

# Package ‘tanggle’

November 28, 2021

**Type** Package

**Title** Visualization of Phylogenetic Networks

**Version** 1.0.0

**Description** Offers functions for plotting split (or implicit) networks (unrooted, undirected) and explicit networks (rooted, directed) with reticulations extending. 'ggtree' and using functions from 'ape' and 'phangorn'. It extends the 'ggtree' package [Yu2017] to allow the visualization of phylogenetic networks using the 'ggplot2' syntax. It offers an alternative to the plot functions already available in 'ape' Paradis and Schliep (2019) <[doi:10.1093/bioinformatics/bty633](https://doi.org/10.1093/bioinformatics/bty633)> and 'phangorn' Schliep (2011) <[doi:10.1093/bioinformatics/btq706](https://doi.org/10.1093/bioinformatics/btq706)>.

**Depends** R (>= 4.1), ggplot2 (>= 2.2.0), ggtree

**Imports** ape (>= 5.0), phangorn (>= 2.5), utils, methods

**Suggests** tinytest, BiocStyle, ggimage, knitr, rmarkdown

**VignetteBuilder** knitr

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

**License** Artistic-2.0

**URL** <https://klausvigo.github.io/tanggle>,  
<https://github.com/KlausVigo/tanggle>

**BugReports** <https://github.com/KlausVigo/tanggle/issues>

**RoxygenNote** 7.1.2

**Roxygen** list(old\_usage = TRUE)

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

**git\_branch** RELEASE\_3\_14

**git\_last\_commit** b54fd2e

**git\_last\_commit\_date** 2021-10-26

**Date/Publication** 2021-11-28

**Author** Klaus Schliep [aut, cre] (<<https://orcid.org/0000-0003-2941-0161>>),  
 Marta Vidal-Garcia [aut],  
 Claudia Solis-Lemus [aut] (<<https://orcid.org/0000-0002-9789-8915>>),  
 Leann Biancani [aut],  
 Eren Ada [aut],  
 L. Francisco Henao Diaz [aut],  
 Guangchuang Yu [ctb]

**Maintainer** Klaus Schliep <klaus.schliep@gmail.com>

## R topics documented:

geom_splitnet . . . . .	2
ggevonet . . . . .	3
ggsplitnet . . . . .	4
minimize_overlap . . . . .	5
node_depth_evonet . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

geom_splitnet	<i>geom_splitnet</i>
---------------	----------------------

---

### Description

add splitnet layer

### Usage

```
geom_splitnet(layout = "slanted", ...)
```

### Arguments

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

### Value

splitnet layer

### Author(s)

Klaus Schliep

### Examples

```
data(yeast, package='phangorn')
dm <- phangorn::dist.ml(yeast)
nnet <- phangorn::neighborNet(dm)
ggplot(nnet, aes(x, y)) + geom_splitnet() + theme_tree()
```

---

ggevonet	<i>ggevonet</i>
----------	-----------------

---

**Description**

drawing phylogenetic tree from phylo object

**Usage**

```
ggevonet(tr, mapping = NULL, layout = "slanted", mrsd = NULL,
  as.Date = FALSE, yscale = "none", yscale_mapping = NULL,
  ladderize = FALSE, right = FALSE, branch.length = "branch.length",
  ndigits = NULL, min_crossing = TRUE, ...)
```

**Arguments**

tr	a evonet object
mapping	aes mapping
layout	one of 'rectangular', 'slanted'
mrsd	most recent sampling date
as.Date	logical whether using Date class in time tree
yscale	y scale
yscale_mapping	yscale mapping for category variable
ladderize	logical
right	logical
branch.length	variable for scaling branch, if 'none' draw cladogram
ndigits	number of digits to round numerical annotation variable
min_crossing	logical, rotate clades to minimize crossings
...	additional parameter

**Value**

tree

**Author(s)**

Klaus Schliep

**See Also**

[evonet](#), [ggtree](#)

**Examples**

```
(enet <- ape::read.evonet(text='((a:2,(b:1)#H1:1):1,(#H1,c:1):2);'))
ggevonet(enet) + geom_tiplab()
```

---

ggsplitnet

*ggsplitnet*


---

### Description

drawing phylogenetic tree from phylo object

### Usage

```
ggsplitnet(tr, mapping = NULL, layout = "slanted", mrsd = NULL,
  as.Date = FALSE, yscale = "none", yscale_mapping = NULL,
  ladderize = FALSE, right = FALSE, branch.length = "branch.length",
  ndigits = NULL, ...)
```

### Arguments

tr	a network object
mapping	aes mapping
layout	so far only 'slanted' is supported.
mrsd	most recent sampling date
as.Date	logical whether using Date class in time tree
yscale	y scale
yscale_mapping	yscale mapping for category variable
ladderize	logical
right	logical
branch.length	variable for scaling branch, if 'none' draw cladogram
ndigits	number of digits to round numerical annotation variable
...	additional parameter

### Value

tree

### Author(s)

Klaus Schliep

### References

Schliep, K., Potts, A. J., Morrison, D. A. and Grimm, G. W. (2017), Intertwining phylogenetic trees and networks. *Methods Ecol Evol.* **8**, 1212–1220. doi:10.1111/2041-210X.12760

### See Also

[ggtree](#), [networkx](#), [consensusNet](#), [neighborNet](#)

**Examples**

```
data(yeast, package='phangorn')
dm <- phangorn::dist.ml(yeast)
nnet <- phangorn::neighborNet(dm)
ggsplitnet(nnet) + geom_tiplab2()
```

---

minimize\_overlap      *minimize\_overlap reduces reticulation lines crossing over in plots*

---

**Description**

minimize\_overlap reduces reticulation lines crossing over in plots

**Usage**

```
minimize_overlap(x)
```

**Arguments**

x                      Tree of class 'evonet'

**Value**

A Tree with rotated nodes of class 'evonet'

**Author(s)**

L. Francisco Henao Diaz

**Examples**

```
fishnet <- ape::read.evonet(text='(Xalvarezi,Xmayae,((Xsignum,((Xmonticolus,
(Xclemenciae_F2,#H25)),(((((((Xgordoni,Xmeyeri),Xcouchianus),Xvariatus),
Xevelynae),(Xxiphidium,#H24)),Xmilleri),Xandersi),Xmaculatus),((Xmontezumae,
(Xcortezi,(Xbirchmanni_GARC,Xmalinche_CHIC2))),((Xnigrensis,Xmultilineatus),
(Xpygmaeus,Xcontinens))))#H24)),(Xhellerii)#H25));')
fishnet$edge.length <- NULL
new_tre <- minimize_overlap(fishnet)

par(mfrow=c(1,2))
ggevonet(fishnet, min_crossing = FALSE)
ggevonet(new_tre)

net2 <- ape::read.evonet(text='(15,(1,((14,(#H1,((12,13),(11,#H3)),(7,
((10)#H3,(8,9)))))),(((2,3)#H2,(6,(5,(#H2,4))))#H1));')
# Cui et al. 2013 Evol.
new_net2 <- minimize_overlap(net2)
ggevonet(net2, min_crossing = FALSE)
ggevonet(new_net2)
```

---

node\_depth\_evonet      *Depth of Nodes*

---

**Description**

These functions return the depths or heights of nodes and tips.

**Usage**

```
node_depth_evonet(x, ...)
```

**Arguments**

x                    an object of class 'evonet'  
...                  Further arguments passed to or from other methods.

**Value**

a vector with the depth of the nodes

**See Also**

[node.depth](#)

**Examples**

```
z <- ape::read.evonet(text = '((1,((2,(3,(4)Y#H1)g)e,  
((Y#H1, 5)h,6)f)X#H2)c)a,((X#H2,7)d,8)b)r;')  
nd <- node_depth_evonet(z)  
z$edge.length <- nd[z$edge[,1]] - nd[z$edge[,2]]  
ggevonet(z)
```

# Index

consensusNet, [4](#)

evonet, [3](#)

geom\_splitnet, [2](#)

ggevonet, [3](#)

ggsplitnet, [4](#)

ggtree, [3](#), [4](#)

minimize\_overlap, [5](#)

neighborNet, [4](#)

networx, [4](#)

node.depth, [6](#)

node\_depth\_evonet, [6](#)