

Package ‘ggmGraphEditor’

October 30, 2026

Type Package

Title Interactive Graph Layout Editor for Mixed Graphs

Version 0.2.0

Depends R (>= 3.6.0)

Imports methods, visNetwork, shiny, igraph

Suggests ggm (>= 2.5.5), graph, testthat (>= 3.0.0)

Description Provides an interactive interface to plot and layout mixed graphs based on the adjacency matrix defined in Sadeghi and Marchetti (2012) <[doi:10.32614/RJ-2012-015](https://doi.org/10.32614/RJ-2012-015)>. Uses 'shiny' and 'visNetwork' to allow dynamic node repositioning and coordinate extraction.

License GPL-2

Encoding UTF-8

URL <https://github.com/StaThin/ggmGraphEditor>

BugReports <https://github.com/StaThin/ggmGraphEditor/issues>

NeedsCompilation no

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Author Giovanni M. Marchetti [aut, cre] (ORCID: <<https://orcid.org/0000-0003-4413-9349>>),
Kayvan Sadeghi [aut] (ORCID: <<https://orcid.org/0000-0001-7314-744X>>)

Maintainer Giovanni M. Marchetti <giovanni.marchetti@unifi.it>

Repository CRAN

Date/Publication 2026-10-30 15:10:02 UTC

Contents

graphLayout	2
grMAT2	4
Index	6

Description

An interactive replacement for `ggm::plotGraph` using shiny and visNetwork. Allows dynamic node repositioning, handles multiple edge types, and outputs node coordinates.

Usage

```
graphLayout(
  ggm_matrix,
  dashed = TRUE,
  bidirected_roundness = 0.25,
  Viewer = if (Sys.getenv("RSTUDIO") == "1") "Rstudio" else "browser",
  coords = NULL
)
```

Arguments

<code>ggm_matrix</code>	<ul style="list-style-type: none"> • an adjacency matrix for mixed graphs, • an igraph object, or • a descriptive character vector.
<code>dashed</code>	Logical. If TRUE (default), bidirected edges <-> are drawn as dashed lines.
<code>bidirected_roundness</code>	Numeric. Curvature of bidirected edges when multiple edges are present. Default is 0.25.
<code>Viewer</code>	character string specifying where to display the Shiny application. Valid options are "Rstudio" (to open it in a native internal window), or "browser" (to open it in the default web browser). If not specified, it automatically detects the environment, defaulting to "Rstudio" inside RStudio and "browser" elsewhere.
<code>coords</code>	Optional initial node positions. Either a numeric matrix with one row per node (in the order of the adjacency matrix) and two columns (x, y), a function such as <code>igraph::layout_nicely()</code> that takes an igraph graph and returns such a matrix, or one of the strings "TB", "BT", "LR", "RL". The graph passed to a function has the same nodes and adjacencies as the mixed graph (edge types are ignored). The strings request a layered layout (<code>igraph::layout_with_sugiyama()</code>) computed on the directed edges only, in the spirit of the <code>rankdir</code> option of GraphViz: arrows point from top to bottom ("TB"), bottom to top ("BT"), left to right ("LR") or right to left ("RL"). Undirected and bi-directed edges do not contribute to the layering, so these layouts are most useful for graphs with many arrows. The positions are enlarged internally to fit the display, but the returned coordinates are in the same units as coords (for the strings, one unit between consecutive layers). If NULL (default), visNetwork chooses the initial positions and the returned coordinates are in display (pixel) units.

Value

A matrix of X and Y coordinates for each node, returned after terminating the Shiny application.

Note

The author acknowledges the help of Google Gemini AI (2026) and Posit Assistant (2026).

Author(s)

Giovanni M. Marchetti

Examples

```
if (interactive()) {
  amat <- matrix(c(0,1,0, 0,0,1, 0,0,0), 3, 3,
    dimnames = list(c("A","B","C"), c("A","B","C")))
  coords <- graphLayout(amat)
  print(coords)

  # Generating the adjacency matrix from a vector and plotting
  exvec <- c(
    "b", 1, 2, "b", 1, 14, "a", 9, 8, "l", 9, 11,
    "a", 10, 8, "a", 11, 2, "a", 11, 9, "a", 11, 10,
    "a", 12, 1, "b", 12, 14, "a", 13, 10, "a", 13, 12
  )
  graphLayout(exvec)

  # Plotting from a direct adjacency matrix
  amat2 <- matrix(c(0, 11, 0, 0, 10, 0, 100, 0, 0, 100, 0, 1, 0, 0, 1, 0), 4, 4)
  graphLayout(amat2)

  # A graph with double and triple edges
  G <- structure(c(
    0, 101, 0, 0, 100, 0, 100, 100, 0, 100, 0, 100, 0, 111, 100, 0),
    dim = c(4L, 4L),
    dimnames = list(c("X", "Z", "Y", "W"), c("X", "Z", "Y", "W")))
  graphLayout(G)

  # A graph with 4 edges between two nodes
  G4 <- matrix(0, 2, 2)
  G4[1, 2] <- 111
  G4[2, 1] <- 111
  graphLayout(G4)

  # Starting from an igraph layout
  graphLayout(G, coords = igraph::layout_nicely)

  # Layered layouts (as rankdir in GraphViz), best for graphs with
  # many arrows
  dag <- c("a", "A", "C", "a", "B", "C", "a", "C", "D",
    "a", "C", "E", "a", "D", "F", "a", "E", "F", "a", "B", "E")
  graphLayout(dag, coords = "TB")
}
```

```

graphLayout(dag, coords = "LR")
xy <- matrix(c(0, 1, 1, 0, 0, 0, 1, 1), 4, 2)
graphLayout(G, coords = xy)
}

```

grMAT2

Graph to adjacency matrix

Description

grMAT2 converts graph objects to a mixed adjacency matrix.

Usage

```
grMAT2(agr)
```

Arguments

agr A graph object. This can be:

- a graphNEL object,
- an `igraph::igraph()` object, or
- a character vector of length $3e$, where e is the number of edges. If it is a vector, it must be a sequence of triples (type, node1label, node2label). The type of edge can be:
 - "a" (arrow from node1 to node2),
 - "b" (bi-directed arc),
 - "l" (undirected line), or
 - "*" (for an isolated node with node1 == node2).

Details

Support for graphNEL objects requires the graph package from Bioconductor, which is a suggested dependency. If the package is missing, passing a graphNEL object will trigger an informative error.

Value

A matrix consisting of 4 different integers representing the ij -elements:

- 0 for a missing edge between i and j ,
- 1 for an arrow from i to j ,
- 10 for a full line between i and j , and
- 100 for a bi-directed arrow between i and j . These numbers are added when multiple edges of different types are present. The matrix is symmetric with respect to full lines and bi-directed arrows.

Author(s)

Kayvan Sadeghi, Giovanni Marchetti

Examples

```
## Generating the adjacency matrix from a vector
exvec <- c(
  "b", 1, 2, "b", 1, 14, "a", 9, 8, "l", 9, 11, "a", 10, 8,
  "a", 11, 2, "a", 11, 10, "a", 12, 1, "b", 12, 14, "a", 13, 10, "a", 13, 12)
grMAT2(exvec)

## Example with graphNEL (requires the 'graph' package from Bioconductor)
if (requireNamespace("graph", quietly = TRUE)) {
  V <- c("a", "b", "c")
  g <- graph::graphNEL(nodes = V, edgemode = "undirected")
  grMAT2(g)
}

g <- c("a", 3, 1, "l", 1, 2, "*", 4, 4)
grMAT2(g)
```

Index

- * **adjacency**
 - grMAT2, [4](#)
- * **graphs**
 - grMAT2, [4](#)
- * **matrix**
 - grMAT2, [4](#)
- * **mixed-graph**
 - grMAT2, [4](#)
- * **vector**
 - grMAT2, [4](#)

ggm::plotGraph, [2](#)
graphLayout, [2](#)
grMAT2, [4](#)

igraph::igraph(), [4](#)
igraph::layout_nicely(), [2](#)
igraph::layout_with_sugiyama(), [2](#)