Package ‘aplot’

July 7, 2026

Title Decorate a 'ggplot' with Associated Information

Version 0.3.1

Description Supports data-driven composition of a main plot with associated subplots that need precise axis alignment. Unlike general layout-focused tools such as 'cowplot' and 'patchwork', it enables related subplots to be integrated on the top, bottom, left, or right sides of a main plot with matched axes, so that the combined panels can be interpreted as a coherent whole. This design was inspired by the 'Method 2' described in 'ggtree' (G Yu (2018) <doi:10.1093/molbev/msy194>).

Depends R (>= 4.1.0)

Imports ggfun (>= 0.1.3), gtable, ggplot2, ggplotify, patchwork, magrittr, methods, utils, yulab.utils (>= 0.1.9), pillar

Suggests ggtree, testthat (>= 3.0.0)

URL https://github.com/YuLab-SMU/aplot, https://yulab-smu.top/aplot/

License Artistic-2.0

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author Guangchuang Yu [aut, cre] (ORCID: <https://orcid.org/0000-0002-6485-8781>), Shuangbin Xu [ctb] (ORCID: <https://orcid.org/0000-0003-3513-5362>), Thomas Hackl [ctb]

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Repository CRAN

Date/Publication 2026-07-07 08:30:02 UTC

Contents

as.patchwork 2
gglist 2

gglistGrob 3
insert_left 3
plot_list 4
set_guide_area 6
set_guide_layout 6
set_panel_spacing 8
xlab2 9
xlim2 9

Index **11**

`as.patchwork` *as.patchwork*

Description

as.patchwork

Usage

```
as.patchwork(x, align = getOption("aplot_align", default = "xy"))
```

Arguments

`x`	object
`align`	"x","y","xy","none", align the axis of x/y or not.

`gglist` *construct a ‘gglist‘ object*

Description

constructure a ‘gglist‘ object that contains a list of plots (‘gglist‘) and parameters (via ‘...‘), the object can be displayed via the ‘plot_list()‘ function.

Usage

```
gglist(gglist, ...)
```

Arguments

`gglist`	a list of plots
`...`	parameters for plotting the ‘gglist‘

Value

gglist object

Author(s)

Guangchuang Yu

`gglistGrob` *This function converts 'gglist' object to grob (i.e. gtable object)*

Description

title gglistGrob

Usage

```
gglistGrob(x)
```

Arguments

`x`	A 'gglist' object

Value

A 'gtable' object

`insert_left` *plot-insertion*

Description

insert an associated plot to left, right, top and bottom of a main plot

Usage

```
insert_left(.data, plot, width = 1)

insert_right(.data, plot, width = 1)

insert_top(.data, plot, height = 1)

insert_bottom(.data, plot, height = 1)
```

Arguments

`.data`	an 'aplot' or 'gg' object
`plot`	a 'gg' plot to be inserted
`width`	relative width to the main plot
`height`	relative height to the main plot

Details

The first input serve as a main plot, and other plots can be progressively inserted to different sides on left, right, top and bottom.

Value

an 'aplot' object

Author(s)

Guangchuang Yu

Examples

```
library(ggplot2)
library(aplot)

p <- ggplot(mtcars, aes(mpg, disp)) + geom_point()
p2 <- ggplot(mtcars, aes(mpg)) +
    geom_density(fill='steelblue', alpha=.5) +
        ggfun::theme_noxaxis()
p3 <- ggplot(mtcars, aes(x=1, y=disp)) +
    geom_boxplot(fill='firebrick', alpha=.5) +
    theme_void()
ap <- p %>%
    insert_top(p2, height=.3) %>%
    insert_right(p3, width=.1)

ap
ap[2, 1] <- ap[2, 1] + theme_bw()
ap[2, 1] <- ap[2, 1] +
            aes(color = as.factor(am)) +
            scale_color_manual(values = c('steelblue', 'darkgreen'))
ap[1, 1] <- ap[1, 1] + theme(axis.line.x.bottom=element_line())
ap
```

`plot_list` *plot a list of ggplot objects*

Description

plot a list of ggplot objects using patchwork, similar to ‘cowplot::plot_grid(plotlist)‘

Usage

```
plot_list(
  ...,
  gglist = NULL,
  ncol = NULL,
```

```
  nrow = NULL,
  byrow = NULL,
  widths = NULL,
  heights = NULL,
  guides = NULL,
  labels = NULL,
  tag_levels = NULL,
  tag_size = 14,
  design = NULL,
  output = "patchwork"
)
```

Arguments

`...`	list of plots to be arranged
`gglist`	(optional) list of plots
`ncol`	number of columns
`nrow`	number of rows
`byrow`	If "FALSE" the plots will be filled in in column-major order
`widths`	relative widths
`heights`	relative heights
`guides`	A string specifying how guides should be treated in the layout.
`labels`	manual specified labels to label plots
`tag_levels`	format to label plots, will be disable if 'labels' is not NULL
`tag_size`	size of tags
`design`	specification of the location of areas in the layout
`output`	one of 'gglist' or 'patchwork'

Value

composite plot

Author(s)

Guangchuang Yu

set_guide_area *set_guide_area*

Description

Set a guide area on an 'aplot' object

Usage

```
set_guide_area(
  x,
 position = c("top-right", "top-left", "bottom-right", "bottom-left", "right", "left",
    "bottom", "top")
)
```

Arguments

x	an 'aplot' object
position	placement of collected guides. Supported values are '"top-right"', '"top-left"', '"bottom-right"', '"bottom-left"', '"right"', '"left"', '"bottom"', and '"top"'. Corner positions use the largest continuous empty rectangular region anchored at the requested corner. Side positions create a single guide region on the requested side, where the guide box is placed as one object rather than as repeated guide cells.

Details

Designate a guide placement region in an 'aplot' layout for collected guides. Corner positions use the continuous empty region anchored at the requested corner. Side positions create a guide region on the requested side. The selected guide area remains empty when guides are not collected.

Value

an 'aplot' object

set_guide_layout *set_guide_layout*

Description

Set collected guide layout on an 'aplot' object

Usage

```
set_guide_layout(
  x,
  legend_ncol = NULL,
  legend_byrow = FALSE,
  legend_title_position = NULL,
  legend_direction = NULL,
  legend_keywidth = NULL,
  legend_keyheight = NULL,
  legend_spacing_x = NULL,
  legend_spacing_y = NULL,
  guides_ncol = NULL,
  guides_byrow = FALSE,
  guides_direction = NULL
)
```

Arguments

`x` an 'aplot' object

`legend_ncol` number of columns used inside each individual legend. Must be a positive integer. If ‘NULL‘, the internal column layout of each legend is left unchanged.

`legend_byrow` logical; whether keys inside each individual legend should be filled row-wise when a multi-column legend is created. The default is ‘FALSE‘.

`legend_title_position` title position used inside each individual legend. Must be one of ‘"top"‘, ‘"bottom"‘, ‘"left"‘, or ‘"right"‘. If ‘NULL‘, the current title position is left unchanged.

`legend_direction` direction used inside each individual legend. Must be one of ‘"horizontal"‘ or ‘"vertical"‘. If ‘NULL‘, the current legend direction is left unchanged.

`legend_keywidth`, `legend_keyheight` legend key size in millimeters for each individual legend. These values are converted to ‘grid::unit(..., "mm")‘ and passed to [ggplot2::guide_legend()]. If ‘NULL‘, the corresponding key dimension is left unchanged.

`legend_spacing_x`, `legend_spacing_y` spacing between legend items in millimeters inside each individual legend. These values are converted to ‘grid::unit(..., "mm")‘ and applied through [ggplot2::theme()] as ‘legend.spacing.x‘ and ‘legend.spacing.y‘. If ‘NULL‘, the corresponding spacing is left unchanged.

`guides_ncol` number of columns used to arrange multiple collected legend boxes as a group. Must be a positive integer. If ‘NULL‘, the group layout stays in a single strip determined by ‘guides_direction‘.

`guides_byrow` logical; whether collected legend boxes should be filled row-wise when a wrapped guide group is created. The default is ‘FALSE‘.

`guides_direction`
: overall direction used to arrange multiple collected legend boxes. Must be one of ‘"horizontal"‘ or ‘"vertical"‘. If ‘NULL‘, the default follows the guide placement side used by [set_guide_area()].

Details

Control collected guides at two semantic levels when an ’aplot’ object is rendered. Use [set_guide_area()] to decide where guides should go, and use ‘set_guide_layout()‘ to decide how they should be arranged once placed.

Value

an ’aplot’ object

`set_panel_spacing` *set_panel_spacing*

Description

Set panel spacing for an ’aplot’ object

Usage

```
set_panel_spacing(x, spacing = 0)
```

Arguments

`x`	an ’aplot’ object
`spacing`	a non-negative numeric value in millimeters, or ‘"asis"‘

Details

Control the spacing between aligned panels when an ’aplot’ object is rendered. Numeric values are interpreted in millimeters and represent the gap between neighboring panels. Use ‘"asis"‘ to preserve the subplot margins already defined in each plot.

Value

an ’aplot’ object

`xlab2` *display x or y axis label as an ordinary text, so that the label will not be aligned with axis label of another plot*

Description

display x or y axis label as an ordinary text, so that the label will not be aligned with axis label of another plot

Usage

```
xlab2(label, fontsize = 11, ...)

ylab2(label, fontsize = 11, ...)
```

Arguments

`label`	axis label
`fontsize`	fontsize of the label
`...`	additional parameter passed to ‘gpar‘

Value

gg object with new label

Author(s)

Guangchuang Yu

`xlim2` *xlim2*

Description

set axis limits (x or y) of a ‘ggplot‘ object (left hand side of ‘+‘) based on the x (‘xlim2‘) or y (‘ylim2‘) limits of another ‘ggplot‘ object (right hand side of ‘+‘). This is useful for using ‘cowplot‘ or ‘patchwork‘ to align ‘ggplot‘ objects.

Usage

```
xlim2(gg, limits = NULL)

ylim2(gg, limits = NULL)
```

Arguments

gg	ggplot object
limits	vector of limits. If NULL, determine from ‘gg‘.

Value

ggplot2 object with new limits

Author(s)

Guangchuang Yu

Examples

```
library(ggplot2)
library(aplot)
p1 <- ggplot(mtcars, aes(cyl)) + geom_bar()
p2 <- ggplot(subset(mtcars, cyl != 4), aes(cyl)) + geom_bar()
p2 + xlim2(p1)
```

Index

as.patchwork, 2

gglist, 2
gglistGrob, 3

insert_bottom *(*insert_left*)*, 3
insert_left, 3
insert_right *(*insert_left*)*, 3
insert_top *(*insert_left*)*, 3

plot_list, 4

set_guide_area, 6
set_guide_layout, 6
set_panel_spacing, 8

xlab2, 9
xlim2, 9

ylab2 *(*xlab2*)*, 9
ylim2 *(*xlim2*)*, 9