

# Package ‘hyperdraw’

October 24, 2025

**Version** 1.61.0  
**Depends** R (>= 2.9.0)  
**Imports** methods, grid, graph, hypergraph, Rgraphviz, stats4  
**SystemRequirements** graphviz  
**Title** Visualizing Hypergraphs  
**Author** Paul Murrell  
**Maintainer** Paul Murrell <p.murrell@auckland.ac.nz>  
**Description** Functions for visualizing hypergraphs.  
**License** GPL (>= 2)  
**Collate** AllClasses.R affine.R draw.R graphBPH.R grid.R hypergraph.R  
legacy.R node.R RagraphBPH.R  
**biocViews** Visualization, GraphAndNetwork  
**git\_url** <https://git.bioconductor.org/packages/hyperdraw>  
**git\_branch** devel  
**git\_last\_commit** e49afb2  
**git\_last\_commit\_date** 2025-04-15  
**Repository** Bioconductor 3.23  
**Date/Publication** 2025-10-24

## Contents

|                            |          |
|----------------------------|----------|
| graphBPH . . . . .         | 2        |
| graphBPH-class . . . . .   | 3        |
| graphLayout . . . . .      | 4        |
| RagraphBPH-class . . . . . | 6        |
| <b>Index</b>               | <b>8</b> |

graphBPH

*Constructor for graphBPH objects***Description**

A convenience constructor for [graphBPH-class](#) objects. This is a generic function.

**Usage**

```
graphBPH(graph, edgeNodePattern, ...)
```

**Arguments**

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| graph           | Some form of graph that is to be converted into a graphBPH object.            |
| edgeNodePattern | A regular expression used to distinguish between normal nodes and edge nodes. |
| ...             | Potential arguments to other methods.                                         |

**Value**

An object of class [graphBPH-class](#)

**Methods**

**graphBPH** signature(graph = "graphNEL", edgeNodePattern = "character"): create a graphBPH object from a (directed) graphNEL object.

**graphBPH** signature(graph = "Hypergraph", edgeNodePattern = "missing"): create a graphBPH object from a Hypergraph object (where all Hyperedges are DirectedHyperedges).

**Author(s)**

Paul Murrell

**References**

Falcon, S. and Gentleman, R. **hypergraph**: A package providing hypergraph data structures.

Gentleman, R. and Whalen, E. and Huber, W. and Falcon, S. **graph**: A package to handle graph data structures.

**See Also**

[graphBPH-class](#)

graphBPH-class

Class "graphBPH"

**Description**

A bipartite representation of a hypergraph. The purpose of this class is to support visualization of the hypergraph; it is not intended for analysis or manipulation of the hypergraph.

**Objects from the Class**

Objects can be created by calls of the form `new("graphBPH", graph, edgeNodePattern, ...)`. There is also a convenience function `graphBPH()`.

A `graphBPH` object consists of a `graphNEL` object, which must obey some strict rules:

- nodes in the graph are divided into two sets: normal nodes and edge-nodes,
- all edges in the graph must connect a normal node to an edge node,
- the graph must be a directed graph.

The `edgeNodePattern` is a regular expression that is used to define the set of edge-nodes.

**Slots**

**graph:** Object of class `graphNEL`. This graph must obey the constraints described above.

**edgeNodePattern:** Object of class `character`. The regular expression used to define edge-nodes.

**nodes:** Object of class `character`. Records which nodes in the graph are normal nodes.

**edgeNodes:** Object of class `character`. Records which nodes in the graph are edge-nodes.

**edgeNodeIO:** Object of class `list`. Records information about which edges enter and exit each edge-node.

**Methods**

**plot** signature(`x` = "graphBPH", `y` = "ANY"): draw a representation of the hypergraph where edges between normal nodes in the graph pass through an intermediate edge-node in a nice smooth curve.

**graphLayout** signature(`graph` = "graphBPH", `layoutType` = "missing"): convert the `graphBPH` object to a `RagraphBPH` object (using a default layout method).

**graphLayout** signature(`graph` = "graphBPH", `layoutType` = "character"): convert the `graphBPH` object to a `RagraphBPH` object (using the specified layout method).

**Author(s)**

Paul Murrell

## References

Gansner, E.R. and North, S.C. (1999) An open graph visualization system and its applications to software engineering, *Software - Practice and Experience*, 30:1203–1233.

Gentleman, R. and Whalen, E. and Huber, W. and Falcon, S. **graph**: A package to handle graph data structures.

Gentry, J. and Long, L. and Gentleman, R. and Falcon, S. and Hahne, F. and Sarkar, D. and Hansen, K. **Rgraphviz**: Provides plotting capabilities for R graph objects.

## See Also

[agopen](#), [graphLayout](#) and [graphNEL](#) [RagraphBPH](#)

## Examples

```
nodes <- c(LETTERS[1:5], paste("R", 1:3, sep=""))
testgnel <- new("graphNEL",
               nodes=nodes,
               edgeL=list(
                 A=list(edges=c("R1", "R2")),
                 B=list(edges="R2"),
                 C=list(),
                 D=list(edges="R3"),
                 E=list(),
                 R1=list(edges="B"),
                 R2=list(edges=c("C", "D")),
                 R3=list(edges="E")),
               edgemode="directed")
testbph <- graphBPH(testgnel, "^R")
plot(testbph)

# A Hypergraph equivalent
require(hypergraph)
dh1 <- DirectedHyperedge("A", "B", "R1")
dh2 <- DirectedHyperedge(c("A", "B"), c("C", "D"), "R2")
dh3 <- DirectedHyperedge("D", "E", "R3")
hg <- Hypergraph(LETTERS[1:5], list(dh1, dh2, dh3))
plot(graphBPH(hg))
```

---

graphLayout

Layout a graph.

---

## Description

This function is designed to layout a graph using the **Rgraphviz** package. The **hyperdraw** package makes this a generic function with a method for graphBPH objects. The function of the same name in the **Rgraphviz** package is used as a method for Ragraph objects.

**Usage**

```
graphLayout(graph, layoutType, ...)
```

**Arguments**

|            |                                                                       |
|------------|-----------------------------------------------------------------------|
| graph      | An graphBPH object, which is to be laid out.                          |
| layoutType | The layout method (e.g., dot or neato).                               |
| ...        | These arguments will be passed to the <code>agopen()</code> function. |

**Value**

An RagraphBPH object.

**Author(s)**

Paul Murrell

**References**

Gansner, E.R. and North, S.C. (1999) An open graph visualization system and its applications to software engineering, *Software - Practice and Experience*, 30:1203–1233.

Gentry, J. and Long, L. and Gentleman, R. and Falcon, S. and Hahne, F. and Sarkar, D. and Hansen, K. **Rgraphviz**: Provides plotting capabilities for R graph objects.

**See Also**

[agopen](#) and [GraphvizLayouts](#)

**Examples**

```
nodes <- c(LETTERS[1:5], paste("R", 1:3, sep=""))
testgnel <- new("graphNEL",
  nodes=nodes,
  edgeL=list(
    A=list(edges=c("R1", "R2")),
    B=list(edges="R2"),
    C=list(),
    D=list(edges="R3"),
    E=list(),
    R1=list(edges="B"),
    R2=list(edges=c("C", "D")),
    R3=list(edges="E")),
  edgemode="directed")
testbph <- new("graphBPH", testgnel, "^R")
testrabph <- graphLayout(testbph)
```

---

|                  |                    |
|------------------|--------------------|
| RagraphBPH-class | Class "RagraphBPH" |
|------------------|--------------------|

---

### Description

The purpose of this class is to represent a laid out version of a graphBPH object. The laying out is performed by the **Rgraphviz** package. This is an intermediate step in the process of drawing a graphBPH object.

### Objects from the Class

Objects of this class should be created via the `graphLayout()` function.

### Slots

**graph:** Object of class `Ragraph`. The laid out graph.  
**allNodes:** Object of class `character`. The names of all nodes in the graph.  
**nodes:** Object of class `character`. Records normal nodes in the graph.  
**edgeNodes:** Object of class `character`. Records edge-nodes in the graph.  
**edgeNodeIO:** Object of class `list`. Records which edges enter and exit each edge-node.

### Methods

**plot** signature(`x` = "RagraphBPH", `y` = "ANY"): draw a representation of the hypergraph where edges between normal nodes in the graph pass through an intermediate edge-node in a nice smooth curve.

**edgeDataDefaults<-** signature(`self` = "RagraphBPH", `attr` = "character", `value` = "ANY"): set the default drawing attributes for all edges.

**edgeData<-** signature(`self` = "RagraphBPH", `from` = "character", `to` = "character", `attr` = "character", `value` = "ANY"): set a specific drawing attribute for one or more edges.

**nodeDataDefaults<-** signature(`self` = "RagraphBPH", `attr` = "character", `value` = "ANY"): set the default drawing attributes for all nodes.

**nodeData<-** signature(`self` = "RagraphBPH", `n` = "character", `attr` = "character", `value` = "ANY"): set a specific attribute for one or more nodes.

**graphDataDefaults<-** signature(`self` = "RagraphBPH", `attr` = "character", `value` = "ANY"): set the default drawing attributes for the graph.

**graphData<-** signature(`self` = "RagraphBPH", `n` = "character", `attr` = "character", `value` = "ANY"): set a specific attribute for the graph.

### Author(s)

Paul Murrell

**See Also**

[graphLayout](#), [graphBPH](#), and [Ragraph](#)

**Examples**

```

nodes <- c(LETTERS[1:5], paste("R", 1:3, sep=""))
testgnel <- new("graphNEL",
               nodes=nodes,
               edgeL=list(
                 A=list(edges=c("R1", "R2")),
                 B=list(edges="R2"),
                 C=list(),
                 D=list(edges="R3"),
                 E=list(),
                 R1=list(edges="B"),
                 R2=list(edges=c("C", "D")),
                 R3=list(edges="E")),
               edgemode="directed")
testbph <- graphBPH(testgnel, "^R")
testrabph <- graphLayout(testbph)
edgeDataDefaults(testrabph, "lwd") <- 1
edgeData(testrabph, c("A", "R1"), c("R1", "B"), "lwd") <- c("3", 5)
edgeDataDefaults(testrabph, "color") <- "black"
edgeData(testrabph, c("A", "R1"), c("R1", "B"), "color") <- "red"
nodeDataDefaults(testrabph, "margin") <- 'unit(2, "mm")'
nodeDataDefaults(testrabph, "shape") <- "circle"
plot(testrabph)
graphDataDefaults(testrabph, "arrowLoc") <- "middle"
graphData(testrabph, "arrowLoc") <- "end"
plot(testrabph)
graphData(testrabph, "arrowLoc") <- "none"
plot(testrabph)

```

# Index

- \* **classes**
  - graphBPH, [2](#)
  - graphBPH-class, [3](#)
  - RagraphBPH-class, [6](#)
- \* **dplot**
  - graphLayout, [4](#)
- agopen, [4](#), [5](#)
- edgeData<- , RagraphBPH, character, character, character, ANY-method  
(RagraphBPH-class), [6](#)
- edgeDataDefaults<- , RagraphBPH, character, ANY-method  
(RagraphBPH-class), [6](#)
- graphBPH, [2](#), [7](#)
- graphBPH, graphNEL, character-method  
(graphBPH), [2](#)
- graphBPH, Hypergraph, missing-method  
(graphBPH), [2](#)
- graphBPH-class, [3](#)
- graphData<- , RagraphBPH, character, ANY-method  
(RagraphBPH-class), [6](#)
- graphDataDefaults<- , RagraphBPH, character, ANY-method  
(RagraphBPH-class), [6](#)
- graphLayout, [4](#), [4](#), [7](#)
- graphLayout, graphBPH, character-method  
(graphBPH-class), [3](#)
- graphLayout, graphBPH, missing-method  
(graphBPH-class), [3](#)
- graphNEL, [4](#)
- GraphvizLayouts, [5](#)
- nodeData<- , RagraphBPH, character, character, ANY-method  
(RagraphBPH-class), [6](#)
- nodeDataDefaults<- , RagraphBPH, character, ANY-method  
(RagraphBPH-class), [6](#)
- plot, graphBPH, ANY-method  
(graphBPH-class), [3](#)
- plot, RagraphBPH, ANY-method  
(RagraphBPH-class), [6](#)
- Ragraph, [7](#)
- RagraphBPH, [4](#)
- RagraphBPH (RagraphBPH-class), [6](#)
- RagraphBPH-class, [6](#)