Package ‘RGraphSpace’

February 28, 2026

Type Package

Version 1.1.1

Title A Lightweight Interface Between 'igraph' and 'ggplot2' Graphics

Description Interface to integrate 'igraph' and 'ggplot2' graphics in a normalized coordinate system. 'RGraphSpace' implements new geometric objects using 'ggplot2' prototypes, customized for side-by-side visualization of multiple graphs. By scaling shapes and graph elements, 'RGraphSpace' helps to create layered visualizations that stay coherent across multiple graphs.

Depends R(>= 4.4), methods, ggplot2

Imports grDevices, scales, grid, igraph, rlang, lifecycle, patchwork, cowplot, gtable, ggrastr, circlize

Suggests knitr, rmarkdown, testthat

Enhances RedeR

License Artistic-2.0

VignetteBuilder knitr

URL https://github.com/sysbiolab/RGraphSpace

BugReports https://github.com/sysbiolab/RGraphSpace/issues

Collate gspaceChecks.R gspaceValidation.R gspaceConstructor.R gspaceSupplements.R gspaceGeom.R gspaceClasses.R gspaceGenerics.R gspaceMethods.R gspaceMisc.R

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Victor Apolonio [ctb],
Vinicius Chagas [ctb],
Mauro Castro [aut, cre] (ORCID:
<https://orcid.org/0000-0003-4942-8131>)

Maintainer Mauro Castro <mauro.a.castro@gmail.com>

Repository CRAN

Date/Publication 2026-02-28 17:10:03 UTC

Contents

`GeomEdgeSpace` *GeomEdgeSpace: a ggplot2 prototype for GraphSpace-class methods*

Description

‘GeomEdgeSpace‘ is the underlying ggproto object used by `geom_edgespace` to draw edge elements in a 2D graph layout, whose properties are controlled by aesthetics such as ‘colour‘ and ‘linewidth‘.

This ‘geom‘ is designed for graph/network diagrams, where edge attributes are often already in their final form (e.g., hex colors). When used with identity scales, the appearance of edges can be fully controlled by the input data.

Usage

```
GeomEdgeSpace
```

Format

An object of class `GeomEdgeSpace` (inherits from `Geom`, `ggproto`, `gg`) of length 5.

Aesthetics

‘GeomEdgeSpace‘ understands the following aesthetics:

- ‘x‘ Starting horizontal position in ‘npc‘ `unit` (required).
- ‘y‘ Starting vertical position in ‘npc‘ `unit` (required).
- ‘xend‘ Ending horizontal position in ‘npc‘ `unit` (required).
- ‘yend‘ Ending vertical position in ‘npc‘ `unit` (required).
- ‘colour‘ Edge colour.
- ‘linewidth‘ Edge width, using ’lwd’ standard graphics unit (see `gpar`).

- ‘linetype‘ Line type, using ’lty’ standard graphics unit (see `gpar`).
- ‘alpha‘ Transparency applied to the edge colour.

These aesthetics follow ggplot2’s conventions for segment-like ‘geoms‘.

See Also

`geom_edgespace`, `geom_segment`

`GeomNodeSpace` *GeomNodeSpace: a ggplot2 prototype for GraphSpace-class methods*

Description

‘GeomNodeSpace‘ is the underlying ggproto object used by `geom_nodespace` to draw node elements in a 2D graph layout. It produces point-like glyphs (typically circles) whose properties are controlled by aesthetics such as ‘x‘, ‘y‘, ‘size‘ and ‘colour‘.

This geom is designed for graph/network diagrams, where node attributes are often already in their final form (e.g., hex colors). When used with identity scales, the appearance of nodes can be fully controlled by the input data.

Usage

```
GeomNodeSpace
```

Format

An object of class `GeomNodeSpace` (inherits from `Geom`, `ggproto`, `gg`) of length 5.

Aesthetics

‘GeomNodeSpace‘ understands the following aesthetics:

- ‘x‘ Horizontal position in ‘npc‘ `unit` (required).
- ‘y‘ Vertical position in ‘npc‘ `unit` (required).
- ‘size‘ Node size in ‘npc‘ `unit` (required).
- ‘fill‘ Node fill colour.
- ‘colour‘ Node line colour.
- ‘shape‘ Integer code between 0 and 25 (see `points`).
- ‘linewidth‘ Line width, using ’lwd’ standard graphics unit (see `gpar`).
- ‘alpha‘ Transparency applied to fill and line colour.

These aesthetics follow ggplot2’s conventions for point-like ‘geoms‘.

Drawing

The ‘draw_panel()‘ method transforms data into panel coordinates and constructs a ‘grid::pointsGrob()‘ for rendering. Node size is expected to be normalized to the range ‘[0, 1]‘, so it can be interpreted in graph space using ‘npc‘ unit.

See Also

`geom_nodespace`, `geom_point`

`geom_edgespace` *Draw edge elements in a 2D graph layout*

Description

Constructor for `GeomEdgeSpace` ggproto objects, a variant of `geom_segment` supporting edge attributes and bidirectional arrows.

This ‘geom‘ is designed to create edge-level aesthetics such as ‘colour‘ and ‘linewidth‘, or any custom aesthetics defined in ‘GeomEdgeSpace‘.

Usage

```
geom_edgespace(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  show.legend = NA,
  inherit.aes = FALSE,
  lineend = "butt",
  linejoin = "mitre",
  arrow_type = 0,
  arrow_size = 1,
  offset_start = 0,
  offset_end = 0
)
```

Arguments

`mapping` Set of aesthetic mappings created by [ggplot2::aes()]. These mappings apply to the layer and may override global aesthetics.

`data` Either a GraphSpace object or a data frame containing edge attributes. If ‘NULL‘, the layer will use the default plot data.

`stat` The statistical transformation to use on the data. Defaults to ‘"identity"‘.

`position` Position adjustment, either as a string or the result of a call to a position adjustment function.

`...` Additional parameters passed to the underlying drawing function in ‘GeomEdgeSpace‘.

`show.legend` Logical or a named logical vector indicating whether this layer should be included in legends.

`inherit.aes` Logical. If ‘FALSE‘ (default), the layer will use aesthetics defined in ‘mapping‘. Set to ‘TRUE‘ if this geom should inherit aesthetics from the plot.

`lineend` Line end style (round, butt, square).

`linejoin` Line join style (round, mitre, bevel).

`arrow_type` Arrowhead types (see ’drawing’ section).

`arrow_size` Numeric scaling factor controlling arrowhead geometry.

`offset_start` Offset arrows as a fraction of graph space at start positions.

`offset_end` Offset arrows as a fraction of graph space at end positions.

Details

Arrowhead size is a numeric scaling factor controlling arrowhead geometry. The value is interpreted in the same numeric space as line width (‘lwd‘), ensuring consistent scaling between edge strokes and arrowheads.

Arrowhead types can be specified using either integer or character codes:
- No arrow: `0 or "---"`
- Forward arrow: `1 or "-->"`
- Backward arrow: `2 or "<--"`
- Bidirectional arrow: `3 or "<->"`
- Forward arrow with backward bar: `4 or "|->"`
- Forward bar: `-1 or "--|"`
- Backward bar: `-2 or "|--"`
- Bidirectional bar: `-3 or "|-|"`
- Backward arrow with forward bar: `-4 or "<-|"`

Value

A ggplot2 layer that renders edge segments defined by `GeomEdgeSpace`.

Aesthetics

‘geom_edgespace‘ understands the following aesthetics:

- ‘x‘ Starting horizontal position in ‘npc‘ `unit` (required).
- ‘y‘ Starting vertical position in ‘npc‘ `unit` (required).
- ‘xend‘ Ending horizontal position in ‘npc‘ `unit` (required).
- ‘yend‘ Ending vertical position in ‘npc‘ `unit` (required).
- ‘colour‘ Edge colour.
- ‘linewidth‘ Edge width, using ’lwd’ standard graphics unit (see `gpar`).
- ‘linetype‘ Line type, using ’lty’ standard graphics unit (see `gpar`).
- ‘alpha‘ Transparency applied to the edge colour.

See Also

GeomEdgeSpace, GraphSpace

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a GraphSpace object
gs <- GraphSpace(gtoy1)

## Not run:

ggplot() + geom_edgespace(aes(x = x, y = y, xend = xend, yend = yend),
  data = gs) + theme(aspect.ratio = 1) +
  scale_x_continuous(limits = c(0, 1)) +
  scale_y_continuous(limits = c(0, 1))


## End(Not run)
```

---

geom_nodespace *Draw node elements in a 2D graph layout*

Description

Constructor for GeomNodeSpace ggproto objects, a variant of geom_point supporting node attributes from GraphSpace objects.

This geom is designed to create node-level aesthetics such as ‘size‘, ‘fill‘, ‘colour‘, or any custom aesthetics defined in ‘GeomNodeSpace‘.

Usage

```
geom_nodespace(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = FALSE
)
```

Arguments

`mapping` Set of aesthetic mappings created by [ggplot2::aes()]. These mappings apply to the layer and may override global aesthetics.

`data` Either a GraphSpace object or a data frame containing node attributes. If ‘NULL‘, the layer will use the default plot data.

`stat` The statistical transformation to use on the data. Defaults to ‘"identity"‘.

`position` Position adjustment, either as a string or the result of a call to a position adjustment function.

`...` Additional parameters passed to the underlying drawing function in ‘GeomNodeSpace‘.

`na.rm` Logical. Should missing values be removed? Defaults to ‘TRUE‘.

`show.legend` Logical or a named logical vector indicating whether this layer should be included in legends.

`inherit.aes` Logical. If ‘FALSE‘ (default), the layer will use aesthetics defined in ‘mapping‘. Set to ‘TRUE‘ if this geom should inherit aesthetics from the plot.

Value

A ggplot2 layer that renders node glyphs defined by ‘GeomNodeSpace‘.

Aesthetics

‘geom_nodespace‘ understands the following aesthetics:

- ‘x‘ Horizontal position in ‘npc‘ `unit` (required).
- ‘y‘ Vertical position in ‘npc‘ `unit` (required).
- ‘size‘ Node size in ‘npc‘ `unit` (required).
- ‘fill‘ Node fill colour.
- ‘colour‘ Node line colour.
- ‘shape‘ Integer code between 0 and 25 (see `points`).
- ‘linewidth‘ Line width, using ’lwd’ standard graphics unit (see `gpar`).
- ‘alpha‘ Transparency applied to fill and line colour.

See Also

`GeomNodeSpace`, GraphSpace

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a GraphSpace object
gs <- GraphSpace(gtoy1)

## Not run:
```

```
ggplot() + geom_nodespace(aes(x = x, y = y), data = gs)


## End(Not run)
```

---

getGraphSpace,GraphSpace-method

Accessors for fetching slots from a GraphSpace object

Description

getGraphSpace retrives information from individual slots available in a GraphSpace object.

Usage

```
## S4 method for signature 'GraphSpace'
getGraphSpace(gs, what = "graph")
```

Arguments

gs	A preprocessed GraphSpace class object
what	A single character value specifying which information should be retrieved from the slots. Options: 'graph','gxy','gxyz','pars','misc','status','summits', 'summit_mask', and 'summit_contour'.

Value

Content from slots in the GraphSpace object.

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a new GraphSpace object
gs <- GraphSpace(gtoy1)

# Get the 'summary' slot in gs
getGraphSpace(gs, what = 'graph')
```

GraphSpace *Constructor of GraphSpace-class objects*

Description

GraphSpace is a constructor of GraphSpace-class objects.

Usage

```
GraphSpace(g, mar = 0.1, layout = NULL, image = NULL, verbose = TRUE)
```

Arguments

g	An igraph object. It must include graph coordinates assigned to x and y vertex attributes, and vertex labels assigned to name vertex attribute.
mar	A single numeric value (in [0,1]) indicating the size of the outer margins as a fraction of the graph space. Note: When an image is provided, mar is a fraction of image margins.
layout	An optional numeric matrix with two columns for x and y coordinates.
image	An optional background image. When provided, x and y coordinates must represent pixel positions in the image space.
verbose	A single logical value specifying to display detailed messages (when verbose=TRUE) or not (when verbose=FALSE).

Value

A GraphSpace class object.

Author(s)

Sysbiolab.

See Also

plotGraphSpace

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

gs <- GraphSpace(gtoy1)
```

GraphSpace-class *GraphSpace: An S4 class for igraph objects*

Description

GraphSpace: An S4 class for igraph objects

Value

An S4 class object.

Slots

`nodes` A data frame with xy-vertex coordinates.

`edges` A data frame with edges.

`graph` An igraph object.

`image` A raster background image matrix.

`pars` A list with parameters.

`misc` A list with intermediate objects for downstream methods.

Constructor

see `GraphSpace` constructor.

gs_nodes,GraphSpace-method *Accessors for applying essential igraph methods to modify attributes of GraphSpace objects.*

Description

Access and modify individual slots of a GraphSpace object. Selected 'igraph' methods are applied to the 'graph' slot and propagated to downstream components.

Usage

```
## S4 method for signature 'GraphSpace'
gs_nodes(x)

## S4 method for signature 'GraphSpace'
gs_edges(x)

## S4 method for signature 'GraphSpace'
gs_vcount(x)
```

```
## S4 method for signature 'GraphSpace'
gs_ecount(x)

## S4 method for signature 'GraphSpace'
names(x)

## S4 replacement method for signature 'GraphSpace'
names(x) <- value

## S4 method for signature 'GraphSpace'
gs_vertex_attr(x, name, ...)

## S4 replacement method for signature 'GraphSpace'
gs_vertex_attr(x, name, ...) <- value

## S4 method for signature 'GraphSpace'
gs_edge_attr(x, name, ...)

## S4 replacement method for signature 'GraphSpace'
gs_edge_attr(x, name, ...) <- value
```

Arguments

x	A GraphSpace class object
value	The new value of the attribute.
name	Name of the attribute.
...	Additional arguments passed to igraph methods.

Value

Updated GraphSpace object.

See Also

vertex_attr, edge_attr

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Create a new GraphSpace object
gs <- GraphSpace(gtoy1)

#--- Usage of GraphSpace attribute accessors:

# Get a data frame with nodes for plotting methods
gs_nodes(gs)
```

```
# Get a data frame with edges for plotting methods
gs_edges(gs)

# Get vertex count
gs_vcount(gs)

# Get edge count
gs_ecount(gs)

# Get vertex names
names(gs)

# Access all vertex attributes
gs_vertex_attr(gs)

# Access a specific vertex attribute
gs_vertex_attr(gs, "nodeLabel")

# Modify a single value within a vertex attribute
gs_vertex_attr(gs, "nodeSize")["n1"] <- 10

# Replace an entire vertex attribute
gs_vertex_attr(gs, "nodeSize") <- 10

# Alternative syntax using `$` accessor
gs_vertex_attr(gs)$nodeSize <- 10

# Access a specific edge attribute
gs_edge_attr(gs, "edgeLineColor")

# Replace an entire edge attribute
gs_edge_attr(gs, "edgeLineWidth") <- 1

 # Alternative syntax using `$` for edge attributes
gs_edge_attr(gs)$edgeLineWidth <- 3
```

gtoys *Toy 'igraph' objects*

Description

Small 'igraph' objects used for workflow demonstrations. All graphs include 'x', 'y', and 'name' vertex attributes.

Usage

```
data(gtoy1)
```

Format

igraph

Value

A pre-processed igraph object.

Source

This package.

Examples

```
data(gtoy1)
data(gtoy2)
```

---

make_gspace_legend *Create standalone legends for RGraphSpace's ggplot objects*

Description

This accessor builds a standalone legend from a named vector of colors. The resulting legend is returned as a 'grob' object that can be used to add external legends to existing ggplots.

Usage

```
make_gspace_legend(
  color_palette,
  breaks = NULL,
  legend_title = "Title",
  legend_shape = 21,
  legend_size = 6,
  text_size = 10,
  orientation = c("vertical", "horizontal"),
  justification = c("left", "right"),
  custom_theme = theme()
)

add_gspace_legend(
  plot,
  legend,
  spacer = unit(2, "mm"),
  position = c("right", "top", "bottom", "left")
)
```

Arguments

`color_palette` A named character vector of colors. Names become legend labels. Values must be valid R color specifications, e.g. ‘color_palette = c(A = "#FF0000", B = "blue")‘.

`breaks` A vector indicating numeric breaks for a continuous legend. If missing, ‘color_palette‘ will instead produce a discrete legend.

`legend_title` The legend title.

`legend_shape` Point shape (default 21, for circle).

`legend_size` Size of legend points (default 6 mm).

`text_size` Text size (default 10 mm).

`orientation` Legend arrangement ("vertical" or "horizontal").

`justification` Legend justification within bounding box ("left" or "right").

`custom_theme` An optional ‘ggplot2‘ theme object applied to the legend panel. This allows customizing the appearance of the legend without affecting the main plot.

`plot` A GraphSpace plot.

`legend` A list with multiple ‘gspace‘ legends.

`spacer` A grid::unit() value specifying space inserted between merged legends.

`position` Placement of legend relative to the main plot.

Details

The object returned by ‘add_gspace_legend()‘ behaves like a ggplot for saving, combining with patchwork, arranging in grids, and rendering in standard graphics devices. Internally, however, the original ggplot is converted into a static ’grob’ embedded inside a blank ggplot. As a result, the returned object cannot accept new layers, scales, or other ggplot modifications; it is no longer an editable ggplot, but a wrapped graphical representation of the composed figure.

Value

The ‘make_gspace_legend()‘ function returns a standalone legend, while the ‘add_gspace_legend()‘ function returns a patchwork object containing a plot combined with the generated legend in the specified position.

Examples

```
library(ggplot2)
library(patchwork)

p <- ggplot(mtcars, aes(wt, mpg)) + geom_point()

# Add a discrete standalone legend to a plot
leg1 <- make_gspace_legend(c(A="red", B="blue"), legend_size = 5)
p + leg1 + plot_layout(widths = c(1, 0.1))

# Add a continuous standalone legend to a plot
leg2 <- make_gspace_legend(c("blue", "white","red"), breaks = c(0, 1, 2))
```

```
p + leg2 + plot_layout(widths = c(1, 0.1))

# Add a legend list to a plot
legs <- list(leg1, leg2)
add_gspace_legend(plot = p, legs, position = "right")
```

`plot.GraphSpace` *Plot GraphSpace objects*

Description

Plot GraphSpace objects

Usage

```
## S3 method for class 'GraphSpace'
plot(x, ...)
```

Arguments

`x`	A GraphSpace class object.
`...`	Additional arguments passed to the `plotGraphSpace` function.

See Also

`plotGraphSpace`

`plotGraphSpace,GraphSpace-method` *Plotting igraph objects with RGraphSpace*

Description

`plotGraphSpace` is a wrapper function to create dedicated ggplot graphics for igraph- and GraphSpace-class objects.

Usage

```
## S4 method for signature 'GraphSpace'
plotGraphSpace(
  gs,
  theme = c("th0", "th1", "th2", "th3"),
  xlab = "Graph coordinates 1",
  ylab = "Graph coordinates 2",
  font.size = 1,
  bg.color = "grey95",
  add.labels = FALSE,
  node.labels = NULL,
  label.size = 3,
  label.color = "grey20",
  add.image = FALSE,
  raster = FALSE,
  dpi = 300,
  dev = "cairo_png"
)

## S4 method for signature 'igraph'
plotGraphSpace(gs, ..., mar = 0.1)
```

Arguments

`gs` Either an `igraph` or GraphSpace class object. If `gs` is an `igraph`, then it must include `x`, `y`, and `name` vertex attributes (see `GraphSpace`).

`theme` Name of a custom RGraphSpace theme. These themes (from 'th1' to 'th3') consist of preconfigured ggplot settings, which the user can subsequently refine using `ggplot2`.

`xlab` The title for the 'x' axis of a 2D-image space.

`ylab` The title for the 'y' axis of a 2D-image space.

`font.size` A single numeric value passed to ggplot themes.

`bg.color` A single color for background.

`add.labels` A logical value indicating whether to plot vertex labels.

`node.labels` A vector of vertex names to be highlighted in the graph space. This argument overrides 'add.labels'.

`label.size` A size argument passed to `geom_text`.

`label.color` A color passed to `geom_text`.

`add.image` A logical value indicating whether to add a background image, when one is available (see `GraphSpace`).

`raster` A logical value indicating whether to rasterize the main plot. See `rasterise` for further specifications.

`dpi` Raster resolution, in dots per inch.

`dev` Device used in the `rasterise` call.

`...` Additional arguments passed to the `plotGraphSpace` function.

`mar` A single numeric value (in `[0,1]`) indicating the size of the outer margins as a fraction of the graph space.

Value

A ggplot-class object.

Author(s)

Sysbiolab.

See Also

GraphSpace

Examples

```
# Load a demo igraph
data('gtoy1', package = 'RGraphSpace')

# Generate a ggplot for igraph
plotGraphSpace(gtoy1)

# Create a GraphSpace object
gs <- GraphSpace(gtoy1)

# Generate a ggplot for gs
plotGraphSpace(gs)
```

Index