

# Package ‘gridGraphviz’

October 13, 2022

**Version** 0.3-1

**Depends** R (>= 2.15.0), grid, graph, Rgraphviz

**Imports** methods

**Suggests** gridSVG

**SystemRequirements** graphviz

**Title** Drawing Graphs with 'grid'

**Description** Functions for drawing node-and-edge graphs that have been laid out by 'graphviz'. This provides an alternative rendering to that provided by the 'Rgraphviz' package, with two main advantages: the rendering provided by 'gridGraphviz' should be more similar to what 'graphviz' itself would draw; and rendering with 'grid' allows for post-hoc customisations using the named viewports and grobs that 'gridGraphviz' produces.

**License** GPL (>= 2)

**URL** <https://r-forge.r-project.org/projects/gridgraph/>

**NeedsCompilation** no

**Author** Paul Murrell [cre, aut],  
Ashley Noel Hinton [aut]

**Maintainer** Paul Murrell <p.murrell@auckland.ac.nz>

**Repository** CRAN

**Date/Publication** 2022-04-11 00:02:29 UTC

## R topics documented:

agopenTrue	2
graphWidth	3
grid.graph	4

<b>Index</b>	<b>6</b>
--------------	----------

---

agopenTrue                      *A function to obtain a layout Ragraph object*

---

### Description

This function is a wrapper for the **Rgraphviz** function `agopen` and will produce a layout Ragraph object with layout more true to the result produced by `graphviz`.

### Usage

```
agopenTrue(graph, name, nodes, edges, kind = NULL, layoutType = "dot",
           attrs = list(), nodeAttrs = list(), edgeAttrs = list(),
           subGList = list(), edgeMode = edgemode(graph),
           recipEdges = c("combined", "distinct"))
```

### Arguments

<code>graph</code>	An object of class <code>graphNEL</code>
<code>nodes</code>	A list of <code>pNode</code> objects
<code>edges</code>	A list of <code>pEdge</code> objects
<code>name</code>	The name of the graph
<code>kind</code>	The type of graph
<code>layoutType</code>	Defines the layout engine. Defaults to <code>dot</code> , and see <code>graphvizCapabilities()\$layoutTypes</code> for possible values.
<code>attrs</code>	A list of <code>graphviz</code> attributes
<code>nodeAttrs</code>	A list of specific node attributes
<code>edgeAttrs</code>	A list of specific edge attributes
<code>subGList</code>	A list describing subgraphs for the <code>graph</code> parameter
<code>edgeMode</code>	Whether the graph is directed or undirected
<code>recipEdges</code>	How to handle reciprocated edges, defaults to <code>combined</code>

### Details

As of `Rgraphviz` version 2.2.1 (2013-01-31) `agopen`:

- Produces graphs of the same size as the current device, or at a default size of 7x7 inches.
- Forces nodes to fixed default height and width.
- Does not pass through edge weight information.

This function returns an Ragraph object with `graph` and node sizes set by `graphviz` or by the user. It also ensures edge weight information is passed through.

### Value

An object of class Ragraph

**Author(s)**

Ashley Noel Hinton

**References**

graphviz

**See Also**

[agopen](#)

**Examples**

```
gnel <- new("graphNEL",
           nodes=letters[1:3],
           edgeL=list(a=list(edges=c("b", "c")),
                    b=list(),
                    c=list()),
           edgemode="directed")
rag <- agopenTrue(gnel, "")
grid.graph(rag)
```

---

graphWidth

*Get width/height of an Rgraph object*

---

**Description**

These functions will return the width or height of an **Rgraphviz** Rgraph object in inches.

**Usage**

```
graphWidth(graph)
graphHeight(graph)
```

**Arguments**

graph            a layout Rgraph object

**Value**

A numeric vector of length 1.

**Author(s)**

Ashley Noel Hinton

**References**

graphviz

**See Also**[Ragraph](#)**Examples**

```
gnel <- new("graphNEL",
           nodes=letters[1:3],
           edgel=list(a=list(edges=c("b", "c")),
                    b=list(),
                    c=list()),
           edgemode="directed")
rag <- agopenTrue(gnel, "")
graphWidth(rag)
graphHeight(rag)
```

---

`grid.graph`*Draw a Node-and-Edge Graph*

---

**Description**

Take a description of the layout of a graph, as produced by the **Rgraphviz** package, and draw it.

**Usage**

```
grid.graph(rag, newpage = FALSE, nodesOnTop = TRUE)
```

**Arguments**

<code>rag</code>	An Ragraph object (must be laid out).
<code>newpage</code>	Logical value indicating whether to draw the graph on a new page.
<code>nodesOnTop</code>	Logical value indicating whether nodes should be drawn after edges (or vice versa).

**Author(s)**

Paul Murrell

**References**[graphviz](#)

**Examples**

```
gnel <- new("graphNEL",
           nodes=letters[1:3],
           edgeL=list(a=list(edges=c("b", "c")),
                     b=list(),
                     c=list()),
           edgemode="directed")
rag <- agopen(gnel, "")
grid.graph(rag)
```

# Index

## \* **dplot**

- agopenTrue, 2
- graphWidth, 3
- grid.graph, 4

agopen, 3

agopenTrue, 2

graphHeight (graphWidth), 3

graphWidth, 3

grid.graph, 4

Ragraph, 4