

Package ‘multigraph’

August 19, 2026

Type Package
Title Plot and Manipulate Multigraphs
Version 1.0
Depends R (>= 4.3.0), multiplex (>= 4.0)
Date 2026-08-18
Author Antonio Rivero Ostoic [aut, cre]
Maintainer Antonio Rivero Ostoic <multiplex@post.com>
Description Functions to plot and manipulate multigraphs, signed and valued graphs, bipar-
tite graphs, multilevel graphs, and Cayley colour graphs with various layout options.
URL <https://github.com/mplex/multigraph/>
BugReports <https://github.com/mplex/multigraph/issues/>
Repository CRAN
License GPL-3
NeedsCompilation no
Date/Publication 2026-08-19 12:20:02 UTC

Contents

multigraph-package	2
bmgraph	3
ccgraph	6
conc	8
fred	9
mlgraph	11
multigraph	14
stsm	17
ucinet	19
Index	20

multigraph-package	<i>Plot and Manipulate Multigraphs</i>
--------------------	--

Description

Functions to create and network manipulate multigraphs, bipartite graphs, multilevel graphs, and semigroup Cayley colour graphs.

Details

Package:	multigraph
Type:	Package
Version:	1.0
Depends:	multiplex (≥ 4.0)
Date:	18 August 2026
License:	GPL-3

The **multigraph** package provides functions for plotting complex network structures, including dichotomous and valued graphs. For one-mode data, multigraph displays signed and valued multigraphs with loops, whereas bmgraph represents bipartite graphs for affiliation networks. The mlgraph function visualizes multilevel graphs that “combine” one- and two-mode network data. Cayley colour graphs are drawn with ccgraph to depict relations among ties in multiplex networks encoded in the associated algebraic semigroup structure.

Author(s)

J. Antonio Rivero Ostoic

Maintainer: Antonio Rivero Ostoic <multiplex@post.com>

References

Ostoic, J.A.R. *Algebraic Analysis of Social Networks: Models, Methods and Applications Using R*, Wiley, 2021

Ostoic, J.A.R. “Algebraic Analysis of Multiple Social Networks with multiplex.” *Journal of Statistical Software*, 91(11), 1-41. <doi:10.18637/jss.v092.i11>

See Also

[multiplex-package](#), [incubs](#), [read.dl](#), [read.gml](#), [transf](#), [edgel](#), [zbind](#), [semigroup](#)

Description

A function to create and manipulate bipartite graphs for affiliation or two-mode network networks.

Usage

```
bmgraph(net, layout = c("bip", "bip3", "bip3e", "bipc", "CA", "force", "rand", "stress",
  "circ", "conc", "circ2"), scope, coord, alpha = c(1, 1, 1), showLbs, showAtts,
  att = NULL, lbat = "1", main = NULL, cex.main, col.main, font.main, directed,
  valued, collRecip, cex, pos, lwd, lty, col, ecol, vcol, vcol0, bg, asp,
  seed = NULL, maxiter = 100, bwd, clu, pch, rot, mirrorX, mirrorY, mirrorV,
  mirrorH, hds, vedist, jitter, sort, add, adc, perm, ffamily, fstyle, fsize, fcol,
  mar, scl, new, ...)
```

Arguments

net	A data frame or array representing an affiliation network incidence matrix.
layout	Visualization layout: • bip Bipartite graph (default) • bip3 Bipartite graph with three columns • bip3e Bipartite graph with three columns for events • bipc “Clustered” bipartite graph • force Apply force-directed algorithm • rand Random • circ Circular • stress Apply stress-majorization algorithm • CA Apply correspondence analysis • circ2 Two semi-circles.
scope	(optional) List with scope of graph, edge and vertex characteristics.
coord	(optional, ignores layout) Two-column data frame of vertex coordinates in graph.
alpha	Vector with the alpha colour transparencies for (vertex, edge, bg).
showLbs	(optional and logical) Show vertex labels in graph when available?
showAtts	(optional and logical) Show vertex attribute labels in graph?
att	(optional) Vector or array representing vertex attributes.
lbat	(optional) Labels for vertex attributes.
main	(optional) Main title of the plot graph.
cex.main	Font size of title in plot graph.
col.main	Font colour of title in plot graph.

font.main	Font family of title in plot graph.
directed	(optional and logical) Is the bipartite graph directed?
valued	Is the bipartite graph valued?
collRecip	(optional and logical) Collapse reciprocated edges in (undirected) graph?
cex	(optional) Vertex size.
pos	(optional) Position of labels from vertex: • 1: Below • 2: Left • 3: Above • 4: Right • 0: Centre
lwd	(optional, ignored for valued) Width of edges.
lty	(optional) Shape of edges.
col	(optional) Alias for vcol.
ecol	(optional) Colour of edges.
vcol	(optional) Colour of vertices.
vcol0	(optional, only for pch=21 through pch=25) Colour of vertices contour.
bg	(optional) Background colour of plot graph.
asp	(optional, asp=NA cover whole canvas) Aspect ratio of plot.
seed	(optional, for force, stress and rand) Random seed number for initial coordinates of vertices.
maxiter	(optional, for force, stress and rand) Maximum number of iterations in layout algorithm.
bwd	(optional, default 1) Real number for width between bundle edges.
clu	(optional) Clustering of vertices (see <i>details</i>).
pch	(optional) Symbol(s) for vertices.
rot	(optional) Clockwise rotation in degrees on graph.
mirrorX	(optional) Mirror by X axis.
mirrorY	(optional) Mirror by Y axis.
mirrorV	“Same as mirrorX”.
mirrorH	“Same as mirrorY”.
hds	(optional, experimental) Head scale for arcs.
vedist	(optional, experimental) Vertex–edge distance.
jitter	(optional, in stress or CA) Amount of jitter to add on coordinates.
sort	(optional and logical) Sort vertex labels?
add	(optional) Nodes to add nodes to network domain.
adc	(optional) Nodes to add nodes to network co-domain.
perm	(optional) List of vectors for permutation of network nodes in domain and codomain.

<code>ffamily</code>	(optional) Font family of vertex labels in graph.
<code>fstyle</code>	(optional) Font style of vertex labels in graph with options: <code>bold</code> , <code>italic</code> , <code>bolditalic</code> .
<code>fsize</code>	(optional) Font size of vertex labels in graph.
<code>fcol</code>	(optional) Font colour of text in graph vertices.
<code>mar</code>	(optional) Plot margins.
<code>scl</code>	(optional, experimental) Numerical scalar (x and y) or vector (x , y) of graph scale.
<code>new</code>	(optional, logical) Plot new graph on canvas afterwards?
<code>...</code>	Additional argument items.

Details

Bipartite graphs serve as visual aids for affiliation or two-mode networks. While affiliation networks are typically represented as data frames, they can also be visualized using three-dimensional arrays. In such case, each level corresponds to a specific type of connection, resulting in parallel edges within the bipartite graph. Additionally, a bipartite network can be generated using a force-directed algorithm to create a visual representation.

With bipartite graphs consisting of two sets of vertices, clustering information, such as vertex colours, can be stored in a list `vclu` with two vectors, one for each vertex set. It is possible to group all members of a vertex set into a single class by setting the corresponding vector to `NULL`.

Value

A plot of the two-mode network as a bipartite graph or multigraph with a projection.

See Also

[multigraph](#), [frcd](#), [stsm](#), [conc](#)

Examples

```
# Two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 3 ) )

# Plot network as bipartite graph
bmgraph(arr)

# Apply force directed algorithm for plot
bmgraph(arr, layout = "force")

# Apply Correspondence Analysis
bmgraph(arr, layout = "CA", asp = NA)
```

ccgraph

*Cayley Colour Graph***Description**

A function to create and manipulate Cayley colour graphs of semigroup structures.

Usage

```
ccgraph(x, main = NULL, seed = 0, maxiter = 100, alpha = c(1, 1, 1),
  scope, loops, collRecip, undRecip, showLbs, cex.main, col.main, font.main,
  conc, coord, clu, cex, lwd, pch, lty, bwd, bwd2, att, bg, mar, pos, asp,
  ecol, vcol, vcol0, lbs, col, lbat, swp, swp2, rot, vedist, vldist,
  mirrorX, mirrorY, mirrorD, mirrorL, mirrorV, mirrorH, ffamily, fstyle,
  fsize, fcol, scl, hds, nr, gens, flip, ...)
```

Arguments

x	An algebraic structure like a ‘Semigroup’ class object (cf. semigroup).
main	(optional) Main title of the plot graph.
seed	(optional) Random seed number for initial coordinates of vertices.
maxiter	(optional) Maximum number of iterations in layout algorithm.
alpha	Vector with the alpha colour transparencies for (vertex, edge, bg).
scope	(optional) List with scope of graph, edge and vertex characteristics.
loops	(optional and logical) Plot loops in graph?
collRecip	(optional and logical) Collapse reciprocated edges in graph?
undRecip	(optional and logical) Plot reciprocated edges as undirected?
showLbs	(optional and logical) Show vertex labels in graph when available?
cex.main	Font size of title in plot graph.
col.main	Font colour of title in plot graph.
font.main	Font family of title in plot graph.
conc	(optional and logical) Is layout concentric?
coord	(optional, ignored if "conc") Two-column data frame of vertex coordinates in graph.
clu	(optional) Clustering of vertices.
cex	(optional) Vertex size.
lwd	(optional) Width of edges.
pch	(optional) Symbol for vertices.
lty	(optional) Shape of edges.
bwd	(optional, default 1) Real number for width between bundle edges.

bwd2	(optional, experimental) width of the bundle loop edges.
att	(optional) Vector or array representing vertex attributes.
bg	(optional) Background colour of plot graph.
mar	(optional) Vector with margins for plot.
pos	(optional) Position of labels from vertex: • 1: Below • 2: Left • 3: Above • 4: Right • 0: Centre
asp	(optional, asp=NA cover whole canvas) Aspect ratio of plot.
ecol	(optional) Colour of edges.
vcol	(optional) Colour of vertices.
vcol0	(optional) Colour of vertices contour.
lbs	(optional) Vertex labels.
col	(optional) Alias for vcol.
lbat	(optional) Labels for vertex attributes.
swp	(optional and logical) Swap bundle patterns?
swp2	(optional and logical) Swap reciprocals?
rot	(optional) Clockwise rotation in degrees on graph.
vedist	(optional) Vertex–edge distance.
vldist	(optional) Vertex–loop distance.
mirrorX	(optional) Mirror by X axis.
mirrorY	(optional) Mirror by Y axis.
mirrorD	(optional) Mirror reflection across diagonal $Y = X$.
mirrorL	(optional) Mirror reflection across diagonal $Y = -X$.
mirrorV	“Same as mirrorX”.
mirrorH	“Same as mirrorY”.
ffamily	(optional) Font family of vertex labels in graph.
fstyle	(optional) Font style of vertex labels in graph with options: bold, italic, bolditalic.
fsize	(optional) Font size of vertex labels in graph.
fcol	(optional) Font colour of text in graph vertices.
scl	(optional) Numerical scalar (x and y) or vector (x, y) of graph scale.
hds	(optional) Head scale for arcs.
nr	(conc layout) Number of radii.
gens	(optional when absent) Semigroup generators in x .
flip	(optional and logical) Flip loops per D -class?
...	Additional argument items.

Details

The Cayley colour graph is a graphical representation of the relationships among relations in the relational structure of a given multiplex network. Both nodes and directed edges represent string relations, and each shape and colour corresponds to a specific generator relation of the semigroup structure.

Value

A plot of the semigroup or group structure.

See Also

[semigroup](#), [multigraph](#), [frcd](#), [conc](#)

Examples

```
# Create an abstract semigroup from random data
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 1 ) )

S <- semigroup(arr)

# plot semigroup Cayley graph
ccgraph(S)
```

conc	<i>Concentric Layout</i>
------	--------------------------

Description

A function to compute the graph coordinated system with a concentric layout.

Usage

```
conc(net, nr, irot, inv, iflip, mirror = c("N", "X", "Y", "D", "L"), ...)
```

Arguments

- | | |
|--------|--|
| net | Array representing the network relations. |
| nr | Scalar with number of radii, or a vector with clustering of actors. |
| irot | Scalar or vector with “internal rotation” for each circle from closer to centre point to further away. |
| inv | (optional and logical) Invert circles ordering? |
| iflip | (optional and logical) Alternate circles flipping? |
| mirror | Mirror transformation: <ul style="list-style-type: none">• N Identity (default) |

- X Reflection through the vertical centre line
- Y Reflection through the horizontal centre line
- D Reflection across diagonal $Y = X$
- L Reflection across diagonal $Y = -X$

... Additional argument items.

Details

An “Euclidean plane” computes the coordinated system with a concentric layout with at least two radii (unless $n = 1$). In case that the number of radii is not specified in `nr`, approx. half of the vertices are located at one radius and half in another one.

The clustering of the actors may be used to establish the location of the vertices in different radii as a numerical, character, or a factor vector.

Value

A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

See Also

[multigraph](#), [bmgraph](#), [frcd](#), [stsm](#)

Examples

```
# Two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 3 ) )

# Coordinates for the concentric layout with two radii
coord <- conc(arr, nr = 2)

# Plot multigraph with customized coordinates
multigraph(arr, coord = coord)
```

frcd

Force Directed Layout

Description

A function to compute the graph coordinated system with a force directed layout algorithm.

Usage

```
frcd(net, seed = seed, maxiter, drp, scl, mov, ...)
```

Arguments

<code>net</code>	Array representing network relations.
<code>seed</code>	(mandatory) Seed of initial layout (see <i>details</i>).
<code>maxiter</code>	(optional) Maximum number of iterations.
<code>drp</code>	(optional, for valued networks) Drop values less than specified.
<code>scl</code>	(optional and experimental) Numerical scalar (x and y) or vector (x, y) of graph scale.
<code>mov</code>	(optional and experimental) Scalar (x and y) or vector (x, y) to move graph coordinates.
<code>...</code>	Additional argument items.

Details

This function is meant as an internal routine for graph visualization with a force-directed layout procedure. However, it can be used to set the coordinate system with the `coord` option in functions [multigraph](#) and in [bmgraph](#). In such case, the coordinate system of the graph starts with a random displacement of nodes where NULL in the `seed` argument implies an initial seed based on the computer clock watch, and the number of iterations in `maxiter` is $60 + n$.

Value

A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

References

Fruchterman, T.M.J., & Reingold, E.M. Graph drawing by force-directed placement. *Software-Practice & Experience*, 21(11), 1129-1164. 1991.

See Also

[multigraph](#), [bmgraph](#), [stsm](#), [conc](#)

Examples

```
# Two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 3 ) )

# Coordinates for the force directed layout with random start
coord <- frcd(arr, seed = NULL)

# Plot multigraph with forced directed coordinates
multigraph(arr, coord = coord)
```

mlgraph

*Multilevel Graph***Description**

A function to create and manipulate multilevel graphs.

Usage

```
mlgraph(net, layout = c("circ", "force", "stress", "rand", "conc", "bip"), main = NULL,
        seed = NULL, maxiter = 100, directed = TRUE, alpha = c(1, 1, 1), scope, collRecip,
        lbs, showLbs, showAtts, cex.main, col.main, font.main, coord, clu, cex, lwd, pch,
        lty, bwd, bwd2, att, bg, mar, pos, asp, ecol, vcol, vcol0, col, lbat, loops, rot,
        mirrorX, mirrorY, mirrorD, mirrorL, mirrorV, mirrorH, hds, scl, vedist, vldist,
        flip, ffamily, fstyle, fsize, fcol, valued, modes, elv, lng, nr, new, jitter, ...)
```

Arguments

net	A “Multilevel” class object or a 3D array with clustering information.
layout	Visualization layout: • circ Circular • force Force-directed • stress Stress-majorization • rand Random • conc Concentric • bip Bipartite graph
main	(optional) Main title of the plot graph.
seed	(optional, for force, stress and rand) Random seed number for initial coordinates of vertices.
maxiter	(optional when seed) Maximum number of iterations.
directed	(logical) Are graph edges in domain directed?
alpha	Vector with the alpha colour transparencies for (vertex, edge, bg).
scope	(optional) Scope of graph, edges, and vertices.
collRecip	(optional, logical and experimental) Collapse reciprocated edges in graph domain?
lbs	(optional) List of customized vertex labels for domain and codomain.
showLbs	(optional and logical) Display vertex labels in graph?
showAtts	(optional and logical) Display vertex attribute labels?
cex.main	Font size of title in plot graph.
col.main	Font colour of title in plot graph.
font.main	Font family of title in plot graph.

coord	(optional) Data frame with graph coordinates for vertices after binomial projection.
clu	(optional) Clustering of vertices as a list of vectors with integers or NULL (see <i>details</i>).
cex	(optional) Vertex size.
lwd	(optional, ignored for valued) Width of edges.
pch	(optional) Symbols for vertices in domain and in codomain.
lty	(optional) Shapes of edges in domain and in codomain.
bwd	(optional, default 1) Real number for width between bundle edges.
bwd2	(optional) Width of bundle loop edges.
att	(optional) Vector or array representing vertex attributes.
bg	(optional) Background colour of plot graph.
mar	(optional) Vector with margins for plot.
pos	(optional) Positions of labels from vertex: • 1: Below • 2: Left • 3: Above • 4: Right • $\emptyset$: Centre
asp	(optional, asp=NA cover whole canvas) Aspect ratio of plot.
ecol	(optional) Colour of edges by type in domain and codomain.
vcol	(optional) Colour of vertices.
vcol0	(optional, only for pch=21 through pch=25) Colour of vertices contour.
col	(optional) Alias for vcol.
lbat	(optional) Labels for vertex attributes.
loops	(optional and logical) Plot loops in graph?
rot	(optional) Clockwise rotation in degrees on graph.
mirrorX	(optional) Mirror by X axis.
mirrorY	(optional) Mirror by Y axis.
mirrorD	(optional) Mirror reflection across diagonal $Y = X$.
mirrorL	(optional) Mirror reflection across diagonal $Y = -X$.
mirrorV	“Same as mirrorX”.
mirrorH	“Same as mirrorY”.
hds	(optional, for directed) Head scale of arcs.
scl	(optional) Numerical scalar (x and y) or vector (x, y) of graph scale.
vedist	(optional) Vertex–edge distance.
vldist	(optional) Vertex–loop distance.
flip	(optional and logical) Flip loops?

<code>ffamily</code>	(optional) Font family of vertex labels in graph.
<code>fstyle</code>	(optional) Font style of vertex labels in graph with options: <code>bold</code> , <code>italic</code> , <code>bolditalic</code> .
<code>fsize</code>	(optional) Font size of vertex labels in graph.
<code>fcol</code>	(optional) Font colour of text in graph vertices.
<code>valued</code>	(optional and logical) Depict a valued graph?
<code>modes</code>	(optional, for “Multilevel” class) Vector of domains and codomains in net matrices.
<code>elv</code>	(experimental for “bip”) Elevation of intra-domain edges.
<code>lng</code>	(experimental for “bip”) Elevation control intra-domain edges.
<code>nr</code>	(optional, for <code>conc</code>) Integer or <code>NULL</code> with number of radii layout.
<code>new</code>	(optional, logical) Plot a new graph on existing canvas?
<code>jitter</code>	(optional, for <code>stress</code>) Amount of jitter to add on coordinates.
<code>...</code>	Additional argument items.

Details

Multilevel graphs represent networks with multiple “levels”, such as distinct domains within a network structure. These networks may include ties both within and between domains.

Because this function accepts many arguments, they may be collected in a list and supplied through the `scope` option. The list can contain vectors, scalars, or combinations of both for rendering.

The bundle widths `bwd` and `bwd2` range from 0, where edges are collapsed, to the default value of 1, depending on vertex size. Values greater than one are allowed when `valued` option is used.

In multilevel structures, argument `clu` serves to assign network members to classes. Setting the corresponding vector to `NULL` places all members of a domain or codomain in a single class. Likewise, setting argument `nr` to `NULL` in the `conc` layout uses two radii, one for each domain.

Value

A plot of the multilevel graph structure for the network.

See Also

[mlvl](#), [multigraph](#), [bmgraph](#), [frcd](#), [stsm](#), [conc](#)

multigraph

*Multigraphs and Valued Multigraphs***Description**

A function to create and manipulate multigraphs and valued multigraphs with different layout options.

Usage

```
multigraph(net, layout = c("circ", "force", "stress", "conc", "rand"), scope,
  directed = TRUE, loops, signed, valued, values, lbs, showLbs, att,
  lbat, showAtts, main = NULL, cex.main, col.main, font.main, coord,
  collRecip, undRecip, seed = NULL, maxiter = 100, clu, cex, cex2,
  pch, lwd, lty, vcol, vcol0, col, ecol, bwd, bwd2, pos, bg, bg2, asp,
  drp, add, swp, swp2, flip, alpha = c(1, 1, 1, 1), rot, mirrorX,
  mirrorY, mirrorD, mirrorL, mirrorV, mirrorH, scl, hds, vedist,
  vldist, mar, ffamily, fstyle, fsize, fsize2, fcol, fcol2, lclu, sel,
  new, mai, lscl, rm.isol, nr, jitter, ...)
```

Arguments

net	Array with usually three dimensions of stacked matrices where the multiple relations are placed.
layout	Visualization layout: • circ Circular • force Force-directed • stress Stress-majorization • conc Concentric • rand Random
scope	(optional) Scope of graph, edges, and vertices (see <i>details</i>).
directed	(logical) Are graph edges directed?
loops	(optional and logical) Plot loops in graph?
signed	(optional and logical) Has the graph a signed structure?
valued	(optional and logical) Depict a valued graph?
values	(optional and logical) Print values of bonds in edges?
lbs	(optional) Vector of customized vertex labels.
showLbs	(optional and logical) Display vertex labels in graph?
att	(optional) Vector or array representing vertex attributes.
lbat	(optional) Vector of labels for vertex attributes.
showAtts	(optional and logical) Display vertex attribute labels?
main	(optional) Main title of the plot graph.

cex.main	Font size of title in plot graph.
col.main	Font colour of title in plot graph.
font.main	Font family of title in plot graph.
coord	(optional) Data frame with graph coordinates for vertices.
collRecip	(optional and logical) Collapse reciprocated edges in graph?
undRecip	(optional and logical) Plot reciprocated edges as undirected?
seed	(optional, for force, stress and rand) Random seed number for initial coordinates of vertices.
maxiter	(optional when seed) Maximum number of iterations.
clu	(optional) Clustering for vertices (see <i>details</i>).
cex	(optional) Vertex size.
cex2	(optional, for values) Size of background in values.
pch	(optional) Symbol(s) for vertices.
lwd	(optional, ignored for valued) Width of edges.
lty	(optional) Shape of edges.
vcol	(optional) Colour of vertices.
vcol0	(optional, only for pch=21 through pch=25) Colour of vertices contour.
col	(optional) Alias for vcol.
ecol	(optional) Colour of edges.
bwd	(optional, default 1) Real number for width between bundle edges.
bwd2	(optional) Width of bundle loop edges.
pos	(optional) Position of labels from vertex: • 1: Below • 2: Left • 3: Above • 4: Right • 0: Centre
bg	(optional) Background colour of plot graph.
bg2	(optional, for values) Background colour.
asp	(optional, asp=NA cover whole canvas) Aspect ratio of plot.
drp	(optional for valued) Drop values in network less than specified.
add	(optional) Vector of vertices to add to graph.
swp	(optional and logical) Swap bundle patterns?
swp2	(optional and logical) Swap reciprocal arcs?
flip	(optional and logical) Flip loops?
alpha	Vector with the alpha colour transparencies for (vertex, edge, bg).
rot	(optional) Clockwise rotation in degrees on graph.
mirrorX	(optional) Mirror by X axis.

<code>mirrorY</code>	(optional) Mirror by Y axis.
<code>mirrorD</code>	(optional) Mirror reflection across diagonal $Y = X$.
<code>mirrorL</code>	(optional) Mirror reflection across diagonal $Y = -X$.
<code>mirrorV</code>	“Same as <code>mirrorX</code> ”.
<code>mirrorH</code>	“Same as <code>mirrorY</code> ”.
<code>scl</code>	(optional) Numerical scalar (x and y) or vector (x, y) of graph scale.
<code>hds</code>	(optional, for directed) Head scale of arcs.
<code>vedist</code>	(optional) Vertex–edge distance.
<code>vldist</code>	(optional) Vertex–loop distance.
<code>mar</code>	(optional) Vector with margins for plot.
<code>ffamily</code>	(optional) Font family of vertex labels in graph.
<code>fstyle</code>	(optional) Font style of vertex labels in graph with options: bold, italic, bolditalic.
<code>fsiz</code>	(optional) Font size of vertex labels in graph.
<code>fsiz2</code>	(optional) Font size for values.
<code>fc01</code>	(optional) Font colour of text in graph vertices.
<code>fc02</code>	(optional) Font colour of text in values.
<code>lclu</code>	(optional) Vector with “levels” in <code>clu</code> (see <i>details</i>).
<code>sel</code>	(optional) Vector with selection of vertex labels to plot.
<code>new</code>	(optional, logical) Plot a new graph on existing canvas?
<code>mai</code>	(optional) Vector with inner margins for plot.
<code>lscl</code>	(optional, for valued) Loop scale in graph.
<code>rm.isol</code>	(optional and logical) Remove isolated vertices?
<code>nr</code>	(optional, for conc) Number of radii.
<code>jitter</code>	(optional, for stress) Amount of jitter to add on coordinates.
<code>...</code>	Additional argument items.

Details

Multigraphs contain parallel edges that represent different relation types within a network. By default, the function uses a circular layout in which tie types are distinguished by shape and a grey colour scale. For undirected multigraphs, reciprocal relations are collapsed automatically, although this behaviour can be disabled. Vertex symbols and colours can be combined by assigning each network member to a class through the clustering option. Vertex sizes can also be specified with a vector whose length matches the network order.

Because the function accepts many arguments, these may be collected in a list and supplied through the `scope` option. The list may include vectors, lists, scalars, or combinations of these to describe graph characteristics.

The bundle width `bwd`, and `bwd2` for loops, ranges from 0, where edges are collapsed, to the default value of 1, depending on vertex size. Values above one are allowed when the `valued` option is used. For large or dense networks, argument `vedist` adjusts the distance between vertices and edges.

For some applications, such as dynamic networks, the ordering of clustering “levels” in `clu` must be specified; this is done with argument `lclu`.

If coordinates are supplied, the `layout` option is ignored.

When argument `new` is used to plot a graph with a background image, previous plots may need to be initialized with package functions equivalent to `graphics::plot.new()`.

Value

A plot of the network as a multigraph or a valued multigraph.

See Also

[bmgraph](#), [ccgraph](#), [frcd](#), [stsm](#), [conc](#)

Examples

```
# Two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 3 ) )

# Plot multigraph of this network
multigraph(arr)

# Plot with a force directed algorithm
multigraph(arr, layout = "force")

# As valued graph
multigraph(arr, valued = TRUE)

# As signed graph
multigraph(arr, signed = TRUE)

# With loops and customized vertex size
multigraph(arr, cex = 3, loops = TRUE)
```

stsm

Stress Majorization Layout

Description

A function to compute the graph coordinated system with a stress majorization layout algorithm.

Usage

```
stsm(net, seed = seed, maxiter = 40, drp, jitter, method, ...)
```

Arguments

<code>net</code>	Array for network relations.
<code>seed</code>	(mandatory) the seed of the initial layout (see <i>details</i>).
<code>maxiter</code>	(optional) Maximum number of iterations.
<code>drp</code>	(optional, for valued networks) Drop values less than specified.
<code>jitter</code>	(optional) The amount of jitter to add on coordinates.
<code>method</code>	(optional, default binary) Initial distance method for graph vertices.
<code>...</code>	Additional argument items.

Details

Like the function `frcd`, this routine serves as an internal tool for graph visualization. It is also designed to establish the coordinate system using the `coord` option within the `multigraph` and `bmgraph` functions.

In this scenario, the graph's coordinate system commences with nodes randomly positioned, and if `NULL` is entered in the `seed` argument. An initial seed will be generated based on the computer clock watch where the number of iterations in `maxiter` is 40.

Value

A data frame with a coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

References

Gansner, E.R., Koren, Y., & North, S. *Graph drawing by stress majorization*. In Graph Drawing 2004: 12th International Symposium, New York, NY, USA, revised selected papers. Springer. pp. 239-250. 2005.

See Also

`multigraph`, `bmgraph`, `frcd`, `conc`

Examples

```
# Two binary relations among three elements
arr <- round( replace( array(runif(18), c(3,3,2)), array(runif(18),
  c(3,3,2))>.5, 3 ) )

# Coordinates for stress majorization layout with random start
coord <- stsm(arr, seed = NULL)

# Plot multigraph with stress majorization coordinates
multigraph(arr, coord = coord)
```

ucinet*External Datasets in dl Format.*

Description

External datasets in **multigraph** package are for illustration purposes, and they have a **Ucinet** dl format. The four datasets need to be transformed with function `read.dl` from **multiplex** package for visualization according to the nature of the network.

Format

- `davis.dl` Southern woman: (Davis, Gardner, Gardner, 1941) affiliation network 18×14 .
- `sampson.dl` Monastery novices: (Sampson, 1969) directed, multiplex, valued, longitudinal network $18 \times 18 \times 10$.
- `padgett.dl` Florentine families: (Padgett and Ansell, 1993) undirected, multiplex network $16 \times 16 \times 2$.
- `padgw.dl` Florentine families' attributes: table 16×3 .

Source

UCINET 6 for windows: *Software for social network analysis.*

Index

- * **IO**
 - multigraph-package, 2
- * **datasets**
 - ucinet, 19
- * **data**
 - multigraph-package, 2
- * **file**
 - multigraph-package, 2
- * **graphics**
 - bmgraph, 3
 - ccgraph, 6
 - conc, 8
 - frcd, 9
 - mlgraph, 11
 - multigraph, 14
 - stsm, 17
- * **graphs**
 - multigraph-package, 2
- * **manip**
 - conc, 8
 - frcd, 9
 - multigraph-package, 2
 - stsm, 17
- * **math**
 - ccgraph, 6
- bmgraph, 3, 9, 10, 13, 17, 18
- ccgraph, 6, 17
- conc, 5, 8, 8, 10, 13, 17, 18
- davis.dl (ucinet), 19
- edgel, 2
- frcd, 5, 8, 9, 9, 13, 17, 18
- graphics::plot.new(), 17
- incubs, 2
- mlgraph, 11
- mlvl, 13
- multigraph, 5, 8–10, 13, 14, 18
- multigraph-package, 2
- padgett.dl (ucinet), 19
- padgw.dl (ucinet), 19
- read.dl, 2, 19
- read.gml, 2
- sampson.dl (ucinet), 19
- semigroup, 2, 6, 8
- stsm, 5, 9, 10, 13, 17, 17
- transf, 2
- ucinet, 19
- zbind, 2