpha = -1/2$ and $\alpha = 1$, respectively .

3. If $\omega(x, y) = \sqrt{((x + y - 2)/xy)}$, then $I(D)$ is the *atom bond connectivity* of D (Estrada et al., 1998).

4. If $\omega(x, y) = \sqrt{xy}/(1/2(x + y))$, then $I(D)$ is the *first geometric-arithmetic index* for D (Vukicevic & Furtula, 2009).

5. If $\omega(x, y) = 2/(x + y)$, then $I(D)$ is the *harmonic index* of D (Favaron et al., 1993).

Value

Index values for a DAG

Author(s)

Ken Aho, Gabor Csardi wrote degree

References

Anthony, B. M., & Marr, A. M. (2021). Directed zagreb indices. *Graphs and Combinatorial Optimization: From Theory to Applications: CTW 2020 Proceedings*, 181-193.

Deng, H., Yang, J., Tang, Z., Yang, J., & You, M. (2021). On the vertex-degree based invariants of digraphs. arXiv Preprint *arXiv:2104.14742*.

Zhou, B., & Trinajstic, N. (2009). On a novel connectivity index. *Journal of Mathematical Chemistry*, 46(4), 1252-1270.

Estrada, E., Torres, L., Rodriguez, L., & Gutman, I. (1998). *An atom-bond connectivity index: Modelling the enthalpy of formation of alkanes*. NISCAIR-CSIR, India.

Vukicevic, D., & Furtula, B. (2009). Topological index based on the ratios of geometrical and arithmetical means of end-vertex degrees of edges. *Journal of Mathematical Chemistry, 46*(4), 1369-1376.

Favaron, O., Maheo, M., & Sacle, J.-F. (1993). Some eigenvalue properties in graphs (conjectures of graffitii). *Discrete Mathematics, 111*(1-3), 197-220.

See Also

degree

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)
I.D(network_a)
```

ICSL — *Integral connectivity scale length (ICSL)*

Description

Integral connectivity scale lengths (ICSL, Western et al. 2013) is the average distance between wet locations using either (1) Euclidean distance or (2) topographically-defined hydrologic distance, e.g., instream hydrologic distance, subsurface distance (Ali and Roy 2009) and outlet distance, in which connected saturated paths must reach the catchment outlet.

Usage

```
ICSL(G, coords = NULL, names = NULL, lengths = NULL,
dist.matrix = NULL, show.dist = FALSE)
```

Arguments

`G`	A graph object of class "igraph", see `graph_from_literal`
`coords`	Spatial coordinates to allow computation of nodal Euclidean distances
`names`	Nodal names
`lengths`	Stream arc lengths or hydrologic arc lengths
`show.dist`	Logical. Show distance matrix?
`dist.matrix`	An optional distance matrix, potentially providing non-Euclidean node distances (e.g., node subsurface distance, etc.). Distance matrix `Labels` in `dist.matrix` must be analogous to those used in `G`. Note that dimensions in `dist.matrix` can be larger than the number of nodes in `G` if, for instance, `dist.matrix` represents distances of the complete wetted network and `G` is a subgraph of the complete wetted network after drying.

Details

Computes either: 1) the average Euclidean distance of connected nodal locations as defined in `G`, if `coords` are provided, 2) if `dist.matrix` is provided, the average nodal distances of a distance matrix provided in `dist.matrix` for nodes remain in `G`, or 3) the instream distances of connected nodal locations if stream lenghts are provided in `lengths`. For 3), the `length` vector will need to trimmed as arcs disappear within intermittent streams (see Examples).

Author(s)

Ken Aho, Gabor Csardi worte underlying functions `distances` and `E`

References

Western, A. W., Bloschl, G., & Grayson, R. B. (2001). Toward capturing hydrologically significant connectivity in spatial patterns. *Water Resources Research, 37*(1), 83-97.

Ali, G. A., & Roy, A. G. (2010). Shopping for hydrologically representative connectivity metrics in a humid temperate forested catchment. *Water Resources Research, 46*(12).

See Also

`distances`

Examples

```
murphy_spring <- graph_from_literal(IN_N --+ M1984 --+ M1909, IN_S --+ M1993,
M1993 --+ M1951 --+ M1909 --+ M1799 --+ M1719 --+ M1653 --+ M1572 --+ M1452,
M1452 --+ M1377 --+ M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823,
M823 --+ M759 --+ M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153,
M153 --+ M91 --+ OUT)

#---- ICSL based on nodal Euclidean distances ----#
data(mur_coords)
ICSL(murphy_spring, coords = mur_coords[,2:3], names = mur_coords[,1])

#---- ICSL based on in-stream length data ----#
data(mur_lengths)
```

```
# Arcs 1 and 3 dry
B <- graph_from_literal(IN_N, M1984, IN_S --+ M1993 --+ M1951 --+ M1909,
M1909 --+ M1799 --+ M1719 --+ M1653 --+ M1572 --+ M1452 --+ M1377 --+ M1254,
M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823 --+ M759 --+ M716,
M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153 --+ M91 --+ OUT)
ICSL(B, lengths = mur_lengths[,2][-c(1,3)], show.dist = TRUE)
```

---

imp.closeness *Improved Closeness Centrality*

Description

Calculates improved closeness centrality of individual nodes in a DAG.

Usage

```
imp.closeness(G)
```

Arguments

G	Graph object of class "igraph", see See `graph_from_literal`.

Details

Improved closeness centrality (Beauchamp, 1965) was developed for weakly connected or disconnected digraphs. The measure is based on the reciprocal of nodal shortest path distances from the *j*th node to the *k*th node, $1/\delta_{j,k}$. For the *j*th node this is:

$$H_j = (n-1)\sum_{j \neq k} 1/\delta_{j,k}$$

where, for disconnected nodes, the reciprocal distance $1/\infty$ is taken to be zero.

Value

Improved closeness centrality of a node

Author(s)

Ken Aho, Gabor Csardi wrote `distances`

References

Beauchamp, M. A. (1965). An improved index of centrality. *Behavioral Science, 10*(2), 161-163.

See Also

`distances`

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)
imp.closeness(network_a)
```

local.summary — *local (generally nodal) summaries of a DAG*

Description

Obtains local (generally nodal) summaries from a DAG

Usage

```
local.summary(G)
```

Arguments

G	Graph of class "igraph". See graph_from_literal

Value

Nodes are returned with values measuring the in degree, alpha centrality, improved path closeness, height, and path length mean, variability, skew, and kurtosis.

Author(s)

Ken Aho, Gabor Csardi wrote degree and alpha_centrality functions.

See Also

degree, alpha_centrality, imp.closeness

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)
local.summary(network_a)
```

mur_arc_pres_abs — *Stream segment presence absence data for Murphy Cr. Idaho*

Description

Simulated multivariate Benroulli outcomes for 27 stream segments, based on their observed marginal probabilities for steam presence and covariance structures. "M"-labelling for nodes indicates "meters above outlet".

Usage

```
data("mur_arc_pres_abs")
```

Format

A data frame with 1000 observations on the following 27 variables.

`IN_N->M1984` a numeric vector

`M1984->M1909` a numeric vector

`M1909->M1799` a numeric vector

`IN_S->M1993` a numeric vector

`M1993->M1951` a numeric vector

`M1951->M1909` a numeric vector

`M1799->M1719` a numeric vector

`M1719->M1653` a numeric vector

`M1653->M1572` a numeric vector

`M1572->M1452` a numeric vector

`M1452->M1377` a numeric vector

`M1377->M1254` a numeric vector

`M1254->M1166` a numeric vector

`M1166->M1121` a numeric vector

`M1121->M1036` a numeric vector

`M1036->M918` a numeric vector

`M918->M823` a numeric vector

`M823->M759` a numeric vector

`M759->M716` a numeric vector

`M716->M624` a numeric vector

`M624->M523` a numeric vector

`M523->M454` a numeric vector

`M454->M380` a numeric vector

`M380->M233` a numeric vector

`M233->M153` a numeric vector

`M153->M91` a numeric vector

`M91->OUT` a numeric vector

`mur_coords` *Coordinates of nodes at Murphy Ck. Idaho*

Description

Coordinates in 12T UTMs of nodes established at Murphy Cr. Idaho.

Usage

```
data("mur_coords")
```

Format

A data frame with 28 observations on the following 3 variables.

`Object.ID` Node name

`x` UTM Easting

`y` UTM Northing

`mur_lengths` *Lengths of Murphy Cr. stream (arc) segments*

Description

Lengths of Murphy Cr. stream (arc) segments

Usage

```
data("mur_lengths")
```

Format

A data frame with 27 observations on the following 2 variables.

`Arcs` Arc names, arrows directionally connect nodes.

`Lengths` Stream segment (arc) length in meters.

Source

Warix, S. R., Godsey, S. E., Lohse, K. A., & Hale, R. L. (2021). Influence of groundwater and topography on stream drying in semi-arid headwater streams. Hydrological Processes, 35(5), e14185.

`mur_node_pres_abs` *Stream node presence absence data for Murphy Cr. Idaho*

Description

A subset of stream node presence absence data from Warix et al. (2019) resulting in binary observations for 28 nodes at 2.5 hr intervals.

Usage

```
data("mur_node_pres_abs")
```

Format

A data frame with 1163 observations on the following 29 variables.

`Datetime` a character vector

`IN_N` a numeric vector

`M1984` a numeric vector

`M1909` a numeric vector

`IN_S` a numeric vector

`M1993` a numeric vector

`M1951` a numeric vector

`M1799` a numeric vector

`M1719` a numeric vector

`M1653` a numeric vector

`M1572` a numeric vector

`M1452` a numeric vector

`M1377` a numeric vector

`M1254` a numeric vector

`M1166` a numeric vector

`M1121` a numeric vector

`M1036` a numeric vector

`M918` a numeric vector

`M823` a numeric vector

`M759` a numeric vector

`M716` a numeric vector

`M624` a numeric vector

`M523` a numeric vector

`M454` a numeric vector

`M380` a numeric vector

`M233` a numeric vector

`M153` a numeric vector

`M91` a numeric vector

`OUT` a numeric vector

References

Warix, S. R., Godsey, S. E., Lohse, K. A., & Hale, R. L. (2021). Influence of groundwater and topography on stream drying in semi-arid headwater streams. *Hydrological Processes, 35*(5), e14185.

mur_seasons_arc_pa *Simulated seasonal arc presence absence data for Murphy Cr*

Description

A data frame with one hundred multivariate Bernoulli simulated outcomes representing spring, summer and fall.

Usage

```
data("mur_seasons_arc_pa")
```

Format

A data frame with 300 observations on the following 28 variables.

'IN_N -> M1984' a numeric vector
'M1984 -> M1909' a numeric vector
'M1909 -> M1799' a numeric vector
'IN_S -> M1993' a numeric vector
'M1993 -> M1951' a numeric vector
'M1951 -> M1909' a numeric vector
'M1799 -> M1719' a numeric vector
'M1719 -> M1653' a numeric vector
'M1653 -> M1572' a numeric vector
'M1572 -> M1452' a numeric vector
'M1452 -> M1377' a numeric vector
'M1377 -> M1254' a numeric vector
'M1254 -> M1166' a numeric vector
'M1166 -> M1121' a numeric vector
'M1121 -> M1036' a numeric vector
'M1036 -> M918' a numeric vector
'M918 -> M823' a numeric vector
'M823 -> M759' a numeric vector
'M759 -> M716' a numeric vector
'M716 -> M624' a numeric vector
'M624 -> M523' a numeric vector
'M523 -> M454' a numeric vector
'M454 -> M380' a numeric vector
'M380 -> M233' a numeric vector
'M233 -> M153' a numeric vector
'M153 -> M91' a numeric vector
'M91 -> OUT' a numeric vector
Season A categorical variable with three levels: "spring" (6/3/2019 - 7/13/2019), "summer" (7/13/2019 - 8/23/2019) and "fall" (8/23/2019 - 10/2/2019)

`path.lengths` *Path Lengths*

Description

Path lengths from all possible nodes to a designated node.

Usage

```
path.lengths(G, node = NULL)
```

Arguments

`G`	Graph of class "igraph". See `graph_from_literal`
`node`	Designated node.

Value

Lengths of paths to a node of interest.

Author(s)

Ken Aho , Gabor Csardi wrote `distances`

`path.lengths.sink` *Path Lengths*

Description

Obtains all path links to a sink

Usage

```
path.lengths.sink(G, sink = NULL, inf.paths = TRUE)
```

Arguments

`G`	Graph object of class "igraph", see: See `graph_from_literal`.
`sink`	sink node from `G`.
`inf.paths`	Logical, consider infinite paths?

Value

Length of path to a sink

Author(s)

Ken Aho, Gabor Csardi wrote `distances`

Examples

```
murphy_spring <- graph_from_literal(IN_N --+ M1984 --+ M1909, IN_S --+ M1993,
M1993 --+ M1951 --+ M1909 --+ M1799 --+ M1719 --+ M1653 --+ M1572 --+ M1452,
M1452--+ M1377 --+ M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823,
M823 --+ M759 --+ M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153,
M153 --+ M91 --+ OUT)

path.lengths.sink(murphy_spring, sink = "OUT")

# with stream lengths as weights
data(mur_lengths)

E(murphy_spring)$weights <- mur_lengths[,2]
path.lengths.sink(murphy_spring, "OUT")
```

`path.visibility` *Path Visibilities*

Description

Shows visibilities of path nodes from one or several source nodes to an sink.

Usage

```
path.visibility(G, degree = "in", source = NULL, sink = NULL, directed = TRUE)

multi.path.visibility(G, degree = "in", source = NULL, sink = NULL, directed = TRUE,
weights = NULL)
```

Arguments

G	Graph of class "igraph". See See `graph_from_literal`
degree	One of "out" for out-degree, "in" for in-degree or "all" for the sum of the two.
source	A starting node for a path. The function `multi.path.visibility` allows multiple starting nodes.
sink	An ending node for a path.
directed	Logical, indicating if `G` is a directed graph.
weights	Refers to $n \times 1$ matrix of weights, with rownames corresponding to node names in `G`.

Value

`multi.path.visibility` Generates tables of path visibilities and visibility summaries for multiple sources to a single sink. The function `path.visibility` calculates path visibilities from single source to a single sink.

Author(s)

Ken Aho, Gabor Csardi wrote `degree`

See Also

degree

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)

multi.path.visibility(network_a, source = c("a","c","f","h"),
sink = "o")

weights <- matrix(nrow = 15, runif(15,0,1),
dimnames = list(letters[1:15], "weights"))
multi.path.visibility(network_a, source = c("a","c","f","h"),
sink = "o", weights = weights)
```

---

plot_degree.dist *Plot degree distributions*

Description

Plots bserved degree distribution against models for uncorrelated random, chaotic and correlated random processes.

Usage

```
plot_degree.dist(G, mode = "all", exp.lambda = c(1.1, 3/2, 2), leg.loc = "topright")
```

Arguments

G	Graph object of class "igraph". See graph_from_literal
mode	Character string, one of "out" for out-degree, "in" for in-degree or "all" for the sum of the two. For undirected graphs this argument is ignored.
exp.lambda	log.lamda = if not NULL, allows specification of chaotic exp.lambda < 3/2 and correlated stochastic processes exp.lambda < 3/2
leg.loc	placement of legend,

Value

Plots processes for observed versus distributions under random or chaotic degrees.

Author(s)

Ken Aho

See Also

degree.dists, degree.distribution

Examples

```
network_a <- graph_from_literal(a --+ b, c --+ d, d --+ e, b --+ e,
e --+ j, j --+ m, f --+ g, g --+ i, h --+ i, i --+ k, k --+ l,
l --+ m, m --+ n,  n --+ o)
plot_degree.dist(network_a)
```

R.bounds — *R bounds Bounds for the correlation of two (or more) Benrnoulli random variables*

Description

Replaces impossible correlations (values too small or too large) with minimum and maximum correlations, respectively.

Usage

```
min_r(p1, p2)
max_r(p1, p2)
R.bounds(p, R, pad = 0.001)
```

Arguments

p1	Probability of success for first random variable
p2	Probability of success for second random variable
p	Vector of marginal probabilities for multivariate Bernoulli random variables, for `R.bounds`.
R	Raw correlation matrix for random variables
pad	Padding (in correlation units) to adjust the returned correlation matrix with respect extremal values.

Details

The functions `r.min` and `r.max` define minimum and maximimum possible correlations. The function `R.bounds` replaces impossibly large or small values with maximally large or small values repectively.

Author(s)

Ken Aho

Examples

```
min_r(0.6, 0.9)
max_r(0.1, 0.2)

x1 <- rep(c(1,0),5)
x2 <- c(rep(1,7), rep(0,3))
x3 <- c(rep(1,3), rep(0,7))
R <- cor(cbind(x1, x2, x3))
R.bounds(c(0.5, 0.7, 1), R)
```

size.intact.to.sink *Size of intact network that includes a particular node*

Description

The size of the subgraph network that include a particular node, e.g., the sink. For a weighted graph, the sum of the weights of the subgraph are given. Thus if weights are stream lengths the function can give the stream length of the portion of the intact stream network that includes the sink.

Usage

```
size.intact.to.sink(G, node = NULL)
```

Arguments

G	A graph object of class "igraph", , see `graph_from_literal`
node	A node of interest.

Value

Returns the size of the graph or subgraph that includes a node of interest.

Author(s)

Ken Aho, Gabor Csardi wrote several important function components including `subgraph`

Examples

```
# Murphy Cr. data, no arc from M1799 to M1719
G <- graph_from_literal(IN_N --+ M1984 --+ M1909, IN_S --+ M1993 --+ M1951,
M1951 --+ M1909 --+ M1799, M1719 --+ M1653 --+ M1572 --+ M1452 --+ M1377,
M1377 --+ M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823 --+ M759,
M759 --+ M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153 --+ M91,
M91 --+ OUT)

data(mur_coords) # coordinate data
spatial.plot(G, mur_coords[,2], mur_coords[,3], names =  mur_coords[,1])

data(mur_lengths) # segment length data

lengths_new <- mur_lengths[-7,] # Drop M1799 -> M1719 arc length
E(G)$weights <- lengths_new[,2]
size.intact.to.sink(G, node = "OUT")
```

spatial.plot *Spatial plot of an igraph object*

Description

Makes a spatial plot of a igraph object, given nodal coordinates

Usage

```
spatial.plot(G, x, y, names = NULL, plot.shapefile = FALSE,
shapefile = NULL, col = 1, text.cex = 0.7, cex = 1, arrow.col = "blue",
arrow.lwd = 1, plot.bg = gray(.9), pch = 21, pt.bg = "orange", grid.lwd = 2, ...)
```

Arguments

G	Graph object, see graph_from_literal.
x	X-coordinates of nodes.
y	Y-coordinates of nodes.
names	Names of nodes, must use the same names as G and correspond to the order of coordinates in x and y.
plot.shapefile	Logical. Plot shapefile (if provided) instead of arrows showing DAG flow. Requires library *ggplot2* and *sf*.
shapefile	Shapefile object brought in using library *sf*
col	point symbol color.
text.cex	Character expansion for text.
cex	Chahracter expnansion of point symbols.
arrow.col	Color of plot arrows.
arrow.lwd	Arrow line width.
plot.bg	Background color of plot.
pch	Plotting character.
pt.bg	Background color for plotting character.
grid.lwd	Grid line width; grid.lwd = 0 suppresses grid.
...	Other arguments to plot

Details

Makes a graph showing arc flow directions to and from spatial node locations, or a spatially explicit graph generated from a stream shapefile showing the stream outlay and node points.

Author(s)

Ken Aho

Examples

```
murphy_spring <- graph_from_literal(IN_N --+ M1984 --+ M1909, IN_S --+ M1993,
M1993 --+ M1951 --+ M1909 --+ M1799 --+ M1719 --+ M1653 --+ M1572 --+ M1452,
M1452--+ M1377 --+ M1254 --+ M1166 --+ M1121 --+ M1036 --+ M918 --+ M823,
M823 --+ M759 --+ M716 --+ M624 --+ M523 --+ M454 --+ M380 --+ M233 --+ M153,
M153 --+ M91 --+ OUT)

data(mur_coords)

x <- mur_coords[,2]
y <- mur_coords[,3]
names <- mur_coords[,1]
spatial.plot(murphy_spring, x, y, names)

## Not run:
# using shapefiles
library(ggplot2); library(sf)
mur_sf <- st_read("Murphy_shapefile/Murphy/Murphy_Creek.shp")
spatial.plot(murphy_spring, x, y, names,
plot.shapefile = TRUE, shapefile = mur_sf)

# modify ggplot
g1 <- spatial.plot(murphy_spring, x, y, names,
plot.shapefile = TRUE, shapefile = mur_sf)
g1 + theme_classic()

## End(Not run)
```

Index