

# Package ‘phyloTop’

February 21, 2018

**Title** Calculating Topological Properties of Phylogenies

**Date** 2018-02-21

**Version** 2.1.1

**Maintainer** Michelle Kendall <michelle.kendall@bdi.ox.ac.uk>

**Description** Tools for calculating and viewing topological properties of phylogenetic trees.

**License** GPL-2

**Depends** R (>= 3.2.2), ape

**Imports** igraph, methods, NHPOisson, phylobase

**Suggests** testthat

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Michelle Kendall [aut, cre],  
Michael Boyd [aut],  
Caroline Colijn [aut]

**Repository** CRAN

**Date/Publication** 2018-02-21 18:18:02 UTC

## R topics documented:

avgLadder . . . . .	2
cherries . . . . .	3
colless.phylo . . . . .	4
configShow . . . . .	5
getDepths . . . . .	6
getLabGenealogy . . . . .	7
ILnumber . . . . .	8
ladderShow . . . . .	8
ladderSizes . . . . .	9
makeEpiRecord . . . . .	10
maxHeight . . . . .	11
nConfig . . . . .	12

nodeDepth . . . . .	13
nodeDepthFrac . . . . .	14
nodeImb . . . . .	15
nodeImbFrac . . . . .	16
phyloTop . . . . .	17
pitchforks . . . . .	18
rTree4 . . . . .	19
sackin.phylo . . . . .	19
splitTop . . . . .	20
stairs . . . . .	21
subtreeShow . . . . .	22
treeImb . . . . .	23
widths . . . . .	24

<b>Index</b>	<b>25</b>
--------------	-----------

---

avgLadder	<i>Average ladder size</i>
-----------	----------------------------

---

## Description

Finds the mean size of ladders in the tree

## Usage

```
avgLadder(tree, normalise = FALSE)
```

## Arguments

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
normalise	option to normalise the result, default is FALSE

## Value

The mean ladder size

## Author(s)

Michelle Kendall <michelle.louise.kendall@gmail.com>

## See Also

[ladderSizes](#)

**Examples**

```
## Find the average ladder size in a random tree with 20 tips:
tree <- rtree(20)
plot(tree)
avgLadder(tree)
# and the normalised average ladder size:
avgLadder(tree, normalise=TRUE)
```

---

cherries	<i>Cherry number</i>
----------	----------------------

---

**Description**

Finds the number of cherries in a tree. A cherry is considered to be a pair of sister tips.

**Usage**

```
cherries(tree, normalise = FALSE)
```

**Arguments**

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
normalise	option to normalise the result, default is FALSE.

**Value**

An integer representing the number of cherries in the tree.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

[configShow](#)

**Examples**

```
## Find the number of cherries in a random tree with 10 tips:
tree <- rtree(10)
plot(tree)
cherries(tree)
# and the normalised cherry number:
cherries(tree, normalise=TRUE)
```

```
## Note that the function configShow can be used to highlight the cherries in the tree:  
configShow(tree, 2, edge.width=2)
```

---

colless.phylo

*Colless number*

---

### Description

Finds the Colless number for a tree. Note that the package apTreeshape has a function `colless` to compute the Colless imbalance with additional options to normalise it based on the model; we include this simple function here for convenience within this package, and for use on objects of class `phylo` and `phylo4`.

### Usage

```
colless.phylo(tree, normalise = TRUE)
```

### Arguments

<code>tree</code>	a tree of class <code>phylo</code> or <code>phylo4</code> . The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using <code>multi2di</code> , if possible.
<code>normalise</code>	option to normalise the result by dividing by the number of tip pairs. Defaults to <code>TRUE</code> .

### Value

The Colless imbalance number of the tree.

### Author(s)

Michael Boyd <mboyd855@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

### Examples

```
## Find the Colless imbalance of a random tree with 10 tips:  
tree <- rtree(10)  
plot(tree)  
colless.phylo(tree)
```

---

configShow	<i>Plot a tree highlighting configurations</i>
------------	--

---

**Description**

Plot a tree, highlighting configurations of a given size.

**Usage**

```
configShow(tree, configSize, mainCol = "black", configCol = "red", ...)
```

**Arguments**

tree	a tree of class phylo4
configSize	an integer giving the configuration size of interest
mainCol	colour for branches which are not in configurations of the chosen size (default is black)
configCol	colour for branches which are in such configurations (default is red)
...	further arguments to be passed to plot.phylo

**Value**

A plot of the tree, highlighting the configurations of the given size.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

Michael Boyd <mboyd855@gmail.com>

**Examples**

```
## Highlight pitchforks in a random tree with 20 tips:  
configShow(rtree(20),3, edge.width=2)
```

---

`getDepths`*Find the depth of each node*

---

**Description**

Determines the depth of each node, defined as the number of steps from the root. (So the root has depth zero, its children have depth one, etc.) The output is given as a list of two vectors: `tipDepths` gives the depths of the tips, and `nodeDepths` gives the depths of the internal nodes. This replaces the deprecated `dists` function. The code is based on that used in the function `computeHeights` in package `treeCentrality`.

**Usage**

```
getDepths(tree)
```

**Arguments**

`tree` a tree of class `phylo` or `phylo4`. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using `multi2di`, if possible.

**Value**

A list of two vectors: `tipDepths` gives the depths of the tips, and `nodeDepths` gives the depths of the internal nodes.

**Author(s)**

Michelle Kendall <[michelle.louise.kendall@gmail.com](mailto:michelle.louise.kendall@gmail.com)>

**See Also**

[nodeDepth](#), [nodeDepthFrac](#)

**Examples**

```
## Find the node depths in a random tree:
tree <- rtree(20)
treeDepths <- getDepths(tree)
## Plot tree with node depths labelled:
tree$tip.label <- treeDepths$tipDepths
tree$node.label <- treeDepths$nodeDepths
plot(tree, show.node.label=TRUE)
```

---

getLabGenealogy      *Create genealogy*

---

## Description

Create a labelled genealogy from an epidemiological record

## Usage

```
getLabGenealogy(epirecord, epsilon = 0.001234)
```

## Arguments

epirecord	an epidemiological record, as output from the function <a href="#">makeEpiRecord</a> . It must be a matrix of at least two rows and with five columns named "Infectee", "Infector", "InfnTime", "RecTime", "DoneFlag".
epsilon	an optional small number to be used for branch lengths which would otherwise be zero.

## Value

An object of class `phylo` representing the transmission tree from infectors to infectees.

## Author(s)

Caroline Colijn <c.colijn@imperial.ac.uk>

Michelle Kendall <michelle.louise.kendall@gmail.com>

## See Also

[makeEpiRecord](#)

## Examples

```
## Generate an epidemiological record:
myepirecord <- makeEpiRecord(c(1,2,3,4))
## make the corresponding genealogy from this record:
mygenealogy <- getLabGenealogy(myepirecord)
plot(mygenealogy)
```

ILnumber                      *IL number*

---

**Description**

Computes the number of internal nodes with a single tip child.

**Usage**

```
ILnumber(tree, normalise = FALSE)
```

**Arguments**

tree                      a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.

normalise                option to normalise the result, default is FALSE

**Value**

The integer number of internal nodes with a single tip child.

**Author(s)**

Michael Boyd <mboyd855@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Find the IL number of a random tree with 10 tips:
tree <- rtree(10)
plot(tree)
ILnumber(tree)
## and the normalised IL number:
ILnumber(tree, normalise=TRUE)
```

---

ladderShow                      *Show ladders*

---

**Description**

Plot a tree, highlighting any ladders within it.

**Usage**

```
ladderShow(tree, mainCol = "black", ladderEdgeCol = "red",
  ladderNodeCol = "red", ...)
```



**Arguments**

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
mainCol	colour for edges which are not ladders (default is black)
ladderEdgeCol	colour for ladder edges (default is red)
ladderNodeCol	colour for ladder nodes (default is red)
...	further arguments to be passed to plot.phylo

**Value**

A plot of the tree, with ladder edges and nodes highlighted by colour.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

Michael Boyd <mboyd855@gmail.com>

**See Also**

[ladderSizes](#)

**Examples**

```
## Highlight in blue the ladders in a random tree with 50 tips:
tree <- rtree(50)
ladderShow(tree, edge.width=2)
# compare to:
ladderSizes(tree)
```

---

ladderSizes

*Ladder sizes*

---

**Description**

Finds the sizes and positions of ladders in the tree. A ladder is here defined to be a series of consecutive nodes in the tree, each of which has exactly one tip child (as counted by [ILnumber](#)). The size of the ladder is given by the number of nodes in the chain.

**Usage**

```
ladderSizes(tree)
```

**Arguments**

tree a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.

**Value**

A list of:

- ladderSizes the sizes of ladders in the tree
- ladderNodes the ladder nodes in the tree
- ladderEdges the edges between ladder nodes of the tree

**Author(s)**

Caroline Colijn <c.colijn@imperial.ac.uk>

Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

[ILnumber](#), [ladderShow](#)

**Examples**

```
## Find ladder sizes in a random tree with 20 tips:
tree <- rtree(20)
plot(tree)
ladderSizes(tree)
# note that the ladders can be highlighted in a plot using ladderShow:
ladderShow(tree)
```

---

makeEpiRecord

*Simulate epidemiological record*

---

**Description**

Create an epidemiological record of infectors and infectees with corresponding infection and recovery times

**Usage**

```
makeEpiRecord(lambda, duration = 1, NumCases = 50)
```

**Arguments**

lambda	a numeric vector specifying the time varying intensity lambda(t) which is passed to <code>simNHP.fun</code> to generate the non-homogeneous Poisson process.
duration	the fixed duration of infection (default is 1)
NumCases	the approximate number of infected cases (default is 50)

**Value**

A matrix with columns "Infected", "Infector", "InfnTime" (infection time), "RecTime" (recovery time), "DoneFlag". Suitable for using with `getLabGenealogy`

**Author(s)**

Caroline Colijn <c.colijn@imperial.ac.uk>  
Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

`getLabGenealogy`

**Examples**

```
myepirecord <- makeEpiRecord(c(1,2,3,4), duration=2, NumCases=100)
```

---

maxHeight	<i>Maximum tree height</i>
-----------	----------------------------

---

**Description**

Find the maximum height of tips in the tree.

**Usage**

```
maxHeight(tree, normalise = FALSE)
```

**Arguments**

tree	a tree of class <code>phylo</code> or <code>phylo4</code> . The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using <code>multi2di</code> , if possible.
normalise	option to normalise the result, default is <code>FALSE</code> .

**Value**

An integer giving the maximum height of tips in the tree.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

Caroline Colijn <c.colijn@imperial.ac.uk>

**Examples**

```
## Maximum height of tips in a random tree with 10 tips:
tree <- rtree(10)
plot(tree)
maxHeight(tree)
maxHeight(tree, normalise=TRUE)
```

---

nConfig

*Configuration sizes in a tree*


---

**Description**

Finds the sizes of configurations in the tree.

**Usage**

```
nConfig(tree, maxClade = NULL)
```

**Arguments**

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
maxClade	an integer between 1 and the number of tips (the default), specifying the maximum clade size of interest.

**Value**

A list with 2 entries:

- cladeSizes is a vector giving the size of the clade descending at each node: tips all have the value 1, internal nodes have their number of tip descendants.
- numClades is a vector where numClades[[i]] is the number of clades of size i in the tree. All clade sizes are calculated, but the output can be restricted using maxClade to just those of size up to 'maxClade'.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Configuration sizes on a random tree with 10 tips:  
tree <- rtree(10)  
plot(tree)  
nodeLabels()  
nConfig(tree)
```

---

nodeDepth

*Node depth*

---

**Description**

Determine the depth of a particular node in a tree, defined as the number of edges between it and the root. (So the root has depth zero, its children have depth one, etc.)

**Usage**

```
nodeDepth(tree, node)
```

**Arguments**

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
node	a number corresponding to a node in the tree.

**Value**

An integer corresponding to the depth of the given node.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Find the depth of node 34 in a random tree with 20 tips:  
tree <- rtree(20)  
plot(tree)  
nodeLabels()  
nodeDepth(tree, 34)
```

---

nodeDepthFrac	<i>Fraction of nodes beyond a given depth</i>
---------------	---

---

**Description**

Calculate the fraction of nodes with a depth greater than or equal to a given threshold.

**Usage**

```
nodeDepthFrac(tree, threshold)
```

**Arguments**

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
threshold	a threshold value for node depth.

**Value**

The fraction of nodes with a depth greater than or equal to the threshold (see [nodeDepth](#) for more details on the meaning of node depth).

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

[nodeDepth](#)

**Examples**

```
## Find the fraction of nodes with a depth of 5 or more, in a random tree with 20 tips:  
tree <- rtree(20)  
tree$edge.length <- rep(1,38) # so that the depths are easier to view in a plot  
plot(tree)  
nodeDepthFrac(tree, threshold=5)
```

---

nodeImb	<i>Node imbalance</i>
---------	-----------------------

---

**Description**

For a given node, this function gives the number of tips descending from each of its two children, as a measure of imbalance.

**Usage**

```
nodeImb(tree, node)
```

**Arguments**

tree	a tree of class <code>phylo</code> or <code>phylo4</code> . The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using <code>multi2di</code> , if possible.
node	a node index between 1 and $2n-1$ , where $n$ is the number of tips in the tree.

**Value**

Two integers corresponding to the number of tip descendants of each of the node's two children. If the node is itself a tip, then the vector (0,0) will be returned.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

[treeImb](#)

**Examples**

```
## Find the imbalance of node 16 in a random tree with 10 tips:  
tree <- rtree(10)  
plot(tree)  
nodeLabels()  
nodeImb(tree,16)
```

---

nodeImbFrac	<i>Fraction of nodes with given imbalance</i>
-------------	---

---

### Description

Calculate the fraction of internal nodes with an imbalance greater than or equal to a given threshold.

### Usage

```
nodeImbFrac(tree, threshold)
```

### Arguments

tree	a tree of class <code>phylo</code> or <code>phylo4</code> . The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using <code>multi2di</code> , if possible.
threshold	a threshold value for node imbalance.

### Value

For any internal node, the function `nodeImb` gives the number of tip descendants of each of the node's descending branches. The function `nodeImbFrac` returns the fraction of internal nodes where the difference between these numbers is greater than or equal to the threshold.

### Author(s)

Michelle Kendall <michelle.louise.kendall@gmail.com>

### See Also

[nodeImb](#)

### Examples

```
## Find the fraction of internal nodes with an imbalance of 5 or more,  
## in a random tree with 20 tips:  
tree <- rtree(20)  
plot(tree)  
nodeImbFrac(tree, threshold=5)
```



---

phyloTop

*phyloTop: topological properties of phylogenies*

---

### Description

Calculate a range of topological properties for one or more phylogenetic trees.

### Usage

```
phyloTop(treeList, funcs = "all", normalise = FALSE)
```

### Arguments

treeList	a list or multiPhylo object, or a single tree of class phylo or phylo4. All trees should be binary and rooted; if not they will be coerced into binary rooted trees using multi2di, if possible.
funcs	a list of functions. The default is to apply all of the topological functions from the package, but a subset can be specified instead. The functions available are: <ul style="list-style-type: none"><li>• <a href="#">avgLadder</a></li><li>• <a href="#">cherries</a></li><li>• <a href="#">colless.phylo</a></li><li>• <a href="#">ILnumber</a></li><li>• <a href="#">maxHeight</a></li><li>• <a href="#">pitchforks</a></li><li>• <a href="#">sackin.phylo</a></li><li>• <a href="#">stairs</a> (note that this adds two columns to the output, "stairs1" and "stairs2")</li></ul>
normalise	option to normalise the results of functions where possible. Default is FALSE

### Value

A matrix where rows correspond to trees and columns correspond to topological properties.

### Author(s)

Michelle Kendall <[michelle.louise.kendall@gmail.com](mailto:michelle.louise.kendall@gmail.com)>

### See Also

[avgLadder](#), [cherries](#), [colless.phylo](#), [ILnumber](#), [maxHeight](#), [pitchforks](#), [sackin.phylo](#), [stairs](#)

### Examples

```
## Apply all of the functions to a list of 10 random trees, each with 50 tips:  
phyloTop(rmtree(10,50))  
## Normalising the results where possible:  
phyloTop(rmtree(10,50), normalise=TRUE)
```

---

pitchforks	<i>Number of pitchforks</i>
------------	-----------------------------

---

**Description**

Finds the number of pitchforks in a tree. A pitchfork is considered to be a clade with three tips.

**Usage**

```
pitchforks(tree, normalise = FALSE)
```

**Arguments**

tree	a tree of class <code>phylo</code> or <code>phylo4</code> . The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using <code>multi2di</code> , if possible.
normalise	option to normalise the result, default is <code>FALSE</code> .

**Value**

An integer representing the number of pitchforks in the tree.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**See Also**

[configShow](#)

**Examples**

```
## Find the number of pitchforks in a random tree with 20 tips:
tree <- rtree(20)
plot(tree)
pitchforks(tree)
# and the normalised pitchfork number:
pitchforks(tree, normalise=TRUE)

## Note that the function configShow can be used to highlight the pitchforks in the tree:
configShow(tree, 3, edge.width=2)
```

---

rtree4	<i>Random phylo4 tree</i>
--------	---------------------------

---

**Description**

Creates a random phylo4 tree, in parallel to ape's rtree function.

**Usage**

```
rtree4(n, ...)
```

**Arguments**

n	an integer giving the number of tips in the tree
...	further arguments to be passed to rtree

**Value**

An object of class "phylo4".

**Author(s)**

Michael Boyd <mboyd855@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Create a random phylo4 tree with 10 tips:  
tree4 <- rtree4(10)
```

---

sackin.phylo	<i>Sackin index</i>
--------------	---------------------

---

**Description**

Finds the Sackin index for a tree. Note that the package apTreeshape has a function sackin to compute the Sackin index with additional options to normalise it based on the model; we include this simple function here for convenience within this package, and for use on objects of class phylo and phylo4.

**Usage**

```
sackin.phylo(tree, normalise = FALSE)
```

**Arguments**

`tree` a tree of class `phylo` or `phylo4`. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using `multi2di`, if possible.

`normalise` option to normalise the result, default is `FALSE`.

**Value**

The Sackin index of the tree.

**Author(s)**

Michael Boyd <mboyd855@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Sackin index of a random tree with 10 tips:
sackin.phylo(rtree(10))

## normalised Sackin index:
sackin.phylo(rtree(10), normalise=TRUE)
```

---

splitTop

*Split topology*

---

**Description**

For each node at a given distance from the root, this function finds the size of its induced subclade, i.e. its number of tip descendants.

**Usage**

```
splitTop(tree, dist)
```

**Arguments**

`tree` a tree of class `phylo` or `phylo4`. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using `multi2di`, if possible.

`dist` integer distance of nodes of interest from the root.

**Value**

A vector of integers, each corresponding to the clade size of a node at the given distance from the root. The clade sizes are given in ascending order and each is labelled by its node name or number. This vector can be considered as a partition of the tips or the "split topology" of the tree at a given depth.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

Michael Boyd <mboyd855@gmail.com>

**Examples**

```
## Find the split topology of a random tree with 20 tips, at a distance 2 from the root:  
tree <- rtree(20)  
plot(tree)  
splitTop(tree,2)
```

---

stairs

*Stairs*

---

**Description**

Calculates the staircase-ness measure defined in Norstrom et al., Evolutionary Bioinformatics online, 8:261 (2012) doi: [10.4137/EBO.S9738](https://doi.org/10.4137/EBO.S9738).

**Usage**

```
stairs(tree)
```

**Arguments**

tree                    a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.

**Value**

Two numbers corresponding to the two staircase-ness measures for a tree. These are two related measures:

- 1: the proportion of subtrees that are imbalanced (i.e. subtrees where the left child has more tip descendants than the right child, or vice versa)
- 2: the average of all the  $\min(l,r)/\max(l,r)$  values of each subtree, where  $l$  and  $r$  are the number of tips in the left and right children of a subtree.

**Author(s)**

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Find the staircase-ness measures in a random tree with 20 tips:  
stairs(rtree(20))
```

---

subtreeShow                      *Highlight a subtree*

---

### Description

Plot a tree, highlighting the clade(s) descending from the given node(s)

### Usage

```
subtreeShow(tree, nodeList, showNodeLabels = "nodeList", mainCol = "black",
  subtreeCol = "red", nodeLabelCol = "lightblue", ...)
```

### Arguments

tree	a tree of class phylo or phylo4. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using multi2di, if possible.
nodeList	a list of one or more internal nodes in the tree.
showNodeLabels	option of whether to show node labels. Default is "nodeList", which only labels the nodes in nodeList. Choosing showNodeLabels="all" will display all node labels; any other arguments will remove all node labels.
mainCol	colour for the edges which are not highlighted (default is black).
subtreeCol	colour for the edges which are highlighted (default is red).
nodeLabelCol	background colour for any node labels shown (default is light blue)
...	further arguments to be passed to plot.phylo

### Value

A plot of the tree, with clade(s) descending from the given node(s) highlighted.

### Author(s)

Michelle Kendall <michelle.louise.kendall@gmail.com>

### Examples

```
## Highlight the clade(s) descending from nodes 23 and 35 in a random tree on 20 tips:
tree <- rtree(20)
subtreeShow(tree, nodeList=c(23,35))
# change aesthetics:
subtreeShow(tree,nodeList=c(23,35), mainCol="navy", subtreeCol="cyan",
  nodeLabelCol="cyan", edge.width=2)
```

---

treeImb	<i>Tree imbalance</i>
---------	-----------------------

---

### Description

Find the imbalance of each node, that is the number of tip descendants of each of its two children. With thanks to Leonid Chindelevitch for use of code from `computeLRValues` from `treeCentrality`.

### Usage

```
treeImb(tree)
```

### Arguments

`tree` a tree of class `phylo` or `phylo4`. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using `multi2di`, if possible.

### Value

A matrix where rows correspond to nodes of the tree. The two column entries correspond to the number of tip descendants of each of its two children. (Note that this is the transform of the output in `phyloTop` version 1.0.0.) Where the row number corresponds to a tip, the entries are (0,0).

### Author(s)

Michelle Kendall <michelle.louise.kendall@gmail.com>

### See Also

[nodeImb](#)

### Examples

```
## Find the imbalance numbers in a random tree with 10 tips:
tree <- rtree(10)
plot(tree)
nodelabels()
treeImb(tree)
```

---

widths	<i>Number of nodes at each depth</i>
--------	--------------------------------------

---

**Description**

Find the number of nodes at each depth in the tree

**Usage**

```
widths(tree)
```

**Arguments**

tree            a tree of class `phylo` or `phylo4`. The tree should be binary and rooted; if not it will be coerced into a binary rooted tree using `multi2di`, if possible.

**Value**

A vector of widths, where entry `i` is the number of nodes at depth `i`. There is a single node at depth 0 (the root) which is not included in the vector, for simplicity.

**Author(s)**

Caroline Colijn <c.colijn@imperial.ac.uk>

Michelle Kendall <michelle.louise.kendall@gmail.com>

**Examples**

```
## Find the node widths in a random tree with 10 tips:
tree <- rtree(10)
tree$edge.length <- rep(1,18) # to make it easier to see the width and depths in the plot
plot(tree)
widths(tree)
```



# Index

avgLadder, [2](#), [17](#)

cherries, [3](#), [17](#)

colless.phylo, [4](#), [17](#)

configShow, [3](#), [5](#), [18](#)

getDepths, [6](#)

getLabGenealogy, [7](#), [11](#)

ILnumber, [8](#), [9](#), [10](#), [17](#)

ladderShow, [8](#), [10](#)

ladderSizes, [2](#), [9](#), [9](#)

makeEpiRecord, [7](#), [10](#)

maxHeight, [11](#), [17](#)

nConfig, [12](#)

nodeDepth, [6](#), [13](#), [14](#)

nodeDepthFrac, [6](#), [14](#)

nodeImb, [15](#), [16](#), [23](#)

nodeImbFrac, [16](#)

phyloTop, [17](#)

pitchforks, [17](#), [18](#)

rtree4, [19](#)

sackin.phylo, [17](#), [19](#)

simNHP.fun, [11](#)

splitTop, [20](#)

stairs, [17](#), [21](#)

subtreeShow, [22](#)

treeImb, [15](#), [23](#)

widths, [24](#)