

# Package ‘ggtree’

December 30, 2024

**Type** Package

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

**Version** 3.14.0

**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, dplyr, ggplot2 (> 3.3.6), grid, magrittr, methods, purrr, rlang, ggfun (>= 0.1.7), yulab.utils (>= 0.1.6), tidyr, tidytree (>= 0.4.5), treeio (>= 1.8.0), utils, scales, stats, cli

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

**VignetteBuilder** knitr

**ByteCompile** true

**Encoding** UTF-8

**License** Artistic-2.0

**URL** <https://www.amazon.com/Integration-Manipulation-Visualization-Phylogenetic-Computational-ebook/dp/B0B5NLZR1Z/>  
(book),  
<http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12628>  
(paper)

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

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

**RoxygenNote** 7.3.2

**Roxygen** list(markdown = TRUE)

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** a309078

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-12-30

**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>),

Lin Li [ctb],

Bradley Jones [ctb],

Justin Silverman [ctb],

Watal M. Iwasaki [ctb],

Yonghe Xia [ctb],

Ruizhu Huang [ctb]

## Contents

|                     |    |
|---------------------|----|
| ggtree-package      | 4  |
| add_colorbar        | 5  |
| applyLayoutDaylight | 5  |
| as.polytomy         | 6  |
| collapse.ggtree     | 6  |
| Date2decimal        | 7  |
| decimal2Date        | 8  |
| expand              | 8  |
| facet_data          | 9  |
| 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    | 18 |
| geom_highlight      | 20 |
| geom_inset          | 22 |
| geom_label2         | 23 |
| geom_motif          | 25 |
| geom_nodelab        | 26 |
| geom_nodelab2       | 27 |
| geom_nodepoint      | 27 |
| geom_point2         | 29 |
| geom_range          | 31 |
| geom_rootedge       | 32 |
| geom_rootpoint      | 33 |
| geom_segment2       | 34 |
| geom_strip          | 36 |
| geom_striplab       | 37 |
| geom_taxalink       | 40 |
| geom_text2          | 42 |
| geom_tiplab         | 43 |

|                             |    |
|-----------------------------|----|
| geom_tiplab2                | 45 |
| geom_tippoint               | 46 |
| geom_tree                   | 47 |
| geom_tree2                  | 48 |
| geom_treescale              | 49 |
| geom_zoom_clade             | 50 |
| get.path                    | 51 |
| getNodeAngle.df             | 51 |
| getNodesBreadthFirst.df     | 52 |
| getSubtree                  | 52 |
| getSubtree.df               | 53 |
| getSubtreeUnrooted          | 53 |
| getSubtreeUnrooted.df       | 54 |
| getTreeArcAngles            | 54 |
| get_clade_position          | 55 |
| get_heatmap_column_position | 55 |
| get_taxa_name               | 56 |
| ggdensitree                 | 56 |
| ggtree                      | 58 |
| gheatmap                    | 60 |
| gzoom                       | 62 |
| gzoom.phylo                 | 62 |
| hexpand                     | 63 |
| label_pad                   | 64 |
| layoutDaylight              | 64 |
| layoutEqualAngle            | 65 |
| layout_rectangular          | 66 |
| msaplot                     | 67 |
| multiplot                   | 68 |
| nodebar                     | 68 |
| nodepie                     | 69 |
| open_tree                   | 70 |
| range_format                | 70 |
| reexports                   | 71 |
| revts                       | 71 |
| rotate                      | 72 |
| rotateTreePoints.df         | 73 |
| rotate_tree                 | 73 |
| scaleClade                  | 74 |
| scale_color                 | 74 |
| scale_color_subtree         | 75 |
| scale_x_ggtree              | 76 |
| scale_x_range               | 76 |
| set_highlight_legend        | 77 |
| StatBalance                 | 77 |
| theme_dendrogram            | 78 |
| theme_inset                 | 78 |
| theme_tree                  | 79 |
| theme_tree2                 | 79 |
| viewClade                   | 80 |
| xlim_expand                 | 81 |
| xlim_tree                   | 81 |

|                     |           |
|---------------------|-----------|
| zoomClade . . . . . | 82        |
| %+>% . . . . .      | 83        |
| %<% . . . . .       | 83        |
| <b>Index</b>        | <b>85</b> |

---

|                |   |
|----------------|---|
| ggtree-package | <i>ggtree: an R package for visualization of tree and annotation data</i> |
|----------------|---|

---

## 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.

## Author(s)

**Maintainer:** Guangchuang Yu <guangchuangyu@gmail.com> ([ORCID](#)) [copyright holder]

Authors:

- Tommy Tsan-Yuk Lam <tylam.tommy@gmail.com> [thesis advisor]
- Shuangbin Xu <xshuangbin@163.com> ([ORCID](#))

Other contributors:

- Lin Li <williamlee2220@qq.com> [contributor]
- Bradley Jones <bjones@cfenet.ubc.ca> [contributor]
- Justin Silverman <jsilve24@gmail.com> [contributor]
- Watal M. Iwasaki <heavy.watal@gmail.com> [contributor]
- Yonghe Xia <xiayh17@gmail.com> [contributor]
- Ruizhu Huang <ruizhuRH@gmail.com> [contributor]

## See Also

Useful links:

- <https://www.amazon.com/Integration-Manipulation-Visualization-Phylogenetic-Computational-ebp/B0B5NLZR1Z/> (book)
- <http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12628> (paper)
- Report bugs at <https://github.com/YuLab-SMU/ggtree/issues>

---

|              |                     |
|--------------|---------------------|
| 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 | <i>applyLayoutDaylight</i> |
|---------------------|----------------------------|

---

**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, 'phylo' object only     |
| feature | selected feature                     |
| fun     | function to select nodes to collapse |

**Value**

polytomy tree

**Author(s)**

Guangchuang Yu

---

collapse.ggtree

*collapse-ggtree*


---

**Description**

collapse a selected clade, which can later be expanded with the 'expand()' fuction if necessary

**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' (default), 'max', 'min' and 'mixed'. 'none' would simply collapse the clade as 'tip' and the rest will display a triangle, whose shape is determined by the farest/closest tip of the collapsed clade to indicate it |
| clade_name | set a name for the collapsed clade. If clade_name = NULL, do nothing   |
| ...        | additional parameters to set the color or transparency of the triangle   |

**Value**

tree view

**Author(s)**

Guangchuang Yu

**See Also**

expand

**Examples**

```
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
p
p1 <- collapse(p, node = 17, mode = "mixed",
               clade_name = "cclade", alpha = 0.8,
               color = "grey", fill = "light blue")
```

---

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 | <i>decimal2Date</i> |
|--------------|---------------------|

---

**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 collapsed 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 object 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



**Examples**

```
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
p1 <- collapse(p, 17)
expand(p1, 17)
```

---

|            |                   |
|------------|-------------------|
| 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**

function to relable selected panels created by 'geom\_facet' or 'facet-plot'

**Usage**

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

**Arguments**

|       |                            |
|-------|----------------------------|
| p     | facet_plot output          |
| label | new 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

exchange the position of 2 clades

## 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 clade 1. It should share a same parent node with node2                   |
| node2     | node number of clade 2. It should share a same parent node with node1                   |

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

```
set.seed(123)
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
  geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
flip(p, 23, 24) ## Depends on the condition of your tree
```

---

geom\_aline

*geom\_aline*

---

**Description**

add horizontal align lines layer to a tree

**Usage**

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

**Arguments**

|           |   |
|-----------|---|
| mapping   | aes mapping                                     |
| linetype  | set line type of the line, defaults to "dotted" |
| linewidth | set width of the line, defaults to 1            |
| ...       | additional parameter                            |

**Details**

'geom\_aline' align all tips to the longest one by adding padding characters to the right side of the tip.

**Value**

aline layer

**Author(s)**

Yu Guangchuang

---

|              |              |
|--------------|--------------|
| geom_balance | geom_balance |
|--------------|--------------|

---

### 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 its two direct descendant               |
| fill     | color to fill in the highlight rectangle, default to "steelblue"             |
| color    | color to outline highlight rectangle and divide balance, defaults to "white" |
| alpha    | alpha (transparency) for the highlight rectangle, defaults to 0.5            |
| extend   | extend xmax of the highlight rectangle by the value of extend                |
| extendto | extend xmax of the highlight rectangle to the value of 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)

For more detailed demonstration, please refer to chapter 5.2.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

**Examples**

```
library(ggtree)
set.seed(123)
tr<- rtree(15)
x <- ggtree(tr)
x + geom_balance(17)
```

---

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, defaults to NULL.   |
| mapping | Set of aesthetic mappings, defaults to NULL. The detail see the following explanation.  |
| geom    | character, one of 'text', 'label', 'shadowtext', 'image' and 'phylopic', defaults to 'text', and the parameter see the Aesthetics For Specified Geom.   |
| parse   | logical, whether parse label to emoji font, defaults to FALSE.  |
| ...     | additional parameters, see also following section.<br>additional parameters can refer the following parameters. <ul style="list-style-type: none"> <li>• <code>offset</code> distance bar and tree, offset of bar and text from the clade, defaults to 0.</li> <li>• <code>offset.text</code> distance bar and text, offset of text from bar, defaults to 0.</li> <li>• <code>align</code> logical, whether align clade lab, defaults to FALSE.</li> <li>• <code>extend</code> numeric, extend the length of bar, defaults to 0.</li> <li>• <code>angle</code> numeric or 'auto', if angle is auto, the angle of text will be calculated automatically, which is useful for the circular etc layout, defaults to 0.</li> <li>• <code>horizontal</code> logical, whether set label to horizontal, defaults to TRUE.</li> </ul> |

- `barsize` the width of line, defaults to 0.5.
- `barcolour` the colour of line, defaults to 'black'.
- `fontsize` the size of text, defaults to 3.88.
- `textcolour` the colour of text, defaults to 'black'.
- `imagesize` the size of image, defaults to 0.05.
- `imagecolor` the colour of image, defaults to 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 to be shown, it is required.
- `colour` the colour of text, defaults to "black".
- `size` the size of text, defaults to 3.88.
- `angle` the angle of text, defaults to 0.
- `hjust` A numeric vector specifying horizontal justification, defaults to 0.
- `vjust` A numeric vector specifying vertical justification, defaults to 0.5.
- `alpha` the transparency of text, defaults to NA.
- `family` the family of text, defaults to 'sans'.
- `fontface` the font face of text, defaults to 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, defaults to 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="label"` (required aesthetics are in bold):

- **node** selected node to hight light, it is required.
- **label** labels to be shown, it is required.
- `colour` the colour of text, defaults to "black".
- `fill` the background colour of the label, defaults to "white".
- `size` the size of text, defaults to 3.88.
- `angle` the angle of text, defaults to 0.
- `hjust` A numeric vector specifying horizontal justification, defaults to 0.
- `vjust` A numeric vector specifying vertical justification, defaults to 0.5.
- `alpha` the transparency of text, defaults to NA.
- `family` the family of text, defaults to 'sans'.
- `fontface` the font face of text, defaults to 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, defaults to 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 shown, it is required.
- **colour** the colour of text, defaults to "black".
- **bg.colour** the background colour of text, defaults to 'black'.
- **bg.r** the width of background text, defaults to 0.1.
- **size** the size of text, defaults to 3.88.
- **angle** the angle of text, defaults to 0.
- **hjust** A numeric vector specifying horizontal justification, defaults to 0.
- **vjust** A numeric vector specifying vertical justification, defaults to 0.5.
- **alpha** the transparency of text, defaults to NA.
- **family** the family of text, defaults to 'sans'.
- **fontface** the font face of text, defaults to 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, defaults to 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 shown, 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, defaults to NULL.
- **size** the size of image, defaults to 0.05.
- **alpha** the alpha of image, defaults to 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-bfe0-4c421fbe51e8"),
  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
  ),
  geom="shadowtext",
  hjust=0.5,
  align=TRUE,
  horizontal=FALSE,
  angle=90,
  show.legend = FALSE
)
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, defaults to 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, defaults to 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",
  to.bottom = FALSE,
  ...
)
```

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

**Arguments**

|           |  |
|-----------|--|
| data      | data.frame, The data to be displayed in this layer, defaults to NULL.  |
| mapping   | Set of aesthetic mappings, defaults to NULL.   |
| node      | selected node to highlight, when data and mapping is NULL, it is required.   |
| type      | the type of layer, defaults to 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), 'gradient' (gradient color), 'roundrect' (round rectangular layer).  |
| to.bottom | logical, whether set the high light layer to the bottom in all layers of 'ggtree' object, default is FALSE.  |
| ...       | additional parameters, see also the below and Aesthetics section. <ul style="list-style-type: none"> <li>align control the align direction of the edge of high light rectangular. Options is 'none' (default), 'left', 'right', 'both'. This argument only work when the 'geom_highlight' is plotting using geom_highlight(mapping=aes(...)).</li> <li>gradient.direction character, the direction of gradient color, defaults to 'rt' meaning the locations of gradient color is from root to tip, options are 'rt' and 'tr'.</li> <li>gradient.length.out integer, desired length of the sequence of gradient color, defaults to 2.</li> <li>roundrect.r numeric, the radius of the rounded corners, when roundrect=TRUE, defaults to 0.05.</li> </ul> |

## 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, defaults to NA.
- **fill** the colour of fill, defaults to 'steelblue'.
- **alpha** the transparency of fill, defaults to 0.5.
- **extend** extend xmax of the rectangle, defaults to 0.
- **extendto** specify a value, meaning the rectangle extend to, defaults to NULL.
- **linetype** the line type of margin, defaults to 1.
- **linewidth** the width of line of margin, defaults to 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, defaults to 'black'.
- **fill** the colour of fill, defaults to 'steelblue'.
- **alpha** the transparency of fill, defaults to 0.5.
- **expand** expands the xspline clade region, defaults to 0.
- **spread** control the size, when only one point.
- **linewidth** the width of line of margin, defaults to 0.5.
- **linetype** the line type of margin, defaults to 1.
- **s\_shape** the shape of the spline relative to the control points, defaults to 0.5.
- **s\_open** whether the spline is a line or a closed shape, defaults to FALSE.

## Author(s)

Guangchuang Yu and Shuangbin Xu

## References

For more detailed demonstration, please refer to chapter 5.2.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

**Examples**

```

library(ggplot2)
set.seed(102)
tree <- rtree(60)
p <- ggtree(tree)
p1 <- p + geom_hilight(node=62) + geom_hilight(node=88, fill="red")
p1
dat <- data.frame(id=c(62, 88), type=c("A", "B"))
p2 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type))
p2
p3 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="left")
p4 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="right")
p5 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), align="both")
# display the high light layer with gradiental color rectangular.
p6 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), type = "gradient", alpha=0.68)
p7 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type),
                      type = "gradient", gradient.direction="tr", alpha=0.68)
# display the high light layer with round rectangular.
p8 <- p + geom_hilight(data=dat, mapping=aes(node=id, fill=type), type = "roundrect", alpha=0.68)
p2/ p3/ p4/ p5 / p6/ p7/ p8

```

---

geom\_inset

*geom\_inset*


---

**Description**

geom\_inset can add subplots to tree by accepting a list of ggplot objects that are ancestral stats or data associated with selected nodes in the tree. These ggplot objects can be any kind of charts or hybrid of of these charts.

add subplots as 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

|           |  |
|-----------|--|
| insets    | a list of ggplot objects, named by node number   |
| width     | width of the inset, relative to the range of x-axis, defaults to .1                        |
| height    | height of the inset, relative to the range of y-axis, defaults to .1                       |
| hjust     | adjust the horizontal position of the charts, charts will go left if hjust > 0             |
| vjust     | adjust the vertical position of the charts, charts will go down if vjust > 0               |
| x         | the position where users want to place the charts, one of 'node' (default) and 'branch'    |
| reverse_x | whether to reverse x axis of the charts by 'ggplot2::scale_x_reverse', defaults to 'FALSE' |
| reverse_y | whether to reverse y axis of the charts by 'ggplot2::scale_y_reverse', defaults to 'FALSE' |
| tree_view | tree view  |

### Details

Users can also use

### Value

inset layer  
tree view with insets

### Author(s)

Guangchuang Yu

### References

For demonstration of this function, please refer to chapter 8.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

---

geom\_label2

geom\_label2

---

### 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       | Set of aesthetic mappings, defaults to NULL.  |
| data          | A layer specific dataset - only needed if you want to override the plot defaults.   |
| ...           | other arguments passed on to 'layer'.   |
| stat          | Name of the 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, defaults to 'FALSE'.   |
| nudge_x       | adjust the horizontal position of the labels.   |
| nudge_y       | adjust the vertical position of the labels.   |
| label.padding | Amount of padding around label, defaults to 'unit(0.25, "lines")'.  |
| label.r       | Use to set the radius of rounded corners of the label, defaults to 'unit(0.15, "lines")'.                                 |
| label.size    | Size of label border, in mm, defaults to 0.25.  |
| na.rm         | If "FALSE" (default), missing values are removed with a warning. If "TRUE", missing values are silently removed, logical. |
| show.legend   | Whether to show legend, logical, defaults to "NA".  |
| inherit.aes   | Whether to inherit aesthetic mappings, logical, defaults to "TRUE".   |

**Details**

'geom\_label2' is a modified version of geom\_label, with subset aesthetic supported

**Value**

label layer



**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration of this function, please refer to chapter A.4.5 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

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

```
library(ggtree)
set.seed(123)
tr<- rtree(15)
x <- ggtree(tr)
x + geom_label2(aes(label = node, subset = isTip == FALSE))
```

---

`geom_motif`*geom\_motif*

---

**Description**

geom layer to draw aligned motif

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

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

**Value**

geom layer

**Author(s)**

Guangchuang Yu

---

geom\_nodelab                      *geom\_nodelab*

---

## Description

add node label layer for a tree

## Usage

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

## Arguments

|         |  |
|---------|--|
| mapping | aesthetic mappings, defaults to NULL   |
| nudge_x | horizontal adjustment to nudge labels, defaults to 0   |
| nudge_y | vertical adjustment to nudge labels, defaults to 0   |
| geom    | one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'   |
| hjust   | horizontal alignment, defaults to 0.5  |
| node    | a character indicating which node labels will be displayed, it should be one of 'internal', 'external' and 'all'. If it is set to 'internal' will display internal node labels, 'external' will display the tip labels, and 'all' will display internal node and tip labels. |
| ...     | additional parameters, see also the additional parameters of <a href="#">geom_tiplab()</a> .   |

## Value

geom layer

## Author(s)

Guangchuang Yu

## References

For demonstration of this function, please refer to chapter A.4.5 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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 | aesthetic mappings, defaults to NULL   |
| nudge_x | horizontal adjustment to nudge labels, defaults to 0   |
| nudge_y | vertical adjustment to nudge labels, defaults to 0   |
| geom    | one of 'text', 'shadowtext', 'label', 'image' and 'phylopic'                                 |
| hjust   | horizontal alignment, defaults to 0.5  |
| ...     | additional parameters, see also the additional parameters of <a href="#">geom_tiplab()</a> . |

### Value

node label layer

### Author(s)

Guangchuang Yu

---

geom\_nodepoint                      *geom\_nodepoint*

---

### Description

add node point layer to a tree

**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 <code>ggplot()</code> .  |
| position    | Position adjustment.   |
| na.rm       | logical. If 'FALSE' (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 (defaults to '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 `library(ggtree) tr<- rtree(15) x <- ggtree(tr) x + geom_nodepoint()`

**References**

For more detailed demonstration, please refer to chapter 4.3.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

---

|             |                    |
|-------------|--------------------|
| geom_point2 | <i>geom_point2</i> |
|-------------|--------------------|

---

### 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

|                          |  |
|--------------------------|--|
| <code>mapping</code>     | 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> . |
| <code>data</code>        | 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 <code>ggplot()</code> .  |
| <code>stat</code>        | Name of the statistical transformation to be used on the data for this layer.  |
| <code>position</code>    | Position adjustment.   |
| <code>na.rm</code>       | logical. If 'FALSE' (default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.  |
| <code>show.legend</code> | 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.                                 |
| <code>inherit.aes</code> | logical (defaults to 'TRUE'). If 'FALSE', overrides the default aesthetics, rather than combining with them.   |
| <code>...</code>         | 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> .               |

### 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, defaults to "black".
- fill the colour of fill, defaults to "black".
- alpha the transparency of fill, defaults to 1.
- size the size of point, defaults to 1.5.
- shape specify a shape, defaults to 19.
- stroke control point border thickness of point, defaults to 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

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

## Usage

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

## Arguments

|        |  |
|--------|--|
| range  | range(interval) to be displayed, 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, linewidth, alpha                           |

## Value

ggplot layer

## Author(s)

Guangchuang Yu

## References

For demonstration of this function, please refer to chapter 5.2.4 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

---

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

---

**Description**

display root edge layer for a tree

**Usage**

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

**Arguments**

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

Additional parameters can be referred to the following parameters:

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

**Details**

geom\_rootedge is used to create a rootedge as ggtree doesn't plot the root edge by default.

**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 detailed demonstration of this function, please refer to chapter A.4.5 of
## *Data Integration, Manipulation and Visualization of Phylogenetic Trees*
## <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

```

---

geom\_rootpoint                      *geom\_rootpoint*

---

## Description

geom\_rootpoint is used to add root point layer to a tree

## 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' (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 (defaults to '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 inherit from geom\_point2, and 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     | Set of aesthetic mappings, defaults to NULL   |
| data        | A layer specific dataset - only needed if you want to override the plot defaults.   |
| stat        | Name of stat to modify data.  |
| position    | The position adjustment to use for overlapping points on this layer.  |
| lineend     | Line end style, one of butt (default), round and square.  |
| na.rm       | If "FALSE" (default), missing values are removed with a warning. If "TRUE", missing values are silently removed, logical. |
| show.legend | Whether to show legend, logical.  |
| inherit.aes | Whether to inherit aesthetic mappings, logical, defaults to "TRUE".   |
| nudge_x     | adjust the horizontal position of the segments.   |
| 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  |

**Details**

'geom\_segment2' is a modified version of geom\_segment, with subset aesthetic supported

**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       | add label alongside the bar (optional)   |
| offset      | offset of bar and text from the clade  |
| offset.text | offset of text from bar  |
| align       | logical, whether to align bars to the most distant bar ,defaults to "TRUE" Note that if "FALSE", the bars might cross the tree |
| barsize     | set size of the bar  |
| extend      | extend bar length vertically   |
| fontsize    | set size of the text   |
| angle       | set the angle of text  |
| geom        | one of 'text' or 'label'   |
| hjust       | adjust the horizontal position of the bar  |
| color       | set color for bar and label  |
| fill        | set color to fill label background, only work with geom='label'  |

|        |  |
|--------|--|
| family | "sans" by default, can be any supported font   |
| parse  | logical, whether to parse labels, if "TRUE", the labels will be parsed into expressions, defaults to "FALSE" |
| ...    | additional parameter   |

**Value**

ggplot layers

**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration of this function, please refer to chapter 5.2.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

**Examples**

```
library(ggtree)
tr<- rtree(15)
x <- ggtree(tr)
x + geom_strip(13, 1, color = "red") + geom_strip(3, 7, color = "blue")
```

---

|               |                      |
|---------------|----------------------|
| geom_striplab | <i>geom_striplab</i> |
|---------------|----------------------|

---

**Description**

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

**Usage**

```
geom_striplab(
  taxa1 = NULL,
  taxa2 = NULL,
  label = NULL,
  data = NULL,
  mapping = NULL,
  geom = "text",
  parse = FALSE,
  ...
)
```

**Arguments**

|         |   |
|---------|---|
| taxa1   | can be label or node number   |
| taxa2   | can be label or node number   |
| 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.<br>additional parameters can refer the following parameters. <b>##</b> <ul style="list-style-type: none"> <li>• <code>offset</code> distance bar and tree, offset of bar and text from the clade, default is 0.</li> <li>• <code>offset.text</code> distance bar and text, offset of text from bar, default is 0.</li> <li>• <code>align</code> logical, whether align clade lab, default is FALSE.</li> <li>• <code>extend</code> numeric, extend the length of bar, default is 0.</li> <li>• <code>angle</code> 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.</li> <li>• <code>horizontal</code> logical, whether set label to horizontal, default is TRUE.</li> <li>• <code>barsize</code> the width of line, default is 0.5.</li> <li>• <code>barcolour</code> the colour of line, default is 'black'.</li> <li>• <code>fontsize</code> the size of text, default is 3.88.</li> <li>• <code>textcolour</code> the colour of text, default is 'black'.</li> <li>• <code>imagesize</code> the size of image, default is 0.05.</li> <li>• <code>imagecolor</code> the colour of image, default is NULL, when geom="phylopic", it should be required.</li> </ul> |

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

- **taxa1** selected tip label or tip node, it is required.
- **taxa2** selected another tip label or tip node, it is required.
- **label** labels to be shown, 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_striplab()` understands the following aesthetics for `geom="label"` (required aesthetics are in bold):

- **taxa1** selected node to hight light, it is required.
- **taxa2** selected another tip label or tip node, it is required.
- **label** labels to be shown, 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_striplab()` understands the following aesthetics for `geom="shadowtext"` (required aesthetics are in bold):

- **taxa1** selected node to hight light, it is required.
- **taxa2** selected another tip label or tip node, it is required.
- **label** labels to be shown, 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\_striplab() understands the following aesthetics for geom="image" or geom="phylopic" (required aesthetics are in bold):

- **taxa1** selected node to hight light, it is required.
- **taxa2** selected another tip label or tip node, it is required.
- **label** labels to be shown, 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(123)
tr <- rtree(10)
dt <- data.frame(ta1=c("t5", "t1"), ta2=c("t6", "t3"), group=c("A", "B"))
p <- ggtree(tr) + geom_tiplab()
p2 <- p +
  geom_striplab(
    data = dt,
    mapping = aes(taxa1 = ta1, taxa2 = ta2,
                  label = group, color=group),
    align = TRUE,
    show.legend = FALSE
  )
p2
```

---

geom\_taxalink

*link between taxa*

---

## Description

geom\_taxalink supports data.frame as input, the colour, linewidth, 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).
- linewidth 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 | geom_tiplab |
|-------------|-------------|

---

**Description**

add tip label layer for a tree

**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, defaults to 0   |
| align    | if TRUE, align all tip labels to the longest tip by adding padding characters to the left side of tip labels, defaults to "FALSE" with a line connecting each tip and its corresponding label, defaults to "FALSE" |
| linetype | set linetype of the line if align = TRUE, defaults to "dotted"   |
| linesize | set line width if align = TRUE, defaults to 0.5  |
| geom     | one of 'text', 'label', 'shadowtext', 'image' and 'phylopic'   |
| offset   | tiplab offset, horizontal adjustment to nudge tip labels, defaults to 0  |
| as_ylab  | display tip labels as y-axis label, only works for rectangular and dendrogram layouts, defaults to "FALSE"   |
| ...      | additional parameter   |

additional parameters can refer the following parameters.  
The following parameters for geom="text".

- size control the size of tip labels, defaults to 3.88.
- colour control the colour of tip labels, defaults to "black".
- angle control the angle of tip labels, defaults to 0.
- vjust A numeric vector specifying vertical justification, defaults to 0.5.
- alpha the transparency of text, defaults to NA.
- family the family of text, defaults to 'sans'.

- `fontface` the font face of text, defaults to 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, defaults to 1.2.
- `nudge_x` horizontal adjustment to nudge labels, defaults to 0.
- `nudge_y` vertical adjustment to nudge labels, defaults to 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, defaults to 3.88.
- `colour` the colour of tip labels, defaults to "black".
- `fill` the colour of rectangular box of labels, defaults to "white".
- `vjust` numeric vector specifying vertical justification, defaults to 0.5.
- `alpha` the transparency of labels, defaults to NA.
- `family` the family of text, defaults to 'sans'.
- `fontface` the font face of text, defaults to 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, defaults to 1.2.
- `nudge_x` horizontal adjustment to nudge labels, defaults to 0.
- `nudge_y` vertical adjustment, defaults to 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, defaults to 'unit(0.25, "lines")'.
- `label.r` Radius of rounded corners, defaults to 'unit(0.15, "lines")'.
- `label.size` Size of label border, in mm, defaults to 0.25.

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

- `bg.colour` the background colour of text, defaults to "black".
- `bg.r` the width of background of text, defaults to 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, defaults to 0.05.
- `colour` the color of image, defaults to NULL.
- `alpha` the transparency of image, defaults to 0.8.

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

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

**Details**

'geom\_tiplab' not only supports using text or label geom to display tip labels, but also supports image geom to label tip with image files or phylogenics.

For adding tip labels to a tree with circular layout, 'geom\_tiplab' will automatically adjust the angle of the tip labels to the tree by internally calling 'geom\_tiplab2'.

**Value**

tip label layer

**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration, please refer to chapter 4.3.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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, defaults to 0  |
| ...     | additional parameter, see geom_tiplab |

**Details**

'geom\_tiplab2' will automatically adjust the angle of the tip labels to the tree with circular layout

**Value**

tip label layer

**Author(s)**

Guangchuang Yu

**References**<https://groups.google.com/forum/#!topic/bioc-ggtree/o35PV3iH0-0>**See Also**[geom\\_tiplab](#)**Examples**

```
library(ggtree)
set.seed(123)
tr <- rtree(10)
ggtree(tr, layout = "circular") + geom_tiplab2()
```

---

`geom_tippoint`*geom\_tippoint*

---

**Description**

add tip point layer for a tree

**Usage**

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

**Arguments**

|                          |  |
|--------------------------|--|
| <code>mapping</code>     | 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> . |
| <code>data</code>        | 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 <code>ggplot()</code> .  |
| <code>position</code>    | Position adjustment.   |
| <code>na.rm</code>       | logical. If 'FALSE' (default), missing values are removed with a warning. If 'TRUE', missing values are silently removed.  |
| <code>show.legend</code> | 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 (defaults to '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`.

### Value

tip point layer

### Author(s)

Guangchuang Yu

### References

For more detailed demonstration, please refer to chapter 4.3.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

### Examples

```
library(ggtree)
tr<- rtree(15)
x <- ggtree(tr)
x + geom_tippoint()
```

---

geom\_tree

*geom\_tree*

---

### 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 of the tree

`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, defaults to "FALSE".  |
| 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<br>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).
- linewidth numeric, control the width of line, default is 0.5 (continuous is "none").

**Author(s)**

Yu Guangchuang

**References**

For demonstration of this function, please refer to chapter 4.2.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

**Examples**

```
tree <- rtree(10)
ggplot(tree) + geom_tree()
```

---

geom\_tree2

geom\_tree2

---

**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 to a tree

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

**Details**

'geom\_treescale' automatically adds a scale bar for evolutionary distance

**Value**

ggplot layers

**Author(s)**

Guangchuang Yu

**References**

For demonstration of this function, please refer to chapter 4.3.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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 to zoom in its corresponding clade   |
| xexpand | numeric, extend x, meaning the ratio of range of the xlim of the original tree, defaults to NULL. |

**Details**

'geom\_zoom\_clade' zooms in on a selected clade of a tree, while showing its on the full view of tree as a separated panel for reference

**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

---

getNodesBreadthFirst.df  
*getNodesBreadthFirst.df*

---

**Description**

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

**Usage**

```
getNodesBreadthFirst.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.

---

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

---

**Description**

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

**Usage**

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

**Arguments**

|      |                      |
|------|----------------------|
| df   | tree data.frame      |
| node | id of starting node. |

**Value**

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

---

|                    |                           |
|--------------------|---------------------------|
| getSubtreeUnrooted | <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**

|      |  |
|------|--|
| tree | ape phylo tree object                                    |
| 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.

---

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  $[\theta, 2]$  (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  $[\theta, 2]$  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 | <i>get_taxa_name</i> |
|---------------|----------------------|

---

### 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 object 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 | <i>ggdensitree</i> |
|-------------|--------------------|

---

### 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, <code>N</code> , to order the tips by the order of the tips in the <code>N</code> th 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 (default) to align trees by their tips and FALSE to align trees by their root  |
| <code>jitter</code>     | deviation to jitter tips  |
| <code>...</code>        | additional parameters passed to <code>fortify</code> , <code>ggtree</code> and <code>geom_tree</code>   |

**Details**

The trees plotted by `'ggdensitree()'` will be stacked on top of each other and the structures of the trees will be rotated to ensure the consistency of the tip order.

**Value**

tree layer

**Author(s)**

Yu Guangchuang, Bradley R. Jones

**References**

For more detailed demonstration of this function, please refer to chapter 4.4.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

**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='mds') +
  scale_colour_manual(values=c("black", "red", "green", "blue"))

```

---

ggtree

*ggtree*


---

## Description

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,
  layout.params = list(as.graph = TRUE),
  hang = 0.1,
  ...
)

```

### 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>layout.params</code>  | list, the parameters of layout, when layout is a function. <code>as.graph=TRUE</code> and layout is a function, the coordinate will be re-calculated as a <code>igraph</code> object, if <code>as.graph=FALSE</code> and layout, the coordinate will be re-calculated keep original object phylo or <code>treedata</code> . |
| <code>hang</code>           | numeric The fraction of the tree plot height by which labels should hang below the rest of the plot. A negative value will cause the labels to hang down from 0. This parameter only work with the 'dendrogram' layout for 'hclust' like class, default is 0.1.   |
| <code>...</code>            | additional parameter<br>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>

For more information, please refer to *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

## See Also

[geom\\_tree\(\)](#)

## Examples

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

---

gheatmap

*gheatmap*

---

## Description

append a heatmap of a matrix to the right side of a 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",  
  custom_column_labels = NULL  
)
```

**Arguments**

|                      |  |
|----------------------|--|
| p                    | tree view  |
| data                 | matrix or data.frame   |
| offset               | set offset of the heatmap to tree  |
| width                | total width of heatmap, compare to width of tree, defaults to 1, which means they are of the same length         |
| low                  | set color of the lowest value, defaults to "green"   |
| high                 | set color of the highest value, defaults to "red"  |
| color                | set color of heatmap cell border, defaults to "white"  |
| colnames             | logical, whether to add matrix colnames, defaults to "TRUE"  |
| colnames_position    | set the position of the colnames, one of 'bottom' (default) or 'top'   |
| colnames_angle       | set the angle of colnames  |
| colnames_level       | set levels of colnames   |
| colnames_offset_x    | set x offset for colnames  |
| colnames_offset_y    | set y offset for colnames  |
| font.size            | set font size of matrix colnames   |
| family               | font of matrix colnames, can be any supported font   |
| hjust                | adjust horizontal position of column names (0: align left, 0.5: align center (default), 1: align right)          |
| legend_title         | title of fill legend   |
| custom_column_labels | instead of using the colnames from the input matrix/data.frame, input a custom vector to be set as column labels |

**Value**

tree view

**Author(s)**

Guangchuang Yu

**References**

For demonstration of this function, please refer to chapter 7.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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

---

|             |              |
|-------------|--------------|
| gzoom.phylo | <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**

|         |               |
|---------|---------------|
| phy     | phylo object  |
| focus   | selected tips |
| subtree | logical       |
| widths  | widths        |

**Value**

a list of ggplot object

**Author(s)**

ygc

---

|         |                |
|---------|----------------|
| hexpand | <i>hexpand</i> |
|---------|----------------|

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

ggexpand object

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(20)
x$tip.label <- paste0('RRRRREEEEEEAAAAALLLLLLYYYYY_Long_Lable_', x$tip.label)
p1 <- ggtree(x) + geom_tiplab()
p1 + ggexpand(1.5, side = "h")
```

---

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

---

**Description**

This function adds padding characters to the left side of taxa labels, adjust their length to the longest label.

**Usage**

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

**Arguments**

|         |  |
|---------|--|
| label   | taxa label   |
| justify | should a character vector be right-justified (default), left-justified, centred or left alone. |
| pad     | padding character (defaults to dots)   |

**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/lXefnfV5AQAJ>

**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

transform circular/fan layout to rectangular layout  
transform rectangular layout to circular layout  
transform rectangular/circular layout to inward circular layout  
transform rectangular/circular layout to fan layout  
transform rectangular layout to dendrogram layout

### Usage

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

### Arguments

xlim                    setting x limits, which will affect the center space of the tree  
angle                    open tree at specific angle

### Author(s)

Guangchuang Yu

### Examples

```
tree <- rtree(20)  
p <- ggtree(tree, layout = "circular") + layout_rectangular()  
tree <- rtree(20)  
p <- ggtree(tree)  
p + layout_circular()  
tree <- rtree(20)  
p <- ggtree(tree)  
p + layout_inward_circular(xlim=4) + geom_tiplab(hjust=1)  
tree <- rtree(20)  
p <- ggtree(tree)  
p + layout_fan(angle=90)  
tree <- rtree(20)  
p <- ggtree(tree)  
p + p + layout_dendrogram()
```

---

`msaplot`*msaplot*

---

**Description**

visualize phylogenetic tree with multiple sequence alignment

**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 that contains multiple sequence alignment information   |
| <code>offset</code>  | set the offset of MSA to tree  |
| <code>width</code>   | total width of alignment, compare to width of tree, defaults to 1, which means they are of the same length |
| <code>color</code>   | set color of the tree  |
| <code>window</code>  | specific a slice of alignment to display   |
| <code>bg_line</code> | whether to add background line in alignment, defaults to "TRUE"  |
| <code>height</code>  | height ratio of sequence, defaults to 0.8  |

**Value**

tree view

**Author(s)**

Guangchuang Yu

**References**

For demonstration of this function, please refer to chapter 7.4 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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       | set the number of column to display the plots |
| widths     | the width of each plot                        |
| labels     | set labels for labeling the plots             |
| label_size | set font size of the label                    |

**Value**

plot

**Author(s)**

Guangchuang Yu

---

nodebar

*nodebar*

---

**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 named "node"                                      |
| cols     | columns of the data.frame that store the stats   |
| color    | set color of bars  |
| alpha    | set transparency of the charts   |
| position | position of bars, if 'stack' (default) make bars stacked atop one another, 'dodge' make them dodged side-to-side |

**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,
  outline.color = "transparent",
  outline.size = 0
)
```

**Arguments**

|               |   |
|---------------|---|
| data          | a data.frame of stats with an additional column of node number named "node" |
| cols          | columns of the data.frame that store the stats                              |
| color         | set color of bars   |
| alpha         | set transparency of the charts  |
| outline.color | color of outline  |
| outline.size  | size of outline   |

**Value**

list of ggplot objects

**Author(s)**

Guangchuang Yu

---

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

---

**Description**

transform a tree in either rectangular or circular layout into the fan layout that opens with a specific angle

**Usage**

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

**Arguments**

|          |  |
|----------|--|
| treeview | tree view in rectangular/circular layout |
| angle    | open the tree at a specific angle        |

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
tree <- rtree(15)
p <- ggtree(tree) + geom_tiplab()
open_tree(p, 180)
```

---

|              |                     |
|--------------|---------------------|
| 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

reexports

*Objects exported from other packages***Description**

These objects are imported from other packages. Follow the links below to see their documentation.

**ape** [read.tree](#), [rtree](#)

**aplot** [plot\\_list](#)

**dplyr** [collapse](#)

**ggfun** [%<+%](#), [identify](#), [td\\_filter](#), [td\\_mutate](#), [td\\_unnest](#)

**ggplot2** [aes](#), [fortify](#), [geom\\_label](#), [geom\\_point](#), [geom\\_text](#), [ggplot](#), [ggsave](#), [guide\\_legend](#), [margin](#), [scale\\_color\\_manual](#), [scale\\_colour\\_manual](#), [scale\\_fill\\_manual](#), [scale\\_x\\_continuous](#), [theme](#), [xlim](#)

**grid** [arrow](#), [unit](#)

**magrittr** [%>%](#)

**tidytree** [groupClade](#), [groupOTU](#), [MRCA](#), [nodeid](#), [nodelab](#)

**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)
```

revts

*revts***Description**

reverse timescale x-axis by setting the most recent tip to 0

**Usage**

```
revts(treeview)
```

**Arguments**

treeview      original tree view

**Details**

'scale\_x\_continuous(labels=abs)' is required if users want to set the x-axis label to absolute value

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
tr <- rtree(10)
p <- ggtree(tr) + theme_tree2()
p2 <- revts(p)
p2 + scale_x_continuous(labels=abs)
```

---

rotate

*rotate*

---

**Description**

rotate selected clade by 180 degree

**Usage**

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

**Arguments**

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

**Value**

ggplot2 object

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
rotate(p, 17)
```



---

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 in a certain angle

**Usage**

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

**Arguments**

|          |                              |
|----------|------------------------------|
| treeview | tree view in circular layout |
| angle    | the angle of rotation        |

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
tree <- rtree(15)
p <- ggtree(tree) + geom_tiplab()
p2 <- open_tree(p, 180)
rotate_tree(p2, 180)
```

---

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

---

**Description**

zoom out/in a selected clade to emphasize or de-emphasize it

**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 object will be used.                          |
| node          | internal node number to specify a clade. If NULL, using the whole tree  |
| scale         | the scale of the selected clade. The clade will be zoom in when scale > 1, and will be zoom out when scale < 1          |
| vertical_only | logical. If TRUE (default), only vertical will be scaled. If FALSE, the clade will be scaled vertical and horizontally. |

**Value**

tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
  geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
scaleClade(p, 24, scale = .1)
```

---

|             |                           |
|-------------|---------------------------|
| 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\_color\_subtree    *scale\_color\_subtree*

---

**Description**

scale tree color by subtree (e.g., output of cutree, kmeans, or other clustering algorithm)

**Usage**

```
scale_color_subtree(group)  
  
scale_colour_subtree(group)
```

**Arguments**

|       |                        |
|-------|------------------------|
| group | taxa group information |
|-------|------------------------|

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

---

scale\_x\_ggtree            *scale\_x\_ggtree*

---

**Description**

scale x for tree with gheatmap

**Usage**

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

**Arguments**

|        |                                 |
|--------|---------------------------------|
| breaks | set breaks for tree             |
| labels | lables for corresponding breaks |

**Details**

Since setting x-axis for tree with gheatmap by using 'theme\_tree2()' is quite tricky, 'scale\_x\_ggtree' can help set the x-axis more reasonably.

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**References**

For more detailed demonstration of this function, please refer to chapter 7.3 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by Guangchuang Yu.

---

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

---

**Description**

add second x-axis for geom\_range

**Usage**

```
scale_x_range()
```

**Details**

notice that the first axis is disabled in the default theme thus users need to enable it first before using scale\_x\_range

**Value**

ggtree object

**Author(s)**

Guangchuang Yu

**References**

For demonstration of this function ,please refer to chapter 5.2.4 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* <http://yulab-smu.top/treedata-book/index.html> by 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     *StatBalance*

---

**Description**

StatBalance

---

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

---

**Description**

dendrogram theme

**Usage**

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

**Arguments**

|         |   |
|---------|---|
| bgcolor | set background color, defaults to "white" |
| fgcolor | set color of 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 |                            |
|                 | set the 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 | set background color, defaults to "white" |
| ...     | additional parameter                      |

**Details**

'theme\_tree' defines a blank background to display tree

**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 | set background color, defaults to "white" |
| fgcolor | set foreground color, defaults to "black" |
| ...     | additional parameter                      |

**Details**

'theme\_tree2' supports displaying phylogenetic distance by setting x-axis

**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 selected clade of tree, clade can be selected by specifying a node number or determined by the most recent common ancestor of selected tips

**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 object will be used. |
| node        | internal node number to specify a clade. If NULL, using the whole tree                         |
| xmax_adjust | adjust the max range of x axis   |

**Value**

clade plot

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(15)
p <- ggtree(x) + geom_tiplab()
viewClade(p, 18, xmax_adjust = 0.)
```



---

|             |                    |
|-------------|--------------------|
| xlim_expand | <i>xlim_expand</i> |
|-------------|--------------------|

---

**Description**

expand x axis limits for specific panel

**Usage**

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

**Arguments**

|       |                             |
|-------|-----------------------------|
| xlim  | x axis limits               |
| panel | name of the panel to expand |

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(30)
p <- ggtree(x) + geom_tiplab()
d <- data.frame(label = x$tip.label,
               value = rnorm(30))
p2 <- p + geom_facet(panel = "Dot", data = d,
                   geom = geom_point, mapping = aes(x = value))
p2 + xlim_expand(c(-10, 10), 'Dot')
```

---

|           |                  |
|-----------|------------------|
| xlim_tree | <i>xlim_tree</i> |
|-----------|------------------|

---

**Description**

set x axis limits specially for Tree panel

**Usage**

```
xlim_tree(xlim)
```

**Arguments**

|      |               |
|------|---------------|
| xlim | x axis limits |
|------|---------------|

**Value**

updated tree view

**Author(s)**

Guangchuang Yu

**Examples**

```
x <- rtree(30)
p <- ggtree(x) + geom_tiplab()
d <- data.frame(label = x$tip.label,
               value = rnorm(30))
p2 <- p + geom_facet(panel = "Dot", data = d,
                  geom = geom_point, mapping = aes(x = value))
p2 + xlim_tree(6)
```

---

zoomClade

*zoomClade*

---

**Description**

zoom in on a selected clade of a tree, while showing its on the full view of tree as a seperated panel for reference

**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 object will be used. |
| node      | internal node number to specify a clade. If NULL, using the whole tree                         |
| xexpand   | numeric, expend the xlim of the zoom area. default is NULL.                                    |

**Value**

full tree with zoom in clade

**Author(s)**

Guangchuang Yu

**Examples**

```
## Not run:
x <- rtree(15)
p <- ggtree(x) + geom_tiplab() +
  geom_nodelab(aes(subset=!isTip, label=node), hjust = -.1, color = "red")
zoomClade(p, 21, xexpand = .2)

## End(Not run)
```

---

%+>%

%+>%

---

### 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

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, 77
- \* **internal**
  - ggtree-package, 4
  - reexports, 71
- %<+% (reexports), 71
- %>% (reexports), 71
- %+>%, 83
- %<+%, 71
- %<%, 83
- %>%, 71
  
- add\_colorbar, 5
- aes, 71
- aes (reexports), 71
- ape::ladderize(), 59
- applyLayoutDaylight, 5
- arrow, 71
- arrow (reexports), 71
- as.polytomy, 6
  
- collapse, 71
- collapse (reexports), 71
- collapse.ggtree, 6
  
- Date2decimal, 7
- decimal2Date, 8
  
- expand, 8
  
- facet\_data, 9
- facet\_labeller, 9
- facet\_plot, 10
- facet\_widths, 11
- flip, 11
- fortify, 71
- fortify (reexports), 71
  
- geom\_aline, 12
- geom\_balance, 13
- geom\_cladelab, 14
- geom\_cladelabel, 17, 19
- geom\_cladelabel2, 18, 18
- geom\_facet (facet\_plot), 10
- geom\_highlight (geom\_hilight), 20
- geom\_hilight, 20
- geom\_inset, 22
- geom\_label, 25, 71
- geom\_label (reexports), 71
- geom\_label2, 23
- geom\_motif, 25
- geom\_nodelab, 26
- geom\_nodelab2, 27
- geom\_nodepoint, 27, 30, 34
- geom\_point, 30, 34, 71
- geom\_point (reexports), 71
- geom\_point2, 29
- geom\_range, 31
- geom\_rootedge, 32
- geom\_rootpoint, 30, 33, 34
- geom\_segment, 35
- geom\_segment2, 34
- geom\_strip, 36
- geom\_striplab, 37
- geom\_taxalink, 40
- geom\_text, 43, 71
- geom\_text (reexports), 71
- geom\_text2, 42
- geom\_tiplab, 43, 46
- geom\_tiplab(), 26, 27
- geom\_tiplab2, 45
- geom\_tippoint, 30, 34, 46
- geom\_tree, 47
- geom\_tree(), 60
- geom\_tree2, 48
- geom\_treescala, 49
- geom\_zoom\_clade, 50
- get.path, 51
- get\_clade\_position, 55
- get\_heatmap\_column\_position, 55
- get\_taxa\_name, 56
- getNodeAngle.df, 51
- getNodesBreadthFirst.df, 52
- getSubtree, 52
- getSubtree.df, 53
- getSubtreeUnrooted, 53
- getSubtreeUnrooted.df, 54
- getTreeArcAngles, 54

- ggdensitree, 56
- ggexpand (hexpand), 63
- ggplot, 71
- ggplot (reexports), 71
- ggsave, 71
- ggsave (reexports), 71
- ggtree, 58
- ggtree-package, 4
- ghemap, 60
- groupClade, 71
- groupClade (reexports), 71
- groupOTU, 71
- groupOTU (reexports), 71
- guide\_legend, 71
- guide\_legend (reexports), 71
- gzoom, 62
- gzoom, ggtree-method (gzoom), 62
- gzoom, phylo-method (gzoom), 62
- gzoom, treedata-method (gzoom), 62
- gzoom.phylo, 62
  
- hexpand, 63
  
- identify, 71
- identify (reexports), 71
- inset (geom\_inset), 22
  
- label\_pad, 64
- layout\_circular (layout\_rectangular), 66
- layout\_dendrogram (layout\_rectangular), 66
- layout\_fan (layout\_rectangular), 66
- layout\_inward\_circular (layout\_rectangular), 66
- layout\_rectangular, 66
- layoutDaylight, 64
- layoutEqualAngle, 65
  
- margin, 71
- margin (reexports), 71
- MRCA, 71
- MRCA (reexports), 71
- msaplot, 67
- multiplot, 68
  
- nodebar, 68
- nodeid, 71
- nodeid (reexports), 71
- nodelab, 71
- nodelab (reexports), 71
- nodepie, 69
  
- open\_tree, 70
  
- package-ggtree (ggtree-package), 4
- plot\_list, 71
- plot\_list (reexports), 71
  
- range\_format, 70
- read.tree, 71
- read.tree (reexports), 71
- reexports, 71
- revts, 71
- rotate, 72
- rotate\_tree, 73
- rotateTreePoints.df, 73
- rtree, 71
- rtree (reexports), 71
  
- scale\_color, 74
- scale\_color, phylo-method (scale\_color), 74
- scale\_color, treedata-method (scale\_color), 74
- scale\_color\_manual, 71
- scale\_color\_manual (reexports), 71
- scale\_color\_subtree, 75
- scale\_colour\_manual, 71
- scale\_colour\_manual (reexports), 71
- scale\_colour\_subtree (scale\_color\_subtree), 75
- scale\_fill\_manual, 71
- scale\_fill\_manual (reexports), 71
- scale\_x\_continuous, 71
- scale\_x\_continuous (reexports), 71
- scale\_x\_ggtree, 76
- scale\_x\_range, 76
- scaleClade, 74
- set\_highlight\_legend, 77
- StatBalance, 77
  
- td\_filter, 71
- td\_filter (reexports), 71
- td\_mutate, 71
- td\_mutate (reexports), 71
- td\_unnest, 71
- td\_unnest (reexports), 71
- theme, 71
- theme (reexports), 71
- theme\_dendrogram, 78
- theme\_inset, 78
- theme\_tree, 79
- theme\_tree2, 79
  
- unit, 71
- unit (reexports), 71
  
- vexpand (hexpand), 63

viewClade, [80](#)

xlim, [71](#)

xlim (reexports), [71](#)

xlim\_expand, [81](#)

xlim\_tree, [81](#)

zoomClade, [82](#)