

Package ‘ggtreeExtra’

March 30, 2021

Type Package

Title An R Package To Add Geom Layers On Circular Or Other Layout Tree
Of ``ggtree''

Version 1.0.2

Description 'ggtreeExtra' extends the method for mapping and visualizing associated data on phylogenetic tree using 'ggtree'. These associated data can be mapped to circular layout, fan layout, or other layout tree built by 'ggtree' with the grammar of 'ggplot2'.

Imports ggplot2, utils, rlang, ggnewscale, stats

Suggests ggtree, treeio, ggstar, patchwork, knitr, rmarkdown,
prettydoc, markdown

License GPL-3

Encoding UTF-8

URL <https://github.com/YuLab-SMU/ggtreeExtra/>

BugReports <https://github.com/YuLab-SMU/ggtreeExtra/issues>

LazyData true

ByteCompile true

Roxygen list(markdown = TRUE)

biocViews Software, Visualization, Phylogenetics, Annotation

RoxygenNote 7.1.1

VignetteBuilder knitr

git_url <https://git.bioconductor.org/packages/ggtreeExtra>

git_branch RELEASE_3_12

git_last_commit 9579685

git_last_commit_date 2021-03-22

Date/Publication 2021-03-29

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

Maintainer Shuangbin Xu <xshuangbin@163.com>

R topics documented:

geom_fruit	2
geom_fruit_list	5
PositionDodgex	7
position_dodgex	7
position_identityx	8
position_points_jitterx	9
position_points_sinax	10
position_raincloudx	11
position_stackx	12

Index	14
--------------	-----------

geom_fruit	<i>geom_fruit</i>
------------	-------------------

Description

plot tree associated data in another method.

Usage

```
geom_fruit(
  mapping,
  data = NULL,
  geom,
  offset = 0.03,
  pwidth = 0.2,
  position = "auto",
  grid.params = NULL,
  axis.params = list(axis = "none", text.angle = 0, text.size = 0.8, text = NULL, title
    = NULL, title.size = 3, title.height = 0.1, title.angle = 0, title.color = "black",
    nbreak = 4, line.size = 0.2, line.color = "grey", line.alpha = 1, ...),
  ...
)

fruit_plot(
  p,
  data = NULL,
  geom,
  mapping,
  offset = 0.03,
  pwidth = 0.2,
  position = "auto",
  ...
)
```

Arguments

mapping aes mapping for 'geom'

data	data to plot by 'geom', the column contained tree tip labels should be as y in mapping.
geom	geom function to plot the data.
offset	numeric, distance between panels, the ratio of distance to tree, default is 0.03.
pwidth	numeric, the width of new geom layer, this is the ratio of the new geom to tree, default is 0.2.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function, default is 'auto'.
grid.params	list, the parameters to control the attributes of grid lines, default is NULL.
axis.params	list, the parameters to control the attributes of pseudo axis.
...	additional parameters for 'geom'

grid.params control the attributes of grid line, it can be referred to the following parameters:

- `vline` logical, whether add the vertical line, default is FALSE.
- `color` color of line, default is grey.
- `size` the width of line, default is 0.2.
- `alpha` the colour transparency of line, default is 1.
- `lineend` Line end style (round, butt, square), default is "butt".
- `linejoin` Line end style (round, butt, square), default is "round".
- `linetype` Type of line, default is 1.

axis.params control the attributes of pseudo axis, it can be referred to the following parameters:

- `axis` character, add the pseudo axis, "none" don't display axis (default), "x" display the x axis, "y" display the (y) axis of tree tip, "xy" display the two axis.
- `text` vector, the text of axis x, default is NULL, it is only valid when the text of axis is single and x is discrete.
- `vjust` numeric, A numeric specifying vertical justification, default is 0.5.
- `hjust` numeric, A numeric specifying horizontal justification, default is 0.5.
- `text.angle` numeric, the angle of axis text, default is 0.
- `text.size` numeric, the size of axis text, default is 0.8.
- `title` character, the title of panel or x-axis label, default is NULL, it is only valid when "x" axis exists.
- `title.size` numeric, the size of title text, default is 3.
- `title.height` numeric, the height of title text position more than tree, default is 0.1, it is relative to height of tree.
- `title.angle` numeric, the angle of title text, default is 0.
- `title.color` character, the color of title text, default is "black".
- `nbreak` numeric, meaning the number of axis to break, integer giving the *desired* number of intervals. Non-integer values are rounded down. It is only valid when x is continuous, default is 4.
- `line.size` numeric, the size of axis line, default is 0.2.
- `line.color` character, the color of axis color, default is "grey".
- `line.alpha` numeric, the colour transparency of line, default is 1.

Details

'geom_fruit()' automatically re-arranges the input 'data' according to the tree structure, visualizes the 'data' on specific 'panel' using the 'geom' function with aesthetic 'mapping' and other parameters, and align the graph with the tree 'p' side by side. The default position parameters is 'auto'. If you want to set manually. You can use 'position_stackx()' or 'position_dodgex()' for 'geom_bar', 'position_identityx()' for 'geom_tile', 'geom_point', 'geom_star', 'geom_symbol' or other layers using 'identity' position in 'ggplot2', and 'position_dodgex()' or 'position_dodgex2()' for 'geom_boxplot' 'geom_violin' or other layers using 'dodge' position in 'ggplot2'. The axis line and text can be added using 'axis.params=list(axis="x",...)', and the grid line also can be added using 'grid.params=list(...)'.

Value

ggplot object

Author(s)

Shuangbin Xu and Guangchuang Yu

Examples

```
library(ggtree)
library(ggplot2)
library(ggstar)
set.seed(1024)
tr <- rtree(100)
dd = data.frame(id=tr$tip.label, value=abs(rnorm(100)))
dt = data.frame(id=tr$tip.label, group=c(rep("A",50),rep("B",50)))
p <- ggtree(tr, layout="circular")

p1 <- p +
  geom_fruit(
    data=dt,
    geom=geom_star,
    mapping=aes(y=id, fill=group),
    size=2.5,
    starstroke=0
  )
p2 <- p1 +
  geom_fruit(
    data=dd,
    geom=geom_bar,
    mapping=aes(x=value, y=id),
    orientation="y",
    stat="identity"
  )

p3 <- fruit_plot(
  p=p,
  data=dt,
  geom=geom_star,
  mapping=aes(y=id, fill=group),
  size=2.5,
  starstroke=0
)
```

```
p4 <- fruit_plot(  
  p=p3,  
  data=dd,  
  geom=geom_bar,  
  mapping=aes(x=value, y=id),  
  orientation="y",  
  stat="identity"  
)  
p <- p %<+% dd %<+% dt  
p5 <- p +  
  geom_fruit(  
    geom = geom_star,  
    mapping = aes(y=id, fill=group),  
    size = 2.5,  
    starstroke = 0  
  ) +  
  geom_fruit(  
    geom = geom_bar,  
    mapping = aes(x=value, y=id),  
    orientation = "y",  
    stat = 'identity'  
  )
```

geom_fruit_list

geom_fruit_list

Description

add the layers to the same position out of ggtree.

Usage

```
geom_fruit_list(fruit, ...)
```

Arguments

`fruit` the layer of geom_fruit.
`...` another layers of geom_fruit, or scales.

Value

ggplot object

Author(s)

Shuangbin Xu and GuangChuang Yu

Examples

```

library(ggplot2)
library(ggtree)
library(ggstar)
library(ggnewscale)
set.seed(1024)
tr <- rtree(100)
dt <- data.frame(id=tr$tip.label, value=abs(rnorm(100)), group=c(rep("A",50),rep("B",50)))
df <- dt
dtf <- dt
colnames(df)[[3]] <- "group2"
colnames(dtf)[[3]] <- "group3"

p <- ggtree(tr, layout="fan", open.angle=0)
# first circle
p1 <- p +
  geom_fruit(
    data=dt,
    geom=geom_bar,
    mapping=aes(y=id, x=value, fill=group),
    orientation="y",
    stat="identity"
  ) +
  new_scale_fill()
# second circle
fruitlist <- geom_fruit_list(
  geom_fruit(
    data = df,
    geom = geom_bar,
    mapping = aes(y=id, x=value, fill=group2),
    orientation = "y",
    stat = "identity",
    addbrink=FALSE
  ),
  scale_fill_manual(values=c("blue", "red")),
  new_scale_fill(),
  geom_fruit(
    data = dt,
    geom = geom_star,
    mapping = aes(y=id, x=value, fill=group),
    size = 2.5,
    color = NA,
    starstroke = 0
  )
)
p2 <- p1 + fruitlist + new_scale_fill()
# third circle
p3 <- p2 +
  geom_fruit(
    data=dtf,
    geom=geom_bar,
    mapping = aes(y=id, x=value, fill=group3),
    orientation = "y",
    stat = "identity"
  ) +

```

```

    scale_fill_manual(values=c("#00AED7", "#009E73"))
  p3

```

PositionDodgex	<i>PositionDodgex</i>
----------------	-----------------------

Description

PositionDodgex
 PositionDodgex2
 PositionIdentityx
 PositionPointsJitterx
 PositionRaincloudx
 PositionPointsSinax
 PositionStackx

Author(s)

Shuangbin Xu

position_dodgex	<i>Dodge overlapping objects side-to-side which can be shifted vertically or horizontally.</i>
-----------------	--

Description

Dodging preserves the vertical position of an geom while adjusting the horizontal position. `position_dodgex2` is a special case of `position_dodgex` for arranging box plots, which can have variable widths. `position_dodgex2` also works with bars and rectangles. But unlike `position_dodgex`, `position_dodgex2` works without a grouping variable in a layer.

Usage

```

position_dodgex(
  width = NULL,
  hexpand = NA,
  vexpand = NA,
  preserve = c("total", "single")
)

position_dodgex2(
  width = NULL,
  preserve = c("total", "single"),
  hexpand = NA,
  vexpand = NA,
  padding = 0.1,
  reverse = FALSE
)

```

Arguments

width	Dodging width, when different to the width of the individual elements. This is useful when you want to align narrow geoms with wider geoms.
hexpand	numeric, Horizon expand for geoms that have a position, default is NA.
vexpand	numeric, Vertical expand for geoms that have a position, default is NA.
preserve	Should dodging preserve the total width of all elements at a position, or the width of a single element?
padding	Padding between elements at the same position. Elements are shrunk by this proportion to allow space between them. Defaults to 0.1.
reverse	If TRUE, will reverse the default stacking order. This is useful if you're rotating both the plot and legend.

Value

position methods

See Also

Other position adjustments: [position_identityx\(\)](#), [position_points_sinx\(\)](#)

Examples

```
library(ggplot2)
library(patchwork)
iris$ID <- rep(c(rep("test1", 15), rep("test2", 15), rep("test3", 20)),3)
p <- ggplot(iris, aes(x=Species,y=Petal.Length,fill=ID))
p1 <- p + geom_bar(stat="identity",position=position_dodgex())
p2 <- p + geom_bar(stat="identity",position=position_dodgex(vexpand=5))
p3 <- ggplot(iris, aes(x=Petal.Length, y=Species, fill=ID)) +
  geom_bar(stat="identity", orientation="y",
           position=position_dodgex(hexpand=5))
p4 <- p1 + p2 + p3
p4
p5 <- p + geom_boxplot(position=position_dodgex2())
p6 <- p + geom_boxplot(position=position_dodgex2(vexpand=5))
p7 <- ggplot(iris, aes(x=Petal.Length, y=Species, fill=ID)) +
  geom_boxplot(orientation="y",
              position=position_dodgex2(hexpand=5))
p8 <- p5 + p6 + p7
p8
```

`position_identityx` *adjust identity position which can be shifted vertically or horizontally.*

Description

adjust identity position which can be shifted vertically or horizontally.

Usage

```
position_identityx(hexpand = NA, vexpand = NA)
```


Arguments

hexpand	numeric, distance to be shifted horizontally for geoms that have a position, default is NA.
vexpand	numeric, distance to be shifted vertically for geoms that have a position, default is NA.

Value

position method.

Author(s)

Shuangbin Xu

See Also

Other position adjustments: [position_dodgex\(\)](#), [position_points_sinx\(\)](#)

Examples

```
library(ggplot2)
library(patchwork)
p <- ggplot(mtcars, aes(x=wt, y=mpg))
p1 <- p + geom_point(position=position_identityx()) + ylim(0, 50)
# whole point layer was shifted vertically (distance=5).
# the label of axis y should be subtracted 5 to get the true value..
p2 <- p + geom_point(position=position_identityx(vexpand=5)) + ylim(0, 50)
# whole point layer was shifted horizontally (distance=5).
# the label of axis x should be subtracted 5 to get the true value.
p3 <- ggplot(mtcars, aes(y=wt, x=mpg)) +
  geom_point(position=position_identityx(hexpand=5)) + xlim(0, 50)
p4 <- p1 + p2 + p3
p4
```

position_points_jitterx

Randomly jitter the points in a ridgeline plot which can be shifted horizontally

Description

This is a position adjustment specifically for 'geom_density_ridges()' and related geoms. It only jitters the points drawn by these geoms, if any. If no points are present, the plot remains unchanged. The effect is similar to [position_jitter\(\)](#): points are randomly shifted up and down and/or left and right. It add 'hexpand' that can control shift horizontally.

Usage

```
position_points_jitterx(
  width = 0,
  height = 0.2,
  yoffset = 0,
  hexexpand = NA,
  adjust_vlines = FALSE,
  seed = NULL
)
```

Arguments

width	Width for horizontal jittering. By default set to 0.
height	Height for vertical jittering, applied in both directions (up and down). By default 0.2.
yoffset	Vertical offset applied in addition to jittering.
hexexpand	numeric, distance to be shifted horizontally for geoms that have a position, default is NA.
adjust_vlines	If TRUE, adjusts vertical lines (as are drawn for quantile lines, for example) to align with the point cloud.
seed	Random seed. If set to NULL, the current random number generator is used. If set to NA, a new random random seed is generated. If set to a number, this number is used as seed for jittering only.

See Also

Other position adjustments for ridgeline plots: [position_points_sinax](#), [position_raincloudx](#)

`position_points_sinax` *adjust ridgeline plot position which can be shifted vertically or horizontally.*

Description

This is a position adjustment specifically for 'geom_density_ridges()', but it add 'hexexpand' that can control shift horizontally.

Usage

```
position_points_sinax(
  rel_min = 0.02,
  rel_max = 0.98,
  seed = NULL,
  hexexpand = NA
)
```

Arguments

rel_min	numeric, the relative minimum value at which a point can be placed.
rel_max	numeric, the relative maximum value at which a point can be placed.
seed	numeric, Random seed, if set to NULL, the current random number generator is used. If set to NA, a new random random seed is generated. If set to a number, this number is used as seed for jittering only, default is NULL.
hexpand	numeric, distance to be shifted horizontally for geoms that have a position, default is NA.

Value

position method.

Author(s)

Shuangbin Xu

See Also

Other position adjustments: [position_dodgex\(\)](#), [position_identityx\(\)](#)

position_raincloudx	<i>Create a cloud of randomly jittered points below a ridgeline plot which can be shifted horizontally</i>
---------------------	--

Description

This is a position adjustment specifically for 'geom_density_ridges()' and related geoms. It only jitters the points drawn by these geoms, if any. If no points are present, the plot remains unchanged. The effect is similar to [position_points_jitterx\(\)](#), only that by default the points lie all underneath the baseline of each individual ridgeline. It add 'hexpand' that can control shift horizontally.

Usage

```
position_raincloudx(
  width = 0,
  height = 0.4,
  ygap = 0.05,
  hexpand = NA,
  adjust_vlines = FALSE,
  seed = NULL
)
```

Arguments

width	Width for horizontal jittering. By default set to 0.
height	Total height of point cloud. By default 0.4.
ygap	Vertical gap between ridgeline baseline and point cloud.
hexpand	numeric, distance to be shifted horizontally for geoms that have a position, default is NA.

adjust_vlines	If TRUE, adjusts vertical lines (as are drawn for quantile lines, for example) to align with the point cloud.
seed	Random seed. See position_points_jitterx .

Details

The idea for this position adjustment comes from Micah Allen, who proposed this type of plot in a [blog post](#) on March 15, 2018.

See Also

Other position adjustments for ridgeline plots: [position_points_jitterx](#), [position_points_sinax](#)

position_stackx	<i>Stack overlapping objects which can be shifted vertically or horizontally</i>
-----------------	--

Description

Stack overlapping objects which can be shifted vertically or horizontally

Usage

```
position_stackx(vjust = 1, hexpand = NA, vexpand = NA, reverse = FALSE)
```

Arguments

vjust	Vertical adjustment for geoms that have a position (like points or lines), not a dimension (like bars or areas). Set to 0 to align with the bottom, 0.5 for the middle, and 1 (the default) for the top.
hexpand	numeric, distance to be shifted horizontally for geoms that have a position, default is NA.
vexpand	numeric, distance to be shifted vertically for geoms that have a position, default is NA.
reverse	If TRUE, will reverse the default stacking order. This is useful if you're rotating both the plot and legend.

Value

position method.

Author(s)

Shuangbin Xu

Examples

```
library(ggplot2)
library(patchwork)
df <- data.frame(trt = c("a", "b", "c"), outcome = c(2.3, 1.9, 3.2))
#
p1 <- ggplot(df, aes(x=trt, y=outcome)) +
  geom_bar(stat="identity",
           position=position_stackx())

p2 <- ggplot(df, aes(x=trt, y=outcome)) +
  geom_bar(stat="identity",
           position=position_stackx(vexpand=5))

p3 <- ggplot(df, aes(x=outcome, y=trt)) +
  geom_bar(stat="identity",
           orientation="y",
           position=position_stackx(hexpand=5))
p <- p1 + p2 + p3
p
```

Index

* datasets

PositionDodgex, 7

* position adjustments

position_dodgex, 7

position_identityx, 8

position_points_sinx, 10

fruit_plot (geom_fruit), 2

geom_fruit, 2

geom_fruit_list, 5

position_dodgex, 7, 9, 11

position_dodgex2 (position_dodgex), 7

position_identityx, 8, 8, 11

position_jitter(), 9

position_points_jitterx, 9, 12

position_points_jitterx(), 11

position_points_sinx, 8–10, 10, 12

position_raincloudx, 10, 11

position_stackx, 12

PositionDodgex, 7

PositionDodgex2 (PositionDodgex), 7

PositionIdentityx (PositionDodgex), 7

PositionPointsJitterx (PositionDodgex),
7

PositionPointsSinax (PositionDodgex), 7

PositionRaincloudx (PositionDodgex), 7

PositionStackx (PositionDodgex), 7