

# Package: tanggle (via r-universe)

May 31, 2026

**Type** Package

**Title** Visualization of Phylogenetic Networks

**Version** 1.18.0

**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> and 'phangorn' Schliep (2011) <doi:10.1093/bioinformatics/btq706>.

**Depends** R (>= 4.1), ggplot2 (>= 3.0.0), ggtree

**Imports** ape (>= 5.0), phangorn (>= 2.12), rlang, utils, methods, dplyr

**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>

**Encoding** UTF-8

**RoxygenNote** 7.3.3

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

**Config/pak/sysreqs** libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev

**Repository** <https://bioc-release.r-universe.dev>

**Date/Publication** 2026-04-28 12:56:49 UTC

**RemoteUrl** <https://github.com/bioc/tanggle>

**RemoteRef** RELEASE\_3\_23

**RemoteSha** 057845ac6d634010b3f8302e785e666642dfba43

## Contents

geom_splitnet . . . . .	2
ggevonet . . . . .	3
ggsplitnet . . . . .	4
minimize_overlap . . . . .	5
node_depth_evonet . . . . .	6
swap_hybrid_minor . . . . .	7
<b>Index</b>	<b>8</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**ggevonet*

---

**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**

<code>tr</code>	a evonet object
<code>mapping</code>	aes mapping
<code>layout</code>	one of 'rectangular', 'slanted'
<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
<code>right</code>	logical
<code>branch.length</code>	variable for scaling branch, if 'none' draw cladogram
<code>ndigits</code>	number of digits to round numerical annotation variable
<code>min_crossing</code>	logical, rotate clades to minimize crossings
<code>...</code>	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()
ggevonet(enet, layout = "rectangular") + 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, angle = 0, ...)
```

**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
angle	rotate the plot.
...	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
- Dress, A.W.M. and Huson, D.H. (2004) Constructing Splits Graphs *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **1(3)**, 109–115
- Bagci, C., Bryant, D., Cetinkaya, B. and Huson, D.H. (2021), Microbial Phylogenetic Context Using Phylogenetic Outlines. *Genome Biology and Evolution.* **13(9)**, evab213
- Potts, A.J. and Hedderson, T.A. and Grimm, G.W. (2013), Constructing Phylogenies in the Presence Of Intra-Individual Site Polymorphisms (2ISPs) with a Focus on the Nuclear Ribosomal Cistron, *Systematic Biology.* **63(1)**, 1–16

## See Also

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

## Examples

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

library(phangorn)
fdir <- system.file("extdata/examples", package = "tangle")
nymania <- read.phyDat(file.path(fdir,
                                "Nymania.capensis.ITS.alignment.fasta"), format="fasta")
nnet <- neighborNet(dist.p(nymania))
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,
(Xcortezzi,(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)
```

---

```
swap_hybrid_minor      swap_hybrid_minor
```

---

**Description**

Swapping the minor edges of an evonet object

**Usage**

```
swap_hybrid_minor(x, hybrid_nodes, node_times = NULL)
```

**Arguments**

x	evonet object
hybrid_nodes	a vector of hybrid nodes to have their minor edges swapped
node_times	an optional argument with node times

**Value**

network

**Examples**

```
(enet <- ape::read.evonet(text='((a:2,(b:1)#H1:1):1,(#H1,c:1):2);'))
ggevonet(enet) + geom_tiplab()
swapped_enet<-swap_hybrid_minor(enet,6)
ggevonet(swapped_enet) + geom_tiplab()
```

# Index

consensusNet, [5](#)

evonet, [3](#)

geom\_splitnet, [2](#)

ggevonet, [3](#)

ggsplitnet, [4](#)

ggtree, [3](#), [5](#)

minimize\_overlap, [5](#)

neighborNet, [5](#)

networx, [5](#)

node.depth, [6](#)

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

swap\_hybrid\_minor, [7](#)