Package ‘multigraph’

September 25, 2018

Type Package
Title Plot and Manipulate Multigraphs
Version 0.91
Depends R (>= 3.5.0), multiplex (>= 2.9)
Imports methods
Date 2018-09-25
Author Antonio Rivero Ostoic [aut, cre]
Maintainer Antonio Rivero Ostoic <multiplex@post.com>
Description Functions to plot and manipulate multigraphs, signed and weighted multigraphs, bipartite graphs, and Cayley graphs with different layout options (devel version).

URL http://github.com/mplex/multigraph/

BugReports https://github.com/mplex/multigraph/issues/
Repository CRAN
License GPL-3
NeedsCompilation no
Date/Publication 2018-09-25 11:40:03 UTC

R topics documented:

  multigraph-package ................................................. 2
  bmgraph ............................................................ 2
  ccgraph ............................................................ 5
  conc ............................................................... 8
  fcrd ............................................................... 9
  multigraph ........................................................ 10
  stsm ............................................................... 14

Index 16
multigraph-package  
Plot and Manipulate Multigraphs

Description
Functions to create and manipulate multigraphs, bipartite graphs, and weighted multigraphs

Details

Package: multigraph
Type: Package
Version: 0.91 (devel)
Depends: multiplex (>= 2.9)
Date: 25 September 2018
License: GPL-3

This package contains functions to create, plot and manipulate multigraphs, signed and weighted multigraphs, bipartite graphs, and Cayley graphs with multiple edges.

Author(s)
J. Antonio Rivero Ostoic
Maintainer: Antonio Rivero Ostoic <multiplex@post.com>

See Also
multiplex-package, incubs, zbind, transf

bmgraph
Bipartite multigraph

Description
A function to create and manipulate bipartite multigraphs

Usage

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

- net: data frame or array representing the two-mode network (see details)
- layout: the visualization layout:
  - bip (default) bipartite graph
  - bip3 bipartite graph with three columns
  - bip3e bipartite graph with three columns for events
  - bipc “clustered” bipartite graph
  - force force-directed algorithm
  - rand random
  - circ circular
  - stress stress-majorization algorithm
  - CA correspondence analysis
  - circ2 two semi-circles
- scope: (optional) the scope of the graph (see details)
- coord: (optional) data frame with the coordinates of the vertices. If coordinates are given then the layout option is ignored
- alpha: vector (vertex, edge, bg) with the alpha color transparency
- showLbs: (optional and logical) whether or not show the vertex labels when dimnames available
- showAtts: (optional and logical) whether or not show the vertex attribute labels
- att: (optional) a vector or an array representing the vertex attributes
- lbat: (optional) the labels for the vertex attributes
- main: (optional) title of the plot
- cex.main: (optional) the size of the plot’s title
- bg: (optional) the background color of the plot
- mar: (optional) the margins of the plot
- directed: (optional and logical) whether or not the graph is directed or undirected
- weighted: (optional and logical) whether or not the graph is weighted or dichotomous
- collRecip: (optional and logical) whether or not collapse reciprocated edges in the undirected graph
- cex: (optional) the size of the vertices
- pos: (optional) the position of the vertices’ labels ($0$ means “at the center of the vertex”)
- lwd: (optional) the width of the edges. Ignored if weighted is set to TRUE
- lty: (optional) the shape of the edges
- col: (optional) alias for vcol
- ecol: (optional) the color of the edges
- vcol: (optional) the color of the vertices
- vcol0: (optional) the color of the vertices’ contour (only works for pch 21 through 25)
asp (optional) the aspect ratio of the plot
seed (optional) the random seed number for the vertices’ initial coordinates. Ignored except for force, stress and rand
maxiter (optional) the maximum number of iterations in layout algorithms. Ignored except for force, stress and rand
bwd (optional) the width of the bundle edges. Ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size). For weighted a value greater than one is possible
clu (optional) the clustering of the vertices (see details)
pch (optional) the symbol representing the vertices
fcol the font color
rot (optional) clockwise rotation of the graph in degrees
mirrorX (optional) mirror of the X axis
mirrorY (optional) mirror of the Y axis
mirrorV same as mirrorX
mirrorH same as mirrorY
hds (optional and experimental) arcs’ head scale
vedist (optional and experimental) a real number with vertex - edge distance
jitter (optional) the jitter in stress or CA
sort (optional and logical) sort the vertex labels
add add nodes to the graph’s domain
adc add nodes to the graph’s codomain
cluc a list of vectors the clustering information in both the domain and the codomain
perm (optional) a list of vectors for the permutation of network members in both the domain and codomain
ffamily the font family
fstyle the font style
fsize the font size
... Additional argument items (see e.g. par)

Details

Bipartite graphs are visualization devices for two-mode networks. Although this type of data would typically record as a data frame, it is possible to use even three dimensional arrays where each level corresponds to a particular type of tie, and thus the bipartite graphs in this case will be depicted with parallel edges. Besides, it is possible to obtain a graph of the bipartite network using the binomial approach to two-mode data, and plot it with a forced directed algorithm.

Value

A plot of the two-mode networks as bipartite graph or multigraph
Author(s)

Antonio Rivero Ostoic

See Also

multigraph, frcd, stsm, conc

Examples

```r
# Create the data: 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 this network as Bipartite graph
bmgraph(arr)

# Now with a Force Directed algorithm
bmgraph(arr, layout = "force")

# And with a Correspondence Analysis method
bmgraph(arr, layout = "CA", asp = NA)
```

Description

A function to create and manipulate bipartite Cayley colour graphs

Usage

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

Arguments

- `x`: an algebraic structure, typically a "Semigroup" object class
- `main`: (optional) title of the plot
- `seed`: (optional) the random seed number for the vertices' initial coordinates. Ignored except for force, stress and rand
- `maxiter`: (optional) the maximum number of iterations in layout algorithms. Ignored except for force, stress and rand
- `alpha`: vector (vertex, edge, bg) with the alpha color transparency
scope (optional) the scope of the graph (see details)
collRecip (optional and logical) whether or not collapse reciprocated edges in the unidirected graph
showLbs (optional and logical) whether or not show the vertex labels when dimnames available
showAtts (optional and logical) whether or not show the vertex attribute labels
cex.main (optional) the size of the plot’s title
coord (optional) data frame with the coordinates of the vertices. If coordinates are given then the layout option is ignored
clu (optional) the clustering of the vertices (see details)
cex (optional) the size of the vertices
lwd (optional) the width of the edges. Ignored if weighted is set to TRUE
pch (optional) the symbol representing the vertices
lty (optional) the shape of the edges
bwd (optional) the width of the bundle edges. Ranges from 0 (edges collapsed) to the default 1 (depending on the vertices’ size). For weighted a value greater than one is possible
att (optional) a vector or an array representing the vertex attributes
bg (optional) the background color of the plot
mar (optional) the margins of the plot
pos (optional) the position of the vertices’ labels (0 means “at the center of the vertex”)
asp (optional) the aspect ratio of the plot
ecol (optional) the color of the edges
vcol (optional) the color of the vertices
vcol0 (optional) the color of the vertices’ contour (only works for pch 21 through 25
hds (optional and experimental) arcs’ head scale
vedist (optional and experimental) a real number with vertex-edge distance
rot (optional) clockwise rotation of the graph in degrees
mirrorX (optional) mirror of the X axis
mirrorY (optional) mirror of the Y axis
col (optional) alias for vcol
labat (optional) the labels for the vertex attributes
swp (optional and logical) whether or not swap the bundle patterns
loops (optional, logical, and experimental) plot graph loops?
swp2 (optional and logical) whether or not swap reciprocals
scl (optional and experimental) numerical scalar (x and y) or vector (x, y) of the graph’s scale
mirrorD (optional) mirror reflection across diagonal Y=X
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 color) corresponds to a specific generator relation of the semigroup structure.

A plot of the semigroup or group structure.

Antonio Rivero Ostoic

See Also

semigroup, multigraph, frcd, conc

Examples

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

## plot its Cayley graph
ccgraph(S)
```
**Description**

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

**Usage**

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

**Arguments**

- `net`: an array representing the network relations
- `nr`: a scalar with the number of radii, or a vector with the clustering of the actors.
- `irot`: a scalar or vector with the "internal rotation" for each circle from closer to the center point to further away
- `inv`: (optional and logical) should the circles be with an inverted ordering?
- `flip`: (optional and logical) should the alternating circles be flipped?
- `mirror`: mirror transformation: N identity (default), X reflection through the vertical center line, Y reflection through the horizontal center line, D reflection across diagonal Y=X, L reflection across diagonal Y=-X
- `...`: Additional argument items

**Details**

In a Euclidean plane computes the coordinated system with a concentric layout with at least two radii (unless \( n = 1 \)). In case \( nr \) is not specified, 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 the coordinated system with two columns representing the abscissa and the ordinate in a two-dimensional rectangular Cartesian coordinate system.

**Author(s)**

Antonio Rivero Ostoic

**See Also**

`multigraph, bmgraph, frcd, stsm`
**Examples**

```r
## Create the data: 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**

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

**Arguments**

- `net`: an array representing the network relations.
- `seed`: (mandatory) the seed of the initial layout (see details).
- `maxiter`: (optional) the maximum number of iterations.
- `drp`: (optional) for weighted networks, drop values less than specified.
- `...`: Additional argument items.
- `scl`: (optional and experimental) numerical scalar (x and y) or vector (x, y) of the graph's scale.
- `mov`: (optional and experimental) numerical scalar (x and y) or vector (x, y) to move the graph.

**Details**

This function was meant as an internal routine for graph visualization; however, it can be used with the `coord` option both in `multigraph` and `bmgraph` where a random seed is stated by `NULL`.

**Value**

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

Author(s)
Antonio Rivero Ostoic

References

See Also
multigraph, bmgraph, stsm, conc

Examples
```r
## Create the data: two binary relations among three elements
arr <- round(replace(array(runif(18), c(3,3,2)), array(runif(18), c(3,3,2)) > .5, 1))

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

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

Description
A function to create and manipulate multigraphs and weighted multigraphs with different layout options

Usage
```r
multigraph(net, layout = c("circ", "force", "stress", "conc", "rand"), scope, directed = TRUE, main = NULL, lbs = NULL, seed = NULL, maxiter = 100, alpha = c(1, 1, 1), collRecip, showLbs, showAtts, cex.main, weighted, weights, coord, clu, cex, lwd, pch, lty, bwd, fcol, att, bg, mar, pos, asp, ecol, vcol1, vcol0, cex2, fsize2, fcol2, hds, vedist, rot, mirrorX, mirrorY, col, lbat, drp, loops, swp, swp2, signed, scl, add, mirrorD, mirrorL, opt, mirrorV, mirrorH, ffamily, fstyle, fsize, ...)
```
Arguments

net an array; usually with three dimensions of stacked matrices where the multiple relations are placed.

layout the visualization layout:
   circ circular
   force force-directed
   stress stress-majorization
   conc concentric
   rand random

scope (optional) the scope of the graph (see details)
directed (logical) whether or not the graph is directed or undirected
main (optional) title of the plot
lbs (optional) the vertex labels
seed (optional) the random seed number for the vertices' initial coordinates. Ignored for circ and conc
maxiter (optional) the maximum number of iterations in layout algorithms. Only for force, stress, and rand
alpha vector (vertex, edge, bg) with the alpha color transparency
collRecip (optional and logical) whether or not collapse reciprocated edges in the undirected graph
showLbs (optional and logical) whether or not show the vertex labels
showAtts (optional and logical) whether or not show the vertex attribute labels
cex.main (optional) the size of the plot's title
weighted (optional and logical) whether or not the graph is weighted or dichotomous
weights (optional and logical) whether or not print the weights of the bonds
coord (optional) data frame with the coordinates of the vertices. If coordinates are given then the layout option is ignored
clu (optional) the clustering of the vertices (see details)
cex (optional) the size of the vertices
lwd (optional) the width of the edges. Ignored if weighted is set to TRUE
pch (optional) the symbol representing the vertices
lty (optional) the shape of the edges
bwd (optional) the width of the bundle edges. Ranges from 0 (edges collapsed) to the default 1 (depending on the vertices' size). For weighted a value greater than one is possible
fcol the font color
att (optional) a vector or an array representing the vertex attributes
bg (optional) the background color of the plot
mar (optional) the margins of the plot
pos  (optional) the position of the vertices’ labels (0 means “in the middle of the vertex”)
asp  (optional) the aspect ratio of the plot
ecol  (optional) the color of the edges
tcol  (optional) the color of the vertices
tcol0  (optional) the color of the vertices’ contour (only works for pch 21 through 25
text2  the size of the background for the weights
fsize2  the font size of the weights
fcolor2  the font color of the weights
hs  (optional and experimental) arcs’ head scale
vedist  (optional and experimental) a real number with vertex - edge distance
rot  (optional) clockwise rotation of the graph in degrees
mirrorX  (optional) mirror of the X axis
mirrorY  (optional) mirror of the Y axis
tcol  (optional) alias for tcol
lbat  (optional) the labels for the vertices’ attributes
drp  (optional) for weighted networks, drop values less than specified
loops  (optional, logical, and experimental) plot graph loops?
swp  (optional and logical) whether or not swap the bundle patterns
swp2  (optional and logical) whether or not swap reciprocals
signed  (optional and logical) whether or not the graph is a signed structure
scl  (optional and experimental) numerical scalar (x and y) or vector (x, y) of the graph’s scale
add  (optional) nodes to add to the graph
mirrorD  (optional) mirror reflection across diagonal Y=X
mirrorL  (optional) mirror reflection across diagonal Y=-X
opt  (optional and experimental) optimize the plot margins
mirrorV  same as mirrorX
mirrorH  same as mirrorY
family  the font family
fstyle  the font style
fsize  the font size
... Additional argument items (see e.g. par)
Details

Multigraph are graphs having parallel edges depicting different types of relations in a network. By default, a circular layout is applied where each type of tie has a distinctive shape and gray color scale. For a better visualization, undirected multigraphs automatically collapse the reciprocal ties, and there is an argument to prevent this from happening. It is possible to combine the symbols and color of vertices by assigning a class to each network member in the clustering option. Vertices can also have different sizes by specifying the argument with a vector with a length size similar to the network order.

Since this function can handle a large number of arguments, these can be stored as a list object that is passed through the scope option. In this case a vector made of lists and scalars or combinations of these are accepted.

Value

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

Author(s)

Antonio Rivero Ostoic

See Also

bmgraph, ccgraph, frcd, stsm, conc

Examples

```r
## Create the data: 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 the multigraph of this network
multigraph(arr)

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

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

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

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

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

Usage

\texttt{stsm(net, seed = seed, maxiter = 40, drp, jitter, method, ...)}

Arguments

- \texttt{net}: an array representing the network relations.
- \texttt{seed}: (mandatory) the seed of the initial layout (see details).
- \texttt{maxiter}: (optional) the maximum number of iterations.
- \texttt{drp}: (optional) for weighted networks, drop values less than specified.
- \texttt{jitter}: (optional) jitter in the layout.
- \texttt{method}: (optional) initial distance method (default binary).
- ...: Additional argument items.

Details

This function was meant as an internal routine for graph visualization; however, it can be used with the \texttt{coord} option both in \texttt{multigraph} and \texttt{bmgraph} where a random seed is stated by \texttt{NULL}.

Value

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

Author(s)

Antonio Rivero Ostoic

References


See Also

\texttt{multigraph, bmgraph, frcd, conc}
Examples

```r
## Create the data: 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 stress majorization layout with random start
coord <- stsm(arr, seed = NULL)

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

*Topic IO
multigraph-package, 2

*Topic data
multigraph-package, 2

*Topic file
multigraph-package, 2

*Topic graphics
bmgraph, 2
ccgraph, 5
cconc, 8
frcd, 9
multigraph, 10
stsm, 14

*Topic graphs
multigraph-package, 2

*Topic manip
conc, 8
frcd, 9
multigraph-package, 2
stsm, 14

*Topic math
ccgraph, 5
bmgraph, 2, 8, 10, 13, 14
ccgraph, 5, 13
conc, 5, 7, 8, 10, 13, 14
frcd, 5, 7, 8, 9, 13, 14
incubs, 2
multigraph, 5, 7, 8, 10, 10, 14
multigraph-package, 2
par, 4, 7, 12
semigroup, 7
stsm, 5, 8, 10, 13, 14
transf, 2
zbind, 2