

# Package ‘ggtree’

October 14, 2021

**Type** Package

**Title** an R package for visualization of tree and annotation data

**Version** 3.0.4

**Maintainer** Guangchuang Yu <guangchuangyu@gmail.com>

**Description** 'ggtree' extends the 'ggplot2' plotting system which implemented the grammar of graphics. 'ggtree' is designed for visualization and annotation of phylogenetic trees and other tree-like structures with their annotation data.

**Depends** R (>= 3.5.0)

**Imports** ape, aplot (>= 0.0.4), dplyr, ggfun, ggplot2 (>= 3.0.0), grid, magrittr, methods, purrr, rlang, scales, tidyr, tidytree (>= 0.2.6), treeio (>= 1.8.0), utils, yulab.utils

**Suggests** emojiFont, ggimage, ggplotify, grDevices, knitr, prettydoc, rmarkdown, stats, testthat, tibble

**Remotes** GuangchuangYu/treeio

**VignetteBuilder** knitr

**ByteCompile** true

**Encoding** UTF-8

**License** Artistic-2.0

**URL** <https://yulab-smu.top/treedata-book/>

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

**biocViews** Alignment, Annotation, Clustering, DataImport, MultipleSequenceAlignment, Phylogenetics, ReproducibleResearch, Software, Visualization

**RoxygenNote** 7.1.1

**Roxygen** list(markdown = TRUE)

**git\_url** <https://git.bioconductor.org/packages/ggtree>

**git\_branch** RELEASE\_3\_13

**git\_last\_commit** 7a83be2

**git\_last\_commit\_date** 2021-08-20

**Date/Publication** 2021-10-14

**Author** Guangchuang Yu [aut, cre, cph]  
 (<<https://orcid.org/0000-0002-6485-8781>>),  
 Tommy Tsan-Yuk Lam [aut, ths],  
 Shuangbin Xu [aut] (<<https://orcid.org/0000-0003-3513-5362>>),  
 Yonghe Xia [ctb],  
 Justin Silverman [ctb],  
 Bradley Jones [ctb],  
 Watal M. Iwasaki [ctb],  
 Ruizhu Huang [ctb]

## R topics documented:

add_colorbar . . . . .	4
applyLayoutDaylight . . . . .	5
as.polytomy . . . . .	5
collapse.ggtree . . . . .	6
Date2decimal . . . . .	6
decimal2Date . . . . .	7
expand . . . . .	8
facet_data . . . . .	8
facet_labeller . . . . .	9
facet_plot . . . . .	10
facet_widths . . . . .	11
flip . . . . .	11
geom_aline . . . . .	12
geom_balance . . . . .	13
geom_cladelab . . . . .	14
geom_cladelabel . . . . .	17
geom_cladelabel2 . . . . .	19
geom_hilight . . . . .	20
geom_inset . . . . .	22
geom_label2 . . . . .	23
geom_motif . . . . .	24
geom_nodelab . . . . .	25
geom_nodelab2 . . . . .	26
geom_nodepoint . . . . .	27
geom_point2 . . . . .	28
geom_range . . . . .	30
geom_rootedge . . . . .	31
geom_rootpoint . . . . .	32
geom_segment2 . . . . .	34
geom_strip . . . . .	35
geom_taxalink . . . . .	36
geom_text2 . . . . .	37
geom_tiplab . . . . .	39
geom_tiplab2 . . . . .	41

geom_tippoint . . . . .	42
geom_tree . . . . .	43
geom_tree2 . . . . .	44
geom_treescale . . . . .	44
geom_zoom_clade . . . . .	45
get.path . . . . .	46
getNodeAngle.df . . . . .	46
getNodesBreadthFirst.df . . . . .	47
getSubtree . . . . .	47
getSubtree.df . . . . .	48
getSubtreeUnrooted . . . . .	48
getSubtreeUnrooted.df . . . . .	49
getTreeArcAngles . . . . .	49
get_clade_position . . . . .	50
get_heatmap_column_position . . . . .	50
get_taxa_name . . . . .	51
ggdensitree . . . . .	52
ggtree . . . . .	53
gheatmap . . . . .	55
gzoom . . . . .	57
gzoom.phylo . . . . .	58
hexpand . . . . .	58
identify.gg . . . . .	59
label_pad . . . . .	60
layoutDaylight . . . . .	61
layoutEqualAngle . . . . .	61
layout_rectangular . . . . .	62
msaplot . . . . .	63
multiplot . . . . .	64
nodebar . . . . .	65
nodepie . . . . .	65
open_tree . . . . .	66
range_format . . . . .	66
revts . . . . .	67
rotate . . . . .	67
rotateTreePoints.df . . . . .	68
rotate_tree . . . . .	68
scaleClade . . . . .	69
scale_color . . . . .	70
scale_x_ggtree . . . . .	70
scale_x_range . . . . .	71
set_highlight_legend . . . . .	71
StatBalance . . . . .	72
td_filter . . . . .	72
td_unnest . . . . .	73
theme_dendrogram . . . . .	74
theme_inset . . . . .	74
theme_tree . . . . .	75

theme_tree2 . . . . .	75
viewClade . . . . .	76
xlim_expand . . . . .	77
xlim_tree . . . . .	77
zoomClade . . . . .	78
%+>% . . . . .	78
%<+% . . . . .	79
%<% . . . . .	80

**Index** **81**

---

add_colorbar	<i>add_colorbar</i>
--------------	---------------------

---

**Description**

add\_colorbar legend

**Usage**

add\_colorbar(p, color, x = NULL, ymin = NULL, ymax = NULL, font.size = 4)

**Arguments**

p	tree view
color	output of scale_color function
x	x position
ymin	ymin
ymax	ymax
font.size	font size

**Value**

ggplot2 object

**Author(s)**

Guangchuang Yu

---

applyLayoutDaylight    *applyLayoutDaylight*

---

**Description**

Apply the daylight algorithm to adjust the spacing between the subtrees and tips of the specified node.

**Usage**

```
applyLayoutDaylight(df, node_id)
```

**Arguments**

df	tree data.frame
node_id	is id of the node from which daylight is measured to the other subtrees.

**Value**

list with tree data.frame with updated layout using daylight algorithm and max\_change angle.

---

as.polytomy    *as.polytomy*

---

**Description**

collapse binary tree to polytomy by applying 'fun' to 'feature'

**Usage**

```
as.polytomy(tree, feature, fun)
```

**Arguments**

tree	tree object
feature	selected feature
fun	function to select nodes to collapse

**Value**

polytomy tree

**Author(s)**

Guangchuang Yu

---

collapse.ggtree      *collapse-ggtree*

---

**Description**

collapse a clade

**Usage**

```
## S3 method for class 'ggtree'  
collapse(x = NULL, node, mode = "none", clade_name = NULL, ...)
```

**Arguments**

x	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node	internal node number
mode	one of 'none', 'max', 'min' and 'mixed'
clade_name	set clade name. If clade_name = NULL, do nothing
...	additional parameters

**Value**

tree view

**Author(s)**

Guangchuang Yu

**See Also**

expand

---

Date2decimal      *Date2decimal*

---

**Description**

convert Date to decimal format, eg "2014-05-05" to "2014.34"

**Usage**

```
Date2decimal(x)
```

**Arguments**

x                      Date

**Value**

numeric

**Author(s)**

Guangchuang Yu

---

*decimal2Date*                      *decimal2Date*

---

**Description**

convert decimal format to Date, eg "2014.34" to "2014-05-05"

**Usage**

`decimal2Date(x)`

**Arguments**

x                      numerical number, eg 2014.34

**Value**

Date

**Author(s)**

Guangchuang Yu

---

expand	<i>expand</i>
--------	---------------

---

**Description**

expand collased clade

**Usage**

```
expand(tree_view = NULL, node)
```

**Arguments**

tree_view	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node	internal node number to specify a clade. If NULL, using the whole tree

**Value**

tree view

**Author(s)**

Guangchuang Yu

**See Also**

collapse

---

facet_data	<i>facet_data</i>
------------	-------------------

---

**Description**

extract data used in facet\_plot or geom\_facet

**Usage**

```
facet_data(tree_view, panel)
```

**Arguments**

tree_view	ggtree object
panel	data plotted in specific panel. If only one dataset used in the panel, return the data frame, else return a list of data frames.



**Value**

data frame or a list of data frames

**Author(s)**

Guangchuang Yu

**References**

G Yu, TTY Lam, H Zhu, Y Guan (2018). Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>

---

facet_labeller	<i>facet_labeller</i>
----------------	-----------------------

---

**Description**

label facet\_plot output

**Usage**

```
facet_labeller(p, label)
```

**Arguments**

p	facet_plot output
label	labels of facet panels

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

facet_plot	<i>facet_plot</i>
------------	-------------------

---

### Description

plot tree associated data in an additional panel

### Usage

```
facet_plot(p, mapping = NULL, data, geom, panel, ...)
```

```
geom_facet(mapping = NULL, data, geom, panel, ...)
```

### Arguments

p	tree view
mapping	aes mapping for 'geom'
data	data to plot by 'geom', first column should be matched with tip label of tree
geom	geom function to plot the data
panel	panel name for plot of input data
...	additional parameters for 'geom'

### Details

'facet\_plot()' 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. 'geom\_facet' is a 'ggplot2' layer version of 'facet\_plot'

### Value

ggplot object

### Author(s)

Guangchuang Yu

### References

G Yu, TTY Lam, H Zhu, Y Guan (2018). Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>

**Examples**

```
tr <- rtree(10)
dd = data.frame(id=tr$tip.label, value=abs(rnorm(10)))
p <- ggtree(tr)
facet_plot(p, 'Trait', data = dd, geom=geom_point, mapping=aes(x=value))
```

---

facet_widths	<i>facet_widths</i>
--------------	---------------------

---

**Description**

set relative widths (for column only) of facet plots

**Usage**

```
facet_widths(p, widths)
```

**Arguments**

p	ggplot or ggtree object
widths	relative widths of facet panels

**Value**

ggplot object by redrawing the figure (not a modified version of input object)

**Author(s)**

Guangchuang Yu

---

flip	<i>flip</i>
------	-------------

---

**Description**

flip position of two selected branches

**Usage**

```
flip(tree_view = NULL, node1, node2)
```

**Arguments**

tree_view	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node1	node number of branch 1
node2	node number of branch 2

**Value**

ggplot2 object

**Author(s)**

Guangchuang Yu

---

geom\_aline

*geom\_aline*

---

**Description**

add horizontal align lines

**Usage**

```
geom_aline(mapping = NULL, linetype = "dotted", size = 1, ...)
```

**Arguments**

mapping	aes mapping
linetype	line type
size	line size
...	additional parameter

**Value**

aline layer

**Author(s)**

Yu Guangchuang

---

geom_balance	<i>geom_balance</i>
--------------	---------------------

---

### Description

highlights the two direct descendant clades of an internal node

### Usage

```
geom_balance(  
  node,  
  fill = "steelblue",  
  color = "white",  
  alpha = 0.5,  
  extend = 0,  
  extendto = NULL  
)
```

### Arguments

node	selected node (balance) to highlight
fill	color fill
color	color to outline highlights and divide balance
alpha	alpha (transparency)
extend	extend xmax of the rectangle
extendto	extend xmax to extendto

### Details

Particularly useful when studying neighboring clades. Note that balances that correspond to multi-chotomies will not be displayed.

### Value

ggplot2

### Author(s)

Justin Silverman and modified by Guangchuang Yu

### References

J. Silverman, et al. *A phylogenetic transform enhances analysis of compositional microbiota data.* (in preparation)

---

geom\_cladelab                      *geom\_cladelab*

---

## Description

annotate a clade with bar and text label or (image)

## Usage

```
geom_cladelab(
  node = NULL,
  label = NULL,
  data = NULL,
  mapping = NULL,
  geom = "text",
  parse = FALSE,
  ...
)
```

## Arguments

node	selected node to annotate, when data and mapping is NULL, it is required.
label	character, character to be showed, when data and mapping is NULL, it is required.
data	data.frame, the data to be displayed in the annotation, default is NULL.
mapping	Set of aesthetic mappings, default is NULL. The detail see the following explanation.
geom	character, one of 'text', 'label', 'shadowtext', 'image' and 'phylopic', default is 'text', and the parameter see the Aesthetics For Specified Geom.
parse	logical, whether parse label to emoji font, default is FALSE.
...	additional parameters, see also following section.

additional parameters can refer the following parameters.

- `offset` distance bar and tree, offset of bar and text from the clade, default is 0.
- `offset.text` distance bar and text, offset of text from bar, default is 0.
- `align` logical, whether align clade lab, default is FALSE.
- `extend` numeric, extend the length of bar, default is 0.
- `angle` numeric or 'auto', if angle is auto, the angle of text will be calculated automatically, which is useful for the circular etc layout, default is 0.
- `horizontal` logical, whether set label to horizontal, default is TRUE.
- `barwidth` the width of line, default is 0.5.
- `barcolour` the colour of line, default is 'black'.
- `fontsize` the size of text, default is 3.88.

- `textcolour` the colour of text, default is 'black'.
- `imagesize` the size of image, default is 0.05.
- `imagecolor` the colour of image, default is NULL, when `geom="phylopic"`, it should be required.

The parameters also can be set in mapping, when data is provided. Note: the `barsize`, `barcolour`, `fontsize`, `textcolour`, `imagesize` and `imagecolor` should not be set in mapping (aesthetics). When the color and size are not be set in mapping, user can modify them to adjust the attributes of specified geom.

### Aesthetics For Specified Geom

`geom_cladelab()` understands the following aesthetics for `geom="text"`(required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **label** labels showed, it is required.
- `colour` the colour of text, default is "black".
- `size` the size of text, default is 3.88.
- `angle` the angle of text, default is 0.
- `hjust` A numeric vector specifying horizontal justification, default is 0.
- `vjust` A numeric vector specifying vertical justification, default is 0.5.
- `alpha` the transparency of text, default is NA.
- `family` the family of text, default is 'sans'.
- `fontface` the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- `lineheight` The height of a line as a multiple of the size of text, default is 1.2 .

when the colour, size are not be set in mapping, and user want to modify the colour of text, they should use `textcolour`, `fontsize` to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthethics for `geom="label"` (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **label** labels to be showed, it is required.
- `colour` the colour of text, default is "black".
- `fill` the background colour of the label, default is "white".
- `size` the size of text, default is 3.88.
- `angle` the angle of text, default is 0.
- `hjust` A numeric vector specifying horizontal justification, default is 0.
- `vjust` A numeric vector specifying vertical justification, default is 0.5.
- `alpha` the transparency of text, default is NA.
- `family` the family of text, default is 'sans'.
- `fontface` the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).

- `lineheight` The height of a line as a multiple of the size of text, default is 1.2 .

when the colour, size are not be set in mapping, and user want to modify the colour of text, they should use `textcolour`, `fontsize` to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthetics for `geom="shadowtext"` (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **label** labels to be showed, it is required.
- `colour` the colour of text, default is "black".
- `bg.colour` the background colour of text, default is 'black'.
- `bg.r` the width of background text, default is 0.1.
- `size` the size of text, default is 3.88.
- `angle` the angle of text, default is 0.
- `hjust` A numeric vector specifying horizontal justification, default is 0.
- `vjust` A numeric vector specifying vertical justification, default is 0.5.
- `alpha` the transparency of text, default is NA.
- `family` the family of text, default is 'sans'.
- `fontface` the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- `lineheight` The height of a line as a multiple of the size of text, default is 1.2 .

when the colour, size are not be set in mapping, and user want to modify the colour of text, they should use `textcolour`, `fontsize` to avoid the confusion with bar layer annotation.

`geom_cladelab()` understands the following aesthetics for `geom="image"` or `geom="phylopic"` (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **label** labels to be showed, it is required.
- **image** the image to be annotated, when `geom="phylopic"`, the uid of phylopic databases, it is required.
- `colour` the color of image, default is NULL.
- `size` the size of image, default is 0.05.
- `alpha` the alpha of image, default is 0.8.

when the colour, size are not be set in mapping, and user want to modify the colour of image, they should use `imagecolour`, `imagesize` to avoid the confusion with bar layer annotation.

## Examples

```
set.seed(2015-12-21)
tree <- rtree(30)
data <- data.frame(id=c(34, 56),
                   annotate=c("another clade", "long clade names"),
                   image=c("7fb9bea8-e758-4986-afb2-95a2c3bf983d",
                           "0174801d-15a6-4668-bfe0-4c421f5e51e8"),
```



```

      group=c("A", "B"),
      offset=c(0.1, 0.1),
      offset.text=c(0.1, 0.2))

p <- ggtree(tree) + xlim(NA, 6)

p + geom_cladelab(node=45, label="test label") +
  geom_cladelab(node=34, label="another clade")
p2 <- p + geom_cladelab(data=data,
  mapping=aes(
    node=id,
    label=annotate,
    image=image,
    color=group,
    offset=offset,
    offset.text=offset.text),
  geom="shadowtext",
  hjust=0.5,
  align=TRUE,
  horizontal=FALSE,
  angle=90
)
p2

```

---

 geom\_cladelabel

*geom\_cladelabel*


---

## Description

annotate a clade with bar and text label

## Usage

```

geom_cladelabel(
  node,
  label,
  offset = 0,
  offset.text = 0,
  extend = 0,
  align = FALSE,
  barsize = 0.5,
  fontsize = 3.88,
  angle = 0,
  geom = "text",
  hjust = 0,
  color = NULL,
  fill = NA,
  family = "sans",
  parse = FALSE,

```

```

    horizontal = TRUE,
    ...
)

```

### Arguments

node	selected node
label	clade label
offset	offset of bar and text from the clade
offset.text	offset of text from bar
extend	extend bar height
align	logical
barsize	size of bar
fontsize	size of text
angle	angle of text
geom	one of 'text' or 'label'
hjust	justify text horizontally
color	color for clade & label, of length 1 or 2
fill	fill label background, only work with geom='label'
family	sans by default, can be any supported font
parse	logical, whether parse label
horizontal	logical, whether set label to horizontal, default is TRUE.
...	additional parameter

### Value

ggplot layers

### Author(s)

Guangchuang Yu

### See Also

[geom\\_cladelabel2](#)

---

geom\_cladelabel2      *geom\_cladelabel2*

---

### Description

annotate a clade with bar and text label

### Usage

```
geom_cladelabel2(
  node,
  label,
  offset = 0,
  offset.text = 0,
  offset.bar = 0,
  align = FALSE,
  barsize = 0.5,
  fontsize = 3.88,
  hjust = 0,
  geom = "text",
  color = NULL,
  family = "sans",
  parse = FALSE,
  horizontal = TRUE,
  ...
)
```

### Arguments

node	selected node
label	clade label
offset	offset of bar and text from the clade
offset.text	offset of text from bar
offset.bar	offset of bar from text
align	logical
barsize	size of bar
fontsize	size of text
hjust	justify text horizontally
geom	one of 'text' or 'label'
color	color for clade & label, of length 1 or 2
family	sans by default, can be any supported font
parse	logical, whether parse label
horizontal	logical, whether set label to horizontal, default is TRUE.
...	additional parameter

**Value**

ggplot layers

**Author(s)**

JustGitting

**See Also**

[geom\\_cladelabel](#)

---

geom\_highlight

*geom\_highlight*

---

**Description**

layer of highlight clade

**Usage**

```
geom_highlight(data = NULL, mapping = NULL, node = NULL, type = "auto", ...)
```

```
geom_highlight(data = NULL, mapping = NULL, node = NULL, type = "auto", ...)
```

**Arguments**

data	data.frame, The data to be displayed in this layer, default is NULL.
mapping	Set of aesthetic mappings, default is NULL.
node	selected node to highlight, when data and mapping is NULL, it is required.
type	the type of layer, default is auto, meaning rectangular, circular, slanted, fan, inward_circular, radial, equal_angle, ape layout tree will use rectangular layer, unrooted and daylight layout tree use will use encircle layer. You can specify this parameter to rect (rectangular layer) or encircle (encircle layer).
...	additional parameters, see also Aesthetics section.

**Details**

geom\_highlight supports data.frame as input. And aesthetics of layer can be mapped. you can see the Aesthetics section to set parameters.

**Value**

a list object.

**Aesthetics**

geom\_highlight() understands the following aesthetics for rectangular layer (required aesthetics are in bold):

- **node** selected node to highlight, it is required.
- **colour** the colour of margin, default is NA.
- **fill** the colour of fill, default is 'steelblue'.
- **alpha** the transparency of fill, default is 0.5.
- **extend** extend xmax of the rectangle, default is 0.
- **extendto** specify a value, meaning the rectangle extend to, default is NULL.
- **linetype** the line type of margin, default is 1.
- **size** the width of line of margin, default is 0.5.

geom\_highlight() understands the following aesthetics for encircle layer (required aesthetics are in bold):

- **node** selected node to highlight, it is required.
- **colour** the colour of margin, default is 'black'.
- **fill** the colour of fill, default is 'steelblue'.
- **alpha** the transparency of fill, default is 0.5.
- **expand** expands the xspline clade region, default is 0.
- **spread** control the size, when only one point.
- **size** the width of line of margin, default is 0.5.
- **linetype** the line type of margin, default is 1.
- **s\_shape** the shape of the spline relative to the control points, default is 0.5.
- **s\_open** whether the spline is a line or a closed shape, default is FALSE.

**Author(s)**

Guangchuang Yu and Shuangbin Xu

**Examples**

```
library(ggplot2)
set.seed(102)
tree <- rtree(60)
p <- ggtree(tree)
p1 <- p + geom_highlight(node=62) + geom_highlight(node=88, fill="red")
p1
dat <- data.frame(id=c(62, 88), type=c("A", "B"))
p2 <- p + geom_highlight(data=dat, mapping=aes(node=id, fill=type))
p2
```

geom\_inset

*geom\_inset***Description**

add subplots to tree  
 add insets in a tree

**Usage**

```
geom_inset(
  insets,
  width = 0.1,
  height = 0.1,
  hjust = 0,
  vjust = 0,
  x = "node",
  reverse_x = FALSE,
  reverse_y = FALSE
)
```

```
inset(
  tree_view,
  insets,
  width,
  height,
  hjust = 0,
  vjust = 0,
  x = "node",
  reverse_x = FALSE,
  reverse_y = FALSE
)
```

**Arguments**

<code>insets</code>	a list of ggplot objects, named by node number
<code>width</code>	width of inset, relative to the range of x-axis
<code>height</code>	height of inset, relative to the range of y-axis
<code>hjust</code>	horizontal adjustment
<code>vjust</code>	vertical adjustment
<code>x</code>	x position, one of 'node' and 'branch'
<code>reverse_x</code>	whether x axis was reversed by <code>scale_x_reverse</code>
<code>reverse_y</code>	whether y axis was reversed by <code>scale_y_reverse</code>
<code>tree_view</code>	tree view

**Value**

inset layer  
tree view with insets

**Author(s)**

Guangchuang Yu  
Guangchuang Yu

---

geom_label2	<i>geom_label2</i>
-------------	--------------------

---

**Description**

geom\_label2 support aes(subset) via setup\_data

**Usage**

```
geom_label2(
  mapping = NULL,
  data = NULL,
  ...,
  stat = "identity",
  position = "identity",
  family = "sans",
  parse = FALSE,
  nudge_x = 0,
  nudge_y = 0,
  label.padding = unit(0.25, "lines"),
  label.r = unit(0.15, "lines"),
  label.size = 0.25,
  na.rm = TRUE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

**Arguments**

mapping	the aesthetic mapping
data	A layer specific dataset - only needed if you want to override the plot defaults.
...	other arguments passed on to 'layer'
stat	Name of stat to modify data
position	The position adjustment to use for overlapping points on this layer
family	sans by default, can be any supported font

parse	if TRUE, the labels will be parsed as expressions
nudge_x	horizontal adjustment
nudge_y	vertical adjustment
label.padding	Amount of padding around label.
label.r	Radius of rounded corners.
label.size	Size of label border, in mm
na.rm	logical
show.legend	logical
inherit.aes	logical

**Value**

label layer

**Author(s)**

Guangchuang Yu

**See Also**

[geom\\_label](#)

---

geom\_motif

*geom\_motif*

---

**Description**

geom layer to draw aligned motif

**Usage**

```
geom_motif(mapping, data, on, label, align = "centre", ...)
```

**Arguments**

mapping	aes mapping
data	data
on	gene to center (i.e. set middle position of the on gene to 0)
label	specify a column to be used to label genes
align	where to place gene label, default is 'centre' and can be set to 'left' and 'right'
...	additional parameters

**Value**

geom layer



**Author(s)**

Guangchuang Yu

---

`geom_nodelab`*geom\_nodelab*

---

**Description**

add node label layer

**Usage**

```
geom_nodelab(  
  mapping = NULL,  
  nudge_x = 0,  
  nudge_y = 0,  
  geom = "text",  
  hjust = 0.5,  
  ...  
)
```

**Arguments**

<code>mapping</code>	aes mapping
<code>nudge_x</code>	horizontal adjustment to nudge label
<code>nudge_y</code>	vertical adjustment to nudge label
<code>geom</code>	one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'
<code>hjust</code>	horizontal alignment, one of 0, 0.5 or 1
<code>...</code>	additional parameters, see also the additional parameters of <a href="#">geom_tiplab()</a> .

**Value**

geom layer

**Author(s)**

Guangchuang Yu

**See Also**[geom\\_tiplab\(\)](#)

---

geom\_nodelab2      *geom\_nodelab2*

---

### Description

add node label for circular layout

### Usage

```
geom_nodelab2(  
  mapping = NULL,  
  nudge_x = 0,  
  nudge_y = 0,  
  geom = "text",  
  hjust = 0.5,  
  ...  
)
```

### Arguments

mapping	aes mapping
nudge_x	horizontal adjustment to nudge label
nudge_y	vertical adjustment to nudge label
geom	one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'
hjust	horizontal alignment, one of 0, 0.5 or 1
...	additional parameters, see also the additional parameters of <a href="#">geom_tiplab()</a> .

### Value

node label layer

### Author(s)

Guangchuang Yu

---

geom_nodepoint	<i>geom_nodepoint</i>
----------------	-----------------------

---

## Description

add node point

## Usage

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

## Arguments

mapping	Set of aesthetic mapping created by <code>aes()</code> . If <code>inherit.aes = TRUE</code> , the mapping can be inherited from the plot mapping as specified in the call to <code>ggplot()</code> .
data	The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to 'ggplot()',
position	Position adjustment.
na.rm	logical. If 'FALSE' (the default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.
inherit.aes	logical (default is 'TRUE'). If 'FALSE', overrides the default aesthetics, rather than combining with them.
...	additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> .

## Value

node point layer

## Author(s)

Guangchuang Yu

geom\_point2

*geom\_point2***Description**

geom\_point2 is a modified version of geom\_point that supports aes(subset)

**Usage**

```
geom_point2(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

**Arguments**

mapping	Set of aesthetic mapping created by aes(). If inherit.aes = TRUE, the mapping can be inherited from the plot mapping as specified in the call to ggplot().
data	The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to 'ggplot()',
stat	Name of the statistical transformation to be used on the data for this layer.
position	Position adjustment.
na.rm	logical. If 'FALSE' (the default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.
inherit.aes	logical (default is 'TRUE'). If 'FALSE', overrides the default aesthetics, rather than combining with them.
...	additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3.

**Details**

geom\_point2 creates scatterplots, just similar to ggplot2::geom\_point. It extends the ggplot2::geom\_point to support filtering via the subset aesthetic mapping (see Aesthetics section).

**Value**

point layer

## Aesthetics

geom\_point2() understands the following aesthetics

- subset logical expression indicating elements or rows to keep: missing values are taken as false; should be in aes().
- colour the colour of point, default is black.
- fill the colour of fill, default is black.
- alpha the transparency of fill, default is 1.
- size the size of point, default is 1.5.
- shape specify a shape, default is 19.
- stroke control point border thickness of point, default is 0.5.

## Author(s)

Guangchuang Yu

## References

1. G Yu, DK Smith, H Zhu, Y Guan, TTY Lam (2017). ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution*, 8(1):28-36. <https://doi.org/10.1111/2041-210X.12628>
2. G Yu\*, TTY Lam, H Zhu, Y Guan\*. Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 2018, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>
3. G Yu. Using ggtree to visualize data on tree-like structures. *Current Protocols in Bioinformatics*, 2020, 69:e96. <https://doi.org/10.1002/cpbi.96>

For more information about tree visualization, please refer to the online book <https://yulab-smu.top/treedata-book/>

## See Also

[geom\\_point](#); [geom\\_rootpoint](#) add point of root; [geom\\_nodepoint](#) add points of internal nodes; [geom\\_tippoint](#) add points of external nodes (also known as tips or leaves).

## Examples

```
library(ggtree)
## add point by aes(subset)
tr <- rtree(10)
# group tip and node
ggtree(tr) + geom_point2(aes(shape=isTip, color=isTip), size=3)
# specify a node to display
ggtree(tr) + geom_point2(aes(subset=(node==15)), shape=21, size=5, fill='green')
# specify a tip to display
ggtree(tr) + geom_point2(aes(subset=(label %in% c("t1", "t3"))), shape=21, size=5, fill='green')

## color point with continuous variables
```

```
library(ggtree)
library(treeio)
library(ggplot2)
beast_file <- system.file("examples/MCC_FluA_H3.tree", package="ggtree")
beast_tree <- read.beast(beast_file)
p <- ggtree(beast_tree) +
  geom_tiplab(hjust = -.1)+
  geom_nodepoint(aes(fill = rate), shape = 21, size = 4) +
  scale_fill_continuous(low = 'blue', high = 'red') +
  theme_tree2() + theme(legend.position = 'right')
p
```

---

geom\_range

*geom\_range*

---

## Description

bar of range (HPD, range etc) to present uncertainty of evolutionary inference

## Usage

```
geom_range(range, center = "auto", ...)
```

## Arguments

range	range, e.g. "height_0.95_HPD"
center	center of the range, mean, median or auto (default, the center of the range)
...	additional parameter, e.g. color, size, alpha

## Value

ggplot layer

## Author(s)

Guangchuang Yu

---

geom_rootedge	<i>geom_rootedge</i>
---------------	----------------------

---

### Description

display root edge

### Usage

```
geom_rootedge(rootedge = NULL, ...)
```

### Arguments

rootedge	length of rootedge; use phylo\$root.edge if rootedge = NULL (by default).
...	additional parameters

Additional parameters can be referred to the following parameters:

- size control the width of rootedge, default is 0.5.
- colour color of rootedge, default is black.
- linetype the type of line, default is 1.
- alpha modify colour transparency, default is 1.

### Details

geom\_rootedge is used to create a rootedge.

### Value

ggtree rootedge layer

### Author(s)

Guangchuang Yu

### References

1. G Yu, DK Smith, H Zhu, Y Guan, TTY Lam (2017). ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution*, 8(1):28-36. <https://doi.org/10.1111/2041-210X.12628>

### Examples

```
library(ggtree)
set.seed(123)
## with root edge = 1
tree1 <- read.tree(text='((A:1,B:2):3,C:2):1;')
ggtree(tree1) + geom_tiplab() + geom_rootedge()
```

```
## without root edge
tree2 <- read.tree(text='((A:1,B:2):3,C:2);')
ggtree(tree2) + geom_tiplab() + geom_rootedge()

## setting root edge
tree2$root.edge <- 2
ggtree(tree2) + geom_tiplab() + geom_rootedge()

## specify length of root edge for just plotting
## this will ignore tree$root.edge
ggtree(tree2) + geom_tiplab() + geom_rootedge(rootedge = 3)

## For more information about tree visualization, please refer to the online book
## https://yulab-smu.top/treedata-book/chapter4.html
```

---

geom\_rootpoint      *geom\_rootpoint*

---

## Description

geom\_rootpoint is used to add root point

## Usage

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

## Arguments

mapping	Set of aesthetic mapping created by aes(). If inherit.aes = TRUE, the mapping can be inherited from the plot mapping as specified in the call to ggplot().
data	The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to 'ggplot()',
position	Position adjustment.
na.rm	logical. If 'FALSE' (the default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.



`inherit.aes` logical (default is 'TRUE'). If 'FALSE', overrides the default aesthetics, rather than combining with them.

... additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`.

### Details

`geom_rootpoint` inherits from `geom_point2`, it is used to display and customize the points on the root

### Value

root point layer

### Author(s)

Guangchuang Yu

### References

1. Guangchuang Yu. Using `ggtree` to visualize data on tree-like structures. *Current Protocols in Bioinformatics*, 2020, 69:e96. doi:10.1002/cpbi.96
2. Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two methods for mapping and visualizing associated data on phylogeny using `ggtree`. *Molecular Biology and Evolution* 2018, 35(12):3041-3043. doi:10.1093/molbev/msy194
3. Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. `ggtree`: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution* 2017, 8(1):28-36. doi:10.1111/2041-210X.12628

For more information, please refer to the online book: Data Integration, Manipulation and Visualization of Phylogenetic Trees. <http://yulab-smu.top/treedata-book/>

### See Also

[geom\\_point](#); [geom\\_rootpoint](#) add point of root; [geom\\_nodepoint](#) add points of internal nodes; [geom\\_tippoint](#) add points of external nodes (also known as tips or leaves).

### Examples

```
library(ggtree)
tr <- rtree(10)
## add root point
ggtree(tr) + geom_rootpoint()
ggtree(tr) + geom_rootpoint(size=2, color="red", shape=2)
```

---

geom\_segment2                    *geom\_segment2*

---

### Description

geom\_segment2 support aes(subset) via setup\_data

### Usage

```
geom_segment2(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  nudge_x = 0,
  arrow = NULL,
  arrow.fill = NULL,
  ...
)
```

### Arguments

mapping	aes mapping
data	data
stat	Name of stat to modify data
position	position
lineend	lineend
na.rm	logical
show.legend	logical
inherit.aes	logical
nudge_x	horizontal adjustment of x
arrow	specification for arrow heads, as created by arrow().
arrow.fill	fill color to use for the arrow head (if closed). NULL means use colour aesthetic.
...	additional parameter

### Value

add segment layer

**Author(s)**

Guangchuang Yu

**See Also**[geom\\_segment](#)

---

*geom\_strip**geom\_strip*

---

**Description**

annotate associated taxa (from taxa1 to taxa2, can be Monophyletic, Polyphyletic or Paraphyletic Taxa) with bar and (optional) text label

**Usage**

```
geom_strip(
  taxa1,
  taxa2,
  label,
  offset = 0,
  offset.text = 0,
  align = TRUE,
  barsize = 0.5,
  extend = 0,
  fontsize = 3.88,
  angle = 0,
  geom = "text",
  hjust = 0,
  color = "black",
  fill = NA,
  family = "sans",
  parse = FALSE,
  ...
)
```

**Arguments**

taxa1	taxa1
taxa2	taxa2
label	optional label
offset	offset of bar and text from the clade
offset.text	offset of text from bar
align	logical

barsize	size of bar
extend	extend bar vertically
fontsize	size of text
angle	angle of text
geom	one of 'text' or 'label'
hjust	hjust
color	color for bar and label
fill	fill label background, only work with geom='label'
family	sans by default, can be any supported font
parse	logical, whether parse label
...	additional parameter

**Value**

ggplot layers

**Author(s)**

Guangchuang Yu

---

geom_taxalink	<i>link between taxa</i>
---------------	--------------------------

---

**Description**

geom\_taxalink supports data.frame as input, the colour, size, linetype and alpha can be mapped. When the data was provided, the mapping should be also provided, which taxa1 and taxa2 should be mapped created by aes, aes\_ or aes\_string. In addition, the hratio, control the height of curve line, when tree layout is circular, default is 1. ncp, the number of control points used to draw the curve, more control points creates a smoother curve, default is 1. They also can be mapped to a column of data.

**Usage**

```
geom_taxalink(
  data = NULL,
  mapping = NULL,
  taxa1 = NULL,
  taxa2 = NULL,
  offset = NULL,
  outward = "auto",
  ...
)
```

**Arguments**

data	data.frame, The data to be displayed in this layer, default is NULL.
mapping	Set of aesthetic mappings, default is NULL.
taxa1	can be label or node number.
taxa2	can be label or node number.
offset	numeric, control the shift of curve line (the ratio of axis value, range is "(0-1)"), default is NULL.
outward	logical, control the orientation of curve when the layout of tree is circular, fan or other layout in polar coordinate, default is "auto", meaning It will automatically.
...	additional parameter.

**Value**

a list object.

**Aesthetics**

geom\_taxalink() understands the following aesthetics (required aesthetics are in bold):

- taxa1 label or node number of tree.
- taxa2 label or node number of tree.
- group group category of link.
- colour control the color of line, default is black.
- linetype control the type of line, default is 1 (solid).
- size control the width of line, default is 0.5.
- curvature control the curvature of line, default is 0.5, it will be created automatically in polar coordinate .
- hratio control the height of curve line, default is 1.
- ncp control the smooth of curve line, default is 1.

---

geom\_text2

*geom\_text2*

---

**Description**

geom\_text2 support aes(subset) via setup\_data

**Usage**

```
geom_text2(  
  mapping = NULL,  
  data = NULL,  
  ...,  
  stat = "identity",  
  position = "identity",  
  family = "sans",  
  parse = FALSE,  
  na.rm = TRUE,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  nudge_x = 0,  
  nudge_y = 0,  
  check_overlap = FALSE  
)
```

**Arguments**

mapping	the aesthetic mapping
data	A layer specific dataset - only needed if you want to override the plot defaults.
...	other arguments passed on to 'layer'
stat	Name of stat to modify data
position	The position adjustment to use for overlapping points on this layer
family	sans by default, can be any supported font
parse	if TRUE, the labels will be passed into expressions
na.rm	logical
show.legend	logical
inherit.aes	logical
nudge_x	horizontal adjustment
nudge_y	vertical adjustment
check_overlap	if TRUE, text that overlaps previous text in the same layer will not be plotted

**Value**

text layer

**Author(s)**

Guangchuang Yu

**See Also**

[geom\\_text](#)

---

geom_tiplab	<i>geom_tiplab</i>
-------------	--------------------

---

**Description**

add tip label layer

**Usage**

```
geom_tiplab(
  mapping = NULL,
  hjust = 0,
  align = FALSE,
  linetype = "dotted",
  linesize = 0.5,
  geom = "text",
  offset = 0,
  as_ylab = FALSE,
  ...
)
```

**Arguments**

mapping	aes mapping
hjust	horizontal adjustment
align	align tip lab or not, logical
linetype	linetype for adding line if align = TRUE
linesize	line size of line if align = TRUE
geom	one of 'text', 'label', 'shadowtext', 'image' and 'phylopic'
offset	tiplab offset, horizontal adjustment to nudge tip labels, default is 0.
as_ylab	display tip labels as y-axis label, only works for rectangular and dendrogram layouts
...	additional parameter

additional parameters can refer the following parameters.

The following parameters for geom="text".

- size control the size of tip labels, default is 3.88.
- colour control the colour of tip labels, default is "black".
- angle control the angle of tip labels, default is 0.
- vjust A numeric vector specifying vertical justification, default is 0.5.
- alpha the transparency of text, default is NA.
- family the family of text, default is 'sans'.
- fontface the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).

- `lineheight` The height of a line as a multiple of the size of text, default is 1.2 .
- `nudge_x` horizontal adjustment to nudge labels, default is 0.
- `nudge_y` vertical adjustment to nudge labels, default is 0.
- `check.overlap` if TRUE, text that overlaps previous text in the same layer will not be plotted.
- `parse` if TRUE, the labels will be parsed into expressions, if it is 'emoji', the labels will be parsed into emoji font.

The following parameters for `geom="label"`.

- `size` the size of tip labels, default is 3.88.
- `colour` the colour of tip labels, default is "black".
- `fill` the colour of rectangular box of labels, default is "white".
- `vjust` numeric vector specifying vertical justification, default is 0.5.
- `alpha` the transparency of labels, default is NA.
- `family` the family of text, default is 'sans'.
- `fontface` the font face of text, default is 1 (plain), others are 2 (bold), 3 (italic), 4 (bold.italic).
- `lineheight` The height of a line as a multiple of the size of text, default is 1.2.
- `nudge_x` horizontal adjustment to nudge labels, default is 0.
- `nudge_y` vertical adjustment, default is 0.
- `check.overlap` if TRUE, text that overlaps previous text in the same layer will not be plotted.
- `parse` if TRUE, the labels will be parsed into expressions, if it is 'emoji', the labels will be parsed into emoji font.
- `label.padding` Amount of padding around label, default is 'unit(0.25, "lines")'.
- `label.r` Radius of rounded corners, default is 'unit(0.15, "lines")'.
- `label.size` Size of label border, in mm, default is 0.25.

The following parameters for `geom="shadowtext"`, some parameters are like to `geom="text"`.

- `bg.colour` the background colour of text, default is "black".
- `bg.r` the width of background of text, default is 0.1 .

The following parameters for `geom="image"` or `geom="phylopic"`.

- `image` the image file path for `geom='image'`, but when `geom='phylopic'`, it should be the uid of phylopic databases.
- `size` the image size, default is 0.05.
- `colour` the color of image, default is NULL.
- `alpha` the transparency of image, default is 0.8.

The following parameters for the line when `align = TRUE`.

- `colour` the colour of line, default is 'black'.
- `alpha` the transparency of line, default is NA.
- `arrow` specification for arrow heads, as created by `arrow()`, default is NULL.
- `arrow.fill` fill color to use for the arrow head (if closed), default is 'NULL', meaning use 'colour' aesthetic.



**Value**

tip label layer

**Author(s)**

Guangchuang Yu

**Examples**

```
require(ape)
tr <- rtree(10)
ggtree(tr) + geom_tiplab()
```

---

geom\_tiplab2

*geom\_tiplab2*

---

**Description**

add tip label for circular layout

**Usage**

```
geom_tiplab2(mapping = NULL, hjust = 0, ...)
```

**Arguments**

mapping	aes mapping
hjust	horizontal adjustment
...	additional parameter, see geom_tiplab

**Value**

tip label layer

**Author(s)**

Guangchuang Yu

**References**

<https://groups.google.com/forum/#!topic/bioc-ggtree/o35PV3iH0-0>

**See Also**

[geom\\_tiplab](#)

---

geom\_tippoint      *geom\_tippoint*

---

### Description

add tip point

### Usage

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

### Arguments

mapping	Set of aesthetic mapping created by <code>aes()</code> . If <code>inherit.aes = TRUE</code> , the mapping can be inherited from the plot mapping as specified in the call to <code>ggplot()</code> .
data	The data to be displayed in this layer. If 'NULL' (the default), the data is inherited from the plot data as specified in the call to 'ggplot()',
position	Position adjustment.
na.rm	logical. If 'FALSE' (the default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? 'NA', the default, includes if any aesthetics are mapped. 'FALSE' never includes, and 'TRUE' always includes.
inherit.aes	logical (default is 'TRUE'). If 'FALSE', overrides the default aesthetics, rather than combining with them.
...	additional parameters that passed on to this layer. These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> .

### Value

tip point layer

### Author(s)

Guangchuang Yu

---

geom_tree	<i>geom_tree</i>
-----------	------------------

---

**Description**

add tree layer

**Usage**

```
geom_tree(
  mapping = NULL,
  data = NULL,
  layout = "rectangular",
  multiPhylo = FALSE,
  continuous = "none",
  position = "identity",
  ...
)
```

**Arguments**

mapping	aesthetic mapping
data	data
layout	one of 'rectangular', 'dendrogram', 'slanted', 'ellipse', 'roundrect', 'fan', 'circular', 'inward_circular', 'radial', 'equal_angle', 'daylight' or 'ape'
multiPhylo	logical, whether input data contains multiple phylo class.
continuous	character, continuous transition for selected aesthetic ('size' or 'color'('colour')). It should be one of 'color' (or 'colour'), 'size', 'all' and 'none', default is 'none'
position	Position adjustment, either as a string, or the result of a call to a position adjustment function, default is "identity".
...	additional parameter some dot arguments: <ul style="list-style-type: none"> <li>• nsplit integer, the number of branch blocks divided when 'continuous' is not "none", default is 200.</li> </ul>

**Value**

tree layer

**Aesthetics**

geom\_tree() understands the following aesthetics:

- color character, control the color of line, default is black (continuous is "none").
- linetype control the type of line, default is 1 (solid).
- size numeric, control the width of line, default is 0.5 (continuous is "none").

**Author(s)**

Yu Guangchuang

---

geom_tree2	<i>geom_tree2</i>
------------	-------------------

---

**Description**

add tree layer

**Usage**

```
geom_tree2(layout = "rectangular", ...)
```

**Arguments**

layout	one of 'rectangular', 'slanted', 'circular', 'radial' or 'unrooted'
...	additional parameter

**Value**

tree layer

**Author(s)**

Yu Guangchuang

---

geom_treescale	<i>geom_treescale</i>
----------------	-----------------------

---

**Description**

add tree scale

**Usage**

```
geom_treescale(
  x = NULL,
  y = NULL,
  width = NULL,
  offset = NULL,
  offset.label = NULL,
  label = NULL,
  color = "black",
  linesize = 0.5,
  fontsize = 3.88,
  family = "sans"
)
```

**Arguments**

x	x position
y	y position
width	width of scale
offset	offset of text to line
offset.label	offset of scale title to line.
label	the title of tree scale, default is NULL.
color	color
linesize	size of line
fontsize	size of text
family	sans by default, can be any supported font

**Value**

ggplot layers

**Author(s)**

Guangchuang Yu

---

geom_zoom_clade	<i>geom_zoom_clade</i>
-----------------	------------------------

---

**Description**

zoom selected clade of a tree

**Usage**

```
geom_zoom_clade(node, xexpand = NULL)
```

**Arguments**

node	internal node number
xexpand	numeric, extend x, meaning the ratio of range of original x, default is NULL.

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

get.path	<i>get.path</i>
----------	-----------------

---

**Description**

path from start node to end node

**Usage**

```
get.path(phylo, from, to)
```

**Arguments**

phylo	phylo object
from	start node
to	end node

**Value**

node vectot

**Author(s)**

Guangchuang Yu

---

getNodeAngle.df	<i>getNodeAngle.df</i>
-----------------	------------------------

---

**Description**

Get the angle between the two nodes specified.

**Usage**

```
getNodeAngle.df(df, origin_node_id, node_id)
```

**Arguments**

df	tree data.frame
origin_node_id	origin node id number
node_id	end node id number

**Value**

angle in range [-1, 1], i.e. degrees/180, radians/pi

---

`getNodeBreadthFirst.df`  
*getNodeBreadthFirst.df*

---

**Description**

Get the nodes of tree from root in breadth-first order.

**Usage**

`getNodeBreadthFirst.df(df)`

**Arguments**

`df`                    tree data.frame

**Value**

list of node id's in breadth-first order.

---

`getSubtree`                    *getSubtree*

---

**Description**

Get all children of node from tree, including start\_node.

**Usage**

`getSubtree(tree, node)`

**Arguments**

`tree`                    ape phylo tree object  
`node`                    is the tree node id from which the tree is derived.

**Value**

list of all child node id's from starting node.

---

<code>getSubtree.df</code>	<i>getSubtree.df</i>
----------------------------	----------------------

---

**Description**

Get all children of node from df tree using breath-first.

**Usage**

```
getSubtree.df(df, node)
```

**Arguments**

<code>df</code>	tree data.frame
<code>node</code>	id of starting node.

**Value**

list of all child node id's from starting node.

---

<code>getSubtreeUnrooted</code>	<i>getSubtreeUnrooted</i>
---------------------------------	---------------------------

---

**Description**

Get all subtrees of specified node. This includes all ancestors and relatives of node and return named list of subtrees.

**Usage**

```
getSubtreeUnrooted(tree, node)
```

**Arguments**

<code>tree</code>	ape phylo tree object
<code>node</code>	is the tree node id from which the subtrees are derived.

**Value**

named list of subtrees with the root id of subtree and list of node id's making up subtree.



---

getSubtreeUnrooted.df    *getSubtreeUnrooted*

---

### Description

Get all subtrees of node, as well as remaining branches of parent (ie, rest of tree structure as subtree) return named list of subtrees with list name as starting node id.

### Usage

```
getSubtreeUnrooted.df(df, node)
```

### Arguments

df	tree data.frame
node	is the tree node id from which the subtrees are derived.

### Value

named list of subtrees with the root id of subtree and list of node id's making up subtree.

---

getTreeArcAngles        *getTreeArcAngles*

---

### Description

Find the right (clockwise rotation, angle from +ve x-axis to furthest subtree nodes) and left (anti-clockwise angle from +ve x-axis to subtree) Returning arc angle in [0, 2] (0 to 360) domain.

### Usage

```
getTreeArcAngles(df, origin_id, subtree)
```

### Arguments

df	tree data.frame
origin_id	node id from which to calculate left and right hand angles of subtree.
subtree	named list of root id of subtree (node) and list of node ids for given subtree (subtree).

### Value

named list with right and left angles in range [0,2] i.e 1 = 180 degrees, 1.5 = 270 degrees.

---

`get_clade_position`      *get\_clade\_position*

---

**Description**

get position of clade (xmin, xmax, ymin, ymax)

**Usage**

```
get_clade_position(treeview, node)
```

**Arguments**

treeview	tree view
node	selected node

**Value**

data.frame

**Author(s)**

Guangchuang Yu

---

`get_heatmap_column_position`  
*get\_heatmap\_column\_position*

---

**Description**

return a data.frame that contains position information for labeling column names of heatmap produced by gheatmap function

**Usage**

```
get_heatmap_column_position(treeview, by = "bottom")
```

**Arguments**

treeview	output of gheatmap
by	one of 'bottom' or 'top'

**Value**

data.frame

**Author(s)**

Guangchuang Yu

---

*get\_taxa\_name*                      *get\_taxa\_name*

---

**Description**

get taxa name of a selected node (or tree if node=NULL) sorted by their position in plotting

**Usage**

```
get_taxa_name(tree_view = NULL, node = NULL)
```

**Arguments**

tree\_view            tree view (i.e. the ggtree object). If tree\_view is NULL, the last ggplot will be used.

node                    internal node number to specify a clade. If NULL, using the whole tree

**Details**

This function extract an ordered vector of the tips from selected clade or the whole tree based on the ggtree() plot.

**Value**

ordered taxa name vector

**Author(s)**

Guangchuang Yu

**Examples**

```
tree <- rtree(30)
p <- ggtree(tree)
get_taxa_name(p)
```

---

`ggdensitree`*ggdensitree*

---

**Description**

drawing phylogenetic trees from list of phylo objects

**Usage**

```
ggdensitree(  
  data = NULL,  
  mapping = NULL,  
  layout = "slanted",  
  tip.order = "mode",  
  align.tips = TRUE,  
  jitter = 0,  
  ...  
)
```

**Arguments**

<code>data</code>	a list of phylo objects or any object with an <code>as.phylo</code> and <code>fortify</code> method
<code>mapping</code>	aesthetic mapping
<code>layout</code>	one of 'slanted', 'rectangular', 'fan', 'circular' or 'radial' (default: 'slanted')
<code>tip.order</code>	the order of the tips by a character vector of taxa names; or an integer, N, to order the tips by the order of the tips in the Nth tree; 'mode' to order the tips by the most common order; 'mds' to order the tips based on MDS of the path length between the tips; or 'mds_dist' to order the tips based on MDS of the distance between the tips (default: 'mode')
<code>align.tips</code>	TRUE to align trees by their tips and FALSE to align trees by their root (default: TRUE)
<code>jitter</code>	deviation to jitter tips
<code>...</code>	additional parameters passed to <code>fortify</code> , <code>ggtree</code> and <code>geom_tree</code>

**Value**

tree layer

**Author(s)**

Yu Guangchuang, Bradley R. Jones

## Examples

```

require(ape)
require(dplyr)
require(tidyr)

# Plot multiple trees with aligned tips
trees <- list(read.tree(text="((a:1,b:1):1.5,c:2.5);"), read.tree(text="((a:1,c:1):1,b:2);"));
ggdensitree(trees) + geom_tiplab()

# Plot multiple trees with aligned tips with tip labels and separate tree colors
trees.fort <- list(trees[[1]] %>% fortify %>% mutate(tree="a"), trees[[2]] %>% fortify %>% mutate(tree="b"));
ggdensitree(trees.fort, aes(colour=tree)) + geom_tiplab(colour='black')

# Generate example data
set.seed(1)
random.trees <- rmtree(5, 10)
time.trees <- lapply(seq_along(random.trees), function(i) {
  tree <- random.trees[[i]]
  tree$tip.label <- paste0("t", 1:10)
  dates <- estimate.dates(tree, 1:10, mu=1, nsteps=1)
  tree$edge.length <- dates[tree$edge[, 2]] - dates[tree$edge[, 1]]
  fortify(tree) %>% mutate(tree=factor(i, levels=as.character(1:10)))
})

# Plot multiple trees with aligned tips from multiple time points
ggdensitree(time.trees, aes(colour=tree), tip.order=paste0("t", 1:10)) + geom_tiplab(colour='black')

# Read example data
example.trees <- read.tree(system.file("examples", "ggdensitree_example.tree", package="ggtree"))

# Compute OTU
grp <- list(A = c("a.t1", "a.t2", "a.t3", "a.t4"), B = c("b.t1", "b.t2", "b.t3", "b.t4"), C = c("c.t1", "c.t2", "c.t3", "c.t4"))
otu.trees <- lapply(example.trees, groupOTU, grp)

# Plot multiple trees colored by OTU
ggdensitree(otu.trees, aes(colour=group), alpha=1/6, tip.order='mids') + scale_colour_manual(values=c("black", "red", "blue", "green", "purple", "orange", "pink", "cyan", "brown", "grey"))

```

---

ggtree

*visualizing phylogenetic tree and heterogenous associated data based on grammar of graphics ggtree provides functions for visualizing phylogenetic tree and its associated data in R.*

---

## Description

If you use ggtree in published research, please cite: Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution* 2017, 8(1):28-36, doi:10.1111/2041-210X.12628

drawing phylogenetic tree from phylo object

### Usage

```
ggtree(
  tr,
  mapping = NULL,
  layout = "rectangular",
  open.angle = 0,
  mrsd = NULL,
  as.Date = FALSE,
  yscale = "none",
  yscale_mapping = NULL,
  ladderize = TRUE,
  right = FALSE,
  branch.length = "branch.length",
  root.position = 0,
  xlim = NULL,
  ...
)
```

### Arguments

<code>tr</code>	phylo object
<code>mapping</code>	aesthetic mapping
<code>layout</code>	one of 'rectangular', 'dendrogram', 'slanted', 'ellipse', 'roundrect', 'fan', 'circular', 'inward_circular', 'radial', 'equal_angle', 'daylight' or 'ape'
<code>open.angle</code>	open angle, only for 'fan' layout
<code>mrsd</code>	most recent sampling date
<code>as.Date</code>	logical whether using Date class in time tree
<code>yscale</code>	y scale
<code>yscale_mapping</code>	yscale mapping for category variable
<code>ladderize</code>	logical (default TRUE). Should the tree be re-organized to have a 'ladder' aspect?
<code>right</code>	logical. If <code>ladderize = TRUE</code> , should the ladder have the smallest clade on the right-hand side? See <a href="#">ape::ladderize()</a> for more information.
<code>branch.length</code>	variable for scaling branch, if 'none' draw cladogram
<code>root.position</code>	position of the root node (default = 0)
<code>xlim</code>	x limits, only works for 'inward_circular' layout
<code>...</code>	additional parameter
	some dot arguments:
	<ul style="list-style-type: none"> <li>• <code>nsplit</code> integer, the number of branch blocks divided when 'continuous' is not "none", default is 200.</li> </ul>

### Value

tree

**Author(s)**

Yu Guangchuang

**References**

1. G Yu, TTY Lam, H Zhu, Y Guan (2018). Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>
2. G Yu, DK Smith, H Zhu, Y Guan, TTY Lam (2017). ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution*, 8(1):28-36. <https://doi.org/10.1111/2041-210X.12628>

**See Also**[geom\\_tree\(\)](#)**Examples**

```
require(ape)
tr <- rtree(10)
ggtree(tr)
```

---

gheatmap

*gheatmap*

---

**Description**

append a heatmap of a matrix to right side of phylogenetic tree

**Usage**

```
gheatmap(
  p,
  data,
  offset = 0,
  width = 1,
  low = "green",
  high = "red",
  color = "white",
  colnames = TRUE,
  colnames_position = "bottom",
  colnames_angle = 0,
  colnames_level = NULL,
  colnames_offset_x = 0,
  colnames_offset_y = 0,
  font.size = 4,
  family = "",
```

```
  hjust = 0.5,  
  legend_title = "value"  
)
```

### Arguments

<code>p</code>	tree view
<code>data</code>	matrix or data.frame
<code>offset</code>	offset of heatmap to tree
<code>width</code>	total width of heatmap, compare to width of tree
<code>low</code>	color of lowest value
<code>high</code>	color of highest value
<code>color</code>	color of heatmap cell border
<code>colnames</code>	logical, add matrix colnames or not
<code>colnames_position</code>	one of 'bottom' or 'top'
<code>colnames_angle</code>	angle of column names
<code>colnames_level</code>	levels of colnames
<code>colnames_offset_x</code>	x offset for column names
<code>colnames_offset_y</code>	y offset for column names
<code>font.size</code>	font size of matrix colnames
<code>family</code>	font of matrix colnames
<code>hjust</code>	hjust for column names (0: align left, 0.5: align center, 1: align right)
<code>legend_title</code>	title of fill legend

### Value

tree view

### Author(s)

Guangchuang Yu



---

gzoom	<i>gzoom method</i>
-------	---------------------

---

## Description

gzoom method

gzoom method

zoom selected subtree

## Usage

```
gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7), ...)
```

```
## S4 method for signature 'ggtree'
```

```
gzoom(object, focus, widths = c(0.3, 0.7), xmax_adjust = 0)
```

```
## S4 method for signature 'treedata'
```

```
gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7))
```

```
## S4 method for signature 'phylo'
```

```
gzoom(object, focus, subtree = FALSE, widths = c(0.3, 0.7))
```

## Arguments

object	supported tree objects
focus	selected tips
subtree	logical
widths	widths
...	additional parameter
xmax_adjust	adjust xmax (xlim[2])

## Value

figure

---

<code>gzoom.phylo</code>	<i>gzoom</i>
--------------------------	--------------

---

**Description**

plots simultaneously a whole phylogenetic tree and a portion of it.

**Usage**

```
gzoom.phylo(phy, focus, subtree = FALSE, widths = c(0.3, 0.7))
```

**Arguments**

<code>phy</code>	phylo object
<code>focus</code>	selected tips
<code>subtree</code>	logical
<code>widths</code>	widths

**Value**

a list of ggplot object

**Author(s)**

ygc

---

<code>hexpand</code>	<i>hexpand</i>
----------------------	----------------

---

**Description**

`hexpand`  
`vexpand`  
 expand xlim (ylim) by ratio of x (y) range

**Usage**

```
hexpand(ratio, direction = 1)
vexpand(ratio, direction = 1)
ggexpand(ratio, direction = 1, side = "hv")
```

**Arguments**

`ratio` expand x (y) limits by amount of `xrange` (`yrange`) \* `ratio`  
`direction` expand x limit at right hand side if `direction` is 1, or left hand side if `direction` is -1  
`side` one of 'h' for horizontal and 'v' for vertical or 'hv' for both.

**Value**

ggexpand object

**Author(s)**

Guangchuang Yu

---

<code>identify.gg</code>	<i>identify</i>
--------------------------	-----------------

---

**Description**

identify node by interactive click

**Usage**

```
## S3 method for class 'gg'  
identify(x = last_plot(), ...)
```

**Arguments**

`x` tree view  
`...` additional parameters

**Value**

node id

**Author(s)**

Guangchuang Yu

---

label_pad	<i>Padding taxa labels</i>
-----------	----------------------------

---

**Description**

This function add padding character to the left side of taxa labels.

**Usage**

```
label_pad(label, justify = "right", pad = ".")
```

**Arguments**

label	taxa label
justify	should a character vector be left-justified, right-justified (default), centred or left alone.
pad	padding character (default is a dot)

**Value**

Taxa labels with padding characters added

**Author(s)**

Guangchuang Yu and Yonghe Xia

**References**

<https://groups.google.com/g/bioc-ggtree/c/INJ0Nfkq3b0/m/1XefnfV5AQAJ>

**Examples**

```
library(ggtree)
set.seed(2015-12-21)
tree <- rtree(5)
tree$tip.label[2] <- "long string for test"
label_pad(tree$tip.label)
```

---

layoutDaylight	<i>Equal daylight layout method for unrooted trees.</i>
----------------	---

---

**Description**

#' @title

**Usage**

```
layoutDaylight(model, branch.length, MAX_COUNT = 5)
```

**Arguments**

model	tree object, e.g. phylo or treedata
branch.length	set to 'none' for edge length of 1. Otherwise the phylogenetic tree edge length is used.
MAX_COUNT	the maximum number of iterations to run (default 5)

**Value**

tree as data.frame with equal angle layout.

**References**

The following algorithm aims to implement the vague description of the "Equal-daylight Algorithm" in "Inferring Phylogenies" pp 582-584 by Joseph Felsenstein.

```
Leafs are subtrees with no children
Initialise tree using equal angle algorithm
tree_df = equal_angle(tree)
```

```
nodes = get list of nodes in tree_df breadth-first
nodes = remove tip nodes.
```

---

layoutEqualAngle	<i>layoutEqualAngle</i>
------------------	-------------------------

---

**Description**

'Equal-angle layout algorithm for unrooted trees'

**Usage**

```
layoutEqualAngle(model, branch.length = "branch.length")
```

**Arguments**

model            tree object, e.g. phylo or treedata  
branch.length   set to 'none' for edge length of 1. Otherwise the phylogenetic tree edge length is used.

**Value**

tree as data.frame with equal angle layout.

**References**

"Inferring Phylogenies" by Joseph Felsenstein.

---

*layout\_rectangular*    *layout\_rectangular*

---

**Description**

tree layout

**Usage**

```
layout_rectangular()  
layout_circular()  
layout_inward_circular(xlim = NULL)  
layout_fan(angle = 180)  
layout_dendrogram()
```

**Arguments**

xlim            x limits  
angle           open tree at specific angle

**Author(s)**

Guangchuang Yu

---

`msaplot`*msaplot*

---

**Description**

multiple sequence alignment with phylogenetic tree

**Usage**

```
msaplot(  
  p,  
  fasta,  
  offset = 0,  
  width = 1,  
  color = NULL,  
  window = NULL,  
  bg_line = TRUE,  
  height = 0.8  
)
```

**Arguments**

<code>p</code>	tree view
<code>fasta</code>	fasta file, multiple sequence alignment
<code>offset</code>	offset of MSA to tree
<code>width</code>	total width of alignment, compare to width of tree
<code>color</code>	color
<code>window</code>	specific a slice to display
<code>bg_line</code>	whether add background line in alignment
<code>height</code>	height ratio of sequence

**Value**

tree view

**Author(s)**

Guangchuang Yu

multiplot

*multiplot*

---

**Description**

plot multiple ggplot objects in one page

**Usage**

```
multiplot(  
  ...,  
  plotlist = NULL,  
  ncol,  
  widths = rep_len(1, ncol),  
  labels = NULL,  
  label_size = 5  
)
```

**Arguments**

...	plots
plotlist	plot list
ncol	number of column
widths	widths of plots
labels	labels for labeling the plots
label_size	font size of label

**Value**

plot

**Author(s)**

Guangchuang Yu



---

nodebar	<i>nodebar</i>
---------	----------------

---

**Description**

generate a list of bar charts for results of ancestral state reconstruction

**Usage**

```
nodebar(data, cols, color, alpha = 1, position = "stack")
```

**Arguments**

data	a data.frame of stats with an additional column of node number
cols	column of stats
color	color of bar
alpha	alpha
position	position of bar, one of 'stack' and 'dodge'

**Value**

list of ggplot objects

**Author(s)**

Guangchuang Yu

---

nodepie	<i>nodepie</i>
---------	----------------

---

**Description**

generate a list of pie charts for results of ancestral stat reconstruction

**Usage**

```
nodepie(data, cols, color, alpha = 1)
```

**Arguments**

data	a data.frame of stats with an additional column of node number
cols	column of stats
color	color of bar
alpha	alpha

**Value**

list of ggplot objects

**Author(s)**

Guangchuang Yu

---

open_tree	<i>open_tree</i>
-----------	------------------

---

**Description**

open tree with specific angle

**Usage**

```
open_tree(treeview, angle)
```

**Arguments**

treeview	tree view
angle	angle

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

range_format	<i>range_format</i>
--------------	---------------------

---

**Description**

format a list of range (HPD, CI, etc that has length of 2)

**Usage**

```
range_format(x, trans = NULL)
```

**Arguments**

x	input list
trans	transformation function

**Value**

character vector of [lower, upper]

**Author(s)**

Guangchuang Yu

---

revts

*revts*

---

**Description**

reverse timescacle x-axis

**Usage**

revts(treeview)

**Arguments**

treeview      treeview

**Value**

updated treeview

**Author(s)**

Guangchuang Yu

---

rotate

*rotate*

---

**Description**

rotate 180 degree of a selected branch

**Usage**

rotate(tree\_view = NULL, node)

**Arguments**

tree\_view      tree view (i.e. the ggtree object). If tree\_view is NULL, the last ggplot will be used.

node            internal node number to specify a clade. If NULL, using the whole tree

**Value**

ggplot2 object

**Author(s)**

Guangchuang Yu

---

rotateTreePoints.df    *rotateTreePoints.data.frame*

---

**Description**

Rotate the points in a tree data.frame around a pivot node by the angle specified.

**Usage**

```
rotateTreePoints.df(df, pivot_node, nodes, angle)
```

**Arguments**

df	tree data.frame
pivot_node	is the id of the pivot node.
nodes	list of node numbers that are to be rotated by angle around the pivot_node
angle	in range [0,2], ie degrees/180, radians/pi

**Value**

updated tree data.frame with points rotated by angle

---

rotate\_tree    *rotate\_tree*

---

**Description**

rotate circular tree

**Usage**

```
rotate_tree(treeview, angle)
```

**Arguments**

treeview	tree view
angle	angle

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

scaleClade	<i>scaleClade</i>
------------	-------------------

---

**Description**

scale clade

**Usage**

```
scaleClade(tree_view = NULL, node, scale = 1, vertical_only = TRUE)
```

**Arguments**

tree_view	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node	internal node number to specify a clade. If NULL, using the whole tree
scale	scale
vertical_only	logical. If TRUE, only vertical will be scaled. If FALSE, the clade will be scaled vertical and horizontally. TRUE by default.

**Value**

tree view

**Author(s)**

Guangchuang Yu

---

scale_color	<i>scale_color method</i>
-------------	---------------------------

---

**Description**

scale\_color method  
 scale color by a numerical tree attribute

**Usage**

```
scale_color(object, by, ...)
```

```
## S4 method for signature 'treedata'
```

```
scale_color(object, by, ...)
```

```
## S4 method for signature 'phylo'
```

```
scale_color(object, by, ...)
```

**Arguments**

object	treedata object
by	one of numerical attributes
...	additional parameter

**Value**

color vector

---

scale_x_ggtree	<i>scale_x_ggtree</i>
----------------	-----------------------

---

**Description**

scale x for tree with gheatmap

**Usage**

```
scale_x_ggtree(breaks = waiver(), labels = waiver())
```

**Arguments**

breaks	breaks for tree
labels	lables for corresponding breaks

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

scale\_x\_range      *scale\_x\_range*

---

**Description**

add second x-axis for geom\_range

**Usage**

scale\_x\_range()

**Value**

ggtree object

**Author(s)**

Guangchuang Yu

---

set\_highlight\_legend      *set\_highlight\_legend*

---

**Description**

set legend for multiple geom\_highlight layers

**Usage**

set\_highlight\_legend(p, color, label, alpha = 1)

**Arguments**

p	ggtree object
color	color vector
label	label vector
alpha	transparency of color

**Value**

updated ggtree object

**Author(s)**

Guangchuang Yu

---

StatBalance	<i>StatBalance</i>
-------------	--------------------

---

**Description**

StatBalance

---

td_filter	<i>td-filter</i>
-----------	------------------

---

**Description**

filter data for tree annotation layer

**Usage**

```
td_filter(..., .f = NULL)
```

**Arguments**

...	Expressions that return a logical value.
.f	a function (if any, default is NULL) that pre-operate the data

**Details**

The 'td\_filter()' function returns another function that can be used to subset ggtree() plot data. The function can be passed to the 'data' parameter of geom layer to perform subsetting. All rows that satisfy your conditions will be retained.

**Value**

A function to filter ggtree plot data using conditions defined by '...'.

**Author(s)**

Guangchuang Yu



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

```
tree <- rtree(30)
## similar to 'ggtree(tree) + geom_tippoint()'
ggtree(tree) + geom_point(data = td_filter(isTip))
```

---

*td\_unnest**td-unnest*

---

**Description**

flatterns a list-column of data frame

**Usage**

```
td_unnest(cols, ..., .f = NULL)
```

**Arguments**

<code>cols</code>	columns to unnest
<code>...</code>	additional parameters that pass to <code>tidyr::unnest</code>
<code>.f</code>	a function (if any, default is NULL) that pre-operate the data

**Details**

The `'td_unnest'` function returns another function that can be used to unnest `ggtree()` plot data. The function can be passed to the `'data'` parameter of a geom layer to flattern list-cloumn tree data.

**Value**

A function to unnest `ggtree` plot data

**Author(s)**

Guangchuang Yu

**See Also**[unnest](#)

---

theme_dendrogram	<i>theme_dendrogram</i>
------------------	-------------------------

---

**Description**

dendrogram theme

**Usage**

```
theme_dendrogram(bgcolor = "white", fgcolor = "black", ...)
```

**Arguments**

bgcolor	background color
fgcolor	color for axis
...	additional parameter

**Author(s)**

Guangchuang Yu

---

theme_inset	<i>theme_inset</i>
-------------	--------------------

---

**Description**

inset theme

**Usage**

```
theme_inset(legend.position = "none", ...)
```

**Arguments**

legend.position	position of legend
...	additional parameter

**Details**

theme for inset function

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

theme_tree	<i>theme_tree</i>
------------	-------------------

---

**Description**

tree theme

**Usage**

```
theme_tree(bgcolor = "white", ...)
```

**Arguments**

bgcolor	background color
...	additional parameter

**Value**

updated ggplot object with new theme

**Author(s)**

Guangchuang Yu

**Examples**

```
require(ape)
tr <- rtree(10)
ggtree(tr) + theme_tree()
```

---

theme_tree2	<i>theme_tree2</i>
-------------	--------------------

---

**Description**

tree2 theme

**Usage**

```
theme_tree2(bgcolor = "white", fgcolor = "black", ...)
```

**Arguments**

bgcolor	background color
fgcolor	foreground color
...	additional parameter

**Value**

updated ggplot object with new theme

**Author(s)**

Guangchuang Yu

**Examples**

```
require(ape)
tr <- rtree(10)
ggtree(tr) + theme_tree2()
```

---

viewClade

*viewClade*

---

**Description**

view a clade of tree

**Usage**

```
viewClade(tree_view = NULL, node, xmax_adjust = 0)
```

**Arguments**

tree_view	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node	internal node number to specify a clade. If NULL, using the whole tree
xmax_adjust	adjust xmax

**Value**

clade plot

**Author(s)**

Guangchuang Yu

---

<code>xlim_expand</code>	<i>xlim_expand</i>
--------------------------	--------------------

---

**Description**

expand x axis limits for specific panel

**Usage**

```
xlim_expand(xlim, panel)
```

**Arguments**

<code>xlim</code>	<code>xlim</code>
<code>panel</code>	<code>panel</code>

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

<code>xlim_tree</code>	<i>xlim_tree</i>
------------------------	------------------

---

**Description**

set x axis limits for Tree panel

**Usage**

```
xlim_tree(xlim)
```

**Arguments**

<code>xlim</code>	<code>xlim</code>
-------------------	-------------------

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

zoomClade	<i>zoomClade</i>
-----------	------------------

---

**Description**

zoom selected clade of a tree

**Usage**

```
zoomClade(tree_view = NULL, node, xexpand = NULL)
```

**Arguments**

tree_view	tree view (i.e. the ggtree object). If tree_view is NULL, the last ggplot will be used.
node	internal node number to specify a clade. If NULL, using the whole tree
xexpand	numeric, extend x, meaning the ratio of range of original x, default is NULL.

**Value**

full tree with zoom in clade

**Author(s)**

Guangchuang Yu

---

%+>%	<i>%+&gt;%</i>
------	----------------

---

**Description**

update data with tree info (y coordination and panel)

**Usage**

```
p %+>% data
```

**Arguments**

p	tree view
data	data.frame

**Details**

add tree information to an input data. This function will setup y coordination and panel info for data used in facet\_plot and geom\_facet

**Value**

updated data.frame

**Author(s)**

Guangchuang Yu

**References**

G Yu, TTY Lam, H Zhu, Y Guan (2018). Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>

---

%<+%

%<+%

---

**Description**

add annotation data to a tree

**Usage**

pg %<+% data

**Arguments**

pg	ggplot2 object
data	annotation data

**Value**

ggplot object with annotation data added

**Author(s)**

Guangchuang Yu

**References**

G Yu, TTY Lam, H Zhu, Y Guan (2018). Two methods for mapping and visualizing associated data on phylogeny using ggtree. *Molecular Biology and Evolution*, 35(2):3041-3043. <https://doi.org/10.1093/molbev/msy194>

**See Also**

geom\_facet

**Examples**

```
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)
p <- ggtree(tree)
dd <- data.frame(taxa=LETTERS[1:13],
  place=c(rep("GZ", 5), rep("HK", 3), rep("CZ", 4), NA),
  value=round(abs(rnorm(13, mean=70, sd=10)), digits=1))
row.names(dd) <- NULL
p %<+% dd + geom_text(aes(color=place, label=label), hjust=-0.5)
```

---

 %<%

 %<%
 

---

**Description**

update tree

**Usage**

pg %<% x

**Arguments**

pg	ggtree object
x	tree object

**Details**

This operator apply the visualization directives in ggtree object (lhs) to visualize another tree object (rhs), that is similar to Format Painter.

**Value**

updated ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

```
library("ggplot2")
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)
p <- ggtree(tree) + geom_tippoint(color="#b5e521", alpha=1/4, size=10)
p %<% rtree(30)
```



# Index

## \* datasets

StatBalance, 72

%+>%, 78

%<+%, 79

%<%, 80

add\_colorbar, 4

ape::ladderize(), 54

applyLayoutDaylight, 5

as.polytomy, 5

collapse.ggtree, 6

Date2decimal, 6

decimal2Date, 7

expand, 8

facet\_data, 8

facet\_labeller, 9

facet\_plot, 10

facet\_widths, 11

filter, 73

flip, 11

geom\_aline, 12

geom\_balance, 13

geom\_cladelab, 14

geom\_cladelabel, 17, 20

geom\_cladelabel2, 18, 19

geom\_facet (facet\_plot), 10

geom\_highlight (geom\_hilight), 20

geom\_hilight, 20

geom\_inset, 22

geom\_label, 24

geom\_label2, 23

geom\_motif, 24

geom\_nodelab, 25

geom\_nodelab2, 26

geom\_nodepoint, 27, 29, 33

geom\_point, 29, 33

geom\_point2, 28

geom\_range, 30

geom\_rootedge, 31

geom\_rootpoint, 29, 32, 33

geom\_segment, 35

geom\_segment2, 34

geom\_strip, 35

geom\_taxalink, 36

geom\_text, 38

geom\_text2, 37

geom\_tiplab, 39, 41

geom\_tiplab(), 25, 26

geom\_tiplab2, 41

geom\_tippoint, 29, 33, 42

geom\_tree, 43

geom\_tree(), 55

geom\_tree2, 44

geom\_treescale, 44

geom\_zoom\_clade, 45

get.path, 46

get\_clade\_position, 50

get\_heatmap\_column\_position, 50

get\_taxa\_name, 51

getNodeAngle.df, 46

getNodesBreadthFirst.df, 47

getSubtree, 47

getSubtree.df, 48

getSubtreeUnrooted, 48

getSubtreeUnrooted.df, 49

getTreeArcAngles, 49

ggdensitree, 52

ggexpand (hexpand), 58

ggtree, 53

ghemap, 55

gzoom, 57

gzoom, ggtree-method (gzoom), 57

gzoom, phylo-method (gzoom), 57

gzoom, treedata-method (gzoom), 57

gzoom.phylo, 58

hexpand, 58

identify.gg, 59

inset (geom\_inset), 22

label\_pad, 60

layout\_circular (layout\_rectangular), 62

layout\_dendrogram (layout\_rectangular), 62

layout\_fan (layout\_rectangular), 62

layout\_inward\_circular (layout\_rectangular), 62

layout\_rectangular, 62

layoutDaylight, 61

layoutEqualAngle, 61

msaplot, 63

multiplot, 64

nodebar, 65

nodepie, 65

open\_tree, 66

package-ggtree (ggtree), 53

range\_format, 66

revts, 67

rotate, 67

rotate\_tree, 68

rotateTreePoints.df, 68

scale\_color, 70

scale\_color, phylo-method (scale\_color), 70

scale\_color, treedata-method (scale\_color), 70

scale\_x\_ggtree, 70

scale\_x\_range, 71

scaleClade, 69

set\_highlight\_legend, 71

StatBalance, 72

td\_filter, 72

td\_unnest, 73

theme\_dendrogram, 74

theme\_inset, 74

theme\_tree, 75

theme\_tree2, 75

unnest, 73

vexpand (hexpand), 58

viewClade, 76

xlim\_expand, 77

xlim\_tree, 77

zoomClade, 78